International Journal of Applied and Physical Sciences
volume 12 pp. 01-16 doi: https://dx.doi.org/10.20469/ijaps.12.50001

Efficient and Secure Anonymous Multi-Server Web Application Login Authentication Scheme

Bello Alhaji Buhari*
Department of Computer Science,
Usmanu Danfodiyo University,
Sokoto – Nigeria.
Afolayan Ayodele Obiniyi
Department of Cybersecurity,
Federal University Lokoja – Nigeria.
Bello Aminu Bodinga
Department of Computer Science,
SAISA University of Medical Sciences and Technology.
Hauwau Jibril
Department of Computing,
Northwest University,
Sokoto – Nigeria.
Sirajo Shehu
Department of Computer Science,
Usmanu Danfodiyo University,
Sokoto – Nigeria.
Aliyu A. Dahiru
Department of Computer Science,
Usmanu Danfodiyo University,
Sokoto – Nigeria.
Maniru M. Umar
Department of Computer Science,
Shehu Shagari College of Education,
Sokoto – Nigeria.
Shamsudeen Mustapha
Department of Computing,
Northwest University,
Sokoto– Nigeria.
Sadiq Aliyu Ahmad
Department of Cybersecurity,
Federal University Dutse,
Jigawa State.

Abstract: To grant users access to secure and customized online resources, web applications must recognize, validate, and authorize users using established methods and due to the growing popularity of distributed systems, more and more multi-server settings are being deployed to deliver practical and effective network and online services. An enhanced

*Correspondence concerning this article should be addressed to Bello Alhaji Buhari, Department of Computer Science, Usmanu Danfodiyo University, Sokoto – Nigeria. E-mail: buhari.bello@udusok.edu.ng
© 2026 The Author(s). Published by KKG Publications. This is an Open Access article distributed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

anonymous web application login authentication mechanism was proposed in Buhari and Obiniyi' scheme using hybrid cryptography. Through cryptanalysis, they showed how their system provides security against impersonation attacks, password protection, user privacy, perfect forward secrecy, and mutual authentication. But their schemesingle server for authentication, it has a significant computational cost and is vulnerable to replay attacks.Consequently, in order to address the shortcomings of the Buhari and Obiniyi's scheme while still adopting its advantages, we proposed an effective and secure anonymous multi-server web application login authentication scheme. Using the AVISPA tool, cryptanalysis, time complexity analysis, and performance analysis, the security of the suggested scheme was confirmed, validated, and assessed in relation to the reviewed scheme in terms of security and performance. The AVISPA simulation's results showed that the suggested approach is reliable, effective, safe from replay attacks, and appropriate for multi-server web applications. Furthermore, the performance evaluation of the suggested scheme revealed a significant 64.10% improvement in computation cost when compared to the examined system.

Keywords: *Web application, Authentication, Multi-server, Anonymity, Distributed system*

Received: 03 February 2026; ***Accepted:*** 18 April 2026; ***Published:*** 15 May 2026

I. INTRODUCTION

The internet and technological advancement have increased the use of web applications, thereby revolutionizing the way business and organizational services are accessed. The high demands for these services resulted in the need for improved security measures. Trustworthiness of user identity is one of the main challenges of the Internet, because it is easier for an attacker to pretend to be a legitimate user [1]. This resulted in the development and deployment of different identification, authentication, and authorization methods on web applications.

A web application is a computer program that can be used on desktop computers, laptop computers, tablets, or mobile phones, running on a server that is accessible through a browser [2]. These applications are developed using well known applications such as PHP, NET, Java or JavaScript.

Web applications must identify, authenticate, and authorize users using well-defined techniques in order to allow them access to protected and personalised web resources. The most common approach is the provision of a username and password to gain access to web application resources [2].

More and more multi-server environments are used to provide convenient and efficient network and web services because of the widespread use of distributed systems [3]. Distributed settings have become more prevalent and are changing quickly in the last few years. Different servers work together in this environment to deliver resources and services for users.

The single-server authentication technique presents a greater challenge in this scenario, particularly for those users who must register with each server independently. Moreover, a multi-server user authentication strategy may be used to solve the multi-registration issue of several servers [4].

A lot of research has been conducted on web application authentication. Bucko et al. [1] enhanced the trustworthiness of user authentication in web applications by using user behaviour history. Kamiński et al. in [5] implemented a voice biometrics system in a web application as a two-factor authentication for user login. Iyanda and Mayokun [6] used dynamic passwords with SMS verification to develop a two-factor login authentication system. Buhari and Obiniyi [7] used hybrid cryptography to propose a new anonymous web application login authentication scheme. Olanrewaju et al. [8] employed a friction-less experience to propose a secure web application user authentication mechanism.

Buhari and Obiniyi [7] employed the SHA-2 hash function algorithm, the ElGamal public key algorithm for scalability and security, and the Blow-fish private key method for efficiency. They demonstrated how their system ensures password protection, user privacy, perfect forward secrecy, mutual authentication, and security against impersonation attacks through crypt-analysis evaluation. But it is still insecure against replay attacks; it is a single-server authentication scheme and exhibits high computational cost.

This research proposes an efficient and secure anonymous multi-server web application login authentication scheme in order to resolve the pointed-out weaknesses of Buhari and Obiniyi's scheme [7] and, at the same time, benefit from its strengths. The proposed scheme is multi-server and anonymous. Elliptic curve cryptography is used instead of El-Gamal in order to make our proposed

scheme more efficient because its key sizes are shorter compared to other algorithms, its private and public keys are shorter, and it has faster computational costs and lower demands on memory and bandwidth. Using cryptanalysis, time complexity analysis, performance analysis, and the AVISPA tool, the security of the proposed scheme was confirmed, validated, and compared to the scheme under evaluation in terms of security and performance. The results of the AVISPA simulation showed that the proposed approach is reliable, effective, safe against replay attacks, and appropriate for multi-server web applications. Furthermore, a significant 64.10% improvement in computation cost was demonstrated by the performance evaluation of the suggested scheme in comparison to the assessed scheme. The contributions of our research are as follows:

1. We enhanced Buhari and Obiniyi's [7] web-based application login authentication scheme by making it multi-server instead of single-server to also be applicable in a distributed environment.

2. We employed elliptical curve cryptography instead of ElGamal cryptography to make the proposed scheme more efficient and also ensure user privacy.

3. We evaluated the security and efficiency of our proposed scheme with the reviewed scheme using AVISPA simulation, crypt-analysis, and time complexity, and found that our scheme is more secure and efficient.

II. PREVIOUS WORKS

Irshad and Alreshoodi [9] identified that the Wu et al. [10] scheme is insecure against impersonation attacks, password guessing attacks, and man-in-the-middle attacks. Therefore, they presented a secure and efficient multi-server authentication protocol for 5G networks called SEMS-5G that provides perfect forward secrecy and other security features not provided by previous schemes. They validated their results using the ProVerif tool and BAN logic. Their results show that their protocol supports all security features at low cost.

Petcu et al. [11]] used a novel browser interaction mechanism with users' wallets to describe a straightforward implementation of anonymous user authentication. The implementation is on decentralized applications. They provide a balance of speed, security, and decentralization by combining both Web3 and block-chain technology. This combination enables secure and efficient transactions without centralized infrastructures. They conducted an experiment to evaluate the proposed mechanism with SMS based and mobile wallet-based solutions. Their results show that implementation of the proposed login mechanism can be achieved, and its login times are significantly faster than other login solutions.

Bucko et al. [1] enhanced the trustworthiness of user authentication in web applications by using user behaviour history. They assign weight or percentage to factors such as number of password attempts, IP address consistency, and user agent type. The percentages are updated and stored on the user account after each transaction. They implemented their proposed approach using .NET Framework, C # programming, and a PostgreSQL database. Their results indicated that their proposed approach effectively increases user authentication trustworthiness.

Kamiński et al. [5] implemented a voice biometrics system in a web application as a two-factor authentication user login. The phases of the web application are registration, pre-processing, feature extraction and normalization, classification, and speaker verification. They adapted a modified Gaussian mixture model (GMM) algorithm in their implementation. They also performed experiment to test the performance reached in different acoustic environment during remote voice login the portal.

Iyanda and Mayokun [6] used dynamic passwords with SMS verification to develop a two-factor login authentication system. They added another level of security by generating a dynamic password called OTP in order to secure the login system against dictionary and man-in-the-middle attacks. They implemented their system using PHP, Apache and MySQL and used reliability, effectiveness, efficiency, usability, expediency, and satisfactorily as metrics for evaluating the system. But they could not generate their OTP based on cryptography functions.

Buhari and Obiniyi [7] used hybrid cryptography to propose a new anonymous web application login authentication scheme. They used Blow-fish private key algorithm because of its efficiency, Elgamal public key algorithm because of its expandability and security and SHA-2 hash function algorithm. They evaluated their scheme using crypt-analysis and shown that ensure password protection, user privacy, perfect forward secrecy, mutual authentication and security against impersonation attack.

Akram et al. [12] proposed a web application user authentication system based on adaptive authentication. The factors considered are behavioural and user context factors. A machine learning model that will evaluate the risk profile of the user will be invoked any time a user logs in to the system. Novel features such as mouse speed

and acceleration that are important in determining the accuracy of the machine language model are taken into consideration by their proposed system. Their machine learning model's user context and user behaviour had accuracy of 98% and 87.36%, respectively.

Olanrewaju et al. [8] employed friction-less experience to propose a secure web application user authentication mechanism. It was at the login interface the uniqueness of user identity is validated in their proposed system, then appropriate user authentication process is applied. Behavioural data, together with login session time, device location, browser and details of accessed web services are used in the user account. These data are processed by their system to generate a user profile, which is used to perform authentication process by the authenticator. The outcome of their study showed that their proposed scheme is better than other authentication schemes under existing web-based applications.

Kurniawan et al. [13] improved the security of username and password by employing AES and Blow-fish algorithms to encrypt the One-Time Password (OTP) verification code, which uses an SMS gateway. They compared the performance of AES and Blow-fish with regard to the developed system and shown that OTP runs faster with AES than Blowfish with a variation of ± 0.015 milliseconds for encryption and ± 0.04 milliseconds for decryption.

A multifactor authentication system that has both ease of use and low cost has been proposed by ALSaleem and Alshoshan [14]. It doesn't require special infrastructure or setup and depends on graphical passwords. The user only needs to choose three images and memories them during registration, and only need to choose the correct images in the correct order during login. Their proposed system is secure against many security attacks such as key-loggers, screen capture attacks and shoulder surfing.

Mohammed and Mohamed [15] presented an Android smartphone web-based application to protect against security attacks. Session management is ensured by implementing a bio-metric system as part of the security technique. They use continuous user verification as the methodology defined for perpetual authentication. They also provide the users with appropriate tools to restored the session environment when not using the computer for specified time limits.

Tanrıverdi [16] developed and implemented block-chain based single sign-on authentication system for web applications. They ensured 2FA by defining public address and private key on the private block-chain network of the users through the mobile application developed.

Combination of both fingerprinting and behavioural dynamics – mouse keyboard in order to enhanced the security of login mechanism has been proposed by Solano et al. [17]. They accomplished this by employing machine learning techniques. They used data-set consisting of mouse, keyboard and session context information of 24 user to evaluate and simulate their proposed technique. They show that context analysis and behavioural analysis when combined reaches up to 0.9 accuracy than individual approaches that produce only 0.7 accuracy each.

Auth.js was designed and implemented by Christou and Athanasopoulos [18] as a framework that enables a web application to provide advanced authentication that powers sophisticated methods compared to typical cryptographically hashed passwords. This can be used implement strong authentication that ensures cryptographic primitives are trusted and executed using browser engine like PAKE and solutions alike. They also improve Mozilla Crypto with scrypt and edwards25519 elliptic curve. Lastly, they evaluate it with real web application like WordPress.

With dApp as token generation system Putri et al. [19] proposed a two-factor authentication framework based on Ethereum block-chain without using third party. Their token system generates up to 3164 different tokens in a second with no collision. They also protect the token from man-in-the-middle attacks. Their results indicated that even if an attacker gets the token value, it is difficult for an attacker to get the original token value.

Two factor graphical password authentication approach which authenticate website accounts called Pass-Page has been proposed by Chu and Chen in [20]. Implicit memories of users' browsing web-pages are used by their proposed approach. The server displays 9 small pages as a challenge when ever a user wants to login. It will ask the user to select all the pages he/she browses in addition to the text password entered. They conducted an experiment on 12 volunteers. The result of their experiment shown that when users are familiar with the login process, the login success rate is over 80% and login success rate does not decline for 6 days suddenly.

An algorithm that will allow users to perform registration and access control on web applications safely has been designed by Mohammed and Mehdi [21]. Their design is based on zero-knowledge proof with chaotic system and RSA algorithm. Their novel 6D-Hyper chaotic system initiated the complex generation of random number.

A multi-factor authentication protocol has been designed and developed by Boonkrong [22]. It starts from registration to actual authentication. They generate authentication factors that are categorized as short-term and

long-term. Only three messages are exchanged between clients and financial institution's server. My cryptography processes such as symmetric and asymmetric cryptography, symmetric key generation, technique for generating and verifying digital signatures are combined in their protocol. They proved and analysed their protocol using GNY and OWASP-AT-009.

A web authentication scheme that employs pervasive mobile devices and a digital camera has been presented by Xie et al. [23]. These combinations are done to secure against various password threats such as phishing and man-in-the-middle attacks. They used mobile device as the second authentication medium to identify the legitimacy of the user. The communication between the mobile device and PC is direct via visible light communication channels. This has no cellular cost attached and is secure against radio frequency attacks. They adopt public key cryptography in order to guarantee a secure authentication process. Their prototype incorporated an Android application, a Chrome web browser, and a Java-based web server.

III. REVIEW OF BUHARI AND OBINIYI'S SCHEME

A new web application login authentication scheme using hybrid cryptography has been proposed by Buhari and Obiniyi [7] in which user privacy is ensured through anonymity. They employ Blow-fish private key cryptography due to its efficiency, ElGamal public key cryptography due to its security and expand-ability, and SHA-2 as the hash function algorithm. It consists of five phases namely: initialization, registration, login, authentication and change password.

A. *Initialization Phase*

The web server WS perform the under-listed activities before a user can have legitimate access to web application by registration.

1. Select two prime numbers p and q such that
$$p = 2q + 1 \tag{1}$$
2. Select a generator g of Z_q^*
3. Select a cryptographic hash function $h()$ that is SHA-2
4. Select $x \in Z_q^*$ as its secret key
5. Select a private key algorithm like Blow-fish; $E_y()$ for encryption and $D_y()$ for decryption using key y
6. Decide the format of identity UN_i

This can be shown in figure 1.

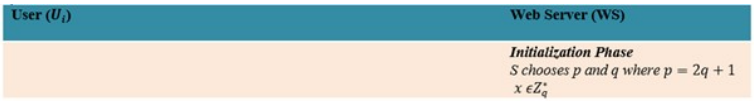


Fig. 1. Initialization Phase [7]

B. *Registration Phase*

For a user U_i to register with the web application server he/she does the following:

1. U_i selects his/her identity UN_i and password PW_i in the specified format and submit to WS securely.
2. U_i selects a random number b_i
3. WS encrypt the identity UN_i as
$$EUN_i = E_x(UN_i) \tag{2}$$
4. WS computes security parameters
$$P_i = (EUN_i)^{h(x,PW_i)} \pmod{p} \tag{3}$$
5. WS stores $\{P_i, EUN_i, h()\}$ to the database

This can be shown in figure 2.

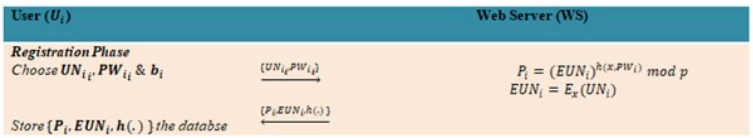


Fig. 2. Registration Phase [7]

C. Login Phase

If a user U_i wants to login to the web application server WS he/she performs the following:

1. Provides identity UN_i and password PW_i
2. U_i computes:

$$C_i = P_i \mid (EUN_i)^{PW_i} \pmod p \tag{4}$$
$$D_i = (EUN_i)^{b_i} \pmod p \tag{5}$$
$$W_i = C_i D_i \pmod p \tag{6}$$
$$M_i = h(EUN_i, C_i, D_i, W_i) \tag{7}$$



Fig. 3. Login Phase [7]

D. Authentication Phase

WS and U_i perform the following steps to authenticate each other

1. On receiving the login request message WS checks the format of identity UN_i, if UN_i is valid it goes to the next step; otherwise it rejects the login request.
2. WS computes:

$$C_i = (EUN_i)^x \pmod p \tag{8}$$
$$W_i' = C_i D_i \pmod p \tag{9}$$
$$M_i' = h(EUN_i, C_i, W_i) \tag{10}$$

3. WS compares M_i' and M_i, if they are equal U_i is authenticated; otherwise, the login request is rejected
4. WS computes:

$$M_s = h(EUN_i, W_i') \tag{11}$$

And sends the mutual authentication message $\{EUN_i, M_s\}$ to U_i

5. On receiving the message, U_i check EUN_i and compares M_s and M_i, if they are equal WS is also authenticated; otherwise the login request is rejected.
6. Now, both U_i and WS compute the session key

$$sk = h(W_i) = h(W_i') \tag{12}$$

This can be shown in figure 4.

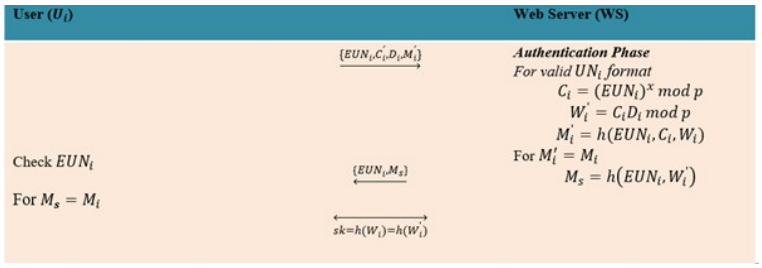


Fig. 4. Authentication Phase [7]

E. Change Password Phase

User U_i performs the following steps to change his/her existing password PW_i to a new password $(PW_i)_{new}$. We assume that the user is already authenticated before he/she can be able to change password.

1. The user U_i provides his/her new password $(PW_i)_{new}$
2. WS re-computes security parameters

$$(P_i)_{new} = (EUN_i)^{h(x,(PW_i)_{new})} \tag{13}$$

3. WS replaces P_i with $(P_i)_{new}$ on the EUN_i data row in the database

This can be shown in figure 5,

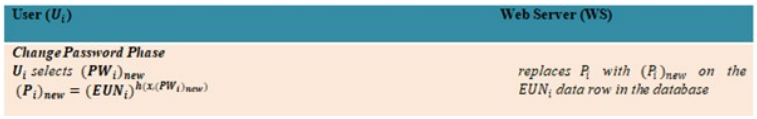


Fig. 5. Change Password Phase [7]

IV. WEAKNESSES OF BUHARI AND OBINIYI'S SCHEME

This section presented the limitations of Buhari and Obiniyi [7] scheme. Based on the review it is found that it is still insecure against replay attack, it is a single server authentication scheme and exhibit high computational cost. These can be discussed in the remaining sub-sections of this section.

A. *Replay Attack*

An attacker can intercept and re-transmit the request messages $\{EUN_i, C_i, D_i, W_i\}$, $\{EUN_i, C_i', D_i, M_i'\}$ and $\{EUN_i, M_s\}$ to impersonate both the user and the user. And the server or the user will not know the Genuity of the re transmitted messages because timestamp has not been used.

B. *Single Server Authentication*

Their scheme is a single server authentication scheme. Single server authentication suffers a significant insufficiency because user has to register with each server separately and remember many usernames and passwords if he/she wishes to access web services from different web servers especially in a distributed computing environment.

C. *High Computational Cost*

Their scheme is based Elgamal public key algorithm. Elgamal is based on discrete logarithm problem and requires modular exponentiation operations. This makes the schemes based on Elgamal cryptographic algorithms to have high computational cost.

V. PROPOSED SCHEME

The proposed scheme deployed Elliptical Key Cryptography (ECC) instead of Elgamal because it has smaller key size and has faster computations than other public key algorithms. It also provides the same security with other public key algorithm with the smaller key size. It is multi-server in order to support distributed or multi-server environments and so, there is going to be registration center (RC) that is in charge of generating system parameters and completing user and server registration.

It consists of six phases namely: initialization, server registration, user registration, login, authentication and change password. The details of these phases will be discussed in the next sub-section of this section.

A. *Initialization Phase*

The RC generates the following parameters before it can both register server and user with the web application. These are:

1. A prime order finite field with an elliptical curve defined over it
2. An additive cyclic group
3. A generator
4. A hash function where
5. Generates which is master secrete key known to RC, which is user secrete key only known to servers and which server secrete key known to users.
6. Decides the format of the username

This can be shown in figure 6.



Fig. 6. Initialization Phase

B. Server Registration Phase

Server register with RC by doing the following:

1. Select identity
2. Also select the current timestamp
3. Send to RC
4. RC encrypt the identity:

(14)

5. RC computes identity value:

(15)

6. RC select the current timestamp
7. RC store and on the database
8. RC sends $\{\dots\}$ to server
9. Stores and on the database

This can be shown in figure 7.

S_i		RC
Server Registration Phase		
S_j select sid_j		
S_j gets timestamp T_j	$\xrightarrow{\{sid_j, T_j\}}$	If $T_j - T \leq \Delta T$
		$esid_j = e_{x_s}(sid_j)$
		$I_j = x(esid_j \| \| H(x_s)) \cdot g$
		RC gets timestamp T_{rc}
	$\xleftarrow{\{I_j, H(), x_u, T_{rc}\}}$	store $esid_j$ and I_j
If $T_{rc} - T \leq \Delta T$		else discard
stores I_j and $H()$		
else discard		

Fig. 7. Server Registration

C. User Registration Phase

User register with RC by doing the following:

1. Select username and password in the format defined
2. Also select the current timestamp
3. Send to RC
4. RC check the current timestamp. If valid, it continues; otherwise, discard the registration request.
5. RC encrypt the username:

(16)

6. RC computes identity value:

(17)

7. RC computes security parameter:

(18)

8. RC send
9. RC stores and to the database

This can be shown in figure 8.

U_i		RC
User Registration Phase		
U_i select un_i and pw_i		
U_i gets timestamp T_i	$\xrightarrow{\{un_i, pw_i, T_i\}}$	If $T_i - T \leq \Delta T$
		$eun_i = e_{x_u}(un_i)$
		$I_i = x(eun_i \| \| H(x_u)) \cdot g$
		$P_i = I_i \oplus H(pw_i)$
	$\xleftarrow{\{eun_i, P_i, H(), x_s, T_{rc}\}}$	RC gets timestamp T_{rc}
If $T_{rc} - T \leq \Delta T$		else discard
Stores eun_i, P_i and $H()$		
else discard		

Fig. 8. User Registration Phase

D. Login Phase

User U_i login to the server S_j by doing the following:

1. Provides username *uni*, password *pwi* and server id sid_i

2. U_i computes:

$$I_i' = P_i \oplus H(pwi) \tag{19}$$
$$C_i = esidi \parallel H(x_s) \tag{20}$$
$$K_i = C_i I_i' \tag{21}$$

3. U_i choose a random number r_i

4. U_i computes:

$$R_u = r_i K_i \tag{22}$$
$$W_i = H(K_i) \oplus R_u \tag{23}$$
$$M_i = H(K_i) \parallel R_u \tag{24}$$

5. At the current timestamp T_i', U_i send login request $\{euni, W_i, M_i, T_i'\}$ to server S_j

This can be shown in figure 9.

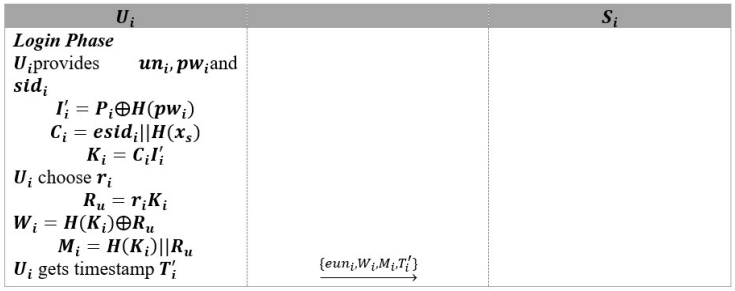


Fig. 9. Login Phase

E. Authentication Phase

The server S_j and user U_i do the following to authenticate each other:

1. After receiving the login request message, the server S_j check the current timestamp. If $T_i' - T \leq \Delta T$ it continues, otherwise, discard the login request.

2. S_j computes

$$C_j = euni \parallel H(x_u) \tag{25}$$
$$K_j = C_j I_j \tag{26}$$
$$R_u' = W_i \oplus H(K_j) \tag{27}$$
$$M_i' = H(K_j) \parallel R_u' \tag{28}$$

3. Check if $M_i' \stackrel{?}{=} M_i$. If they are equal to user is authenticated, otherwise, login request is rejected

4. S_j choose a random number r_j

5. S_j computes:

$$R_s = r_j K_j \tag{29}$$
$$W_j = H(K_j) \oplus R_s \tag{30}$$
$$M_j = H(K_j) \parallel R_s \tag{31}$$

6. At the current timestamp T_j', S_j send login request $\{W_j, M_j, T_j'\}$ to user U_i

7. After receiving the authentication request message, the user U_i check the current timestamp. If $T_j' - T \leq \Delta T$ it continues, otherwise, discard the authentication request.

8. U_i computes:

$$R_s' = W_j \oplus H(K_i) \tag{32}$$
$$M_j' = H(K_i) \parallel R_s' \tag{33}$$

9. Check if $M_j' \stackrel{?}{=} M_j$. If they are equal to server is authenticated, otherwise, authentication request is rejected

10. Both the user U_i and server S_j generate session

$$ski = H(r_i R_u) = H(r_j R_s) \tag{34}$$

This can be shown in figure 10.

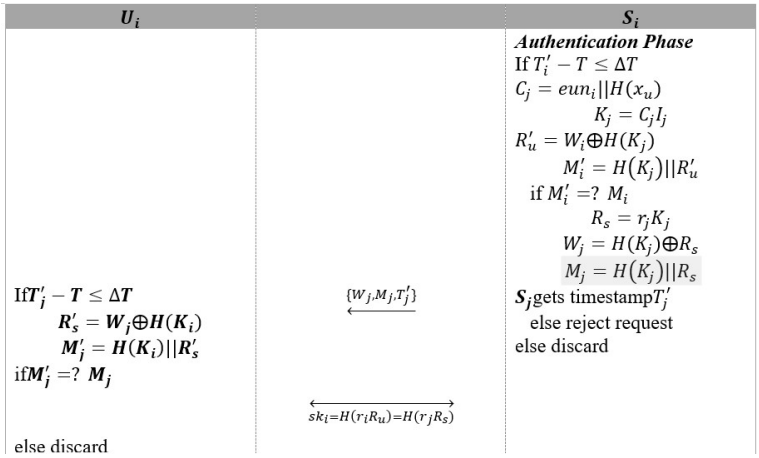


Fig. 10. Authentication Phase

F. Change Password Phase

User U_i has to be authenticated before he/she can change his/her old password pwi to new password pwi''. He/she will go to RC and perform the following:

1. U_i provides uni', old password pwi' and new password pwi'' to RC
2. At the current timestamp T_i', U_i send change password request $\{uni', pwi', pwi'', T_i'\}$ to RC
3. After receiving the change password request message, the RC check the current timestamp. If $T_i' - T \leq \Delta T$ it continues, otherwise, discard the change password request.
4. RC computes:

$$euni' = E_{x_u}(uni') \tag{35}$$

$$I_i'' = x(euni' \parallel H_{x_u}) \cdot g \tag{36}$$

$$P_i' = I_i'' \oplus H(pwi') \tag{37}$$

5. RC gets P_i from the database using $euni'$
6. RC checks if $P_i' \stackrel{?}{=} P_i$. If they are equal it continues, otherwise, request is rejected
7. RC computes:

$$P_i'' = I_i'' \oplus H(pwi'') \tag{38}$$

8. RC replaces P_i with P_i'' on the database.

This can be shown in figure 11.

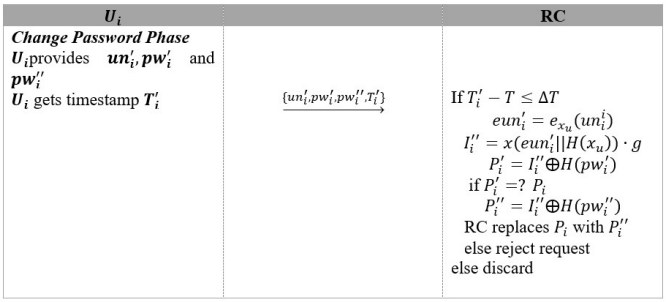


Fig. 11. Change Password Phase

VI. STRENGTHS OF THE PROPOSED SCHEME

We analyse and discuss the strengths of our proposed scheme in this section. The improvement of the reviewed research to be discussed include: multi-server authentication, secure and efficient authentication scheme and security against replay attack. These are going to be discussed in the next sub-sections of this section.

A. Multi-server Authentication

Buhari and Obiniyi [7] presented web application login authentication scheme is single user scheme. In multi-server environment like distributed web-based system the user U_i has to register with all the servers, remember

many login credentials and he/she has to login any time he/she switches from one server to another in a distributed environment.

But our proposed scheme is multi-server web application login authentication scheme. In our scheme there is going registration center (RC) where user U_i and servers S_j register once. So, the user U_i login to each server by just specifying the server id sid_j as shown in the login phase of the proposed scheme.

B. *Secure and Efficient Authentication Scheme*

Our proposed scheme is based on Elliptical Curve Cryptography (ECC). The basic EC operations is the Point Multiplication which is the result of adding the same point P to itself n times which is given as:

$$nP = \sum_{i=1}^{n} P = R \tag{39}$$

Given a point R it is hard to find the initial point P or the number of n iterations. This the power of this operation and is called Elliptic Curve Discrete Logarithm Problem (ECDLP). Since the key sizes of ECC are shorter compared to other algorithms, its private and public keys are shorter and so has faster computational costs, and lower demands on memory and bandwidth.

C. *Security Against Replay Attack*

An attacker A_i can intercepts the previous messages sent during registration, login and authentication such as $\{I_j, H(), x_u, T_{rc}\}$, $\{euni, W_i, M_i, T_i'\}$, etc., in order to impersonate the legitimate user. Our proposed scheme uses the concept of timestamp to preserve the freshness of the messages in each request attempt. Any time an attacker A_i attempt to replay previous message will have a failed attempt because the user, RC or server check the freshness of the timestamp sent together with the request message. Also, comparisons such as $M_i' \stackrel{?}{=} M_i$ and $M_j' \stackrel{?}{=} M_j$ will not hold if an attacker A_i replaces only timestamp T_i' and T_j' in messages $\{euni, W_i, M_i, T_i'\}$ and $\{W_j, M_j, T_j'\}$ with T_a and T_a'. Therefore, replay of intercepted messages in any way is not successful in our proposed scheme.

VII. FORMAL VERIFICATION USING AVISPA SIMULATION

AVISPA was used here to verify and validate the proposed scheme protocol. Using the HLPSL specification, a security animator known as SPAN for HLPSL was utilized to interactively create a message sequence chart (MSC) for the proposed scheme protocol execution. If the proposed protocol is unsafe, the AVISPA verification tool will automatically create an attack MSC on the HLPSL specification.

The simulation was run on an HP laptop running Windows 10 Pro 64-bit, with an Intel(R) Celeron(R) N4100 CPU running at 1.10 GHz, 4.00 GB of RAM, and an x64 processor. After installing VirtualBox, SPAN on Ubuntu 10.10 was installed in order to configure the simulation environment.

A. *Proposed Scheme Verification Results*

The agents involved were the web user, web server and registration centre. So, these roles were first implemented using the HLPSL language to show how the web server register with the registration centre, web user register with the registration centre, web user send login request to the web server and the web server send mutual authentication request to the web user.

Figure 12 displays the OFMC verification, which demonstrates if the proposed protocol is safe against session key communication. The outcome demonstrated that the proposed technique guarantees mutual authentication and is safe and secure against session key transmission.

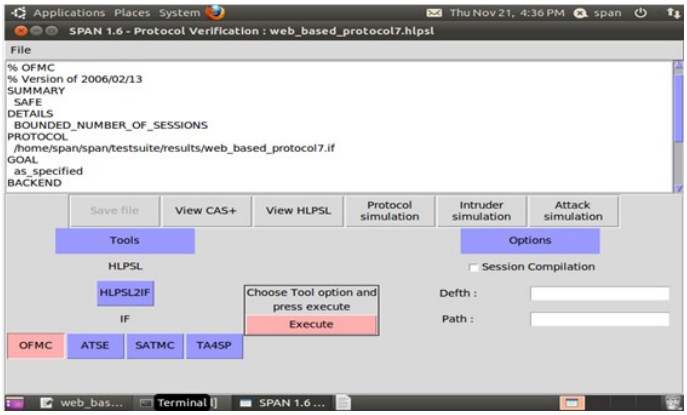


Fig. 12. OFMC Verification Results

Figure 13 displays the CL-AtSe verification, which is actually used to demonstrate if the proposed protocol is safe against adversary attacks. The outcome demonstrated that the proposed scheme is safe and secure from man-in-the-middle and replay attacks.

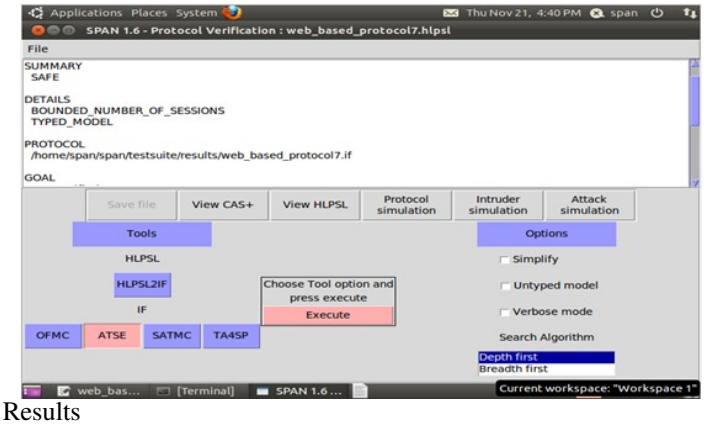


Fig. 13. CL-AtSe Verification Results

Following protocol verification, Security Protocol Animator (SPAN) for AVISPA under OFMC and CL-AtSe back-ends is used to simulate the proposed method and show the transitions between the three interacting entities, web user, registration centre and web server. In accordance with the proposed scheme's interactive user options, SPAN is utilized to construct one potential execution (an MSC drawing) as shown in Figure 14.

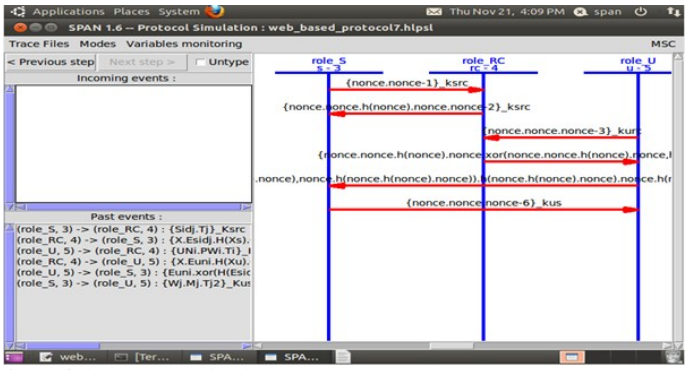


Fig. 14. SPAN+AVISPA simulation of the proposed scheme

VIII. PERFORMANCE EVALUATION OF THE PROPOSED SCHEME

Performance of the proposed scheme and Buhari and Obiniyi's scheme are analysed and compared in this section. The analysis and evaluation are based on computation cost and communication cost. These will be discussed in the next sub-sections of this section.

A. *Computation Cost Evaluation*

The symbols of the computations to be used in the evaluation of the proposed scheme and reviewed scheme is shown in table 1.

TABLE 1
Defined Computation Time for Different Operations [3]

S/NO	OPERATION	DESCRIPTION	COST (s)
1	T_H	One way hash function	0.023
2	T_{Asym}	Cost for asymmetric encryption/decryption	3.85
3	T_{Sym}	Cost for symmetric encryption/decryption	0.0046
4	T_{ECC}	Cost for scalar point multiplication of ECC	2.226
5	T_{Exp}	Cost of modular exponentiation	3.85
6	T_{pa}	Cost of point addition over G	0.0288
7	T_{grg}	Cost of generator and random number on Z_q^*	0.539

B. Computation Cost of the Proposed Scheme

This sub-section describes the computation cost of our proposed scheme. During initialization phase, $1T_{pa} + 1T_H$ equivalent to 0.0518s are needed. $1T_{Sym} + 1T_{ECC} + 1T_H$ equivalent to 2.2536s are needed for server registration phase. User registration phase needed $1T_{Sym} + 1T_{ECC} + 2T_H$, which are equivalent to 2.2766s. Only $4T_H$ equivalent to 0.092s are needed for login phase. Authentication phase needed $1T_{Sym} + 9T_H$, equivalent to 0.2116s. $1T_{Sym} + 3T_H + 1T_{ECC}$, equivalent to 2.2996s are needed for change password phase. And the total computation cost needed for our proposed scheme are $4T_{Sym} + 20T_H + 4T_{ECC}$, equivalent to 7.1852s. This can be shown in Table 2 and Figure 15.

TABLE 2
Computation Cost of Proposed Scheme

Scheme Phases	**Computation Cost**	**Computation Time (s)**
Initialization Phase	$1T_{pa} + 1T_H$	0.0518
Server Registration Phase	$1T_{Sym} + 1T_{ECC} + 1T_H$	2.2536
User Registration Phase	$1T_{Sym} + 1T_{ECC} + 2T_H$	2.2766
Login Phase	$4T_H$	0.092
Authentication Phase	$1T_{Sym} + 9T_H$	0.2116
Change Password Phase	$1T_{Sym} + 3T_H + 1T_{ECC}$	2.2996
TOTAL	$4T_{Sym} + 20T_H + 4T_{ECC}$	**7.1852**

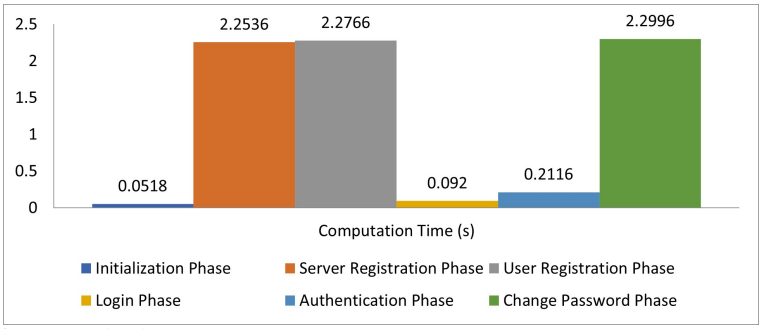


Fig. 15. Computation cost of proposed scheme

C. Comparison of Computation Cost

Comparison of computation cost of the proposed scheme and reviewed scheme is presented in this sub-section as shown in Table 3. The initiation phase of reviewed scheme is $2T_{grg} + 1T_H + 1T_{Sym}$ and that of our proposed scheme is $1T_{pa} + 1T_H$. The reviewed scheme has no server registration because it is single server scheme but our scheme server registration is $1T_{Sym} + 1T_{ECC} + 1T_H$ because it is a multi-server scheme. The user registration phase of the reviewed scheme is $1T_{Sym} + 1T_{exp} + 1T_H$ and that of our proposed scheme is $1T_{Sym} + 1T_{ECC} + 2T_H$. The login phase of the reviewed scheme is $1T_{Sym} + 2T_{exp} + 1T_H$ and our proposed scheme is $4T_H$. The authentication phase of the reviewed scheme is $2T_{Sym} + 1T_{exp} + 4T_H$ and our proposed scheme is $1T_{Sym} + 9T_H$. And lastly the change password phase of the reviewed scheme is $1T_{Sym} + 1T_{exp}$ and our proposed scheme is $1T_{Sym} + 3T_H + 1T_{ECC}$. This can be shown in Table 3.

TABLE 3
Computation Cost Comparison

Scheme Phases	**Buhari and Obiniyi [7] Scheme**	**Proposed Scheme**
Initialization Phase	$2T_{grg} + 1T_H + 1T_{Sym}$	$1T_{pa} + 1T_H$
Server Registration Phase	–	$1T_{Sym} + 1T_{ECC} + 1T_H$
User Registration Phase	$1T_{Sym} + 1T_{exp} + 1T_H$	$1T_{Sym} + 1T_{ECC} + 2T_H$
Login Phase	$1T_{Sym} + 2T_{exp} + 1T_H$	$4T_H$
Authentication Phase	$2T_{Sym} + 1T_{exp} + 4T_H$	$1T_{Sym} + 9T_H$
Change Password Phase	$1T_{Sym} + 1T_{exp}$	$1T_{Sym} + 3T_H + 1T_{ECC}$
Total	$2T_{grg} + 5T_{exp} + 6T_{Sym} + 7T_H$	$4T_{Sym} + 20T_H + 4T_{ECC}$

Comparison of computation time of the proposed scheme and reviewed scheme is presented in this subsection as shown in Table 4. The initiation phase of reviewed scheme is 1.1056s and that of our proposed scheme is 0.0518s. The reviewed scheme has no server registration because it is single server scheme but our scheme server registration is 2.2536s because it is a multi-server scheme. The user registration phase of the reviewed scheme is 3.8776s and that of our proposed scheme is 2.2766s. The login phase of the reviewed scheme is 7.2278s and our proposed scheme is 0.092s. The authentication phase of the reviewed scheme is 3.9512s and our proposed scheme is 0.2116s. And lastly the change password phase of the reviewed scheme is 3.8546s and our proposed scheme is 2.2996s. This can be shown in Table 4 and Figure 16.

TABLE 4
Computation Time Comparison

Scheme Phases	**Buhari and Obiniyi [7] Scheme**	**Proposed Scheme**
Initialization Phase	1.1056	0.0518
Server Registration Phase	0	2.2536
User Registration Phase	3.8776	2.2766
Login Phase	7.2278	0.092
Authentication Phase	3.9512	0.2116
Change Password Phase	3.8546	2.2996
Total	**20.0168**	**7.1852**

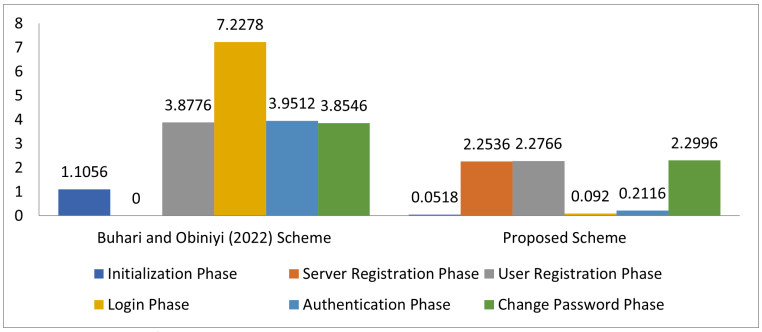


Fig. 16. Phases Computation Cost Comparison

The reviewed scheme has a total computation time of 20.0168s and the proposed scheme a total computation cost of 7.1852s as shown in Figure 17. This shown that the proposed scheme is more efficient with 64.10% improvement.

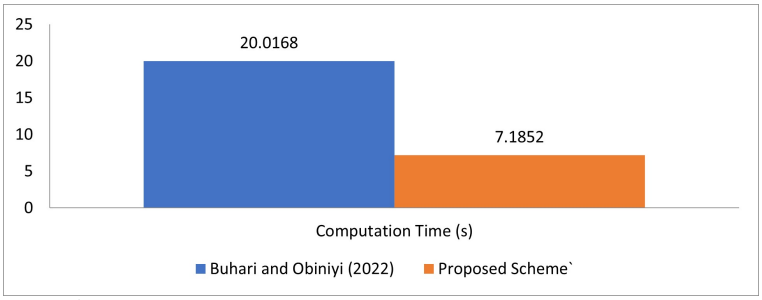


Fig. 17. Computation Cost Comparison

IX. CONCLUSION

A web application is a computer program that runs on a server and is accessed through a browser and may be used on desktop, laptop, tablet, or mobile devices. The resulted development and deployment of different identification, authentication and authorization methods on web applications are because high demand on web services.

Also, the single-server authentication method employed by the reviewed research is more difficult to use, especially for users who have to register separately with each server. Furthermore, the multi-registration problem of many servers may be resolved by employing a multi-server user authentication technique as employed by our

proposed scheme.

This study offered an effective and safe anonymous multi-server web application login authentication technique to address the shortcomings of the Buhari and Obiniyi [7] scheme while also utilizing its advantages. The suggested scheme is anonymous and multi-server. To make our proposed technique more efficient, we replace El-Gamal encryption with elliptic curve cryptography, which has smaller key sizes than previous algorithms, shorter private and public keys, faster computing costs, and reduced memory and bandwidth needs. The security of the proposed scheme was verified, validated, and evaluated against the reviewed scheme in terms security and performance using AVISPA tool, cryptanalysis, time complexity analysis and performance analysis. The outcome of the AVISPA simulation demonstrated that the proposed scheme is validated, efficient, secure from replay attacks, and compatible with multi-server web applications. Additionally, compared to the evaluated scheme, the performance evaluation of the proposed scheme shown a considerable improvement in computing cost of 64.10%.

DECLARATIONS

AVAILABILITY OF DATA AND MATERIAL

Not applicable

FUNDING

Not applicable

ACKNOWLEDGMENTS

Not applicable

REFERENCES

[1] A. Bucko, K. Vishi, B. Krasniqi, and B. Rexha, "Enhancing jwt authentication and authorization in web applications based on user behavior history," *Computers*, vol. 12, no. 4, p. 78, 2023.

[2] M. Dul, Ł. Gugała, and Ł. A. B. A. Kamil, "Protecting web applications from authentication attacks," *Advances in Web Development Journal*, vol. 1, no. 1, 2023.

[3] D. He, "Robust biometric-based user authentication scheme for wireless sensor networks," *Cryptology ePrint Archive*, 2012.

[4] S. Zhou, Q. Gan, and X. Wang, "Authentication scheme based on smart card in multi-server environment," *Wireless Networks*, vol. 26, no. 2, pp. 855–863, 2020.

[5] K. A. Kamiński, A. P. Dobrowolski, Z. Piotrowski, and P. Ścibiorek, "Enhancing web application security: Advanced biometric voice verification for two-factor authentication," *Electronics*, vol. 12, no. 18, p. 3791, 2023.

[6] A. R. Iyanda and E. F. Mayokun, "Development of two-factor authentication login system using dynamic password with sms verification," *International Journal of Education and Management Engineering*, vol. 12, no. 3, p. 13, 2022.

[7] B. A. Buhari and A. A. Obiniyi, "Web applications login authentication scheme using hybrid cryptography with user anonymity," *Int. J. Inf. Eng. Electron. Bus. (IJIEEB)*, vol. 14, no. 5, pp. 42–50, 2022.

[8] R. F. Olanrewaju, B. U. I. Khan, M. A. Morshidi, F. Anwar, and M. L. B. M. Kiah, "A frictionless and secure user authentication in web-based premium applications," *IEEE Access*, vol. 9, pp. 129 240–129 255, 2021.

[9] A. Irshad and M. Alreshoodi, "Sems-5g: A secure and efficient multi-server authentication scheme for 5g networks," *IEEE Access*, 2024.

[10] F. Wu, X. Li, L. Xu, A. K. Sangaiah, and J. J. Rodrigues, "Authentication protocol for distributed cloud computing: An explanation of the security situations for internet-of-things-enabled devices," *IEEE Consumer Electronics Magazine*, vol. 7, no. 6, pp. 38–44, 2018.

[11] A. Petcu, B. Pahontu, M. Frunzete, and D. A. Stoichescu, "A secure and decentralized authentication mechanism based on web 3.0 and ethereum blockchain technology," *Applied Sciences*, vol. 13, no. 4, p. 2231, 2023.

[12] S. V. Akram, S. K. Joshi, and R. Deorari, "Web application based authentication system," in *2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC)*. IEEE, 2022, pp. 1439–1443.

[13] D. E. Kurniawan, M. Iqbal, J. Friadi, F. Hidayat, and R. D. Permatasari, "Login security using one time password (otp) application with encryption algorithm performance," in *Journal of Physics: Conference Series*, vol. 1783, no. 1. IOP Publishing, 2021, p. 012041.

[14] B. O. ALSaleem and A. I. Alshoshan, "Multi-factor authentication to systems login," in *2021 National Computing Colleges Conference (NCCC)*. IEEE, 2021, pp. 1–4.

[15] I. T. E. J. Mohammed and Y. A. Mohamed, "A new system for user authentication using android application," in *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*. IEEE, 2021, pp. 1–5.

[16] M. Tanrıverdi, "Design and implementation of blockchain based single sign-on authentication system for web applications," *Sakarya University Journal of Computer and Information Sciences*, vol. 3, no. 3, pp. 343–354, 2020.

[17] J. Solano, L. Camacho, A. Correa, C. Deiro, J. Vargas, and M. Ochoa, "Combining behavioral biometrics and session context analytics to enhance risk-based static authentication in web applications," *International Journal of Information Security*, vol. 20, no. 2, pp. 181–197, 2021.

[18] N. Christou and E. Athanasopoulos, "auth. js: Advanced authentication for the web," in *Emerging Technologies for Authorization and Authentication: Third International Workshop, ETAA 2020, Guildford, UK, September 18, 2020, Proceedings 3*. Springer International Publishing, 2020, pp. 33–49.

[19] M. C. I. Putri, P. Sukarno, and A. A. Wardana, "Two factor authentication framework based on ethereum blockchain with dapp as token generation system instead of third-party on web application," *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 6, no. 2, pp. 74–85, 2020.

[20] X. Chu, H. Sun, and Z. Chen, "Passpage: graphical password authentication scheme based on web browsing records," in *Financial Cryptography and Data Security: FC 2020 International Workshops, AsiaUSEC, CoDeFi, VOTING, and WTSC, Kota Kinabalu, Malaysia, February 14, 2020, Revised Selected Papers 24*. Springer International Publishing, 2020, pp. 166–176.

[21] S. J. Mohammed and S. A. Mehdi, "Web application authentication using zkp and novel 6d chaotic system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 3, pp. 1522–1529, 2020.

[22] S. Boonkrong, "Internet banking login with multi-factor authentication," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 11, no. 1, pp. 511–535, 2017.

[23] M. Xie, Y. Li, K. Yoshigoe, R. Seker, and J. Bian, "Camauth: securing web authentication with camera," in *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*. IEEE, 2015, pp. 232–239.