IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ENHANCED ONLINE MEETINIG SCHEDULER SYSTEM BASED ON VISUAL CRYPTOGRAPHY AND ZOOM CLONING

B. Hemakumari, P. Vignesh, K. Vinish Rohan, J. Nitish Kumar

Sreenidhi Institute of Science and Technology,

Yamnampet, Ghatkesar, Hyderabad, Telangana-501301.

ABSTRACT :

Visual Cryptography is employed to securely distribute meeting credentials among participants. This cryptographic technique ensures that no single participant possesses complete access to the meeting credentials, thereby mitigating the risk of unauthorized access. Additionally, Zoom Cloning is utilized to create multiple instances of the meeting environment, each isolated from the others, providing an added layer of security against potential breaches. Visual Cryptography is employed to securely distribute meeting credentials among participants. Additionally, Zoom Cloning is utilized to create multiple instances of the meeting environment, each isolated from the others, providing an added layer of security against potential breaches. [1] Visual cryptography is an encryption technique that decomposes secret images into multiple shares. These shares are digitally or physically overlapped to recover the original image, negating the need for complex mathematical operations or additional hardware. [2] Visual cryptography has applications in secure image sharing, watermarking, and authentication. It is often used in scenarios where security and privacy are critical, such as in medical imaging, document protection, and secure communications. Since visual cryptography divides the secret image into shares and encrypts them to generate individual encrypted shares, this approach of generating partitions can be employed to ensure the privacy and integrity of a meeting or video conference. As zoom meeting app is one of the sophisticated online video conferencing platform, we have implemented the approach of privacy preservation using visual cryptography along with the other features of it. The platform generates keys based on the secret image or text given at the time of scheduling. Then, the internal mechanism divides the key according to number of participants, and shares the key for meeting or conference. [3] With WebRTC, we added real-time communication capabilities to our application that works on top of an open standard, which supports video, voice, and generic data to be sent between peers, allowing developers to build powerful voice- and video-communication solutions. In this way, the application ensures the privacy, integrity of meeting and members of the meeting, and employs video conferencing.

KEYWORDS :

Online meeting Scheduler , Visual Cryptography , Zoom Cloning , Security Enhancement , Privacy Protection , Authentication , Confidentiality , Meeting Credentials , Cryptographic Techniques , Multi-layered Security , Access Control , Virtual Meeting Environment , Secure Communication , Digital Authentication , Meeting Authorization , Secure Scheduling , Secure Collaboration , Cybersecurity , Encryption , Access Management

1. INTRODUCTION :

[4] In the digital era, data transmission plays a crucial role, encompassing various forms such as text, images, and videos. Industries such as banking, medical, marketing, social media, and business applications heavily rely on images for communication with customers and others. However, this increased reliance on digital communication has attracted Cyber Attackers (CAs), individuals or entities that exploit network weaknesses and disturb data integrity, leading to vulnerabilities in data transmission. To address these challenges, this paper introduces an Enhanced Online Meeting Scheduler System (EOMSS) that leverages advanced cryptographic techniques, namely Visual Cryptography, and innovative technology such as Zoom Cloning, to enhance the

security and privacy of online meetings. Related to video streaming, [5]WebRTC has emerged as the primary protocol for the most demanding, ultra real-time video streaming scenarios, such as telepresence, conferencing, surveillance, and drone control. It has also found massive adoption in the media & entertainment industry, both as a production collaboration tool and for streaming live events. By integrating these cutting-edge methodologies, the EOMSS offers a multi-layered approach to safeguarding meeting credentials, protecting sensitive data, and mitigating the risks associated with online communication. [6] Visual cryptography is a method of data hiding in which the data is encrypted and concealed using visuals. [7] It is a cryptographic technique in which no cryptographic computation is needed at the decryption end and the decryption is performed by the human visual system (HVS). By employing Visual Cryptography, the system ensures that meeting credentials are securely distributed among participants, preventing any single entity from gaining unauthorized access to sensitive information. This approach not only enhances authentication but also guarantees confidentiality, as meeting credentials remain protected even in the event of a security breach.

2. RELATED WORK :

The Enhanced Online Meeting Scheduler System (EOMSS) aims to build upon existing research and address the evolving security concerns in online meeting environments. Through the integration of Visual Cryptography and Zoom Cloning, the EOMSS offers a comprehensive solution to enhance security, privacy, and usability in online collaboration platforms.

Existing Web-Based Meeting Scheduler Systems: [8] Modern video conferences are increasingly moving to the WebRTC technology, which works in a browser. [9] Video conferencing applications allow two or more people to connect and conduct video calls. During the call, people can see and hear each other. Therefore, to make a video call we need a device equipped with a camera and a microphone. One of the technologies that permits us to send video and audio stream is WebRTC (Web Real-Time Communication). Numerous web-based meeting scheduler systems have been developed, each offering unique features and functionalities. Studies have surveyed and evaluated popular meeting scheduler tools, highlighting their strengths, weaknesses, and user experiences.

User Interface Design for Web-Based Meeting Schedulers: Research efforts have explored the importance of user interface design in web-based meeting scheduler systems. Works like have investigated user preferences, usability challenges, and design principles to enhance the user experience and efficiency of scheduling meetings online.

Automated Scheduling Algorithms: Studies have focused on developing automated scheduling algorithms to optimize meeting scheduling processes in web-based systems. Research such as has proposed algorithms based on factors like availability, preferences, and constraints to facilitate efficient and conflict-free scheduling of meetings.

Integration with Calendar and Email Systems: Integration with existing calendar and email systems is a common feature in web-based meeting schedulers. Research efforts, including , have examined the integration capabilities, interoperability, and user experiences of meeting schedulers with popular calendar and email platforms.

3. METHEDOLOGY :

3.1 SYSTEM DESIGN :

Starting from the user interaction services to typical backend, as in figure 1, the implementation of ENHANCED ONLINE MEETING SCHEDULER SYSTEM BASED ON VISUAL CRYPTOGRAPHY AND ZOOM CLONING involves various stages or pipelines, they are; user interface for user-interaction, a server for enabling connection and transfer of data from and to database, a server to allow users to join video-conferencing or a meeting, and implementation of abundant activities related to various events occurring on the frontend, such as, click of a button, link, rendering a page, reload, and more.

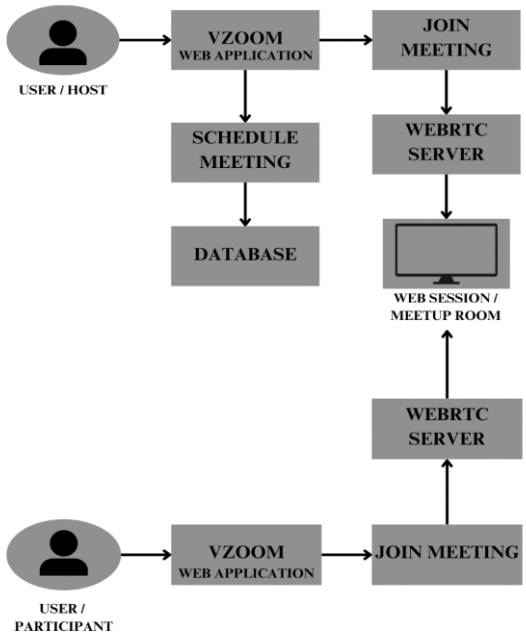

Fig 1:

System Design

3.2 ARCHITECTURE :

The workflow of the system goes like as in figure 2.

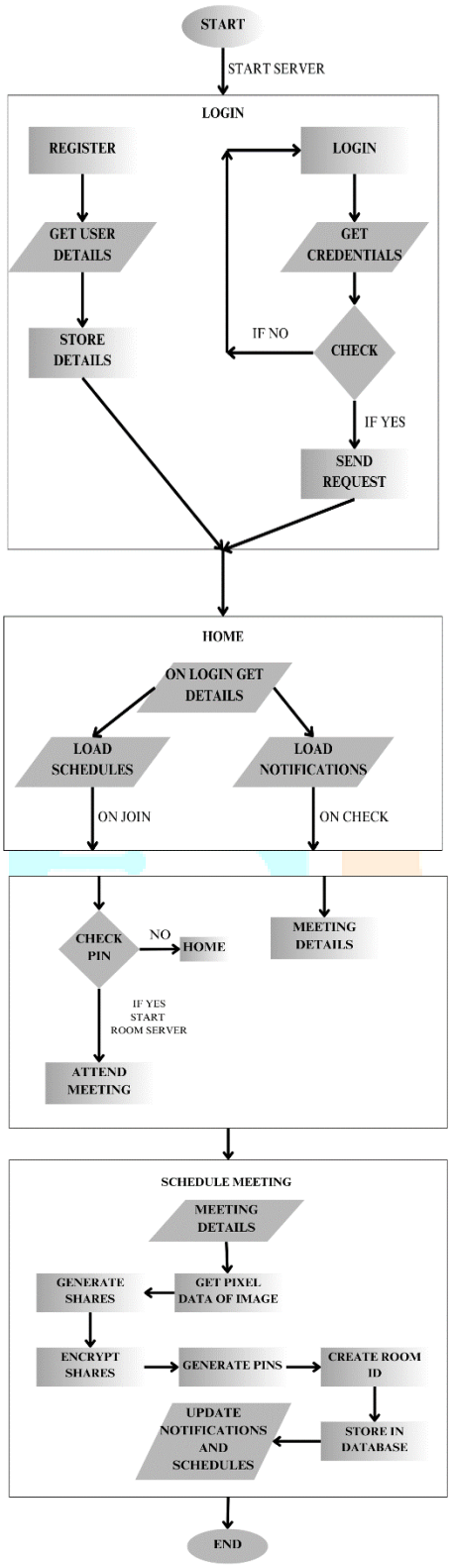

Fig 2:

Application Architecture

3.3 WORKING :

3.3.1 Register and Login:

ENHANCED ONLINE MEETING SCHEDULER SYSTEM BASED ON VISUAL CRYPTOGRAPHY AND ZOOM CLONING is a web application that enables the users in conducting and attending online meetings and video-conferences. As in figure 3, prior to conduct meetings or attend meetings, the user or the customer has to register with their respective details and a password, by which, the user will gain access to log into their respective accounts and interact in a secure mode.

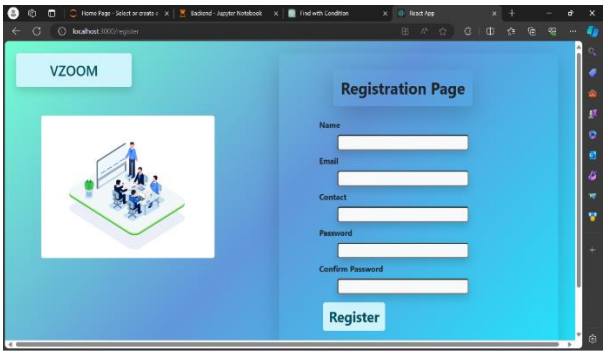

Fig 3: Registration Page

Upon registering, using the credentials at the time of logging in, the user can get into their respective accounts and can securely conduct and attend meetings.

3.3.2 Home page and Meeting Scheduling :

A user after logging in to his/her account using their respective credentials, based on the credentials, the application will gather all the data and information related to user from database and will build the interface, i.e., details of the user, meetings scheduled by the user, notifications about meetings scheduled, and more. Such as shown in figure 4.

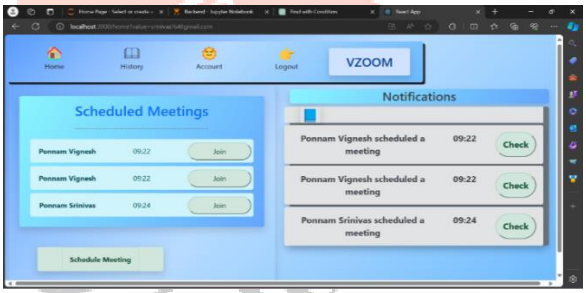

Fig 4: Login Page

As shown in figure 5, to schedule a meeting, the user has to navigate to meeting scheduling page by clicking “Schedule Meeting” button. Upon navigation, the user will be asked to provide several details regarding meetup, such as, name, email, contact, employment status, date and time, and a secret image. After filling all the required details, the user will be prompted to add participants, then, by clicking on “Schedule Meeting” button, the meetup will be scheduled according to the details given. Moreover, it will send notifications to the participants with some details of meeting that was scheduled.

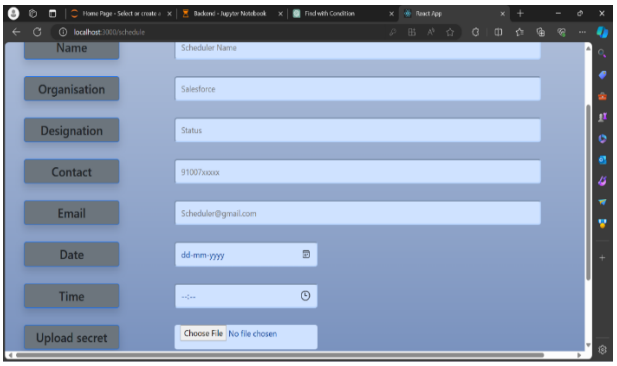

Fig 5: Scheduling Page

3.3.3 Join Meeting :

The client / user can join the meeting or conference by clicking on join and giving the secret pin, that was sent to the participant during scheduling of a meeting. Upon successful event, as shown in figure 6, the server will navigate the user or client to a separate dialog in the web, that consists of unique room id, which was created while scheduling the meeting. All the participants of the meeting will be redirected to same room, thus, video conferencing will be engaged. Here, WebRTC plays a key role. [10] WebRTC is responsible for two major aspects of peer-to-peer conferencing. First, it is responsible for media capture on your device. That means that WebRTC is the technology that tells your device to start recording. Second, it is responsible for transmitting the data between the two devices.

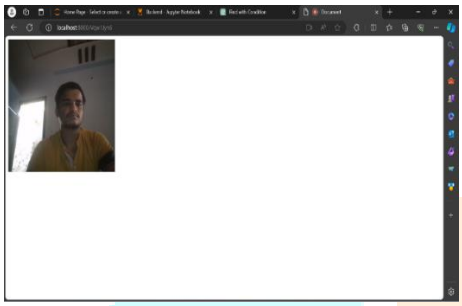

Fig 6: Conference Room

3.4 ANALYSIS METRICS :

- User Interface – The user interface is quite responsive and will adopt to any dimensions. The home page of the user will consists of all the information or data related to user alone. Such as, meetings scheduled, notifications, user-details, and more.

- Room Creation – Upon scheduling the meeting, the flask app at the backend will generate a unique alphanumerical room id and will insert it into the record. Thus, every meeting will be scheduled with a unique room id. Hence, at the time of meetup, the clients will be redirected to dialogs comprised of respective room-ids.

- Real-time Conferencing and Security – WebRTC(Web Real-Time Communication) is a technology that enables real-time communication directly between web browsers or mobile applications without the need for plugins. WebRTC uses encryption protocols like Datagram Transport Layer Security (DTLS) and Secure Real-Time Transport Protocol (SRTP) to ensure secure communication between peers, protecting against eavesdropping and data tampering. [11] PeerJS wraps the browser's WebRTC implementation to provide a complete, configurable, and easy-to-use peer-to-peer connection API. Equipped with nothing but an ID, a peer can create a P2P data or media stream connection to a remote peer.

- Password Generation – Visual Cryptography provides a form of security through obscurity. Each share appears as random noise or patterns, making it difficult for unauthorized individuals to decipher the original image without possessing all the shares. Every time a user schedules a online meetup, the user will be asked to provide a secret image, which will be encrypted using visual cryptography functionalities and generates unique passwords for each participant.

3.5 EVALUATION :

- Interface Usability – The user interface is greatly responsive and will adapt to the dimensions of the dialog or window. The labels on the components will directly show the function behind them. For example, “Schedule Meeting” button will navigate the user to the window where the user can schedule meetings.

- WebRTC Real-time Communication – WebRTC is a powerful framework for real-time communication, it enables built-in support in major web browsers, eliminating the need for plugins, provides direct peer-to-peer communication, reducing latency and enhancing privacy, supports real-time audio, video, and data transmission, offers encryption for secure communication, and compatible with both web and mobile platforms.

- Visual Cryptography – Upon providing a secret image, the application will call functionalities of visual cryptography to split the image into shares according to the number of participants, encrypts them to generate unique passwords or pins according to values of each share of image, and after successful execution, the application will send the passwords to respective participants in needy scenarios.

- Server and Database – Every detail about the user will get stored, i.e., user data, meetings scheduled, login credentials, and more. Server will act as an interface between the user interface and database. For every event on the frontend, there will be a consequent action takes place at server side. The Flask app will enhance connection between frontend and database. It gets requests from the user interface and then fetches related results as a response. On the other-hand, the NodeJS server helps in navigating to conference rooms. For every meeting scheduled, the server will navigate to a unique room, which ensures privacy, isolation and integrity.

4.IMPLEMENTATION ANALYSIS:

4.1 Register and Login :

Every new user has to register first in-order to enter into the home page and conduct or attend meetings. At the time of registration, the details given by the user or customer will be fetched to the backend server, i.e., flask app as a post request, which inserts the data into the database.

At the time of logging in, the user will be asked to enter user credentials, i.e., email-id and password. Upon providing the details, similar to the registration process, the application will send a post request along with the credentials to the server. The task will be achieved by the code in figure 7. Then the server will check the credentials using the data that it possesses from database, and gives out a result as a response to the application. If the credentials are satisfied, then the user will be navigated to respective home page.

```
const Signup = ()=>{

    const [res, setRes]=useState('');
    const [path, setPath]=useState('');
    async function fetchData() {
      try {
        const result = await axios.post('http://127.0.0.1:5000/', {
          key: "value",
          "User":{user},
          "Password":{password},
        });
        setRes(result.data);
        console.log(res);
        if(result.data===1){
          setPath("/home");
          window.location.href = '/home?value='+encodeURIComponent(user);
        }
        else if(result.data===0){
          setPath("/");
          window.location.href = '/';
        }
      } catch (error) {
        console.error('Error fetching data:', error);
      }
    }
```

Fig 7: Login code

4.2 Home Page Rendering :

After successfully logging into the user's account, the user will be redirected to respective home page. Home page comprises of navigation components at the top, (i.e., home, history, logout, account / profile), notifications component, and schedules component. As soon as the user logs into their account, the application captures the email-id, and at the time of loading or rendering home page, the application will send a post request along with the email to the server. The tasks will be achieved by the code in figure 8. Then the server will get all the co-related data from "notifications" and "schedules" tables from database, finally sends the data acquired as a response to the application. Now, upon receiving the records, the rows of the notifications and schedules components will be filled accordingly.

```
function Home() {
  const url="http://127.0.0.1:5000/home";
  const [da, setDa]=useState("");
  const location = useLocation();
  const [email, setEmail]=useState("");
  useEffect(() => {
     const searchParams = new URLSearchParams(location.search);
     const valueReceived = searchParams.get('value');
     console.log('Received value:', valueReceived);
     setEmail(valueReceived);
     const fetchData = async () => {
       try {
         const response = await axios.post(url,{
             email:valueReceived,
         });
         const fetchedData = response.data;
         setDa(fetchedData);
       } catch (error) {
         console.error('Error fetching data:', error);
       }
     };
     fetchData();
   }, [location.search]);
```

Fig 8: Home page code

4.3 Schedule Meeting :

To schedule a meeting, the user has to navigate to schedule page by clicking "Schedule Meeting" on home. Then, the user will be asked to enter various details about the meeting to be scheduled, i.e., name, email, contact, occupation, organization, date, time, and a secret image. Then, the application will prompt the user to add participants and schedule the conference. As soon as the user schedules a meeting, a function the application employs functionalities of visual cryptography, i.e., the function takes in the uploaded image, divides them into shares according to the number of participants, and encrypts them to generate a unique password for each participant. In addition to that, a corresponding post request will be sent to backend server along with the meeting details and passwords. Entire details will be captured and request sent by the code in figure 9. Upon receiving the request, the server will generate a unique random alphanumerical room id, and inserts the whole data as a record into "schedules" table in database. Some selected details about meeting will be inserted into notifications table also. According to the designated time and date, the user can start or join the conference.

```
async function fetchData() {
  try {
    const result = await axios.post('http://127.0.0.1:5000/schedule', {
      key: "value",
      "Name":{name},
      "Organization":{org},
      "Designation":{desig},
      "Contact":{contact},
      "Email":{email},
      "Date":{date},
      "Time":{time},
      "Room":{room},
      "Pin":{pin}

    });

    setData(result.data);
  } catch (error) {
    console.error('Error fetching data:', error);
```

Fig 9: Schedule Meeting code

4.4 Join Meeting :

Selected details about the meetings that are scheduled by the user or with the user will be seen on notifications column and schedules column. As soon as a meeting is scheduled, a notification with some selected details about meetup will be sent to respective participants. Upon checking the notifications, the meeting details will be seen in schedules column. According to the scheduled date and time, the user can join the meeting. At the time of joining, the user will be asked to enter their password. If the password gets validated, the code in figure 10 will start the server, then the user or client will be navigated to the conference room, or-else, will be redirected to home page again.

```
io.on('connection', socket => {
  socket.on('join-room', (roomId, userId) => {
    socket.join(roomId)
    socket.to(roomId).broadcast.emit('user-connected', userId)
    console.log(roomId)
    socket.on('disconnect', () => {
      socket.to(roomId).broadcast.emit('user-disconnected', userId)
    })
  })
})

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PowerShell Extension v2024.0.0
Copyright (c) Microsoft Corporation.

https://aka.ms/vscode-powershell
Type 'help' to get help.

PS C:\Users\vagni\OneDrive\Desktop\meeting app\Zoom-Clone-With-WebRTC-master
> peerjs --port 3001
Started PeerServer on ::, port: 3001, path: /
```

Fig 10: Meeting Room server

5. CONCLUSION :

Incorporating WebRTC and visual cryptography into our project has revolutionized the landscape of virtual communication and security. By leveraging WebRTC's peer-to-peer communication capabilities, we have streamlined real-time interactions, eliminating the need for intermediary servers and enhancing performance. Additionally, integrating visual cryptography ensures the utmost privacy and integrity of our meetings, safeguarding sensitive information from

unauthorized access. Together, these advancements have propelled our project to the forefront of secure and efficient online collaboration, offering users a seamless and protected experience in their virtual interactions.

6. FUTURE ENHANCEMENTS :

To further enhance our project, we can implement several key features. Firstly, integrating end-to-end encryption ensures that all data exchanged during meetings remains confidential and secure. Additionally, introducing advanced authentication methods such as multi-factor or biometric authentication enhances the verification process, preventing unauthorized access. Integration with popular collaboration tools like project management software and document sharing platforms improves productivity during meetings. Incorporating artificial intelligence capabilities for features such as real-time language translation and transcription adds value to the platform. Enhancing accessibility features such as live captioning and screen reader compatibility ensures inclusivity for all users. Moreover, continuous optimization for better audio and video quality, as well as scalability improvements to support a larger number of users, contributes to a seamless and reliable experience across different devices and operating systems. These enhancements collectively elevate our project, meeting the evolving needs of remote collaboration with enhanced security, productivity, and accessibility.

7. REFERENCES :

[1] "An overview of visual cryptography techniques" by Dyala R. Ibrahim, Je Sen Teh, and Rosni Abdullah (2021) provides a comprehensive survey of visual cryptography methods.

[2] "Visual Cryptography Techniques: Short Survey" (2020) reviews works from 2012 to 2020, focusing on various visual cryptography applications such as image encryption, visual authentication, and digital watermarking.

[3] WebRTC Official Website: https://webrtc.org/

[4] "Progressive meaningful visual cryptography for secure communication of grayscale medical images" (2023)

[5] "IIT RTC Conference 2022: WebRTC and Real-Time Applications Track"

[6] "Visual Cryptography: An Emerging Technology" (2023) introduces the concept of visual cryptography, where secret images are split into noise-like shares.

[7] "A New Approach to Enhance Security of Visual Cryptography" (2021) explores the combination of steganography and visual cryptography for secure data transmission over public media.

[8] WebRTC based Platform for Video Conferencing in An Educational Environment" (2020)

[9] "Reliability of Video Conferencing Applications Based on the WebRTC Standard Used on Different Web Platforms" (2020).

[10] "The Ultimate Guide to WebRTC (Web Real-Time Communication) in 2022"

[11] The WebRTC-PeerJs-Demo repository showcases a WebRTC demo using PeerJs