Chapter 4
THE ADAPTIVE MUTUAL MULTILEVEL AUTHENTICATION PROTOCOL (AMAP)

This chapter proposes a new authentication method, with an intention to provide secure access to computational resources. The proposed method in this chapter works on mutual and multifactor authentication between the server and the user through authentication center in a cloud environment. The proposed multi-factor authentication protocol provides security, considering multiple parameters like OTP, session password, and so on. The protocol includes two phases: Registration and Authentication. At first, the cloud user and the server are registered under Authorization Centre (AC), which authenticates the user and the server using four different messages. The verification of the user is done by hashing function and ECC following is a six-level authentication process to allocate the virtual machines to access the computational resources in the cloud. This method of authentication provides resistance against Server spoofing attack, Stolen verifier attack, Password guessing attack, Impersonation attack and replay attack. Thus, the protocol allows the cloud user to gain secure access offering robustness with the utilization of Elliptic Curve Cryptography (ECC) and a hashing function.

4.1 Introduction

In the Information Technology field, cloud computing has become a rapidly developing technology from a guaranteed business perception [84]. It offers various services to the clients through the Internet. Some kinds of services that cloud computing offers to the end users are software services, platform services, and infrastructure services [85]. The concept of virtualization provides resources to simulate physical devices and thereby, offering various services. But, in cloud computing, virtualization is one of the challenges during the organization of a cloud computing system. The cloud can be categorized into three, namely, private, public and hybrid cloud, based on the data it has
This technology satisfies multiple end users by enhancing the demands of users to access the resources available via the Internet [87]. Even though it is advantageous, it offers several issues related to security and reliability [88]. Security has become a critical problem affecting the growth of this technology [89]. Hence, it is important to design techniques for authentication, so as to provide security [90]. Authentication is the two-party identity management scheme, in which participating parties need to prove the trustworthy relation with each other. Authentication identifies the legitimate users. In the computer world, authentication is very simple, as giving identity to the physical local equipment to gain the access. In case of cloud providing, the identity is a little bit difficult, because once the company decides to go with the cloud, then the company will be going to lose the control over the data, whatever they wanted to store on the cloud. This is termed as lack of control. The cloud providers may spatially have different servers to provide services to users. But to get the services from the cloud, the users want the smooth and continual security from cloud providers. So, cloud provider normally adopts the cryptographic techniques to provide security to the users. Authentication is one of the cryptographic methods to prove the identity between the cloud provider and user. It guarantees the secured initial phase of communication between the cloud provider and user. Many classical authentication mechanisms like private key cryptography, public key cryptography, and single sign-on, password-based protection are available in theory and are practice to provide security to the cloud users. These classical mechanisms work well in private cloud environment where users of the cloud are within the organization. However, the public cloud has a different scenario where multiple users are accessing the cloud resources over the internet. In such condition, it is very difficult to provide security for the resources over the cloud.

One of the potential mechanisms to establish the trust between cloud providers and users is mutual authentication. This enables the user to secure access to the cloud resources. A mutual authentication follows two approaches.

1. Symmetric key encryption
2. Public key encryption
4.1.1 Symmetric Key Encryption

In symmetric key authentication, the same key is used between two or more to encrypt and decrypt the messages. Symmetric key encryption includes the following components:

- **Input text**: It is the plaintext. Original data in the form of the message are given as input to the algorithm.
- **Key**: It is the secret key generated using some method. This is used as input to the algorithm to produce the different text which is encrypted message.
- **Encryption Algorithm**: The algorithm is created using some unique permutation and combinations of plain text. The algorithm produces the output as an encrypted message.
- **Decryption algorithm**: It is essentially the same algorithm one used in encryption, but it should be used reversely to get the original message or plain text.
- **Ciphertext**: It is the output of the encryption algorithm. It depends on the key and algorithm used for encryption.

Symmetric key cryptography provides a good degree of authentication. The algorithms used in these techniques are inexpensive as they will not include any overhead and they will take less time to produce the ciphertext. This reduces the significant delay and produces the strong output text that is sufficient enough to authenticate the users. This technique is good as far as secret key is kept with the participating users. Here one more disadvantage of this technique is that exchange of secret key between the two users is little bit difficult tasks as the public network is not safe.

4.1.2 Public Key Encryption

In public key authentication, users have two different keys. One is the public key, shared between the users and private key, which is known to the owner of the key. The message that the two users want to exchange should be encrypted using public key in this technique. At the receiver, side message must be decrypted using the private key.
In public key cryptography, the sender encrypts the plaintext with the receiver’s public key. This encrypted plain text becomes a cipher text which should be sent over the public networks. At the receiver side, the ciphertext is converted into plain text using the receiver’s private key. In this technique, different keys are used to decrypt and encrypt the message. There are many public key cryptographic algorithms available in practice, name by RSA, DES, MDA, etc.

The public key cryptography eliminates the key distribution problem. It also increases the security. However, the speed is the problem for such algorithms. These algorithms take much time to produce output if some have more mathematical calculations. Another problem is the way used for digital signatures. It creates repudiation problem with the previously signed message.

Even though, the above-mentioned authentication mechanisms have the potential to identify the participating users correctly, but due to some leakages in the techniques, which enables the attackers to users log and specific information. Due to this reason, the one more advanced step is introduced to make it more secure, that is the multi-factor mutual authentication mechanism.

Multifactor authentication is a technique used to access cloud services over the network. This technique uses more than one user’s credentials to authenticate the users as shown in Figure 4.1. It works using two or more following methods of authentication.

✓ What you know (maybe a password)
✓ What you have (may be any device or phone)
✓ What you are (maybe biometrics)

This helps the cloud providers to protect their resources from adversaries. For example, to access the services of cloud, first the user is asked to enter the username and password, then as a second step, once the user is verified with username and password, OTP is sent to their registered mobile number. The user is supposed to enter this OTP to get access to the services. This is the two-factor user verification method to access the cloud services. Like this, we can have two or more authentication factor are in practice to provide a more secure cloud environment to users.
Figure 4.1: Multifactor authentication mechanism

The multifactor authentication mechanism is the improvement over the cryptographic authentication. Implementation of this technique is costlier, and even we are not sure that whether users may use or not. The major problem with this technique is that this is applicable only the user side. What about the cloud provider and existing users, they may be the adversaries. Therefore, the strong authentication mechanism is needed to verify both the user and the cloud provider.

One of the solutions to the above mentioned security problem is to create secured mutual authentication protocol. Various authentication protocols are developed to have control over the attackers from using cloud resources to solve security issues in the cloud [91]. Depending on the service that supports security, the server generates a secret key as soon as a user requests for service to the server. Some of the attacks like Denial of Service (DoS) attacks, malware injection attack, side channel attack and authentication attack, lead to several security issues in the cloud [92].

This chapter proposes a novel authentication protocol, which works on mutual and multi-level authentication between two parties: the server and the user in the cloud environment. The proposed protocol works in multi-level to provide security,
considering multiple parameters like OTP, session password, and so on. The protocol includes two phases: Registration and Authentication. At first, the cloud user and the server are registered under AC, which authenticates the user and the server using four different messages. The verification of the user is by hashing function and ECC following a six-level authentication process. Data integrity, Data confidentiality and mutual and multi-level authentication are the properties used that offer security to the proposed protocol.

The main contribution of the proposed authentication protocol is to design a novel authentication protocol, in a cloud environment to lessen outsider attacks. Such attacks include password guessing, replay attacks, Denial of service, impersonation attack, password guessing attack, and allocate virtual machines to users with the utilization of various security parameters, such as OTP, session password, a hashing function, ECC, etc.

4.2 Related Work

This section deliberates the literature associated with the authentication protocols and techniques used to mitigate the few authentication related attacks. Here, the research papers [93-97] deal with the protocols for securing cloud privacy issues.

Authors in [93], developed a protocol, Shared Authority based Privacy-preserving Authentication protocol (SAPA) to solve the privacy problem for cloud storage. Some of the features of SAPA were, i) Anonymous access request matching mechanism had provided the authority for shared access by considering the security and privacy features, ii) It had access control based on attributes so that the user could access only in their data fields, iii) data sharing was possible among multiple users, as the cloud server utilized proxy re-encryption mechanism. Moreover, a model, called Universal Compensability (UC), was designed to show that SAPA had design correctness in the theoretical analysis. This suggested that the protocol can be used for various multi-user collaborative cloud applications. However, SAPA adopts an identity token based authentication, and there is no interoperable capability between the tokens.
In [94], developed a protocol with an assumption that the data were prone to privacy invasion and attacks, as the servers in the cloud might be insecure. It utilized cloud-based Radio Frequency IDentification (RFID) authentication protocol, such that the cloud data were preserved, and cannot be compromised. This protocol had attained mutual authentication between all the elements like, cloud server, a reader and a tag, which were involved in the communication. The protocol was analyzed both formally and informally with the utilization of a privacy model and CasperFDR. The major limitation of this protocol is that it is susceptible to location tracking of attack, and impersonation attack.

Authors in [95], had designed a smart card based authentication scheme combined with an access control function to preserve users’ data and information. The scheme was based on the objectives as follows: Confidential data transmission: It considered the confidentiality issue during user login or the transmission of data to/from the cloud environment. Smart card authentication mechanism: The protocol designed a smart card authentication scheme between the cloud server and the user’s equipment so that authorized users could log in to the multimedia cloud environment. Combined multimedia cloud access procedure and access control mechanism to ensure proper data access by the corresponding user. Homegroup privacy and content preservation mechanism: Even though the multimedia cloud was suitable for family use, privacy and control in media content were essential. Here, a group-role based access control scheme was used combined with the authentication mechanism to ensure privacy and content control protect mechanism. However, the technique has various disadvantages like User management issue, Failure to consider security in channel issue, Failure in providing a solution for the lost smart card phase, and the verification process requires large consumption of cloud resources.

In [96], presented a cloud-based authentication protocol that was aware of both internal and external DoS attackers to protect the cloud resource from attackers. It used a multilevel adaptive technique to state the efforts of the participants in the protocol. This could identify the valid requests of a user keeping them at first in the authentication
process queue. The design was made such that the cloud servers were aware of the threats due to DoS attacks. Even though the protocol provides authentication by detecting the risks at an early stage, it did not consider security in side-channel problems.

In [97], designed a Yoking-Proof-based Authentication Protocol (YPAP) for authenticating the devices in the cloud environment. Mutual authentication was provided between the wearable devices and a smart phone using the lightweight cryptographic operators and a physically unclonable function. Mutual authentication along with the yoking-proofs establishment performs simultaneous verification. The YPAP ensure the theoretical design correctness by performing formal analysis and Rubin logic-based security. Thus, YPAP was known to be flexible wearable devices which lightweight in various IoT (Internet of Things) applications. The drawback is that the verification process is complex.

4.3 Proposed AMAP Security Protocol

The proposed protocol that reduces authentication attacks for the security in the cloud is explained in this section. The protocol works in two phases, namely Registration and Authentication. Figure 4.2 illustrates the block diagram of the proposed authentication protocol. The followings are the components of the system:

1. Cloud server
2. Physical machine
3. Virtual machine
4. Cloud authorization center
5. Mutual authentication parameters
6. Cloud user

- **Cloud Authentication Center**: It provides IaaS services to the end user. It authenticates the user and server with mutual multifactor authentication parameters. Once the server and users’ asking for services are authenticated, then this center allocates the virtual machine for users. So users gain authorization to the resources of the center. This is denoted in the proposed work as AC. This has many physical machines on which virtual machines are created and maintained.
In this, AC can run multiple users on the single physical machine sharing all the available resources.

- **Cloud Server:** This is a machine running multiple authentication parameters to authenticate the user. The user one who wants to gain access to the resources, first need to register to this server using password and username. Once registration is done, the user will be provided the one-time password (OTP) for further authentication. Once the registration is successful, the user and server are asked to authenticate with AC. Therefore, a server also gives authentication to the AC to provide the services to the user.

- **Physical Machine:** This is the actual physical machine which has a resource to run multiple virtual machines. These physical machines are present on AC.

- **Virtual Machines:** These are the machines having multiple instances of physical machines. These are allocated to users once authentication becomes successful.

- **Mutual Authentication Parameters:** In this proposed protocol multifactor mutual authentication are used. This technique is the way of getting confirmation from three parties involved. The parameters like, user key provided by the user at the time of registration, user password which is encrypted and stored at server side, server key and password provided to AC at the time of server registration, session key and password created using one-way hashing function and encrypted key of AC are used for authentication. Also, this protocol requires identity-verification mutually during the authentication. Server spoofing attack resistance is possible with the proposed protocol as the server wants to prove its identity to the authorization center. Stolen verifier attack resistance, Password guessing attack resistance, and Session Key (SK)-security are possible because the hashing function and ECC are proven robust against these attacks. The proposed protocol utilized these mechanisms to provide the proof against these attacks. Impersonation attacks resistance and replay attack can be easily achievable through the mutual verification process which is utilized in the proposed protocol. DoS attack in the proposed work is reduced because the use of an OTP - based
mechanism utilized, even though the users access their system in different environments. Cache side channel attack resistance is provided by the proposed protocol as it utilizes the ECC mechanism which is strong against the side channel attacks [28].

Cloud User: These are the end users, who want to gain access to the cloud resources. These users are first asked to do the registration, and then they are authenticated using multi-factors. Once the mutual multifactor authentication is completed successfully, then these users are allocated to the virtual machine. This is called the authorization.

Different symbols that are used in this work are listed in Table 4.1 along with its description as shown below.

Table 4.1: Notations with description

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{ID}$</td>
<td>Server ID</td>
</tr>
<tr>
<td>$S_{PW}$</td>
<td>Server Password</td>
</tr>
<tr>
<td>$S_{ePW}$</td>
<td>Session Password</td>
</tr>
<tr>
<td>$K_{AC}$</td>
<td>Authorization Center Key</td>
</tr>
<tr>
<td>$K_{S}$</td>
<td>Server Key</td>
</tr>
<tr>
<td>$K_{U}$</td>
<td>User Key</td>
</tr>
<tr>
<td>$U_{ID}$</td>
<td>User ID</td>
</tr>
<tr>
<td>$U_{PW}$</td>
<td>User Password</td>
</tr>
<tr>
<td>$P(K)$</td>
<td>Public Key</td>
</tr>
<tr>
<td>$\bullet$</td>
<td>Embedding Operator</td>
</tr>
<tr>
<td>$\Theta$</td>
<td>Extraction Operator</td>
</tr>
</tbody>
</table>

The following sections elaborate the two phases of the protocol in providing authentication for the security against attackers.
The first step in the protocol is registration of the user and server. Only the registered user and server is considered as authenticated, where the AC permits the server to serve the resource required for the user. In registration, the server and the AC share the identity so as to register the cloud server in AC through a secured channel. The steps involved in the user and the server registration are as follows: Each server has its ID and password, represented as, $S_{id}$ and $S_{pw}$, which are stored in AC for the authorization. AC generates a session password $S_{epw}$ by applying a hash function as given below. Hashing is usually used for data administration purpose or security. The hashing process changes messages or text into a fixed string of digits. The hashing function means that it is almost
not possible to retrieve the original data from the hashed value. The one-way hash function in proposed methodology is used to authenticate and identify the sender and also identifies the messages which are distributed digitally. The one-way hash function is created using digital signatures. Also, stronger one-way hash functions in authentication are that they can quickly and securely reduce the size of the clear text.

\[ S_{ePW} = h[K_{AC} \| r \| S_{PW}] \]  \hspace{1cm} (4.1)

Where, \( h[\cdot] \) is the hash function, \( K_{AC} \) is the encryption key of AC, \( r \) is a random number and \( S_{PW} \) is the server password.

Then, AC generates the server key by hashing its key \( K_{AC} \) along with the server ID as follows,

\[ K_s = h[K_{AC} \| S_{ID}] \]  \hspace{1cm} (4.2)

The session password and the secret key generated in AC are submitted to the server, which stores the server key \( K_s \). Then, the server generates an OTP based on the device number \( DN \) of the user and check if the session password matches. OTP-based mechanism utilized in the proposed protocol can easily overcome the DoS attacks, even though the users access their system in different environments. Then, AC compares \( S^*_{ePW} \) with \( S_{ePW} \) and if both are equivalent, i.e. \( S^*_{ePW} = S_{ePW} \), the server is registered.

To verify whether the user is a registered user or not, the generated \( OTP \) is sent to the user. The server generates a key by adopting the user ID stored in the server as,

\[ K_u = h[K_s \| U_{ID}] \]  \hspace{1cm} (4.3)

Where, \( U_{ID} \) is the user ID.

The user stores its private key \( K_u \) and forwards the \( OTP \) to the server. The user is identified as registered, if the server finds \( OTP^* = OTP \). Figure 4.3 illustrates the registration phase where the user and the cloud server are registered.
4.3.2 Multifactor Authentication

The second phase is the authentication phase, which performs mutual and adaptive verification using six levels. Once it registers both the user and the server, it executes authentication based on four different messages such as $X_1, Y_1, W$ and $V$ is using various functions via the communication channel. The first two levels of verification utilize the hash function and first message. Initially, the user forwards a message $X_1$ to the server by concatenating the user ID with the hashed user password, as given below.

$$X_1 = [U_{ID} \| h(U_{pw})]$$  \hspace{1cm} (4.4)

Where, $U_{ID}$ is the user ID and $U_{pw}$ is the user password.
Then, the server computes $X_1^*$ by hashing $U_{pw}$ based on the length of $U_{pw}$ obtained from the user as in the following equation,

$$X_1^* = [U_{id} \| h^*(U_{pw})]$$  \hspace{1cm} (4.5)

Once the server finds $X_1 = X_1^*$, it is considered as the first level of verification.

The server by hashing the server password and then concatenating with the IDs of server and user creates a message as $X_2$

$$X_2 = [S_{id} \| U_{id} \| h(S_{pw})]$$  \hspace{1cm} (4.6)

To calculate $X_2^*$, AC utilizes the stored $S_{pw}$ that is obtained based on its length and applies the hashing function, i.e. $h^*(S_{pw})$, as given below $X_2^*$,

$$X_2^* = [S_{id} \| U_{id} \| h^*(S_{pw})]$$  \hspace{1cm} (4.7)

When $X_2 = X_2^*$, it is assumed that the server has cleared the second level of authentication.

The third and the fourth level of authentication are based on stegno function. Let $Y_1$ be a message computed in AC by embedding the server key with the server ID, as follows,

$$Y_1 = [S_{id} \bullet K^*_S]$$  \hspace{1cm} (4.8)

Where, $\bullet$ is the embedding operator.

The message is then transmitted to the server, where it evaluates the server ID by extracting $Y_1$ and the server key.

$$S_{id}^* = Y_1 \Theta K^*_S$$  \hspace{1cm} (4.9)

Where, $K_s$ is the server key and $\Theta$ is the operator representing extraction.

If the server finds $S_{id} = S_{id}^*$, it is the third level of authentication. Then, the server embeds the user ID with the user key to compute the message $Y_2$ as,

$$Y_2 = [U_{id} \bullet K^*_u]$$  \hspace{1cm} (4.10)

The above message is then forwarded to the user, where it performs the extraction on $Y_2$ and $K^*_u$. 


When the evaluated ID matches with that user, i.e., $U_{id} = U_{id}^*$, then it forms the fourth level of verification.

The final two verification levels follow an XOR-based authentication. It adopts an encryption technique, called ECC, for privacy preservation. The main benefits of elliptic curves rather than cryptographic algorithms based on fixed fields, such as RSA or DSA include a smaller size of the key for security equivalence, the probability to implement without a crypto processor, and the faster execution in some cases when using a crypto processor. Importantly, ECC was proved robust against the side channel attacks [28] which, has been developed in the last years to allow the attacker to obtain all or part of the secrete material included in a cryptographic device by noticing elements such as computing time or power consumption.

The user computes a message $W$ by encrypting the user password using ECC and then applies XOR operator over the hashed keys, as given in Equation (4.12),

$$W = ECC[U_{pw}] \oplus h(K_U) \oplus h(P(K))$$  \hspace{1cm} (4.12)

Where, $P(K)$ is the public key of the user and $ECC[U_{pw}]$ is the ECC applied user password.

The user forwards this message to the server, where it computes $W_1$ as,

$$W_1 = W \oplus h(P(K))$$  \hspace{1cm} (4.13)

Where, $h(P(K))$ is the hashed user public key. Then, the server evaluates $ECC(U_{pw})$ to check if it matches with $ECC(U_{pw})$, to verify whether the user is a registered user and thus, creating the fifth level of authentication.

$$ECC(U_{pw})^* = W_1 \oplus h(K_U)$$  \hspace{1cm} (4.14)

To execute the final authentication level, the server transmits a message, which is computed using ECC applied $S_{pw}$, hashed server key and hashed public key, to the AC as given below,

$$V = ECC[S_{pw}] \oplus h(K_S) \oplus h(P(K))$$  \hspace{1cm} (4.15)

$$V_1 = V \oplus h(P(K))$$  \hspace{1cm} (4.16)
The verification is done here by computing $ECC[S_{pw}]$ to check if it matches with $ECC[S_{pw}]$ using Equation (4.17).

$$ECC[S_{pw}] = V_i \oplus h(K_S)$$ (4.17)

The mutual authentication phase performed in six levels of verification is depicted in Figure 4.4 to allocate virtual machines in a cloud environment to access the data.

### 4.4 Working of AMAP protocol

This section explains the two steps working of the proposed AMAP security protocol with an example.

At registration, every cloud server has unique ID and password, for example server ‘S1’ and ‘s1abc’ as a password. Initially this S1 server has to authenticate itself to the AC to verify the server authenticity by registering with AC. Here, $S_{id}$ and $S_{pw}$ are S1 and ‘s1abc’ respectively. These two values are stored in AC for further communication. After the server registration AC has to generate the session password for S1 for further communication using Equation (4.1). This session password and hash value are created by concatenating encrypted key of AC, some random number generated using ran() function and password ‘s1abc’. Therefore,

**Input:**

```
Hash[xyx1234eaxb||4508||s1abc].
```

**Output:**

```
033ddeac62d
```

In the same way the secret key for the server S1 is generated.

**Input:**

```
Hash[xyx1234eaxb||s1abc]
```

**Output:**

```
07834bb
```

Further, when user having an ID as U1 wants to access the resources of server S1, the U1 has to register with S1 by sending U1 and password ‘u1abc’. Using this ID and
password server S1 creates the user key using the hash function as shown in Equation (4.3). Further verification is done by OTP comparison. With these steps User U1 is registered to server S1 and server S1 is registered to AC.

To allocate the computation resources by launching the VM for user U1 follows the authentication mechanism so that U1 is confirmed as a legitimate user. The authentication goes for six levels and uses five messages. This increases the security of the proposed work. Five messages passed between three parties in six levels as mentioned earlier will protect both, the user's credentials and the resources the user can access. Such protocol can make the attacker harder to crack and brute force attacks resistant. In first level message $X_1$ is created using Equation (4.4). This $X_1$ is initially sent by user U1 to Server S1 in the example. In reply to this server S1 computes the $X_1$ to match the length of the message using Equation (4.5). If $1$ and $X_2$ are matched, then it is confirmed as the first level.

Now server S1 has to allocate the resources to U1. For this purpose the S1 has to communicate to AC for the resources. This is done in second level by producing the message $X_2$, which uses Equation (4.6). In response to this the AC produces message $X'_2$ and verify the same for further levels. Third and fourth levels are completed by the server S1 and AC by producing messages $Y_1$ and $Y_2$. These levels use the stegno function as given in Equation (4.8) and (4.9). Remaining two levels are completed between server S1 and User U1. This uses ECC and XOR functions to produce messages W and V is using Equations (4.12), (4.15) respectively. Finally, messages need to be matched when the reverse calculation is performed. This completes the authentication between server S1 and AC at first and then between user U1 and server S1. In this way finally resources are allocated to user U1 at server S1, which resides at AC.
Figure 4.4: The sequence of steps of authentication phase
4.5 Results and Discussion

This section presents the results of the protocol providing cloud security to offer virtual machines to the user. Moreover, the experimental setup and the performance of the proposed approach, evaluated in a comparative analysis are discussed in the following subsections.

4.5.1 Experimental Setup

The simulation of the proposed technique is executed in a system operated with Windows 10 having the following configurations: Intel processor of CPU 2.16 GHz, the memory of 2GB and 64-bit OS. AC and cloud servers are simulated using CloudSim [79] tool, while, the cloud users and the authentication of cloud users are programmed using JAVA and interfaced with CloudSim. A random model considering a specific number of attackers is also simulated. Depending on the multiple parameters considered, the behavior of the attackers will be simulated.

- **Setup 1**: The cloud setup 1 consists of five PMs that consist of 12 VMs through which the users can access the cloud computational resource.
- **Setup 2**: The cloud setup 2 is designed with 10 servers and 24 VMs for the cloud computational resource access via the internet.
- **Setup 3**: The cloud setup 3 is designed with 20 servers and 44 VMs for the cloud computational resource access via the internet.

4.5.2 Evaluation Metrics

The proposed AMAP protocol for authentication considers three parameters, such as the number of genuine/authenticated users, the number of attackers, and accuracy, to evaluate its performance, as defined below.
The number of genuine users: From an assumed percentage of attackers in the cloud, the number of genuine users accessing the cloud over a simulation time can be defined as follows,

\[ N_G = \frac{N_g}{T_G} \]  

(4.18)

Where \( N_g \) is the number of genuine users identified over time and \( T_G \) is the total number of genuine users in the cloud.

The number of attackers: Similarly, the number of attackers in the cloud can be estimated using the representation given below.

\[ N_A = \frac{N_a}{T_A} \]  

(4.19)

Where, \( N_a \) is the number of attackers identified over time and \( T_A \) is the total number of attackers in the cloud.

Accuracy: The well-known parameter accuracy is used to evaluate the degree of performance of the proposed system. It defines the closeness of a value measured to a standard value as,

\[ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \]  

(4.20)

Where, \( TP \) is True Positive, which identifies the correct cache attacker, \( TN \) is True Negative, used to identify the genuine users as attackers, \( FP \) is False Positive, which is falsely identified as the attacker, and \( FN \) is False Negative, which identifies the falsely genuine users as attackers.

Cloud Setup 1

Figure 4.5 presents a graph showing the estimated details regarding the number of genuine user’s details. The details are shown for the time instants 10, 20, 30 and 40 sec. This graph is shown in the cloud setup 1, in which there are five cloud servers, and twelve users virtual machines are running. The number of genuine users’ details
identified 70.06% and 69.82% in the proposed protocol, at time 10 and 30 sec, respectively. The maximum increment is obtained at time 20 sec with 71.63% in proposed protocol. After that, the protocol is run for 40 sec with the same results as of 30 sec. which means that the constant rate is achieved from 30 secs onward.

Figure 4.5: Number of genuine user details for cloud setup 1

The same experimental set up is continued to get some attackers details and accuracy. Figure 4.6 illustrates the number of attacker details estimated in the comparative analysis, where maximum attacker details are identified at time 20 sec. with 75.19% result in proposed protocol. As the time is kept to its maximum, the details regarding the number of attackers provided are 72.39%, in proposed. After that, the protocol is run for 40 sec. with the same results as of 30 sec. which means that the constant rate is achieved from 30 sec. onward.
Lastly, in Figure 4.7, the accuracy analysis result is illustrated, where the maximum accuracy of 73.41% is found in proposed, at time 20 sec. After that, the protocol is run for 40 sec with the same results as of 30 sec. which means that the constant rate is achieved from 30 sec onward.
Cloud Setup 2

Figure 4.8 presents a graph showing the estimated details regarding the number of genuine user’s details. The details are shown for the time instants 10, 20, 30 and 40 sec. This graph is shown in the cloud set up 2, in which there are ten cloud servers and twenty-four users virtual machines are running. The numbers of genuine user details identified are 74.94% and 75.58% in the proposed protocol, at time 10 and 20 sec, respectively. The maximum increment is obtained at time 20 sec, with 76.39% in proposed protocol. After that, the protocol runs for 40 sec. with the same results as of 30 sec. which means that the constant rate for genuine user details, the attacker details, and maximum accuracy is achieved from 30 secs onward.

![Graph showing number of genuine user’s details for cloud setup 2](image)

Figure 4.8: Number of genuine user’s details for cloud setup 2

The same experimental set up is continued to get a number of attacker’s details and accuracy. Figure 4.9 illustrates the number of attacker details estimated in the comparative analysis, where maximum attacker details are identified at time 30 sec. proposed could provide 79.84% result. As the time is kept to its maximum, the details regarding the number of attackers provided are 79.84%, in proposed. After that, the protocol is run for 40 sec with the same results as of 30 sec. which means that the constant rate is achieved from 30 sec onward.
Figure 4.9: Number of attacker’s details for cloud setup 2

Lastly, in Figure 4.10, the accuracy analysis result is illustrated, where the maximum accuracy of 78.11% is found in proposed, at time 30 sec. After that, the protocol is run for 40 sec with the same results as of 30 sec. which means that the constant rate is achieved from 30 sec onward.

Figure 4.10: Accuracy of the proposed protocol for cloud setup 2

Cloud Setup 3

Figure 4.11 presents a graph showing the estimated details regarding the number of genuine user’s details. The details are shown for the time instants 10, 20, 30 and 40 sec. This graph is shown in the cloud setup 3, in which there are twenty cloud servers and first four users virtual machines are running. The number of genuine user details identified in 74.94% and 75.58% in the proposed protocol, at time 10 and 20 sec,
respectively. The maximum increment is obtained at time 30 sec, with 76.39% in proposed protocol. After that, the protocol is run for 40 sec with the same results as of 30 sec. which means that the constant rate is achieved from 30 sec onward.

Figure 4.11: Number of Genuine user details for cloud setup 3

The same experimental set up is continued to get a number of attacker’s details and accuracy. Figure 4.12 illustrates the number of attacker details estimated in the comparative analysis, where maximum attacker details are identified at time 30 sec. proposed could provide 79.84% result. As the time is kept to its maximum, the details regarding the number of attackers provided are 79.84%, in proposed. After that, the protocol is run for 40 sec with the same results as of 30 sec. which means that the constant rate is achieved from 30 sec onward.

Figure 4.12: Number of attackers’ details for cloud setup 3
Lastly, in Figure 4.13, the accuracy analysis result is illustrated, where the maximum accuracy of 78.11% is found in proposed, at time 30 sec. After that, the protocol is run for 40 sec with the same results as of 30 sec. which means that the constant rate is achieved from 30 sec onward.

![Figure 4.13: Accuracy of the proposed protocol for cloud setup 3](image)

4.6 The Method Employed for Comparison Analysis of Proposed AMAP

The level of efficiency of the protocol can be checked with that of an existing algorithm to estimate how far the method is better than the previous. The proposed authentication approach is compared with a multi-server authentication approach, in [98]. In [98], an authentication protocol using the biometrics-based smart card and ECC was developed for the security. This method is taken here as the existing approach for the comparison with the proposed multi-level authentication protocol.

4.6.1 Performance Comparison

The performance of the proposed adaptive authentication protocol is explained in this section with the simulation results obtained for the cloud setup. These are discussed based on analyses of simulation, as deliberate below.

- **Simulation Analysis:** In this analysis, the simulation results of the proposed protocol, which are carried out in three different cloud setups are discussed. The
results are evaluated based on the three parameters in the proposed: genuine/authenticated users, the number of attackers, and accuracy as well as the existing biometric-based multi-server protocol.

- **Functionalities:** The analysis based on several functionality features regarding techniques used in two existing techniques, [93] and [97]. A comparison is performed based on the features such as mutual authentication, verification table, and various attack resistance features, to check the functionality of the proposed method with the other methodologies. Table 4.2 lists the functional features of the three mentioned methods as a comparative analysis.

The AMAP protocol provides the mutual authentication by way of getting confirmation from three parties involved. Also, AMAP protocol requires identity-verification mutually during the authentication. Server spoofing attack resistance is possible with the proposed AMAP protocol as the server wants to prove its identity to the authorization center. Password guessing attack resistance, stolen verifier attack resistance, and Provides Session Key (SK)-security are possible because the hashing function and ECC are proved robust against these attacks. The proposed AMAP protocol utilized these mechanisms to provide the proof against these attacks. Impersonation attacks resistance and replay attack can be easily achievable through the mutual verification process which is utilized in the proposed AMAP protocol. Even though the users access their system in different environments, OTP-based mechanism utilized in the proposed AMAP protocol can easily reduce the DoS attacks.

Table 4.2 presents observations, which the proposed AMAP protocol provides all the considered features, ensuring the security. Moreover, the proposed approach considers all possible external attacks from users on the cloud, which both the existing techniques fail to consider. Thus, the proposed AMAP authentication protocol outperforms the existing technology in providing security with various functionality features.
Table 4.2: Functionality analysis

<table>
<thead>
<tr>
<th>Features</th>
<th>He-Wang [93]</th>
<th>Vanga Odelu et al. [97]</th>
<th>AMAP protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-level and mutual authentication</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Identity-verification table</td>
<td>X</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Resistance against Server spoofing attack</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Resistance against Stolen verifier attack</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Resistance against Password guessing attack</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Provides Session Key (SK)-security</td>
<td>X</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Resistance against impersonation attack</td>
<td>X</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Resistance against replay attack</td>
<td>X</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Resistance against Man-in-the-middle attack</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Provision for re-registration and revocation</td>
<td>X</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Resistance against DoS attack</td>
<td>X</td>
<td>X</td>
<td>✔</td>
</tr>
<tr>
<td>Resistance against SQL Injection</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Resistance against EDoS attacks</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

4.6.2 Evaluation Metrics

The proposed AMAP protocol for authentication considers three parameters, such as the number of genuine/authenticated users, the number of attackers, and accuracy, to evaluate its performance. Details about the evaluation metrics are presented in Section 4.5.2. After applying the evaluation metrics, the results of proposed protocol performance are compared with existing using three different cloud configurations.
Cloud Setup 1

Figure 4.14 presents a graph showing the estimated details regarding the number of genuine user’s details and also depicts the comparative study based on the genuine user's details between the proposed protocol and existing Vanga Odelu. The details are shown for the time instants 10, 20, 30 and 40 sec. This graph is shown in the cloud set up 1, in which there are five cloud servers, and twelve users virtual machine are running. The number of genuine user details identified in Vanga Odelu, is 69.45%, and 68.46%; and 70.06% and 69.82% in the proposed protocol, at time 10 and 30 sec, respectively. The maximum increment is obtained at time 20 sec, with 70.49% in Vanga Odelu et al. and 71.63% in proposed protocol, which is 0.0159% more than in the existing protocol. After that, the protocol runs for 40 secs with the same results as of 30 sec. which means that the constant rate is achieved from 30 secs onwards.

![Graph showing genuine user analysis](image)

Figure 4.14: Genuine user analysis for cloud setup 1

The same experimental set up is continued to get some attackers details and accuracy. Figure 4.15 illustrates the number of attacker details estimated in the comparative analysis, where maximum attacker details are identified at time 20 sec. When Vanga Odelu, showed the maximum 73.08% result, proposed could provide 75.19% result. As the time is kept to its maximum, the details regarding the number of attackers provided are 71.11% and 72.39%, in Vanga Odelu et al. and proposed. After
that, the protocol is run for 40 secs. with the same results as of 30 sec. which means that the constant rate is achieved from 30 secs onwards.

Figure 4.15: Analysis of Attacker details for cloud setup 1

Lastly, in Figure 4.16, the accuracy analysis result is illustrated, where the maximum accuracy of 73.41% is found in proposed, at time 20 sec, while the existing protocol could attain only 71.78%. After that, the protocol runs for 40 secs with the same results as of 30 sec. which means that the constant rate is achieved from 30 secs onwards.

Figure 4.16: Accuracy Analysis of proposed protocol for cloud setup 1

Cloud Setup 2

Figure 4.17 presents a graph showing the estimated details regarding the number of genuine user’s details and also depicts the comparative study based on the genuine user's details between the proposed protocol and existing Vanga Odelu. The details are shown
for the time instants 10, 20, 30 and 40 sec. This graph is shown in the cloud set up 2, in which there are ten cloud servers and twenty-four users virtual machine are running. The number of genuine user details identified in Vanga Odelu is 73.16%, and 75.62%; and 74.94% and 76.39% in the proposed protocol, at time 10 and 30 sec, respectively. The maximum increment is obtained at time 30 sec, with 75.62% in Vanga Odelu and 76.39% in proposed protocol, which is 0.77% more than in the existing protocol. After that, the protocol runs for 40 secs with the same results as of 30 sec. which means that the constant rate for genuine user details, the attacker details, and maximum accuracy is achieved from 30 secs onwards.

The same experimental set up is continued to get a number of attacker’s details and accuracy. Figure 4.18 illustrates the number of attacker details estimated in the comparative analysis, where maximum attacker details are identified at time 30 sec. When Vanga Odelu showed the maximum 77.93% result, proposed could provide

![Graph](image)

Figure 4.17: Analysis of number of genuine user’s details cloud setup 2

78.11% result. As the time is kept to its maximum, the details regarding the number of attackers provided are 79.84% and 77.93%, in Vanga Odelu and proposed. After that, the protocol is run for 40 secs with the same results as of 30 sec. which means that the constant rate is achieved from 30 secs onwards.
Lastly, in Figure 4.19, the accuracy analysis result is illustrated, where the maximum accuracy of 78.11% is found in proposed, at time 30 sec, while the existing protocol could attain only 77.19%. After that, the protocol runs for 40 secs with the same results as of 30 sec. which means that the constant rate is achieved from 30 secs onwards.

Figure 4.18: Analysis of number of attackers’ details for cloud setup 2

Figure 4.19: Accuracy analysis of proposed protocol for cloud setup 2
Cloud Setup 3

Figure 4.20 presents a graph showing the estimated details regarding the number of genuine user’s details and also depicts the comparative study based on the genuine user’s details between the proposed protocol and existing Vanga Odelu for cloud setup 3. The details are shown for the time instants 10, 20, 30 and 40 sec. This graph is shown in the cloud set up 2, in which there are twenty cloud servers and first four users virtual machine are running. The number of genuine user details identified in Vanga Odelu. is 70.16%, and 72.22%; and 74.95% and 75.58% in the proposed protocol, at time 10 and 20 sec, respectively. The maximum increment is obtained at time 30 sec, with 73.56% in Vanga Odelu and 76.39% in proposed protocol, which is 3.17% more than in the existing protocol. After that, the protocol runs for 40 secs with the same results as of 30 sec. which means that the constant rate is achieved from 30 secs onwards.

![Figure 4.20: Analysis of genuine user details for cloud setup 3](image)

The same experimental set up is continued to get a number of attacker’s details and accuracy. Figure 4.21 illustrates the number of attacker details estimated in the comparative analysis, where maximum attacker details are identified at time 20 sec. When Vanga Odelu showed the maximum 78.05% result, proposed could provide 79.49% result. As the time is kept to its maximum, the details regarding the number of attackers provided are 79.84% and 78.05%, in Vanga Odelu and proposed. After that, the
protocol is run for 40 secs with the same results as of 30 sec. which means that the constant rate is achieved from 30 secs onwards.

![Figure 4.21: Analysis of number of attackers’ details for cloud setup 3](image)

Lastly, in Figure 4.22, the accuracy analysis result is illustrated, where the maximum accuracy of 78.11% is found in proposed, at time 30 sec, while the existing protocol could attain only 77.23%. After that, the protocol is run for 40 secs with the same results as of 30 sec. which means that the constant rate is achieved from 30 secs onwards.

![Figure 4.22: Accuracy Analysis of proposed protocol cloud setup 3](image)
4.6.3 Discussion

This section discusses the comparative analysis results, which is obtained for the three cloud setups based on the performance evaluation metrics. Table 4.3 presents the comparative performance results showing the best results obtained to evaluate the performance of the proposed AMAP protocol.

Table 4.3: Performance analysis represented in time T with 10, 20, 30 and 40 seconds

<table>
<thead>
<tr>
<th>Cloud setup</th>
<th>Vanga Odelu</th>
<th>Proposed Protocol AMAP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10sec</td>
<td>20sec</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Genuine user analysis</td>
<td>69.45</td>
<td>70.49</td>
</tr>
<tr>
<td>Attacker analysis</td>
<td>72.35</td>
<td>73.08</td>
</tr>
<tr>
<td>Accuracy</td>
<td>70.9</td>
<td>71.78</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Genuine user analysis</td>
<td>73.16</td>
<td>74.22</td>
</tr>
<tr>
<td>Attacker analysis</td>
<td>76.63</td>
<td>77.93</td>
</tr>
<tr>
<td>Accuracy</td>
<td>74.89</td>
<td>76.06</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Genuine user analysis</td>
<td>70.16</td>
<td>72.22</td>
</tr>
<tr>
<td>Attacker analysis</td>
<td>76.63</td>
<td>77.93</td>
</tr>
<tr>
<td>Accuracy</td>
<td>74.89</td>
<td>76.06</td>
</tr>
</tbody>
</table>

From the overall performance comparison analysis, the proposed protocol seems to have better performance regarding genuine user details, attacker details, and accuracy, than that of the existing protocol at all the time intervals. However, at time T=20 sec, maximum results are obtained, with 71.63% genuine user details, 75.19% attacker details, and 73.41% accuracy, for the cloud setup 1. 75.58% genuine user details, 78.49% attacker
details, and 77.03% accuracy, for the setup 2. Same results are continued for the cloud set 3, which is constant. Hence, it can be suggested that the proposed authentication approach offers better result ensuring security.

4.7 Threats to the Validity of the Proposed AMAP Protocol

In networking environment, the most common attack that is found is the authentication attack. Since the cloud computing technology is based on the network concepts, the authentication attack is very common in cloud computing. To handle such attacks the AMAP protocol is proposed in this research work. The AMAP protocol is the first level of the proposed research to handle the outsider attacks by means of authentication. The authentication in the proposed system has classified the threats in two categories as follows:

1. The internal user causing an attack and
2. The external user causing an attack.

The proposed AMAP protocol, tried to eliminate the attacks caused by the external users. But there is a threat to this protocol is internal user. This is due to violation and ignorance of security policy set by an employee of the organization. Such an ignorance of an employee can cause an authentication level attack. This type of authentication attack is performed on a host machine. The impact of this attack is that the attacker would be able to control the privilege level of a guest Virtual Machine, thereby allowing the guest user to misuse resources allocated to it. Since the detection of fault in this case is very simple, so the effect of such an attack is lesser than other forms of attack.

Summary

The proposed protocol provides the mutual authentication by way of getting confirmation from three parties involved. Also, the protocol requires identity-verification mutually during the authentication. Server spoofing attack resistance is possible with the proposed protocol as the server wants to prove its identity to the authorization center. Also, the proposed protocol is resistance against attacks like password guessing, stolen verifier and provides session key. OTP-based mechanism utilized in the proposed
protocol can easily overcome the DoS attacks. The privacy is preserved using ECC encryption applied over the considered four messages. ECC together with hashing improves the robustness of the protocol. Authentication is performed here by six levels of verification. The performance of the proposed AMAP protocol is compared with that of Vanga Odelu, considering the number of genuine user details, attacker details, and accuracy, as three parameters. The proposed approach could attain a maximum of 75.58% genuine user details, 78.49% attacker details, and 77.03% accuracy. Chapter 5 presents the methodology to handle cache based side channel attack in the cloud environment.

**********