Encrypted real-time communication web (WEBRTC) using the principle of virtual DOM rendering cycles

K T Dermawan¹, I M D Maysanjaya², I M Putrama³

¹ Department of Informatics Education, Faculty of Engineering and Vocational, Universitas Pendidikan Ganesha, Bali, Indonesia – 81116
² Department of Information System, Faculty of Engineering and Vocational, Universitas Pendidikan Ganesha, Bali, Indonesia – 81116
³ Department of Computer Science, Faculty of Engineering and Vocational, Universitas Pendidikan Ganesha, Bali, Indonesia – 81116

Email: teguh.demawan@undiksha.ac.id¹, dendi.ms@undiksha.ac.id², made.putrama@undiksha.ac.id³

Abstract. Website is a collection of formatted pages that display text, image, audio or video content to provide information. Various fields in industry, science, government, non-profit agencies use the website as a medium for presenting information to public or stakeholders. Generally, websites present data synchronously so that it requires an effort to request data until it is displayed on a web page. Industries in the investment, stock and even weather reports require to present information with real-time results. This study aims: (1) Create and design architecture Web Real-time Communication using Virtual DOM Rendering Cycles principles. (2) Present encrypted data to real-time decryption process to render into web page. This research purpose is to provide the solution of real-time rendering encrypted data through process rendering with virtual DOM manipulation. Data encrypted collection have real-time decryption process; its results can be rendered in less interval time of request, with the whole process running on asynchronous process. Technical and non-technical testing steps were conducted, namely (1) White-box, (2) Black-box testing, and (3) Performance consistency testing.

1. Introduction
Website is collection of formatted pages that display text, image, audio or video content. The website technology is also used in the development of information systems. Various fields in industry, science, government, non-profit agencies use the website as a medium for presenting information to public or stakeholders. In some cases, website technology is needed to present information in real time. Such as industries in the investment, stock and even weather reports need to present information with low latency, and it must be real time. Interoperability is a component ability that contains data and information among the integrated systems. It provides an interface to each information system that will enable information system to share data and information. It will also support data and information integrity. In 2017, we have research about development of Prototype Datacenter ECR Systems using messaging service principle. The result is prototyping architecture and functional asynchronous integration between various information systems with real-time capability. This prototype was used to handle interoperability information system in Ganesha University of Education, Indonesia. The results
of our previous research focused on communication between the server side to ensure that each data set can be distributed in real-time with low latency.

Based on research results in 2017, we have a blueprint regarding the process of data distribution in real-time for ensuring interoperability processes can be carried out by utilizing asynchronous communication schemes. Asynchronous communication actually can be implemented with combination between HTTP protocol and messaging services principle [6]. In its application, architecture and principle of our research can be used into the client side, creating worker services to send data with asynchronous and render it into page in the real-time.

Encryption and Decryption are a series of processes that make meaningful phrases into random secret phrases that contain a series of hidden information. Encrypted information can only be opened between end points. There are some technique for implementing encryption message such as in the Symmetric and Asymmetric form. AES and DES method belong to the Symmetric algorithm [4][5], and RSA methods belong to Asymmetric Algorithm. It important to ensure all information system that shared must be secured.

Website uses HTML format as a render component to show in the browser. All content must be formatted into as DOM (Document object model). DOM can be rendered into the browser. The problem of implementation real-time data rendering is to maintain DOM rendering cycles. Generally, DOM runs on with re-rendering after our page receives new response by the server. It means all the content of the DOM is re-rendered. The DOM rendering process takes time to ensure our browser can be fully presented as a page.

Based on the above explanation, a technology was then developed to provide solution to maintain real-time rendering, which must be combined with the principle of rendering cycles. The main tasks in our research is to ensure the web-page does not fully re-render the DOM. This present study, entitled "Encrypted Real-time Communication Web (WebRTC) using The Principle of Virtual DOM Rendering Cycles" focuses on creating and designing architecture Web Real-time Communication using Virtual DOM rendering cycles and ensuring the process decryption message in real-time to present in the pages. This is based on the SocketIO and ReactJS framework.

2. Definition of Terms

**Document Object Model**

The Document Object Model (DOM) is a programming API with formatted tag for XML and HTML documents. Formatted tags define the logical structure of documents that contains the way a document is manipulated and accessed. The structure of DOM is like "document" that can be used in the broad sense. In other documents, format like XML is being used as a way of representing kinds of information that must be stored in diverse systems to implement interoperability, the tags that contains in XML will be shared systematically, it is possible to fetch some information using XML [1]. HTML uses DOM rules to represent object document with tree structure. HTML DOM is a model structure of tree that can be rendered with browser. Each node in the DOM includes a tag that represents information of the part HTML or XML[2].

**E2EE Schema during data coverage**

E2EE is a concept for implementing data security that can hide information message from one end point to another. Encryption is a process of randomizing plain messages to be ciphered so they cannot be read with normal methods. The process to read an encrypted message can only be done using derivative key[8][9]. The principle of End-to-End Encryption is widely used to guarantee data security that can verify producer and consumer message during data communication process[4].

**SocketIO and ReactJS**

Framework of SocketIO is open sources library for real-time web applications in the development based on javascript languages. It enables communication in real-time communication between client
and servers with bi-directional schema. It has two parts of main role component. There are role as client-side, in which the client side must run in the browser, and a role of the server-side, which must run in NodeJS environment. React is open sources javascript library that create by Facebook, running with ability to use NodeJS package manager. ReactJS has ability to support state management. React can be used to maintain structure of DOM manipulation by applying the principle of Virtual DOM Life Cycles[5]. React also can be used to develop SPA (Single Page Application). It also has pattern for optimizing data fetch between client and server sides of the data communication, especially ones with lazy reload during data fetching. React usually makes use of additional libraries for routing component/page in SPA, lazy load interaction API and UI, and state management application.

3. Architectures of Asynchronous Client Side

The initial stage is designing the architecture between client and server with asynchronous communication. The client performs a listening to the server steps, and the server pushes/publishes data to a particular network. The server opens the communication path via the HTTP protocol. The client connected to network will receive real-time response by listening to a specific port on the network. Response fetched can be rendered in to the virtual DOM [3]. Virtual DOM has the ability to update pages with rest and concurrences.

![Figure 1. Asynchronous Client Side Communication Architecture.](image)

Table 1. The Scenario Of The Encryption, Decryption and Rendering Process.

| Process               | Explanations                                                                                                                                                                                                 |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| **Message Encryption**| The process encryption message is used by Cipher Block Chaining (CBC) mode encryption. In this process, a message is encrypted with initial vector key to maintain random character vector to process securing key encryption [4][7]. Initialization vector (IV) is a fixed sizes input to cryptographic that is typically required for randomization. Randomization schema aims at achieving semantic security. It will prevent attacker from collecting pattern of the encrypted message. |
Figure 2. Encryption Process With Initialization Vector (IV).

The process of publishing the encrypted message involves SocketIO to distribute it. The server side will publish the message simultaneously. Message publications remain encrypted while on the network. SocketIO communication runs in realtime to ensure all messages sent through the server can be listened to by the client side.

Figure 3. Publish Message Encrypted with SocketIO.

The client gets an encrypted message through fetching process; the message encrypted must be in block state form, then the state is ready to be manipulated in the management state of ReactJS. The block state is used to inform Virtual DOM to prepare updated component of the page.

Figure 4. Fetch Data as the block state.

The client gets an encrypted message, then the client decrypts the message. The decryption results are still in the form of a block state. This block state will be published as an event on changed component to ReactJS virtual dom.

Rendering Cycles

In the rendering cycles, we use terms of App as main component that contain all of event and virtual DOM information. App is a module created to perform state management processes. State can be read, changed, added, and deleted (CRUD State).

When the client page is first opened, the App module will build the initial state builds to the virtual DOM, the DOM will replicate the contents of the state in the virtual DOM. Furthermore, DOM gives the delivery status to the virtual DOM
that the state has been successfully rendered. The Virtual DOM then sends events states that have been rendered to the App module. The App Module then initiates the state as the current state. The client still listens for data to the Network Channel. When data changes occur, the App module will modify the current state to be updated based on data changes provided by the server.

**Figure 5.** State Management Rendering Cycles.

### 4. Compilation Architecture Design

In this study we combined two architectures to simulate a Real-time-communications web prototype module with encrypted messages. NodeJS environment is used to implement an asynchronous communication architecture design between server and client. SocketIO is used to create server worker to publish message on the network, and ReactJS is used to maintain rendering cycle after data fetched and message decrypted.

**Figure 6.** Compilation Architecture SocketIO and ReactJS.
5. Result

5.1. Software Implementation

a. Interface of Web Real-time simulation
The interface of page simulates the process of decrypt and rendering cycle with real-time and low latency.

![Figure 7. Implementation of page with real-time rendering.](image)

b. Command Line Interface of Distribution Encrypted Message
The interface shows process multi distribution encrypted message with initial vector, sending it with time interval.

![Figure 8. Distribution Message Encrypted.](image)

c. Real-time rendering Encrypted Messages
This is the activity of distribution encrypted messages and process real time rendering, that shows up socket-server and socket client communicate in bi-direction, with messages decrypted in low latency and real-time rendering.

![Figure 9. Real-time rendering Encrypted Messages.](image)
5.2. Review Results of Encrypted Message WebRTC Simulation
The phase of this review includes tabulation analysis of the amount of data tested by the system's resilience in responding to data in real-time, accompanied by the duration of the running time of SocketServer and SocketClient.

| No. | Data  | Success (%) |
|-----|-------|-------------|
| 1   | 246   | 100%        |
| 2   | 6704  | 100%        |

Furthermore, during the testing process in this study, the accuracy of the test sequence was measured by comparing the accumulative average with the average amount of data published by SocketServer. Average accumulative is obtained from the average difference between a lot of data with data loss (cannot be rendered) when rendering in Real-time.

![Success vs Loss Realtime Render](image_url)

**Figure 10. Success vs Loss Realtime Render.**

This study uses 50 test cases, with a random amount of data, with a range of minimum data numbers of 246 to 6704 records. The level of rendering accuracy in the development of NodeJS-based Real-Time Communication (WebRTC) prototype architecture is **99.997%**.

5.3. Problem
During development, there such problem that met, such as:
1. Testing process need more effort and resources to record concurencies UI responses.
2. No have UI team to make complex UI pages, we need to make complex UI to simulate routing and navigation behavior after fetch and render page in the real-time.

6. Conclusion

6.1. Conclusion
Based on the analysis, the implementation and testing in the development of Encrypted Real-time Communication Web (WebRTC) using The Principle of Virtual DOM Rendering Cycles was conducted. In the process of encryption and distribution, message can be published over network in real-time, and in the client side must listen to all encrypted message into distributed channel created by SocketIO.
In this research, message decryption in client side can be done with low latency with real-time DOM update. Updating of the DOM was conducted using virtual DOM that works using ReactJS framework. Overall reliability testing to measure success and loss message decrypted was conducted and the result of rendering success during process real-time decryption was 99.9997%.

6.2. Recommendation

Based on the observations, there are several recommendations to be considered for further development, as follows:

1. In a further development, it is advisable if it can simulate routing and page navigation with more complex UI.
2. In a further development, it is advisable it can simulate data stream such as video and audio data form.

References

[1] Chun Fan, “Research on Development and Evaluation of WebRTC Signaling based on XMPP”, 2017.
[2] Kun Ma & Runyuan Sun, “Introducing WebSocket-Based Real-Time Monitoring System for Remote Intelligent Building,” Int. J. Distributed Sensor Networks, 2013.
[3] Zafran M R M. Gunathunga L G K M, Rangadhari M I T, Gunarathne M D D J, Kuragala K R S C B, Mr Dishan Dhammearathci, “Real Time Information and Communication Center based on webRTC” International Journal of Scientific and Research Publications, vol. 6, no. 4, pp. 644–649, 2016.
[4] Obaida Mohammad and A. Al-Hazaimeh, “A New Approach for Complex Encrypting and Decrypting Data,” Int. J. Comput. Networks Commun., vol. 5, no. 2, p. 88, 2013.
[5] G. Singh, “A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security,” Int. J. Comput. Appl., vol. 67, no. 19, pp. 975–8887, 2013.
[6] Kadek Teguh Dermawan, I. M Putrama and I G D M Darmawiguna, “Penerapan E-CR Pada Pengembangan Prototipe Data Center (Studi Kasus : Unit Pelayanan Teknis Teknologi Informasi dan Komunikasi Universitas Pendidikan Ganesha)”, Jurnal Pendidikan Teknik Informatika (JANAPATI), pp. 25-32, 2018.
[7] Kadek Yota Ernanda, Kadek Teguh Dermawan and I M Putrama, “Fundamental Architectures of Datacenter ECR System Prototype By Applying Principle Of Messaging Services”, The 1st International Conference on Vocational Educational and Technology, 2018.
[8] Pahal Ritu, Vikas Kumar, “Efficient Implementation of AES”, Internation Journal of Advanced Research Computer Science and Software Engineering, vol. 3, no. 5, pp. 875–878, 2013.
[9] Reijo M. Savola, Habtamu Abie, “Development Measureable Security for a Distributed Messaging System”, Internation Journal on Advance in Security, vol. 2, no. 4, pp. 358–380, 2009.
[10] Ata Aburajab, Adnan Salman, “Interactive Blackboard for Web-based Real-time Tutoring System”, IEEE Jordan Internation Joint Conference on Electrical Engineering and Information Technology (JEEIT), pp. 63–68, 2019.