International Journal of Advanced Technology and Engineering Exploration, Vol 11(119)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2024.111100516

Research Article

An approach to cloud user access control using behavioral biometric-based authentication and continuous monitoring

A. Riyaz Fathima[1*] and A. Saravanan[2]
Research Scholar, Department of Computer Science, Sree Saraswathi Thyagaraja College, Pollachi, Coimbatore Tamil Nadu, India[1]
Associate Professor, Department of Computer Science, Sree Saraswathi Thyagaraja College, Pollachi, Coimbatore Tamil Nadu, India[2]

Received: 11-April-2024; Revised: 15-October-2024; Accepted: 17-October-2024

Abstract
Cloud computing, enabling remote access to services and resources, poses a critical challenge in user authentication and access control, as users can access resources from anywhere with an internet connection. Traditional authentication methods, such as passwords and tokens, are vulnerable to attacks like brute-force, phishing, and man-in-the-middle (MITM). Researchers are exploring biometric authentication methods, but security and privacy concerns arise due to cloud environment control and potential data breaches or theft. To address these concerns, a comprehensive multifactor authentication (MFA) framework was proposed with an authorization scheme to enhance data security in a cloud environment. The proposed methodology comprises three phases: user registration, login, and continuous authentication. During the registration phase, users provide significant data, resulting in the assignment of a unique 6-digit personal identification number (PIN) upon successful registration. In the login process, authentication is achieved using a combination of static (primary user credentials), dynamic (color-based physical action verification), and possession factors (one-time password). Additionally, a trust score is calculated based on the evaluation of inherence factors (IFs), including user and typing behavior, to assign access control. The continuous authentication phase involves the use of a secure PIN for critical operations, evaluation of risk values, and reauthentication requests when necessary. The proposed model demonstrated superior performance, achieving 99.4% robustness, 99.7% accuracy, and a 0.3% error rate on a closed dataset, and 99.8% robustness, 99.8% accuracy, and a 0.2% error rate on an open dataset. The model's effectiveness was further demonstrated by its ability to prevent unauthorized access and mitigate security risks through the use of behavioral biometrics and access control strategies. The proposed MFA effectively addressed security concerns in cloud systems. It offered valuable benefits to cloud service providers and end users by enhancing data security and mitigating potential threats.

Keywords
Multifactor authentication, Behavioral biometrics, Access control, Authorization, Continuous authentication, Cloud users.

1.Introduction
Computing on the cloud, coupled with big data, artificial intelligence (AI), and other related technologies, completely transforms how businesses store and manage their data. Cloud computing also provides potential advantages like flexibility, scalability, and accessibility, which are made possible by the growing availability of the internet [1]. However, cloud storage data is vulnerable to unauthorized access, data breaches, and cyberattacks, all of which can compromise its authenticity and privacy.

Additionally, the proliferation of such security breaches has increased with the adoption of the Internet and the rising use of technology [2]. As businesses become excessively dependent on cloud services for the management of sensitive information, it becomes paramount to protect the data stored in the cloud from unauthorized access. Authentication is the primary security barrier, and if an attacker can breach this wall, the security system becomes ineffective [3]. This applies not only to cloud servers but also to most applications. Credential-based authentication is the primary method used in numerous applications [4]. According to the findings of the survey, more than 30% of users utilize the same password (PWD)

*Author for correspondence

across various applications [5]. An intruder can quickly gain access to the system, data, and resources related to data after they have stolen a user account PWD. Accounts with elevated permissions are more vulnerable to this security risk. Moreover, traditional authentication techniques, including PWD-based systems, are vulnerable to security flaws like credential theft, phishing, and PWD guessing [6]. The need for additional security in user identity verification has given rise to multifactor authentication (MFA), which aims to address the shortcomings of single-factor authentication methods [7]. To increase security and reduce unauthorized entry, these approaches demand users enter multiple forms of authentication, such as biometric data, bot verification, and one-time passwords (OTPs).

While MFA has the potential to increase security, its ineffective and poorly designed implementation might make it difficult for both users and service providers. Furthermore, since these methods are limited to protecting the security walls, even with a successful approach, the data is only secured from unauthorized outside access [8]. Nevertheless, even with the additional protection, it is not possible to protect the data and the system from attacks that originate within the system. As a result, protecting sensitive data even after a user has successfully authenticated into a system has become of paramount significance due to the growing dependence on cloud services for critical operations [9]. The limitations of existing MFA frameworks have been evident in various studies. For instance, while MFA can mitigate risks associated with PWD vulnerabilities, it does not fully address the issue of insider threats or attacks that exploit weaknesses in the authentication process itself [10]. Additionally, many MFA implementations rely heavily on short message service (SMS)-based verification, which is susceptible to spoofing attacks, thereby compromising the intended security benefits [11].

Recent studies have highlighted the need for continuous user authentication to address the limitations of traditional MFA methods. Rayani and Changder (2023) conducted a survey on continuous user authentication on smartphones using behavioral biometrics, emphasizing the importance of ongoing monitoring to enhance security [12]. Finnegan et al. (2024) reviewed the utility of behavioral biometrics in user authentication and demographic characteristic detection, suggesting that these techniques can provide a more comprehensive security solution compared to traditional methods [13]. A survey on quantitative risk estimation approaches for secure and usable user authentication on smartphones highlighted the need for adaptive and context-aware authentication mechanisms [14]. Moreover, a survey highlighted the difficulties in establishing strong authentication in decentralized settings in the comprehensive evaluation of cloud infrastructure with multiple factor authentication [15]. Thus, cloud service providers must implement robust authentication procedures to protect user data and prevent unauthorized access amid a surge in cyber threats and insider data breaches [10, 16]. This research is motivated by the need to address these gaps in current MFA frameworks. As cyber threats become more sophisticated and pervasive, it is crucial to develop an authentication system that not only strengthens initial verification but also provides ongoing protection.

The increasing reliance on cloud services for critical operations requires a solution that integrates strong authentication with continuous monitoring to maintain complete security. While several frameworks for authentication have been developed and proven to be effective in many aspects, they frequently fail to offer complete protection against constantly emerging cyber threats and require continuous authentication upon accessing critical data, even after successful authentication. The limitations of existing frameworks highlight the need for a more adaptive and resilient approach to user authentication, particularly in the context of cloud computing, where both external and internal threats pose significant risks [8, 11].

This study has been motivated by the need to address these research gaps in current MFA frameworks by developing and deploying a system that enhances cloud data security through the integration of advanced authentication methods, continuous authentication procedures, and access control mechanisms. The primary objective of this study is to tackle the research challenge of developing a robust authentication framework to secure data in a cloud environment. The proposed MFA framework involves registration, login, and continuous authentication phases, specifically designed for cloud users. To improve data security in cloud environments, the method employs users' trust for access control and risk evaluation for continuous authentication during critical cloud data operations. Thus, based on MFA and behavioral biometrics literature, the proposed framework addresses security

threats, protects sensitive data, and improves cloud user security, usability, and scalability.

In order to achieve the identified research objectives, this study makes several key contributions:

- Proposes a novel MFA framework integrating advanced authentication methods with behavior analysis.
- Implement a continuous authentication procedure that balances security and usability.
- Analyze the users' and their typing behaviors to determine user trust and enable access control.
- Provides a comprehensive evaluation of the model's performance and effectiveness in protecting cloud-stored data.

The paper is structured as follows: Section 2 presents a comprehensive discussion of the studies related to the proposed work. Section 3 details the proposed authentication model along with the detailed architecture. The various phases of the proposed architecture, including the registration phase, user login phase, and continuous authentication phase, are explained in subsequent subsections. The results, including comparative and performance analysis, along with cryptanalysis, are explained in section 4. Section 5 discusses the study's findings and limitations. Finally, the paper concludes the proposed work with suggestions for future enhancement.

2.Literature review

The work associated with the proposed research study on authentication models is covered in this section. Studies emphasized that several elements, including security, privacy, and usability, must be properly considered while developing user authentication [17]. Based on a review of the available literature, it was clear that no authentication model was able to meet all three of these criteria [18]. For instance, authentication schemes such as PWDs, personal identification numbers (PINs), or unique tokens have higher usability, whereas schemes such as possession-based authentication improve privacy. On the other hand, biometric-based authentication and encryption-based communication enhance security [19].

2.1Password-based authentication

Historically, several security weaknesses in authentication systems prompted the development of PWD-authenticated key agreements [20] and enhanced identity-based two-party authentication key exchange mechanisms [21, 22]. These methods may still be vulnerable to attacks if users choose weak PWDs or fail to implement additional security measures, such as rate limiting or account lockouts. Additionally, some of these methods employ data randomization to make it difficult for attackers. To protect mobile client-server applications, elliptic curve cryptography (ECC) is frequently used. The use of ECC as a solution for the authentication and security of the system is becoming increasingly common in spite of its computational complexity. ECC is often used for establishing a mutual authentication system between mobile devices and the cloud [23], authenticating users using a lightweight and anonymous method [24], and securing patients' privacy [25, 26]. However, these methods are susceptible to attacks involving stolen devices and known-key security attacks. This indicates that even with strong encryption, if a device is compromised, the authentication system may be breached. Moreover, although ECC is effective, its reliance on secure key management practices is crucial; any compromise in key management can lead to vulnerabilities. Furthermore, the anonymity methods used may not be suitable for all applications, especially those requiring user accountability.

By utilizing a two-factor dynamic identifier, an effective technique for authenticated key exchange was suggested, yet the method is vulnerable to various advanced attacks [27]. Similar to encryption, hashing techniques are also commonly employed. Chang and Le (2015) employed a one-way hash function for effectively authenticating users [28]. While one-way hash functions improve security, they are still vulnerable to attacks such as rainbow table attacks if users select weak PWDs or if the hash function is not sufficiently complex. Thus, these models are unable to resist masquerade and spoofing attacks, indicating the need for additional layers of security, such as behavioral biometrics or continuous monitoring, to enhance authentication reliability. PWD-based authentication for web-based graphics computing service retrieval in the cloud was explored [29]. While this method offers simplicity and user familiarity, it is vulnerable to phishing and brute-force attacks, necessitating additional security measures to safeguard sensitive data. Additionally, PWD-based systems are susceptible to phishing and brute-force attacks and may not offer adequate security when used alone.

2.2Behavioral biometric authentication

In addition to encryption and hashing, behavioral analysis has now become a reliable technique for user authentication in cloud environments [30]. This

behavioral analysis includes the typing behavior of the user by extracting keystroke dynamics for effective user authentication [31]. While behavioral analysis using keystroke dynamics is a promising technique for user authentication in cloud environments, these methods may be susceptible to impersonation attacks if an attacker can closely mimic the user's typing patterns. Additionally, factors such as user fatigue, stress, or device changes can affect the consistency of keystroke dynamics, leading to more false positives (FP) or false negatives (FN) than true positives (TPs) and true negatives (TNs). Ahmadi et al. (2021) proposed a model employing various biometrics as input for effective authentication [32]. This model integrates a person's behavioral, physiological, or both biometric features, defining process steps at each level, thereby enhancing verification and security in fog computing through various biometric features. However, the implementation of this multi-modal biometric system may be complex and resource-intensive, especially in resource-constrained fog environments. Additionally, the model does not address the issue of continuous authentication.

Generally, login credentials are often used at application entry points for traditional authentication rather than being used continuously throughout the application. Uslu et al. (2023) propose behavioral biometrics for continuous authentication using deep learning algorithms for binary classification, with multilayer perceptron and convolutional long short-term memory (LSTM) algorithms showing superior performance [33]. In spite of providing higher accuracy, the implementation of this multi-modal biometric system may be complex and resource-intensive, especially in resource-constrained fog environments. Additionally, the model does not address the issue of continuous authentication. A hybrid verification method that makes use of encryption and biometric technologies was developed [34]. This study utilized fingerprints as a biometric technique and advanced encryption standards (AES) to enhance the security of cloud computing authentication, ensuring a secure and robust method. However, fingerprint biometrics may not be suitable for all users, and the system's performance can be affected by environmental factors such as dirt or moisture on the sensor. Additionally, the study does not address the issue of continuous authentication.

A crypto-biometric cloud computing system was proposed that employs the optimization blowfish algorithm (OBA) by Uddin et al. (2023) without disclosing any personal biometric information [35]. While this approach ensures privacy, it may introduce computational overhead and latency, especially in scenarios with a large number of users. The study does not provide a comprehensive evaluation of the system's performance and scalability in real-world cloud deployments. Through continuous authentication, Yao et al. (2020) proposed a dynamic access control and user authentication method that evaluates trust scores based on behavioral analysis [8]. Similar to the trust score, a risk score was also used to authenticate or reauthenticate the user based on behavioral biometrics before accessing critical data [36]. However, these models have not been thoroughly evaluated in large-scale cloud environments, and their effectiveness in detecting sophisticated attacks or adapting to changes in user behavior over time remains unclear.

2.3Multifactor authentication (MFA)

A study proposed multi-level recognition and OBA to solve data security issues in cloud storage [37]. The process involves various multistage authentication, data security, and data recovery, using OBA encryption and binary crow search for security. The method lacks extensive experimental analysis to prove its reliability, and the effectiveness of the OBA in real-world applications remains untested. Additionally, the complexity of the algorithm may lead to performance issues in resource-constrained environments. A novel secure two-factor authentication scheme was proposed that integrates traditional user IDs, PWDs, and OTP verification into a single authentication model similar to the one proposed by Kaur et al. (2022) [38], Anitha and Jayarekha (2021) [39]. While the authentication model was proven to be resistant to brute force, session hijacking, man-in-the-middle (MITM), and replay attacks, it may still be vulnerable to phishing attacks and social engineering tactics. The reliance on OTPs also introduces potential vulnerabilities if the OTP delivery mechanism is compromised.

Mostafa et al. (2023) proposed a multi-factor, multi-layer authentication framework to enhance cloud security by incorporating access control, intrusion detection, and automated authentication methods [40]. Though it uses AES– based encryption to protect login information from disclosure, with user factors, geolocation, and browser confirmation contributing to its improvement, it fails to provide continuous authentication. Moreover, the complexity of the multi-layer approach may lead to user

frustration and decreased usability. In remote and interoperable contexts, a new approach to data integrity and authentication was proposed [41]. The study is suitable only for examining security models in large distributed environments, suggesting a private virtual network for transit security and data encryption and integrity algorithms for user identity protection. However, it does not address the challenges of implementing such a model in diverse cloud environments or the potential performance overhead.

A secure, lightweight protocol was proposed based on temporary identity and a cumulative keyed-hash chain [42]. While the method is suitable for IoT-smart home environments, its applicability to broader cloud computing scenarios is limited. The protocol may also face challenges in terms of scalability and adaptability to varying network conditions. A secure cryptosystem using improved identity-based encryption with multimodal biometric authentication and authorization in cloud environments was proposed [43]. This approach enhances security by integrating multiple biometric modalities, but it may introduce implementation complexity and require significant computational resources. A novel three-factor authentication (3FA) framework for secure medical big data transmission over the cloud was introduced [44]. This framework enhances security by combining three authentication factors, but it may face challenges in user adoption and implementation complexity in diverse healthcare settings. The summary of notable studies from recent literature is summarized in *Table 1*.

Table 1 Summary of notable literature

Author(s) (Year)	Focus	Factors Used	Results	Advantages	Limitations
Alshahrani and Traore (2019) [42]	Secure mutual authentication for IoT smart homes	Cumulative keyed-hash chain	Developed a lightweight protocol for IoT environments.	Provides a secure method for IoT devices with low resource consumption.	Limited scalability and adaptability to broader cloud computing scenarios.
Jan and Qayum (2020) [30]	Behavioral analysis for user authentication	Typing behavior, keystroke dynamics	Established behavioral analysis as a reliable technique for user authentication.	Enhances security by using unique user behavior patterns for authentication.	Susceptible to impersonation attacks; affected by user variability.
Yao et al. (2020) [8]	Dynamic access control and user authentication	Trust scores based on behavioral analysis	Evaluated user trust scores for authentication and access control.	Provides a flexible approach to access control based on user behavior.	The effectiveness of detecting advanced attacks in real-world scenarios is not thoroughly evaluated.
Buriro et al. (2021) [36]	Risk-driven behavioral biometric-based authentication scheme	Behavioral biometrics	Used risk scores for user authentication before accessing critical data.	Offers a dynamic method of authentication based on risk assessment.	Lacks a thorough evaluation and may struggle to adapt effectively to changes in user behavior.
Kaur et al. (2022) [38]	Secure two-factor authentication scheme	User IDs, PWDs, OTP verification	Proved resistant to brute force, session hijacking, MITM, and replay attacks.	Combines multiple authentication factors for increased security.	Vulnerable to phishing and social engineering tactics; relies on OTP delivery mechanisms.
Hossain and Al (2022) [34]	Hybrid verification method combining encryption and biometrics	Fingerprints, AES encryption	Improved security of cloud computing authentication through a hybrid approach.	Enhances security by combining biometric and encryption methods.	Fingerprint biometrics may not be suitable for all users; performance affected by environmental factors.
Saravanan and Bama (2023) [31]	Keystroke dynamics for effective user authentication	Keystroke dynamics	Demonstrated effective extraction of typing behavior for authentication.	Provides a non-intrusive method for continuous user authentication.	Performance can be affected by user fatigue and environmental factors.
Uslu et al. (2023)	Continuous	Deep learning	Achieved superior	Utilizes advanced	Requires extensive

Author(s) (Year)	Focus	Factors Used	Results	Advantages	Limitations
[33]	authentication using behavioral biometrics	algorithms, LSTM	performance in binary classification for continuous authentication.	algorithms for improved accuracy in user verification.	training data and computational resources; may struggle with adaptability.
Durga et al. (2023) [37]	Multi-level recognition and OBA	OBA encryption, binary crow search	Addressed data security issues in cloud storage through a multi-stage authentication process.	Provides a structured approach to enhance data security in cloud environments.	Lacks extensive experimental analysis to prove reliability.
Mostafa et al. (2023) [40]	Multi-factor, multi-layer authentication framework	Access control, intrusion detection	Enhanced cloud security through a comprehensive authentication framework.	Integrates multiple security measures for a robust authentication process.	Fails to provide continuous authentication; complexity may frustrate users.
Sarkar and Roychowdhury (2023) [11]	Authentication, authorization, and security	Various authentication methods	Discussed various aspects of cloud computing, including security risks and potential solutions.	Identifies key challenges and provides security solutions for cloud computing environments.	Lacks empirical evidence on the effectiveness of proposed solutions across all applications.
Arumugam (2023) [45]	Hybrid encryption model utilizing biometric keys	Fingerprint-based biometric key, AES, Elgamal Encryption using ECC	Developed a hybrid encryption model that combines symmetric and asymmetric encryption methods.	Provides enhanced data security through the use of a unique biometric key along with traditional methods.	Lacks comprehensive security analysis against various attacks; high processing time for encryption and decryption.
Arasan et al. (2024) [46]	Anonymous authentication scheme for cloud users	ECC, Bilinear pairing	Efficient scheme providing strong security and user anonymity	Suitable for resource-constrained IoT and preserves privacy	Lacks formal security analysis and scalability not investigated
Konwar et al. (2024) [47]	A Two-Factor Authentication with Novel OTP Generation	OTP authentication	Introduced a novel OTP generation algorithm to enhance cloud application security.	Increases security through dynamic, time-sensitive authentication.	Reliance on OTP delivery mechanisms; potential vulnerabilities in the OTP generation process.
Aburbeian and Fernández-veiga (2024) [48]	Integration of MFA and Machine Learning	MFA, Machine Learning	Proposed a framework that combines MFA with machine learning for secure financial transactions.	Enhances security by adapting to user behavior and detecting anomalies.	Requires extensive training data for machine learning models; may face challenges in real-time implementation.

The literature review discusses various cloud authentication methods, including PWD-based, biometric, behavioral biometric, and MFA, while highlighting their lack of continuous authentication or comprehensive solutions. PWD-based schemes offer high usability but are susceptible to various attacks. Biometric authentication enhances security but raises privacy concerns. Behavioral biometrics analyze user patterns for authentication, but current methods lack continuous monitoring. Multi-factor schemes combine multiple techniques but primarily focus on initial login rather than ongoing verification. The proposed research aims to address these limitations by developing a novel approach to cloud user access control using behavioral biometric-based authentication and continuous monitoring. By integrating continuous authentication with behavioral biometrics, the proposed solution seeks to provide a more robust and user-friendly alternative to existing authentication models in cloud environments.

3.Methods

The proposed novel MFA framework was discussed in this section. This framework involves three main phases: the user registration phase, the user login phase, and the continuous authentication phase. In the registration phase, the user registers their username (UID), PWD, email ID (EID), and registered mobile number (RMN) with the cloud server. During this phase, the server stores all the required details about the user, along with behavioral profiling such as login behavior, device verification, usage behavior, and behavioral biometrics like keystroke dynamics. *Figure 1* displays the simple block diagram for the proposed model.

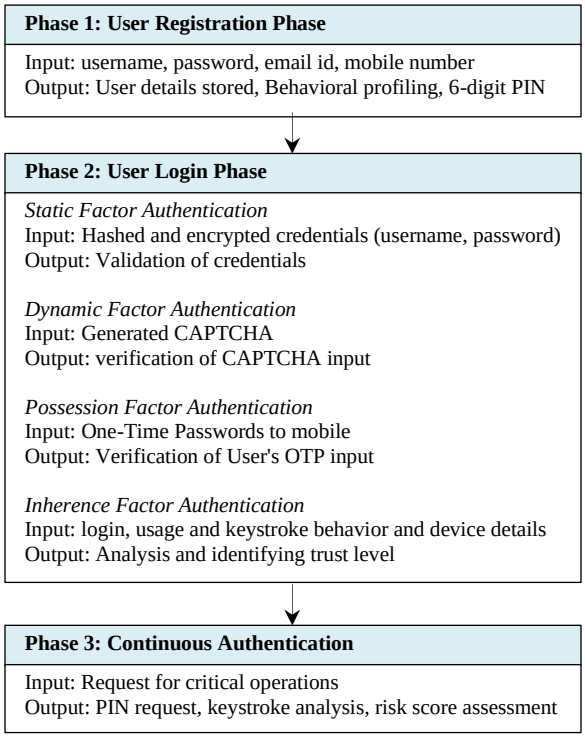

Figure 1 Block diagram of the proposed authentication framework

Upon successful registration, the cloud server generates and sends a unique 6-digit PIN to the user. In the user login phase, the user is authenticated using a static factor (SF) such as the UID and PWD, a dynamic factor (DF) involving a color-based physical action verification (PAV) system, a possession factor (PF) like an OTP sent to the RMN, and an inherence factor (IF) like behavioral biometrics involving typing behavior. By assessing the behavioral profiling, the server computes the trust factor, based on which access control is provided to the user. To improve data security in the cloud environment, the model also includes continuous authentication, where the server demands the PIN during critical operations on the cloud data that the user is authorized to access. The server computes the keystroke analysis and assesses the risk value. If the risk is high, the server requests the user to reauthenticate before performing critical operations. *Figure 2* illustrates the detailed description of the proposed authentication model. This study extends the concept of 3FA developed by [31].

3.1User registration phase

During the registration phase, the user's information is collected and saved in a user profile database. The process begins when the user sends a registration request to the server. If the request is successful, the server responds by sending the registration page to the user. If the request fails, the server sends an error message. This interaction ensures that the user's registration details are processed and handled appropriately based on the success or failure of the initial request. The user's personal identity information, such as UID, PWD, RMN, and EID, is submitted by the user. Any system needs this set of information to provide personalized services, ensure account security, maintain user communication, and comply with legal and regulatory requirements regarding user data protection.

Additionally, the PWD is hashed using the Argon2 hashing algorithm to offer an additional layer of security [49]. This step is essential for protecting PWDs against data breaches. Since Argon2 is a powerful and contemporary PWD hashing technology, it enhances PWD security. The hashing process incorporates random data through an integrated salt generator, rendering it resistant to several types of attacks, such as brute-force, rainbow, and dictionary attacks. The hashed PWD, along with other submitted user information, is encrypted before being sent to the server. By applying encryption, sensitive data is secured during network transmission, making the registration process more secure. AES symmetric key cryptography is applied in this work, as it has proven effective and has widespread support [40]. Specifically, AES-256 in Galois/counter mode (GCM) is used. GCM is chosen for its efficiency and performance in providing both data confidentiality and integrity.

Moreover, a key management service (KMS) is used to securely manage encryption keys, ensuring compliance with security policies and minimizing the risk of key compromise. The transport layer security

(TLS) protocol enhances data transmission by encrypting data end-to-end, preventing eavesdropping, tampering, and message forgery. After being encrypted with AES-256-GCM, the data is transferred to the server. Upon receipt, the server decrypts the data using the corresponding AES decryption key. The decrypted information is then saved in the profile database, ensuring secure storage of user information.

The server then generates and sends the OTP to the RMN for verification. A time-based technique is used to generate the OTP, applying the hash-based message authentication code using secure hash algorithm (HMAC-SHA1), a cryptographic hash function to the current time, and a shared secret key. To obtain a 6-digit PIN that is shorter, the output is further truncated. Upon entering the OTP, the server records the user's behavior such as login behavior, device details, and keystroke behavior. Next, the server produces a six-digit PIN and sends it to the user securely after verifying that the received OTP matches. Algorithm 1 presents the pseudocode for the user registration phase. Section 3.2.3 provides the detailed procedure for OTP generation. *Figure 3* illustrates the user registration procedure with a sequence diagram for better understanding.

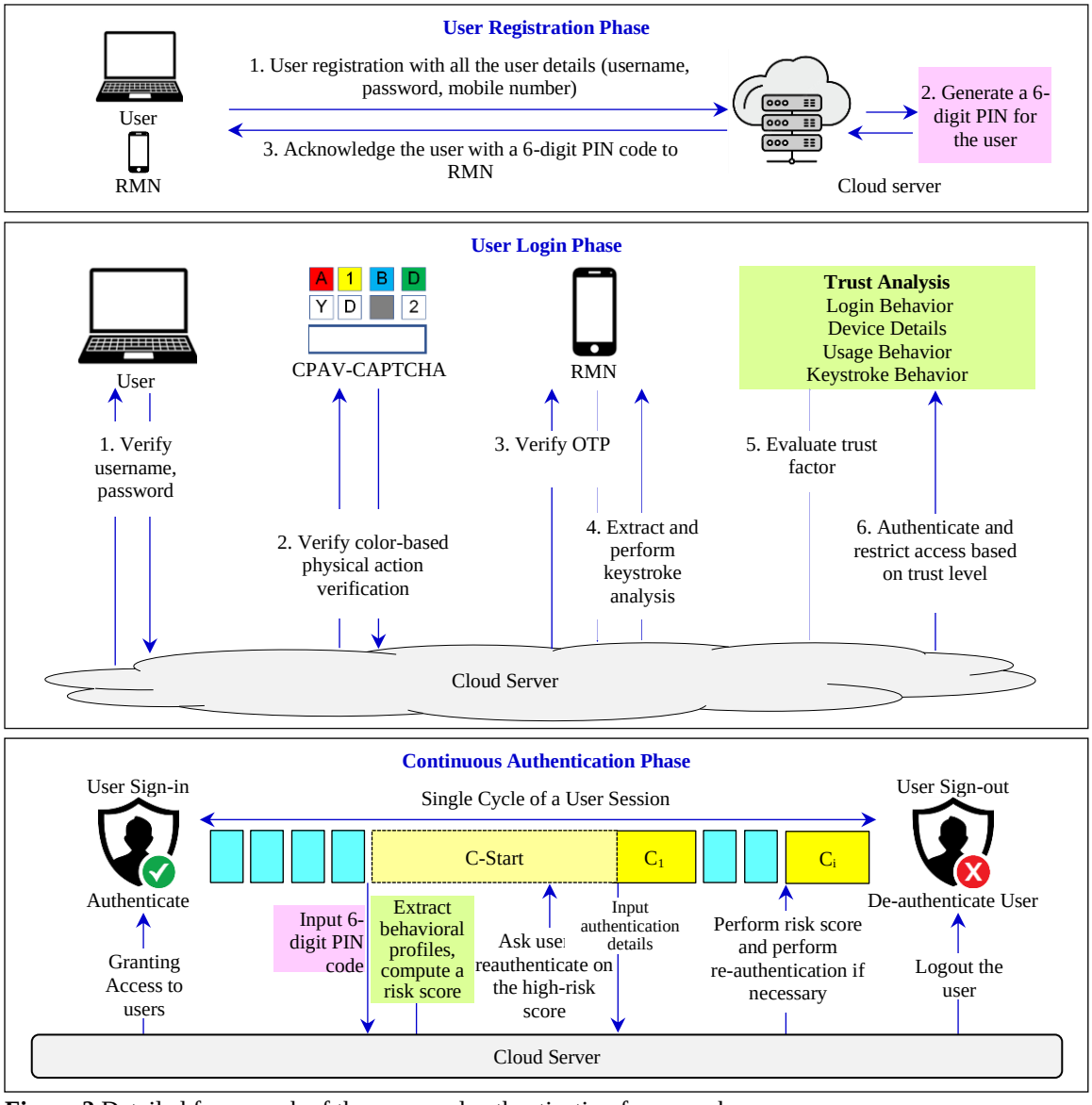

Figure 2 Detailed framework of the proposed authentication framework

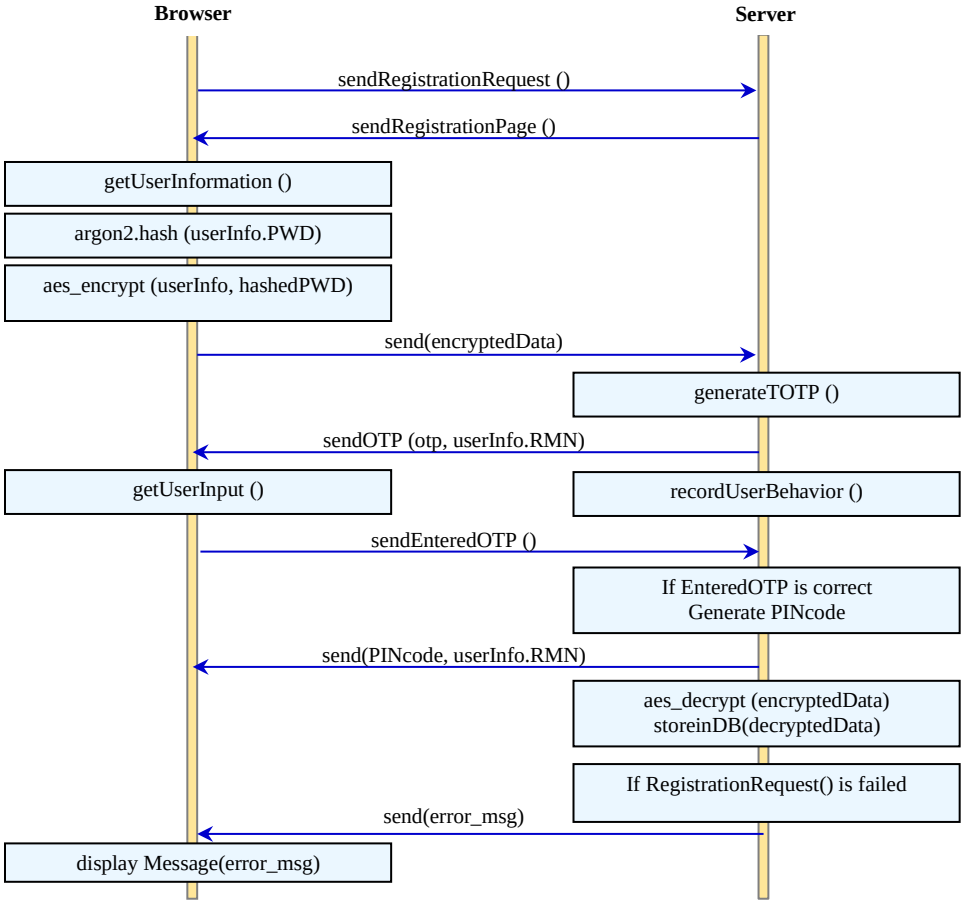

Figure 3 Sequence diagram of proposed user registration phase

Algorithm 1: Pseudocode for user registration

Input: web page
Output: Register the user profile
Procedure userRegistration()
Begin
```
  // Initialize variables
   requestPage = null; registrationPage = null; userInfo = null; hashedPWD = null;
   encryptedData = null; otp = null; enteredOTP = null; random_pin = 0;
   PINcode = 0; decryptedData = null;
   //User requests registration from the server
   requestPage = sendRegistrationRequest()
   If request failed then
      // Log the error message and notify the user
      logError(errorMessage)
      displayErrorMessage(errorMessage)
   Else
      registrationPage = sendRegistrationPage(requestPage)
      userInfo = getUserInformation(registrationPage)
      //Apply Argon2 for hashing technique
      hashedPWD = argon2.hash(userInfo.PWD)
      //encrypt using AES algorithm
      encryptedData = aes.encrypt(userInfo, hashedPWD)
      send(encryptedData) // send the data to the server
      // Generate a time-based OTP
      otp = generateTOTP(secretKey, interval)
```

```
        sendOTP(otp, userInfo.RMN)
        enteredOTP = getUserInput()
        // Record user behavior (login behavior, device details)
        recordUserBehavior(loginBehavior, devicedetails, keystrokeBehavior)
        // Verify the OTP
        If(enteredOTP==otp) then
           random_pin = select_in_range(100000, 999999)
           while isDuplicate(random_pin) do
                random_pin = select_in_range(100000, 999999)
           PINcode = random_pin //Generate a 6-digit PIN code
           //Send the PIN code to the RMN
           send(PINcode, userInfo.RMN)
           // Decrypt the received encrypted data
           decryptedData = aes.decrypt (encryptedData)
           storeinDB(decryptedData)
        End If
    End If
End Procedure
```

3.2Multifactor user authentication phase

Each time a user attempts to log in, an authentication step is performed to verify their identity. If the login request is successful, the server responds by returning the login page to collect additional user information needed for authentication. If the login request fails, the server displays an error message to the user. On the login screen, information such as the UID and PWD is entered by the user. Here, the cloud server implements an MFA system during the user login procedure. Initially, it assesses only the knowledge component, which comprises conventional verification of the login and PWD. This approach is static. Second, a color-based PAV system is used by the knowledge approach for dynamic verification. Utilizing the PF, the third step sends the RMN together with the OTP, and authentication is performed using the entered OTP. Subsequently, the method records the user's actions, such as device information, usage patterns, typing patterns, and login behaviors. Finally, behavior profiling is used to assess the trust factor, which is typically used to authenticate users and grant access control-based permissions or restrict users' access to critical operations. In this case, the user's authorization is granted within a predefined range of threshold values based on biometric behavior.

3.2.1Primary credentials-based authentication

One of the most popular methods of authorizing users is static authentication, which requires them to provide their registered UID and PWD. The user's profile information, in addition to their UID and PWD, is always required throughout the registration process. Hashing the PWD during the login process is done to make it secure during transit. Furthermore, to protect user credentials during transit, they undergo a cryptographic procedure. More specifically, the proposed scheme utilizes Argon2 hashing techniques and the AES encryption algorithm to enhance security. First, the user PWD is hashed using Argon2, and then AES encryption is applied to the UID and hashed PWD, respectively. The Argon2 hashing technique includes six parameters: PWD, salt, memory cost, time cost, parallelism factor, and hash length. These parameters determine the algorithm's memory use, execution time, number of repetitions, and overall performance. Although it takes more time, Argon2 is effective for offline key derivation since it is a memory-hard PWD hashing algorithm. Data is secured from being revealed by utilizing the AES encryption technique. Upon successful static authentication, the binary value 1 is returned. After the user logs in, the server receives encrypted ciphertext that contains the user's UID, PWD, and any other information gathered during the login process. The server then decrypts the ciphertext using AES decryption. A comparison is made between the hashed PWD retrieved from the decrypted data and the value saved in the database. If a match is found, the remaining authentication steps are carried out.

3.2.2Color-based PAV system

The second common authentication factor is to distinguish humans from bots. In the proposed model, a color-based PAV system is employed. When the user attempts to perform the login form submission, the server generates a series of colored shapes embedding letters and incomplete captcha characters. The user is required to physically interact with the system by clicking the specified color, which then fills in the missing character of the captcha. By requesting the response to a color recognition challenge, this layer augments the verification process with an extra layer. Selecting the appropriate color completes the captcha character slot and

enables the captcha entry field by prompting the user to input the complete captcha sequence. Then the system verifies the user's input with the correct captcha code. If the user's response matches the correct code, they are given the next authentication stage. On the other hand, if there is a mismatch, the user is prompted to try again with another challenge. Here, the user is given a limited number of chances for the alternative challenge, say three times. *Figure 4* illustrates the method. At this point, the user is prompted to click the red box, and after they do so, the character in the box labeled "a" fills into the missing slot of the captcha sequence, which enables the user to enter the captcha. When the user enters the captcha code as 'YDa2', they are taken to the next authentication step. On the other hand, if the user clicks the wrong color, say the yellow box, the captcha is filled with '1' and the captcha code becomes 'YD12'. The user is presented with an additional choice if the value provided does not match the initial captcha, resulting in unsuccessful authentication.

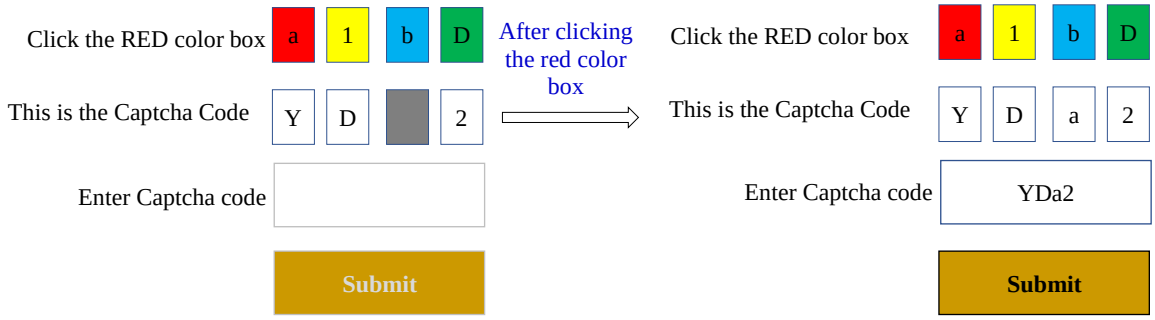

Figure 4 Proposed color-based PAV system

Through user validation, this approach reduces the risk of fraudulent or automated spam submissions and unauthorized access by strengthening security measures. At the same time, while enhancing security and ensuring validity, the interactive aspect of the user experience provides an enjoyable and engaging challenge for the user.

3.2.3Dynamic PF - based authentication

This method uses a PF, which refers to the device the user possesses and uses to verify their identity. The global reach and dominance of mobile phones in human life has led to increased usage of authentication for effective identification among users and organizations. The user is authenticated using an OTP token instead of relying only on a given PWD for stored details verification. Thus, as the next step of authentication, the proposed model utilizes OTP, a unique PWD generated once for each communication. Although effective, it should not be used as primary authentication but as part of MFA. Generally, unique, temporary, and secret tokens are generated and sent to users' mobile phones as a challenge. And because this secret token is only valid for one login session, users need to enter it immediately for authentication. Once the user enters an OTP, which is verified with the generated OTP, if a match is found, the user moves to the next level of authentication. In the meantime, the server retrieves the user's OTP keystrokes and stores them in the database when they enter the system. The various parameters include dwell time (time between keystrokes), flight time (time between releasing one key and pressing the next), key hold time, key release time, and typing speed.

There are several methods for generating OTPs, including random numbers and HMAC-based OTPs. However, the proposed model used time-based OTPs, a widely used algorithm that utilizes the current time along with the shared secret key and interval as input parameters for generating OTPs. The parameter interval specifies the time duration for which the OTP remains valid. Further, the method applies HMAC-SHA1, a cryptographic hash function, to these parameters. This approach provides sufficient security with little computing cost, in contrast to other cryptographic hash functions, by maintaining a balance between security and efficiency. Typically, the generated hash is rather lengthy; thus, it is truncated to produce a shorter numerical code, usually a 6-digit number that functions as an OTP. Because of its adaptability, security, scalability, and compatibility, this authentication strategy is ideal for use with online accounts. Algorithm 2 presents the pseudocode for generating time-based OTP.

Algorithm 2. Time-based OTP generation

Input: secretKey shared between server and user device, interval
Output: Generated OTP
Procedure generateTOTP(secretKey, interval)
Begin
// Initialize variables
current_time = 0; counter = 0; counter_bytes = null; hmacHash = null; offset = 0;
truncatedHash = null; otpValue = 0; otpString = "";
// getting the current time in seconds
current_time = getCurrentTimestamp()
// calculating counter value
counter = current_time/interval
// converting to a byte array
Counter_bytes = convert_bytearray(counter)
hmacHash = calculateHMACSHA1(secretKey, counterBytes) //hashing the values
//Extracting last 4 bytes
offset = LSB_4bytes(hmacHash)
truncatedHash = extract(hmacHash, offset)
//Applying modulo 10^6 to get a 6-digit OTP
otpValue = convert(truncatedHash)
//Converting the OTP value to a 6-digit string
otpString = format(otpValue)
Return otpString
End Procedure

3.2.4Behavioral analysis-based user trust computation

This step involves computing the user's trust level to provide access control or authorization based on the user's behavior. Behavioral profiling attempts to create a distinct behavioral profile for every user by tracking various behavior-related data points, which may then be used for identity verification and authentication. Utilizing the user's behavioral biometrics, the authorization for data access is modified accordingly. Generally, attackers' behaviors deviate significantly from typical user behavior, even if they manage to breach the user's authentication mechanism. Therefore, these deviations may be evaluated by extracting different user behaviors that are identified as attributes and comparing them with past user behavior by computing scores using attributes. Higher-degree deviations make it possible to identify abnormal behavior and present the user with restrictions on accessing the data. An overview of various user behavioral data is presented in *Table 2*. Each dimension of the user's behavioral data represents a different degree of significance.

Table 2 Overview of various user behavioral data

Behavioral dimensions	Attribute/ Specific Behavior	Details
Login Behavior	Login method	The method used to log in to the system (PWD-based or social login)
	Login time	The timestamp of the login attempt
	Login duration	The duration of the login session
	Geographic location	Geographic origin of the login attempt
Device Details	Device used	Information about the device used to log in
	IP address	The network address from which the login attempt originated
	Browser	Browser used to log in
Operational Behavior	Login attempts	Number of logins attempts by a user over time
	Frequency of interaction	Frequency of the user's interactions with the system over time
	Typical usage patterns	Identifies user interaction patterns, including the sequence of actions, features accessed, and the user's navigation paths.
Typing Behavior	Dwell time	The time interval between the key press and the key release
	Flight time	The time it takes for the user to move from one key to the next while typing
	Typing speed	The maximum number of characters a user can type within a specific timeframe

Behavioral dimensions	Attribute/ Specific Behavior	Details
	Error Rate	The frequency of typing errors, such as mistyped characters, within a specific timeframe

The trust computation is based on the ideas proposed by Buriro et al. (2021) [36] and Yao et al. (2020) [8]. Each aspect of the user's behavioral data is accompanied by historical data that falls within a specific time frame. Consider that the user is logging in to session *N*, and the historical data covers all the previous attribute data from *N-1* sessions with *K* attributes. The user's trust is then computed for the *M* categorical attributes by comparing each attribute value (*AV*) of the current session with the previous *N* sessions based on their frequency of occurrence. This is evaluated by computing the attribute trust AT_i as in Equation 1.

$$AT_i^c = \frac{\sum_{j=1}^{N-1}(AV_{ij}=AV_{iN})}{N-1} \; \forall \; i \in \; M \; cat.attr. \quad (1)$$

However, in the case of numerical attributes, instead of matching the number of occurrences, the attribute deviation (*AD*) is computed for the current AV with the previous AVs from *N-1* sessions, as in Equation 2. Here, μ_i and σ_i are the mean and standard deviation of the AVs for *N-1* sessions.

$$AD_i^n = \sum_{j=1}^{N-1} \frac{(AV_{iN}-\mu_i)}{\sigma_i} \; \forall \; i \in \; P \; num. \; att. \quad (2)$$

Then the trust score of the numerical attributes is computed by determining the threshold values as in Equation 3. The first threshold is set at the point that is exactly halfway between the mean and the maximum or minimum value, and the second threshold is set at the point that is exactly the maximum or minimum value.

$$AT_i^n = \begin{cases} 0.0 \; if \; \frac{\mu_i \pm Min_i}{2} < AD_i < \frac{\mu_i \pm Max_i}{2} \\ 0.5 \; if \; Min_i < AD_i < Max_i \\ 1.0 \; if \; Max_i < AD_i < Min_i \end{cases} \quad (3)$$

The value of attribute trust 0 indicates lower trust, and 1 indicates higher trust. For effective analysis, weights are assigned to the attribute dimensions. Finally, the trust score is computed by evaluating the weighted average of the attribute trust as shown in Equation 4.

$$Trust \; Score = \frac{\sum_{i=1}^{m} AT_i \times W_i}{\sum_{i=1}^{m} W_i} \quad (4)$$

Higher values indicate that the user has higher trust and lower risk. By computing the trust score, users are provided with different permissions to access data, which are determined by their trust degree. In the proposed approach, the range of trust values is divided into three distinct levels, each representing a different degree of trustworthiness. Consequently, users are categorized into these levels based on their calculated trust values that fall within the predefined thresholds. The trust threshold for critical operations is presented in *Table 3*. Users with a trust score below 0.3 can access the data but cannot download or modify it. At the medium trust level, users are permitted to view or download the data. However, high-trust users are granted full access to all critical operations.

The SF encrypted UID and PWD using Argon2 hashing and PINs, providing strong defense against brute-force and guessing attacks. Color-based PAV in DF prevented automated attacks and differentiated humans from bots. PF secured against replay and phishing, and IF, using behavioral analysis, detected anomalies and resisted impersonation. Despite these multiple authentication measures, the user's trust level is determined by behavioral analysis, which assesses various behavioral attributes—such as typing patterns, login behaviors, operational behaviors, and device usage—and compares them with historical data. The behavioral trust score reflects how consistently and reliably the user behaves compared to past interactions. The rationale behind using behavioral analysis in trust score computation is that it adapts to changes in user behavior, enhancing security by providing dynamic, real-time trust assessments. An overview of the procedure for MFA and trust-based user access control is shown in Algorithm 3. Furthermore, the sequence diagram for the user authentication phase is provided in *Figure 5.*

Table 3 Access control based on the trust value

Trust level	Trust score range	Allowed operations
Low	0.0 – 0.3	Only View
Medium	0.3 – 0.7	Only View or Download
High	0.7 – 1.0	View, Download, Add, Modify or Delete

Algorithm 3: Multifactor User Authentication

```
Procedure authenticateUser():
  // Initialize variables
   requestPage = null; errorMessage = ""; loginPage = null; username = ""; password = ""; \
   hashedPWD = ""; encryptedData = ""; DynCaptcha = ""; userDynCaptcha = "";
   OTP = ""; userOTP = ""; trustScore = 0.0
   isCorrectCAPTCHA = false //User requests login page from the server
   requestPage = sendLoginRequest()
   If request failed then
      // Log the error message and notify the user
      displayErrorMessage(errorMessage)
   Else
      loginPage = sendLoginPage(requestPage)
      // Step 1: Static Authentication (Uname and Pwd)
      // get user's credentials
      username, password = getUserCredentials()
      //Apply Argon2 for hashing technique
      hashedPWD = argon2.hash(password)
      //encrypt using AES algorithm
      encryptedData = aes.encrypt(username, hashedPWD)
      send(encryptedData) // send the data to the server
      // verifies whether the credentials are valid
      If isvalid(username, hashedPWD) then
     // Step 2: Dynamic Authentication (Color-based
       CAPTCHA)
        //generates color based CAPTCHA
        DynCaptch=generateColorCAPTCHA()
        //sends color-based CAPTCHA to the user
        sendToUser(DynCaptch)
        //User performs CAPTCHA
        userDynCaptch=performColorCAPTCHA(DynCaptch)
        // send color-based CAPTCHA to the server
        send(userDynCaptch)
        // verifies the received CAPTCHA
        isCorrectCAPTCHA = receivedCaPTCHA()
        //verifies whether the CAPTCH is correct
        If isCorrectCAPTCHA then
          // Step 3: Possession Factor Authentication
          //Generate time based OTP
          OTP = generateOTP()
          //send the OTP to the user's RMN
          sendOTPToRMN(OTP)
          //Records user typing and other behaviors
          recordUserBehavior()
          send(userOTP) //Send OTP to the server
          // verifies whether the OTP is correct
          If isCorrectOTP(userOTP) then
            // Step 4: Compute User Trust Level
            trustScore = computeTrustScore()
           //Computes trust score as in section 3.2.4
           // Step 5: Determine acc. ctrl based TS
            If trustScore >= 0.7 then grantAccess()
            Else if trustScore >= 0.3 then
                 restrictAccess()
            Else return noAccess()
End Procedure
```

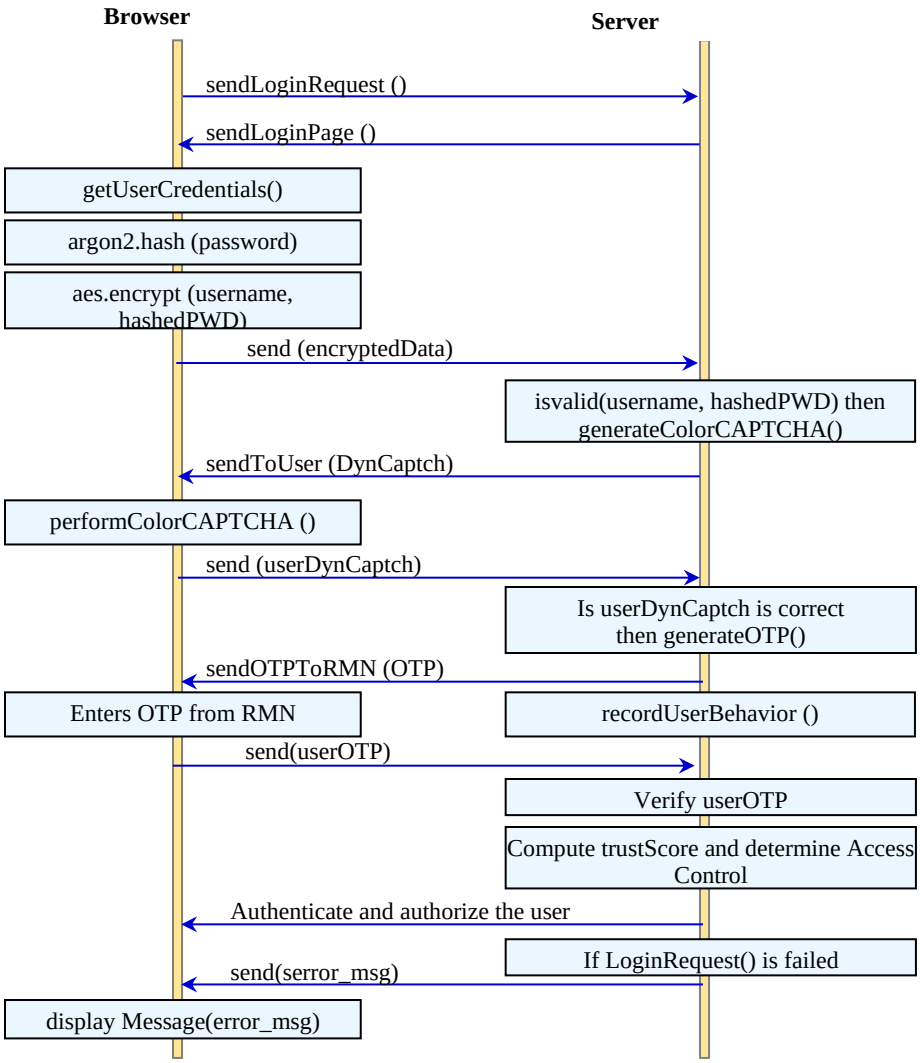

Figure 5 Sequence diagram of proposed MFA

3.3Risk analysis-based continuous authentication phase

The first two phases are static procedures that occur when the user first registers with the cloud application or each time they log in to the system. However, even after the user has been authenticated by the system, the data still needs to be secured against any unusual behavior. Thus, a dynamic authentication procedure is required to secure the data while it is being operated on. This phase intends to perform continuous user authentication during critical operations in the cloud data, even after a successful login. Whenever the user tries to perform critical operations during their login session, the server requires them to enter the 6-digit secret PIN given to them during registration. While the user inputs the PIN, the server fetches the typing behavior, other login behaviors, device details, and operational behaviors of the user. With these extracted behavioral AVs, the risk score for the authenticated user during their critical operations is calculated. Here, the risk score is computed by summing up the uncertainty related to each of the user behavioral factors listed in *Table 3*. There are varying degrees of importance for each user behavior attribute through which the risk can be assessed.

This phase is continuous and executes dynamically and periodically when the user tries to access or modify the data during their login session. The computation of the risk score is similar to the mathematical formulas used in trust score computation, assessing the present value and the previously recorded values. While the trust score computes the frequency of occurrence of AVs in the past *N-1* sessions, the risk score computes the

frequency of non-occurrence in the past *N-1* sessions. It is feasible to detect increased risk and request user reauthentication before executing crucial data operations when higher degree deviations are detected.

The risk computation is based on the concepts proposed by [8, 36]. Every component of the user's behavioral data has historical data associated with it for a certain period. Assume that the user is logged into session *N* and that all of the prior attribute data from *N−1* sessions with *K* attributes is covered by the historical data. Next, depending on the frequency of occurrence, the risk associated with the user for the *M* categorical attributes is calculated by comparing the *AV* of the present session with those of the previous *N* sessions. This is evaluated by computing the attribute risk AR_i as shown in Equation 5.

$$AR_i^c = \frac{\sum_{j=1}^{N-1}(AV_{ij} \neq AV_{iN})}{N-1} \; \forall \; i \in \; M \; cat. \; att. \quad (5)$$

However, in the case of numerical attributes, instead of matching the number of non-occurrences, the *AD* is computed for the current session's *AV* compared to the previous sessions' AVs from *N−1* sessions, as shown in Equation 6. Here, μ_i and σ_i are the mean and standard deviation of the AVs for the previous *N-1* sessions.

$$AD_i^n = \sum_{j=1}^{N-1} \frac{(AV_{iN} - \mu_i)}{\sigma_i} \; \forall \; i \in \; P \; num. att. \quad (6)$$

Next, the threshold values for the numerical attributes are determined, as shown in Equation 7, to compute the risk score. The first threshold is established at the precise midpoint between the mean and the highest or lowest value, while the second threshold is set at the exact point of the greatest or lowest value.

$$AR_i^n = \begin{cases} 0.0 \; if \; \frac{\mu_i \pm Min_i}{2} < AD_i < \frac{\mu_i \pm Max_i}{2} \\ 0.5 \; if \; Min_i < AD_i < Max_i \\ 1.0 \; if \; Max_i < AD_i < Min_i \end{cases} \quad (7)$$

High risk is indicated by a value of 1 for attribute risk, while low risk is indicated by a value of 0. Weights are then applied to the attribute dimensions to enable effective analysis. Ultimately, the risk score is calculated using the weighted average of the attribute risk, as shown in Equation 8.

$$Risk \; Score = \frac{\sum_{i=1}^{m} AR_i \times W_i}{\sum_{i=1}^{m} W_i} \quad (8)$$

Higher values suggest that the user is exposed to greater risk. To authorize users to execute crucial operations on the data, it is necessary to forecast their behavior during these operations using the risk score and then request reauthentication from them. The computed risk score is compared with the threshold value before allowing the user to perform critical operations. If the risk score, calculated by comparing the current session's *AV* with the previous *N−1* sessions' AVs, is higher than the predefined threshold, then reauthentication, as presented in Section 3.2, is required. The proposed risk analysis-based continuous user authentication is outlined in algorithm 4.

Algorithm 4: Risk Analysis-based Continuous Authentication

```
Procedure continuousAuthentication()
  If criticalOperation() then
    // Prompt user to enter their 6-digit PIN
    pin = getUserPIN()
    //Record user typing, login, device behavior
    record UserBehavior()
    send(pin) //send PIN to the server
    //Verifies whether the PIN is correct
    If isValidPIN(pin) then
      //If the PIN is correct, compute the risk score
      riskScore = computeRiskScore()
      //Compare with the risk threshold
      If riskScore > riskThreshold then
          // prompt reauthentication for higher score
          request Reauthentication()
      //All access for lower risk score
      Else allowAccess()
      End IF
End Procedure
```

Thus, the proposed authentication scheme utilizes the concept of MFA, leveraging credentials, an OTP, a color-based PAV, and behavioral analysis-based trust score computation to authenticate the user and provide access control or restrictions for accessing the data. Furthermore, it includes continuous authentication mechanisms by dynamically computing the risk associated with the user through behavioral analysis. The proposed system features robust error handling mechanisms for smooth operation, generating detailed error logs for analysis and debugging upon error occurrences. It is designed to handle various failure scenarios, such as incorrect user input, server downtime, and network issues. Automated alerts are sent to system administrators in the event of a system failure, allowing for immediate resolution.

4.Results

4.1Experimental setup

An extensive experimental study was conducted to illustrate the performance of the proposed MFA framework with continuous authentication and behavioral biometric-based access control employing a web application. Using the Apache hypertext transfer protocol (HTTP) server as a framework, the web application utilizes MySQL for database management, Python for server-side programming, and JavaScript for client-side programming. The simulation was performed on an Intel Core i3-4005U x64 central processing unit (CPU) with a clock speed of 1.70 GHz and 4 GB of RAM on a Windows 8 machine.

The efficacy of the proposed methodology was assessed by creating a dataset that enables authentic users to register their accounts, stores their hashed PWDs and PINs, and extracts user behavioral attributes. When a user logs in, information is retrieved about their device, location, IP address, typing speed, error rate, dwell time, flight duration, and mode of login. Additionally, further information such as login duration, frequency of interaction, and typical usage patterns is gathered throughout their session and during logout. The experimental analysis was performed for the proposed method, and the results were compared with the 3FA reported by Saravanan and Bama (2023) [31].

The created web application has 30 registered users, with records generated through MFA credentials from genuine users during the registration phase. Following the lead of Mondal and Bours (2016) [50], the proposed approach uses both closed and open-set records to evaluate the model's efficacy, involving 30 registered users and 10 unknown users. The login process in a closed set is restricted to registered users, who act as both genuine users and impostors, while in an open set, it includes both registered and unknown users. The sample dataset is provided in *Table 4*. Here, the variable *UserID* serves as the unique identifier for each user. The *UID* is chosen by the user during registration, and the PWD is stored as a hashed version using Argon2. The *EID* and *RMN* are the email address and RMN provided by the user. *TypingSpeed* (ms) represents the average typing speed recorded in milliseconds, and *LoginAttempts* indicates the number of login attempts made by the user. The variables *IsGenuine* and *IsImpostor* indicate whether the user is a genuine registered user or if the user is acting as an impostor.

Table 4 Sample records in the dataset

UserID	UID	HashPWD	EID	RMN	TypingSpeed (ms)	LoginAttempts	IsGenuine	IsImpostor
001	user1	$argon2id...	user1@mail.com	1234567890	250	1	Yes	No
002	user2	$argon2id...	user2@mail.com	1234567891	300	2	Yes	No
003	user3	$argon2id...	user3@mail.com	1234567892	270	4	Yes	No
...	...	...	...	...	...	...	...	...
030	user30	$argon2id...	user30@mail.com	1234567829	260	3	Yes	No
031	unknown1	$argon2id...	-	-	280	2	No	Yes
...	...	...	...	...	...	...	...	...
040	unknown10	$argon2id...	-	-	275	3	No	Yes

Python is used by the server program to decrypt and compare credentials with the database data, while JavaScript, along with jQuery, is used by the client software to hash PWDs, obtain typing behaviors, and encrypt credentials. For the color-based PAV, the server generates a random combination of colors and shapes. The generation process involves the following steps:

- *Color Selection:* The server selects colors from a predefined palette, ensuring high contrast and distinctiveness. This palette is designed to be accessible to most users, including those with

color blindness, by incorporating color patterns or text labels alongside the colors.

- *Shape Selection:* Shapes are chosen from a set of basic geometric figures, such as circles, squares, and triangles. The selection is random but ensures that each shape is paired with a distinct color to facilitate easy identification and verification by users.
- *Combination Display:* The server then combines these selected colors and shapes into a visual challenge presented to the user during the login process. Users must perform specific physical actions based on the color-shape combinations to complete the authentication.

Table 5 presents the various features utilized in the proposed model, and *Table 6* presents the sample records involving various behavioral dimensions.

Table 5 Features involved in the proposed model

Authentication type	Feature dimensions	# Features
SF	Primary Credentials	2
	PIN verification	1
DF	Color-based PAV	2
PF	OTP	1
IF	Login Behavior	4
	Device Details	2
	Operational Behavior	3
	Typing Behavior	4

Table 6 Sample records of IF

Behavioral dimensions	Features	Record1	Record2	Record3	Record4	Record5
User ID	-	user01	user02	user03	user04	user05
Login Behavior	Method	PWD -based	PWD-based	PWD-based	PWD-based	PWD -based
	Time	7/24/2023 8:15	7/24/2023 9:30	7/24/2023 10:45	7/24/2023 11:00	7/25/2023 12:15
	Duration	2 hours	1 hour	30 minutes	3 hours	20 minutes
Device Details	Device	Laptop, Windows 10	Smartphone, Android	Desktop, Windows 10	Laptop, Windows 10	Desktop, Windows 10
	IP Address	192.168.1.1	172.16.0.2	10.0.0.1	192.168.100.1	10.0.1.2
	Browser	Chrome	Chrome	Chrome	Edge	Edge
Operational Behavior	Attempts	3	1	5	2	4
	Frequency	Daily	Weekly	Daily	Monthly	Daily
	Usage	Regular document editing	Infrequent operations	Attempted unauthorized access	Regular file management	Attempted unauthorized access
Typing Behavior	Dwell Time	100 ms	120 ms	150 ms	110 ms	140 ms
	Flight Time	80 ms	90 ms	110 ms	100 ms	130 ms
	Speed	60 WPM	50 WPM	55 WPM	65 WPM	45 WPM
	Error Rate	2%	3%	10%	1%	5%

4.2Performance indicators

To evaluate the performance of the authentication models, several key metrics, such as accuracy, error rate, robustness, and efficiency, are utilized. These metrics provide a comprehensive view of the model's effectiveness and reliability. Each metric is defined and calculated based on the following components [51]:

- TP: The number of genuine users correctly authenticated by the system.
- TN: The number of imposters correctly identified as unauthorized by the system.
- FP: The number of imposters incorrectly authenticated as genuine users.
- FN: The number of genuine users incorrectly classified as imposters.

Accuracy measures the proportion of correctly classified samples out of the total, indicating the model's overall correctness, as given in Equation 9. Higher accuracy reflects better performance, and a value close to 100% signifies high effectiveness.

$$Accuracy = \frac{TP+TN}{N} \quad (9)$$

Error rate represents the percentage of incorrectly classified samples, highlighting the model's error rate, as shown in Equation 10. A lower error rate indicates better performance and is crucial for the

reliability of the authentication system.

$$Error\ Rate = \frac{FP+FN}{N} \quad (10)$$

Robustness is defined as the proportion of correctly identified impostors (TN) relative to the total number of impostors (TN plus FP), as shown in Equation 11. This metric assesses the model's ability to effectively reject unauthorized users, thereby enhancing system security.

$$Robustness = \frac{TN}{FP+TN} \quad (11)$$

Efficiency measures the proportion of genuine users correctly identified (TP) compared to the total number of genuine users (TP plus FN), as shown in Equation 12. It indicates the model's ability to authenticate legitimate users accurately while minimizing incorrect rejections.

$$Effeciency = \frac{TP}{TP+FN} \quad (12)$$

4.3Comparative analysis

The comparison was made by analyzing the individual authentication factors and the proposed MFA model, similar to the study conducted by Saravanan and Bama (2023) [31]. The closed set allows users to log in to the web application at different times and for a maximum of five attempts. The closed-set experiment involved registered users acting as both genuine users and impostors, with the experiment evaluated by varying the number of genuine users and impostors in three sets: maximum, equal, and minimum. The results obtained for various authentication models, such as SF, DF, PF, and IF, and their various combinations along with 3FA [31] are presented in *Table 7*. The values include three different input sets by varying the number of genuine users and impostors from the closed set and were analyzed for various authentication models.

Table 7 Performance analysis with closed dataset

Authentication model	#Samples (#Genuinue, #Imposter)	Error rate	Accuracy	Robustness	Efficiency
SF	200 (150,50)	6.00	94.00	76.00	100.00
	200 (100,100)	10.00	90.00	80.00	100.00
	200 (50,150)	10.00	90.00	86.67	100.00
	Avg. value	*8.70*	*91.30*	*80.90*	*100.00*
SF+DF	200 (150,50)	5.00	95.00	80.00	100.00
	200 (100,100)	8.00	92.00	84.00	100.00
	200 (50,150)	9.50	90.50	87.33	100.00
	Avg. value	*7.50*	*92.50*	*83.80*	*100.00*
SF+DF+PF	200 (150,50)	2.00	98.00	92.00	100.00
	200 (100,100)	3.00	97.00	94.00	100.00
	200 (50,150)	5.00	95.00	93.33	100.00
	Avg. value	*3.30*	*96.70*	*93.10*	*100.00*
SF+IF	200 (150,50)	5.00	95.00	90.00	96.67
	200 (100,100)	7.50	92.50	94.00	91.00
	200 (50,150)	7.00	93.00	92.00	96.00
	Avg. value	*6.50*	*93.50*	*92.00*	*94.60*
SF+DF+IF	200 (150,50)	4.50	95.50	94.00	96.00
	200 (100,100)	6.50	93.50	95.00	92.00
	200 (50,150)	5.00	95.00	94.67	96.00
	Avg. value	*5.30*	*94.70*	*94.60*	*94.70*
3FA	200 (150,50)	4.50	95.50	96.00	95.33
	200 (100,100)	3.00	97.00	96.00	98.00
	200 (50,150)	3.50	96.50	96.00	98.00
	Avg. value	*3.70*	*96.30*	*96.00*	*97.10*
Proposed	200 (150,50)	0.00	100.00	100.00	100.00
	200 (100,100)	0.50	99.50	99.00	100.00
	200 (50,150)	0.50	99.50	99.33	100.00
	Avg. value	*0.30*	*99.70*	*99.40*	*100.00*

From the obtained values, it was clear that the performance of the authentication system improved whenever PF and IF were included with SF and DF authentication. The authentication system, initially using only SF and DF, achieved an average accuracy of 92.5% and robustness of 83.8%. When including PF, the accuracy improved to 96.7% and the robustness to 93.1%. Further inclusion of IF with SF

and DF resulted in enhanced accuracy of 94.7% and robustness of 94.6%. DF mainly helped to authenticate humans from bots, whereas PF and IF provided an additional security layer. Although the IF variables that involve login, usage, and typing behaviors initially produced inaccurate results, after a few logins, they could identify patterns and provided precise authentication. Thus, the proposed model offered an average of 99.4% robustness, 100% effectiveness, 99.7% accuracy, and a 0.3% error rate.

The experiments were also performed with open sets involving registered users as genuine users and unregistered users as impostors. Similarly, the model was evaluated by assessing different authentication models by varying the number of genuine users in three sets: maximum, equal, and minimum. The average performance evaluation of error rate, accuracy, robustness, and efficiency obtained for SF, DF, PF, and IF and their various combinations, along with existing 3FA model [31], for the proportion of open datasets is presented in *Figure 6*. The obtained results indicated that the performance of the authentication system improved with PF and IF in the open set. After several login sessions, the suggested model enhanced performance by allowing for the identification of user behaviors and providing reliable authentication. Moreover, the proposed model offered 99.8% robustness, 100% effectiveness, and 99.8% accuracy with a 0.2% error rate.

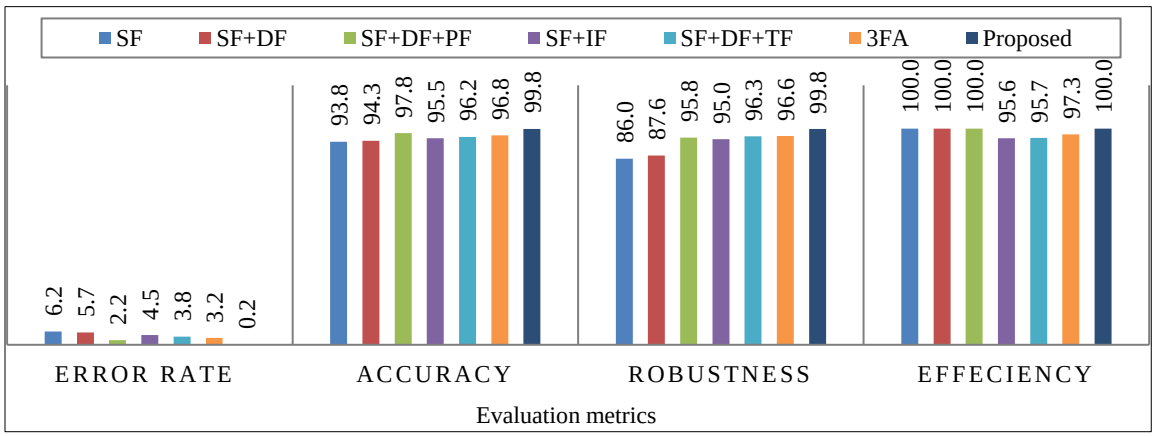

Figure 6 Average performance analysis with open dataset

4.4Performance analysis

Other important factors to consider while assessing the performance of the authentication system are computational and communication overheads. Furthermore, the performance of the proposed model was compared with other existing authentication models using evaluation metrics such as computation cost and communication cost.

4.4.1Computational overhead

Generally, the time required to perform an authentication procedure is represented by its computation cost. This encompasses both the moment the user requests the login page and the moment the server authenticates the user. The computational complexity of the proposed model, represented by O(n), where n is the number of authentication steps, was compared with other state-of-the-art authentication models. The average time taken for five genuine authentications was used for the comparison. Thus, for a complete authentication procedure, the proposed MFA phase involved an average time of 4.048 milliseconds. The various state-of-the-art authentication models used for comparison were identity-based authentication [21], improved two-factor authentication [25], dynamic ID-based anonymous scheme [27], remote user authentication [26], anonymous user authentication [23], lightweight anonymous authentication [24], and CloudSec(3FA) [31]. *Figure 7* displays the computational overhead of different state-of-the-art and proposed authentication models. From the analysis, it was clear that the computational overhead of the proposed model was better than that of many existing authentication models, demonstrating its superiority.

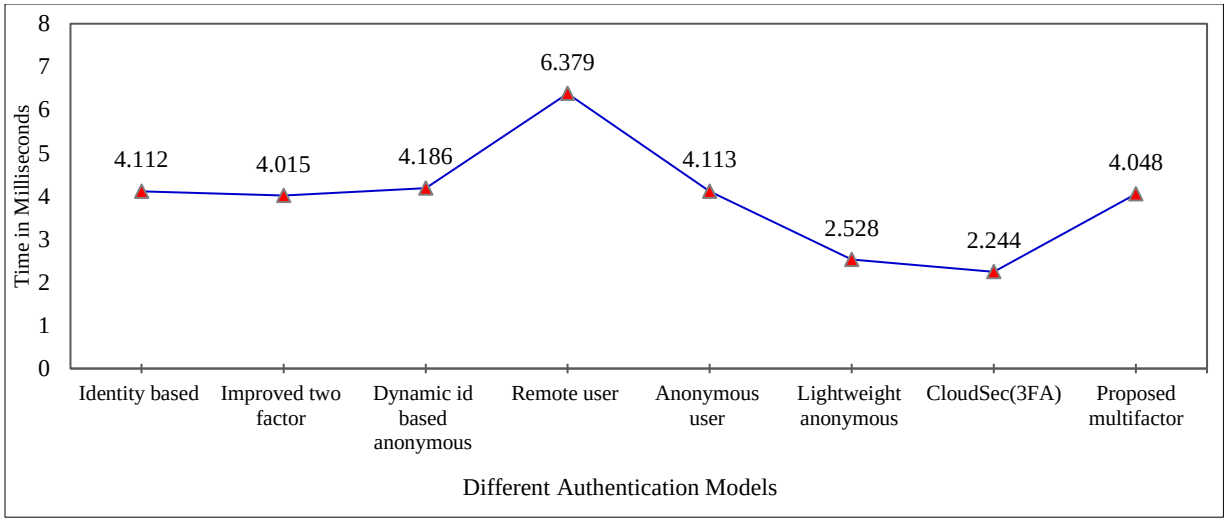

Figure 7 Performance analysis with computational overhead

4.4.2Communication overhead

The number of bits transmitted between the user and the cloud server represents the communication cost. Analyzing the performance of the proposed approach involved comparing its communication cost to other authentication models found in the literature. For the proposed model, the number of bits communicated throughout the entire MFA was 952 bits, including 384 bits for static authentication in transmitting user credentials, 32 bits for dynamic authentication, 24 bits for OTPs, and 512 bits for data on behavioral analysis. Moreover, the space complexity of the proposed model, in terms of the amount of data stored and processed, is O(m), where m is the number of authentication factors, totaling 952 bits.

Furthermore, the results of the communication cost were also compared with other state-of-the-art authentication models, including the data randomization trick [20], identity-based authentication [21], improved two-factor authentication [25], improved authentication [22], flexible authentication [28], dynamic ID-based anonymous scheme [27], remote user authentication [26], anonymous user authentication [23], lightweight anonymous authentication [24], biometric-based [30], and CloudSec-three factor [31] authentications. *Figure 8* displays the communication overhead of different state-of-the-art and proposed authentication models. From the analysis, it was clear that although the communication overhead of the proposed model was slightly higher than that of biometric-based, CloudSec (3FA), and flexible authentication, it was better than many other existing authentication models, demonstrating its superiority.

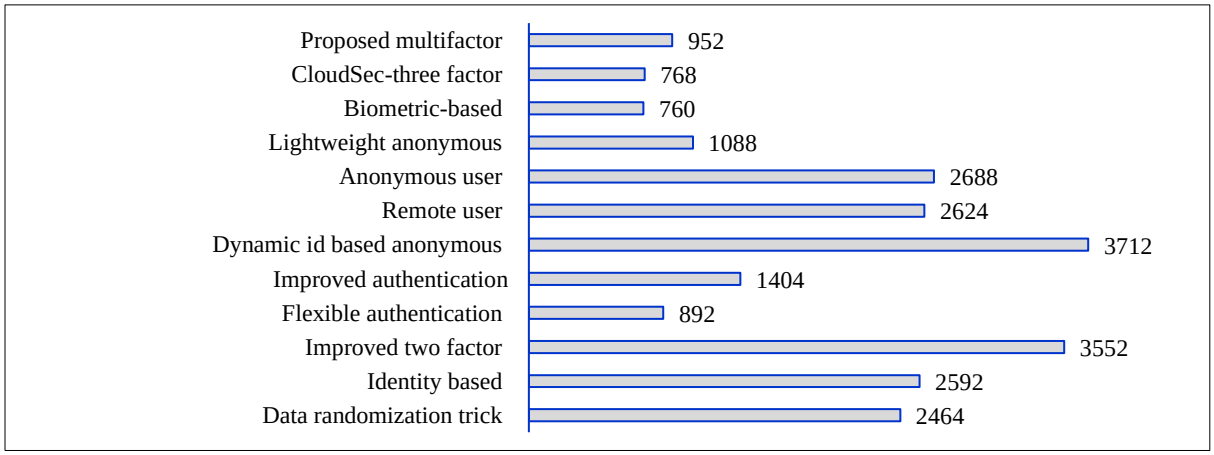

Figure 8 Performance analysis with communication overhead

4.5Cryptanalysis of the proposed model

This section presents the cryptoanalysis of various factors used in MFA.

- *SF:* Initially, users were authenticated with a primary UID and PWD verification. It is a widely used method, providing a baseline level of security. Since the hashed PWD and other credentials were encrypted before being transmitted over the network, the method is resistant to *brute-force attacks*, *dictionary attacks*, *guessing attacks*, and *rainbow table attacks*. Furthermore, since Argon2 hashing algorithms have memory-hardness properties, they are also resistant to attacks based on graphics processing units (GPUs) and application-specific integrated circuits (ASICs). Similarly, a 6-digit PIN adds another layer of security against *guessing* and *brute-force attacks*. Moreover, by implementing lockout methods, limiting the number of incorrect attempts, encouraging periodic PWD changes, and generating alphanumeric PINs, the system can be protected against *brute-force and guessing attacks*.
- *DF:* A challenge-and-response algorithm is used by the captcha to differentiate between human and automated users. Security is enhanced by ensuring human interaction with the unpredictable nature of the challenge-response process. The resilience of captcha against *automated attacks* like *optical character recognition* (OCR) or *machine learning-based approaches* is influenced by its randomness and unpredictability.
- *PF:* By creating OTPs that are only valid for a short time, this technique improves security. This factor is resistant to *replay attacks*. Additionally, the implementation of secure transmission protocols such as hypertext transfer protocol secure (HTTPS) protects authentication tokens like time-based one-time password (TOTPs) from being intercepted and manipulated by adversaries who are attempting to carry out *MITM attacks*. Furthermore, including a second factor—for instance, a temporary code—into the authentication process, makes *phishing attacks* less successful.
- *IF:* For authentication, behavioral profiling and keystroke dynamics examine each person's unique behavior and typing patterns, considering typing activity as a biometric identification. This component can withstand *impersonation attacks*, in which a hacker tries to replicate a genuine user's typing patterns. This provides insights into user activity patterns, enabling the detection of anomalies and the prevention of fraud. However, by regularly updating user behavior and incorporating diversity into behavioral data collection, the likelihood of bias is reduced.

Generally, credential stuffing involves hackers gaining unauthorized access to accounts using compromised UIDs and PWDs, while MFA prevents successful logins even if credentials are compromised. Furthermore, MFA enhances security against session hijacking attacks, as attackers require both user credentials and an additional authentication factor to gain access. MFA reduces the likelihood of keylogging attacks, in which hackers steal user credentials, but it still necessitates an extra layer of security. Social engineering attacks require an additional authentication factor to bypass authentication, thereby reducing the risk of users being manipulated into disclosing sensitive information. MFA is also resistant to insider threats. It adds another layer of defense against insider risks when authorized users misuse their privileges, as the second factor is still necessary for successful authentication even if an attacker has valid credentials. Even if a device is lost or stolen, an attacker would need both the user's credentials and the additional authentication factor to access their accounts or data, reducing the risk of unauthorized access.

Additionally, the use of dynamic captcha and OTPs in MFA helps adapt to evolving security threats. To counter adaptive attacks, implementing continuous learning mechanisms in captcha and OTP algorithms is crucial, as these can adjust complexity based on attack patterns. AI-driven attacks are mitigated through robust Argon2 hashing and regularly updated captcha systems. AI models designed to bypass captchas face significant challenges due to increasing entropy and complexity. Integrating AI-driven behavioral analysis further enhances the system’s ability to detect and mitigate advanced AI-based attacks. Moreover, the multi-layered approach of MFA, integrated with regular key rotations and secure key management practices, provides strong protection against advanced persistent threats (APTs). However, continuous monitoring and anomaly detection systems are essential for identifying patterns indicative of persistent attacks, thereby reducing their effectiveness.

4.6Considerations and challenges in model implementation

Several aspects should be considered when implementing the system in a real-time environment.
Hardware and Software Requirements: To deploy

and run the proposed MFA system effectively, the following hardware and software requirements were recommended: a high-performance server, modern client devices, specific operating systems, web server software, a robust database system, and appropriate development and security tools.

Data Protection and Privacy: Biometric data and personal identifiers are sensitive, making user privacy and data protection crucial. The system should adhere to regulations pertaining to data protection, such as the general data protection regulation (GDPR), digital personal data protection (DPDP) act, and data protection act (DPA). It should use strong encryption for data in transit and at rest, obtain user consent for data collection, and implement access controls to prevent unauthorized access. Regular security audits and compliance checks should be carried out to ensure adherence to these regulations.

Scalability and Load Handling: The system must be designed to efficiently scale with increasing user numbers and high request loads, using load balancing techniques to distribute traffic evenly across multiple servers, ensuring no single server is overloaded. The infrastructure should be horizontally scalable, enabling the addition of more servers to handle increased demand. Network reliability is critical to ensure uninterrupted authentication processes. Redundant systems and backup servers should be in place to ensure continuity of service and minimize downtime.

System Testing and Validation: The system should be validated in various phases. The performance of the model was evaluated across different environments and use cases, considering factors such as device type, network conditions, and user behavior variability. This comprehensive assessment helped ensure that the model remained effective and reliable under diverse conditions. A small group of users monitored login times and system responsiveness during a pilot deployment. User feedback was collected via surveys and interviews to assess usability and satisfaction. Furthermore, appropriate error handling mechanisms were validated by generating detailed error logs and responding to failure scenarios like incorrect user input, server downtime, and network issues with automated alerts. Regular performance testing, including stress and load testing, was conducted to evaluate the system's capacity to handle peak loads and ensure responsiveness under heavy usage. Information security audits and penetration testing identified vulnerabilities and ensured data protection compliance. These tests informed system robustness and user experience improvements.

Real-Time Implementation Challenges: However, implementing the proposed MFA framework in a real-time cloud environment presented several challenges and considerations. Ensuring scalability and optimizing CPU and memory usage were crucial to handling increased data volumes and maintaining performance. Real-time data processing for continuous authentication and behavioral biometrics must be efficient to avoid latency. Maintaining security and privacy is essential, requiring robust encryption and compliance with data protection regulations. Seamless integration with existing systems is necessary to facilitate adoption. Continuous monitoring and maintenance, including regular updates and performance optimizations, are also essential for long-term success. Moreover, the framework can be further enhanced to reduce the impact of distributed denial of service (DDoS) attacks by limiting the number of authentication requests that can be made in a given timeframe. These challenges and considerations must be taken into account to ensure the robustness, usability, and effectiveness of the proposed model in enhancing data security within a real-time cloud environment.

5.Discussion

The proposed MFA framework involves three phases: user registration, user login, and continuous authentication. During registration, users provide essential details and behavioral data, including login behavior and keystroke dynamics, which are used to establish a comprehensive user profile. In the login phase, the system employs a combination of SF, DF, PF, and IF. The trust score is derived from evaluating the IF, which considers typing patterns and user behavior to determine the level of user authorization. Additionally, the continuous authentication phase allows users to enter a secure PIN for critical operations while assessing risk levels and requesting reauthentication as needed.

The effectiveness of the framework was evaluated through a comparative analysis of various authentication models using both closed and open datasets. The proposed model demonstrated significant improvements in performance metrics with the inclusion of PF and IF. Specifically, adding PF increased accuracy from 92.5% to 96.7%, while incorporating IF further enhanced accuracy to 94.7% and improved robustness. The final model achieved

exceptional results with 99.7% accuracy, 99.4% robustness, and a 0.3% error rate. Comparative analysis with existing models, including the 3FA model, shows that the proposed MFA framework excels, particularly with open datasets. After multiple login sessions, the model demonstrates 99.8% robustness and 99.8% accuracy, highlighting its reliability and effectiveness in real-world scenarios. These findings highlight the model's ability to offer strong security while adapting to evolving attack patterns and user behaviors.

The observed errors and misclassifications during the experiments underscore several key issues affecting the performance of the authentication system. The error rates in the proposed model are attributed to FPs and FNs. FPs, where legitimate users are incorrectly flagged as unauthorized, often result from variability in user behavior or anomalies in the data, such as sudden changes in typing speed or login times. FNs, where unauthorized users are mistakenly classified as legitimate, occur due to insufficient differentiation between legitimate behaviors and those of attackers, particularly when attackers mimic legitimate user patterns. These errors are attributable to several factors: inherent variability in user behavior, initial inaccuracies in IF, and limited or unrepresentative training data. Generally, these errors significantly impact system accuracy and robustness, affecting user experience and security. The system's current error rate of 0.2% reflects these challenges, indicating room for improvement. Mitigation strategies involve adaptive learning mechanisms, dataset expansion, and regular updates of authentication algorithms to address evolving behaviors and attack patterns.

Moreover, the proposed model demonstrates strong performance in computational and communication efficiency. The average authentication time is 4.048 milliseconds, showing superior efficiency compared to other state-of-the-art models. The model’s communication cost totals 952 bits, encompassing various authentication factors. While the communication overhead is slightly higher than that of some biometric and CloudSec (3FA) models, it remains competitive. Furthermore, the proposed model outperforms existing models in computational and communication efficiencies, demonstrating its effectiveness and robustness in user authentication procedures and confirming its suitability for efficient and secure operations.

The cryptanalysis of the proposed framework reveals robust security measures. Specifically, SF uses hashed and encrypted credentials that are resistant to brute-force and advanced attacks. DF, such as CAPTCHA, enhances security by distinguishing between humans and bots, while PF utilizes OTPs and secure transmission to prevent replay and MITM attacks. Finally, IF leverages behavioral biometrics, resisting impersonation and fraud through continuous updates. The multi-layered approach of MFA improves resilience against credential stuffing, session hijacking, keylogging, and insider threats. Additionally, dynamic CAPTCHAs and OTPs adapt to evolving threats, with AI-driven analysis strengthening defenses against advanced attacks and APTs.

5.1Implications and recommendations

The proposed MFA framework demonstrates significant advancements in authentication security and efficiency. The findings highlight several implications for both practical applications and future research. To further enhance the system's performance and usability, several key recommendations are suggested. Optimizing algorithm efficiency and evaluating CPU and memory usage are critical for ensuring that the system scales effectively and performs well under varying conditions. Enhancing usability by addressing color-blind accessibility and incorporating additional input methods, such as mouse movements and touchscreen gestures, makes the system more inclusive. Testing the system in real-world environments and gathering user feedback through surveys provides valuable insights for refinement, ensuring the system meets practical needs and user expectations. Implementing continuous learning mechanisms and regular updates enables the system to adapt to evolving threats and maintain high robustness. Expanding the dataset to include diverse scenarios improves the model's generalizability and effectiveness across different contexts. Finally, reinforcing cryptographic measures strengthens the system's defense against sophisticated attacks, ensuring long-term security. These recommendations aim to address current limitations, enhance the system's performance, and adapt to future challenges in the field of authentication.

5.2Limitations of the study

The study has several limitations that necessitate enhancements to the proposed work. Firstly, the current model lacks fine-tuning and scalability optimization to ensure improved performance with increased data volumes. Secondly, while the model's performance was assessed in terms of computational

and communication overhead, it has not been evaluated for CPU and memory usage, which are crucial for understanding its resource efficiency in real-world applications. Thirdly, the model has not been implemented in a real-time environment, making it necessary to test its performance and identify challenges arising under real-time operating conditions. Fourth, the use of color-based PAV may pose challenges for color-blind users, potentially impacting the model's usability for all users. Fifth, the user trust computation based on behavioral analysis could be further enhanced by incorporating additional relevant attributes such as mouse movements or touchscreen gestures, which are currently not included, potentially limiting the comprehensiveness of the behavioral analysis. Finally, the study lacks a usability analysis through user satisfaction surveys, which could provide valuable insights into the user-friendliness of the proposed MFA system. Incorporating user feedback would assist in creating a more practical and user-centric authentication system for real-world applications. The duration of this research study is another limitation, as this restricted timeframe might not capture evolving security challenges and user habits, potentially affecting the overall robustness and adaptability of the system. These limitations highlight areas for future improvement to ensure the robustness, usability, and effectiveness of the proposed authentication framework.

A complete list of abbreviations is listed in *Appendix I.*

6.Conclusion and future work

This study proposed a systematic framework for MFA and authorization for cloud users, intending to improve data security in cloud environments. The proposed framework consisted of three primary phases: user registration, login, and continuous authentication. Users supplied important information throughout the registration process, and if their registration was approved, a unique 6-digit number was provided. Users were validated during the login process by employing a variety of elements, including SF (primary user credentials), DF (color-based PAV), and PF (OTP) for authentication. Moreover, the trust score was determined by assessing the IF, which encompassed user behavior as well as typing behavior, to determine the authorization level of users. In addition, the continuous authentication phase not only allowed users to input a secure PIN for critical operations but also assessed risk levels and requested reauthentication if necessary. The MFA model achieved an impressive average accuracy of 99.7%, robustness of 99.4%, and a minimum error rate of 0.3% on a closed dataset, demonstrating its exceptional performance. On an open dataset, the model exhibited even greater effectiveness, with an average accuracy of 99.8%, robustness of 99.8%, and a minimum error rate of 0.2%. The computational complexity of the proposed model was measured at 4.048 milliseconds, and its communication overhead totaled 952 bits, confirming its overall effectiveness. The originality of the method lay in employing behavioral biometrics and access control strategies to prevent unauthorized access and security risks.

Future work will focus on fine-tuning the proposed model and optimizing its scalability for improved performance and increased data volumes. Furthermore, the behavioral analysis could be enhanced by including additional relevant attributes such as mouse movements or touchscreen gestures. While the model's performance was evaluated using computational and communication overhead, future work will also assess CPU and memory usage. Additionally, the model needs to be implemented in real-time to thoroughly evaluate its performance. An alternative solution, such as patterns or text labels, should also be analyzed to ensure suitability for all users, including color-blind individuals. Furthermore, future work should analyze factors like device type, network conditions, and user behavior variability, which are crucial for model effectiveness and require future research to enhance robustness and adaptability. To gain a more comprehensive understanding of user behavior and the long-term effectiveness of this authentication method, future research should consider extending the observation period. This would enhance the robustness and adaptability of the authentication framework.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Data availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Author’s contribution statement

A. Riyaz Fathima: Conceptualization and design of the work, literature review, data collection, implementation, experimental analysis and interpretation of results, and manuscript preparation. **A. Saravanan:** Conceptualization

and design of the work, Supervision, review of the results and critical assessment of the manuscript.

References

[1] Sunyaev A. Cloud computing. Internet computing: principles of distributed systems and emerging internet-based technologies. 2020:195-236.

[2] Ande R, Adebisi B, Hammoudeh M, Saleem J. Internet of things: evolution and technologies from a security perspective. Sustainable Cities and Society. 2020; 54:101728.

[3] Jiang P, Wang Q, Huang M, Wang C, Li Q, Shen C, et al. Building in-the-cloud network functions: security and privacy challenges. Proceedings of the IEEE. 2021; 109(12):1888-919.

[4] Ali HS, Sridevi R. Credential-based authentication mechanism for IoT devices in fog-cloud computing. In ICT analysis and applications 2022 (pp. 307-18). Springer Singapore.

[5] Zhou C, Lin Z. Study on fraud detection of telecom industry based on rough set. In 8th annual computing and communication workshop and conference 2018 (pp. 15-9). IEEE.

[6] Wang X, Yan Z, Zhang R, Zhang P. Attacks and defenses in user authentication systems: a survey. Journal of Network and Computer Applications. 2021; 188:103080.

[7] Suleski T, Ahmed M, Yang W, Wang E. A review of multi-factor authentication in the internet of healthcare things. Digital health. 2023; 9:20552076231177144.

[8] Yao Q, Wang Q, Zhang X, Fei J. Dynamic access control and authorization system based on zero-trust architecture. In proceedings of the 1st international conference on control, robotics and intelligent system 2020 (pp. 123-7). ACM.

[9] Saranya N, Sakthivadivel M, Karthikeyan G, Rajkumar R. Securing the cloud: an empirical study on best practices for ensuring data privacy and protection. International Journal of Engineering and Management Research. 2023; 13(2):46-9.

[10] Shahidinejad A, Ghobaei-arani M, Souri A, Shojafar M, Kumari S. Light-edge: a lightweight authentication protocol for IoT devices in an edge-cloud environment. IEEE Consumer Electronics Magazine. 2021; 11(2):57-63.

[11] Sarkar S, Roychowdhury S. Authentication authorization and security issues in cloud computing. International Journal for Research in Applied Science & Engineering Technology. 2023; 11(XI):1275-83.

[12] Rayani PK, Changder S. Continuous user authentication on smartphone via behavioral biometrics: a survey. Multimedia Tools and Applications. 2023; 82(2):1633-67.

[13] Finnegan OL, WhiteIII JW, Armstrong B, Adams EL, Burkart S, Beets MW, et al. The utility of behavioral biometrics in user authentication and demographic characteristic detection: a scoping review. Systematic Reviews. 2024; 13(1):61.

[14] Papaioannou M, Pelekoudas-oikonomou F, Mantas G, Serrelis E, Rodriguez J, Fengou MA. A survey on quantitative risk estimation approaches for secure and usable user authentication on smartphones. Sensors. 2023; 23(6):1-34.

[15] Otta SP, Panda S, Gupta M, Hota C. A systematic survey of multi-factor authentication for cloud infrastructure. Future Internet. 2023; 15(4):1-20.

[16] Saranya A, Naresh R, Karuppiah S, Jenifer M. Development of trust-based authorization and authentication framework for secure electronic health payment in cloud environment. Soft Computing. 2024:1-6.

[17] Gupta S. Next-generation user authentication schemes for IoT applications. PhD thesis, University of Trento, Italy. 2020.

[18] Halunen K, Häikiö J, Vallivaara V. Evaluation of user authentication methods in the gadget-free world. Pervasive and Mobile Computing. 2017; 40:220-41.

[19] Erdogan O, Saran NA. A survey on server-based electronic identification and signature schemes to improve eIDAS: with a new proposal for Turkey. Peer Journal Computer Science. 2021; 7:e734.

[20] Li X, Qiu W, Zheng D, Chen K, Li J. Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. IEEE Transactions on Industrial Electronics. 2009; 57(2):793-800.

[21] Farash MS, Attari MA. A secure and efficient identity-based authenticated key exchange protocol for mobile client–server networks. The Journal of Supercomputing. 2014; 69: 395–411.

[22] Farash MS. Security analysis and enhancements of an improved authentication for session initiation protocol with provable security. Peer-to-Peer Networking and Applications. 2016; 9:82-91.

[23] Mo J, Hu Z, Chen H, Shen W. An efficient and provably secure anonymous user authentication and key agreement for mobile cloud computing. Wireless Communications and Mobile Computing. 2019; 2019(1):4520685.

[24] Hammami H, Yahia SB, Obaidat MS. A lightweight anonymous authentication scheme for secure cloud computing services. The Journal of Supercomputing. 2021; 77(2):1693-713.

[25] Chaudhry SA, Mahmood K, Naqvi H, Khan MK. An improved and secure biometric authentication scheme for telecare medicine information systems based on elliptic curve cryptography. Journal of Medical Systems. 2015; 39:1-12.

[26] Chaudhry SA, Naqvi H, Mahmood K, Ahmad HF, Khan MK. An improved remote user authentication scheme using elliptic curve cryptography. Wireless Personal Communications. 2017; 96:5355-73.

[27] Xie Q, Wong DS, Wang G, Tan X, Chen K, Fang L. Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model. IEEE Transactions on Information Forensics and Security. 2017; 12(6):1382-92.

[28] Chang CC, Le HD. A provably secure, efficient, and flexible authentication scheme for ad hoc wireless

sensor networks. IEEE Transactions on Wireless Communications. 2015; 15(1):357-66.
[29] Alkhalifah ES. Password based authentication for web based graphics computing services retrieval in cloud. Multimedia Tools and Applications. 2024:1-23.
[30] Jan SU, Qayum F. An authentication scheme for distributed computing environment. International Journal of Information and Computer Security. 2020; 13(2):227-48.
[31] Saravanan A, Bama SS. CloudSec (3FA): a multifactor with dynamic click colour-based dynamic authentication for securing cloud environment. International Journal of Information and Computer Security. 2023; 20(3-4):269-94.
[32] Ahmadi F, Gupta G, Zahra SR, Baglat P, Thakur P. Multi-factor biometric authentication approach for fog computing to ensure security perspective. In international conference on computing for sustainable global development 2021 (pp. 172-6). IEEE.
[33] Uslu U, İncel ÖD, Alptekin GI. Evaluation of deep learning models for continuous authentication using behavioral biometrics. Procedia Computer Science. 2023; 225:1272-81.
[34] Hossain MA, Al HMA. Improving cloud data security through hybrid verification technique based on biometrics and encryption system. International Journal of Computers and Applications. 2022; 44(5):455-64.
[35] Uddin MA, Kaif M, Zubair MA, Ali MR. Data Repossession by optimized blow fish algorithm in Ml and multistage authentication in cloud. Mathematical Statistician and Engineering Applications. 2023; 72(1):1360-6.
[36] Buriro A, Gupta S, Yautsiukhin A, Crispo B. Risk-driven behavioral biometric-based one-shot-cum-continuous user authentication scheme. Journal of Signal Processing Systems. 2021; 93(9):989-1006.
[37] Durga KK, Rejeti VK, Chandra GR, Ramesh R. Utilizing multi-stage authentication and an optimized blowfish algorithm for effective secure date retrieval on cloud computing. Journal of Data Acquisition and Processing. 2023; 38(4):1418-31.
[38] Kaur S, Kaur G, Shabaz M. A secure two-factor authentication framework in cloud computing. Security and Communication Networks. 2022; 2022(1):7540891.
[39] Anitha HM, Jayarekha P. Multistage authentication to enhance security of virtual machines in cloud environment. International Journal of Advanced Computer Science and Applications. 2021; 12(10):615-23.
[40] Mostafa AM, Ezz M, Elbashir MK, Alruily M, Hamouda E, Alsarhani M, et al. Strengthening cloud security: an innovative multi-factor multi-layer authentication framework for cloud user authentication. Applied Sciences. 2023; 13(19):10871.
[41] Megouache L, Zitouni A, Djoudi M. Ensuring user authentication and data integrity in multi-cloud environment. Human-centric Computing and Information Sciences. 2020; 10:1-20.
[42] Alshahrani M, Traore I. Secure mutual authentication and automated access control for IoT smart home using cumulative keyed-hash chain. Journal of Information Security and Applications. 2019; 45:156-75.
[43] Jasmine RM, Jasper J, Geetha MR. An efficient secure cryptosystem using improved identity based encryption with multimodal biometric authentication and authorization in cloud environments. Wireless Networks. 2024:1-21.
[44] Rajeshkumar K, Dhanasekaran S, Vasudevan V. A novel three-factor authentication and optimal mapreduce frameworks for secure medical big data transmission over the cloud with shaxecc. Multimedia Tools and Applications. 2024: 1-29.
[45] Arumugam S. An effective hybrid encryption model using biometric key for ensuring data security. The International Arab Journal of Information Technology. 2023; 20(5):796-807.
[46] Arasan A, Sadaiyandi R, Al-turjman F, Rajasekaran AS, Selvi KK. Computationally efficient and secure anonymous authentication scheme for cloud users. Personal and Ubiquitous Computing. 2024; 28(1):111-21.
[47] Konwar R, Jha D, Agrawal R, Purkayastha R, Banerjee I. A two-factor authentication mechanism using a novel OTP generation algorithm for cloud applications. In 14th international conference on cloud computing, data science & engineering 2024 (pp. 245-50). IEEE.
[48] Aburbeian AM, Fernández-veiga M. Secure internet financial transactions: a framework integrating multi-factor authentication and machine learning. AI. 2024; 5(1):177-94.
[49] George AT, Scholar PG, Mathew J. Argon2: the secure password hashing function. Proceedings of the national conference on emerging computer applications 2021 (pp. 81-4).
[50] Mondal S, Bours P. Combining keystroke and mouse dynamics for continuous user authentication and identification. In international conference on identity, security and behavior analysis 2016 (pp. 1-8). IEEE.
[51] Saravanan A, Bama SS, Kadry S, Ramasamy LK. A new framework to alleviate DDoS vulnerabilities in cloud computing. International Journal of Electrical & Computer Engineering. 2019; 9(5):4163-75.

A. Riyaz Fathima had obtained M. Sc(CS)., M.Phil.(CS) from Bharathiar University, Coimbatore, India. She is pursuing Ph.D (Computer Science) from Bharathiar University, Coimbatore, India. She has more than eight years of teaching experience in Computer Science. She is currently employed in Sree Saraswathi Thyagaraja College (Autonomous), Pollachi. Her specializations include Computer Networks, Software Engineering, Database Management Systems, and current area of research interest

is Information Security. She has published research papers and presented many contemporary analysis researches papers.
Email: riyazfathimarf@gmail.com

A. Saravanan completed his Doctor of Philosophy in Computer Applications under Anna University Chennai, Tamil Nadu, India. He is currently working as a Director and Professor in the Department of Computer Science and Applications, Sree Saraswathi Thyagaraja College, Pollachi, Coimbatore Tamil Nadu, India. He has an experience of 25 years in teaching and 14 years in research with a good number of publications. His area of interest includes Web Security, Network Security, Web Mining, Software Engineering and Database. He completed his Doctor of Philosophy in Computer Applications under Anna University Chennai, Tamil Nadu, India. He is currently working as a Director and Professor in the Department of Computer Science and Applications, Sree Saraswathi Thyagaraja College, Pollachi, Coimbatore Tamil Nadu, India. He has an experience of 25 years in teaching and 14 years in research with a good number of publications. His area of interest includes Web Security, Network Security, Web Mining, Software Engineering and Database.
Email: a.saravanan21@gmail.com

Appendix I

S. No.	Abbreviation	Description
1	3FA	Three-Factor Authentication
2	AD	Attribute Deviation
3	AES	Advanced Encryption Standard
4	AI	Artificial Intelligence
5	APTs	Advanced Persistent Threats
6	ASCI	Application-Specific Integrated Circuits
7	AV	Attribute Value
8	CPU	Central Processing Unit
9	DDoS	Distributed Denial of Service
10	DF	Dynamic Factor
11	DPA	Data Protection Act
12	DPDP	Digital Personal Data Protection
13	ECC	Elliptic Curve Cryptography
14	EID	Email ID
15	GDPR	General Data Protection Regulation
16	GPU	Graphics Processing Units
17	HMAC-SHA1	Hash-Based Message Authentication Code using Secure Hash Algorithm
18	HTTP	Hypertext Transfer Protocol
19	HTTPS	Hypertext Transfer Protocol Secure
20	FN	False Negative
21	FP	False Positive
22	GCM	Galois/Counter Mode
23	IF	Inherence factor
24	KMS	Key Management Service
25	LSTM	Long Short-Term Memory
26	MFA	Multifactor Authentication
27	MITM	Man-In-The-Middle
28	OBA	Optimization Blowfish Algorithm
29	OCR	Optical Character Recognition
30	OTP	One-Time Passwords
31	PAV	Physical Action Verification
32	PF	Possession Factor
33	PIN	Personal Identification Number
34	PWD	Password
35	RMN	Registered Mobile Number
36	SF	Static Factor
37	SMS	Short Message Service
38	TLS	Transport Layer Security
39	TN	True Negative
40	TOTP	Time-based One-Time Password
41	TP	True Positive
42	UID	Username