A Brief Review on Cloud Computing Authentication Frameworks

Abdul Raouf Khan
Department of Computer Sciences, King Faisal University, Saudi Arabia
raoufkhan@kfu.edu.sa
(corresponding author)

Latifa Khalid Alnwihel
Aljaber Finance, Saudi Arabia
latifaalnwihel@gmail.com

Received: 10 November 2022 | Revised: 23 November 2022 | Accepted: 26 November 2022

ABSTRACT

Cloud computing is among the most widely used technologies worldwide as it helps individual users and organizations to store and share information conveniently and cost-effectively. However, data security is a major concern in cloud computing. Security involves guaranteed access to the data only to authorized users and protection against various threats and attacks. Security is generally achieved through the appropriate and efficient implementation of access control, authentication, and authorization mechanisms. Various frameworks have been proposed and implemented for authentication and access control in cloud computing. This paper discusses some of the main authentication frameworks in cloud computing, highlighting their methodologies, algorithms, and problems and identifying the issues that should form the focus of research in the area.

Keywords-cloud security; cloud computing; security framework; cloud authentication

I. INTRODUCTION

The reliance on cloud for storage and computing makes it a prominent target for online attacks. Sharing resources over the Internet puts shared data at risk of being compromised or misused. Thus, users are not comfortable sharing sensitive data online and it is important to ensure that they clearly understand the security-related requirements [1]. Regardless of the clear risks, customers use the cloud due to its many convenient features [2]. Therefore, the security of the cloud needs to be improved. Many researchers have considered various security issues related to cloud computing and proposed various ways to improve it [3-4]. The following sections consider and categorize the research on security-related issues concerning the cloud.

II. AUTHENTICATION IN CLOUD COMPUTING

Authentication is an important aspect of cloud computing in the context of data security. Authentication in an information system can be achieved in multiple ways, for example, based on something the user "has" such as smartcards, something he "knows" such as passwords, and/or something the user "is" such as biometrics (fingerprint, iris scan, etc.).

A. Biometric Authentication Frameworks

In [5], a biometric-based authentication scheme was proposed to offer secure user identification and authentication using elliptic curve cryptography for the generation and exchange of secure keys. An authentication scheme based on a contactless smart card was proposed for the user to improve security and avoid problems with key management. In the registration phase, the user must enter his biometric data, which are recorded in the service provider database. The user identifier was based on the username and the data associated with the server and was used to verify his identity with the cloud service provider. The latter collects the user's identifier, biometric data, and hashed password to store in the smart card during the registration phase. Operation is assumed to be supervised by a monitor, and a proxy is issued at the end of each access case. The reader's sensor had a liveness detector for the data to confirm that they were not collected from an earlier scan. Furthermore, the transferred data were encrypted to prevent attackers from obtaining or tracking them, and data transmission occurred only if the server and the customer met the validation criteria. The biometric scheme involved two phases: biometric verification and biometric linking. To avoid the introduction of a replay attack and guarantee the creation of a session, the nonce value is generated during the handshake message. When a user attempts to log in by inserting his biometric data, a Biohash is generated and associated with their ID and known password. To generate a more secure biometric pattern, the Biohash is XORed with the hashed password and then the smartcard compares its data with the hashed password. The authentication server matches the biometric data and the

hashed password with the relevant data in the database and checks the client identifier. Following this, the authentication server generates an encrypted session key that is transferred with a hashed value. By computing the nonce value, the client can decrypt the data sent by the server and verify its origin. The client can then prove their validity by submitting a signed "hello" message with the nonce value. The authentication server then validates the message and submits proxy credentials to access the service. This scheme has low computational complexity, requires scant communication, is scalable over a large range of numbers of clients, and involves little computation on the client side, which renders it suitable for a cloud environment. However, the complexity of the used hash algorithms is high.

In the context of multimodal biometric systems, a framework for biometrics-based mobile cloud computing was proposed in [6], using two-step verification based on fingerprints and iris recognition. If the user cannot be authenticated through these biometric techniques, the system supports a backup authentication code. The fingerprint recognition system takes an image of the user's fingertip through sensors. The iris recognition system scans their iris using a high-definition camera. The system converts the image of the user's fingertip to obtain a ridge structure, and their iris scan is matched with the available database. Images of the fingertip and the iris of the user are recorded in each access case and compared with the saved input to authenticate them. The user's fingerprint is validated before their iris scan. Backup code-based verification serves as an emergency access mode, as the user is given five to ten unique codes at registration, where each is valid for a single use to access the system. This method helps maintain data security in the cloud. The user can generate new unique codes by using the system, which notifies him of each code generation instance. However, the backup codes are at risk of attacks, and code maintenance induces complexity from the user's perspective.

B. Password-based Authentication Frameworks

In [7], a model was proposed that unified the mathematical approaches to single- and two-server Password-based Authentication Systems (PAS). The reliability and availability of security services were measured to ensure secure data access through Web applications. A single authentication server was involved in the single-server architecture to authenticate the client, storing a dictionary of all client passwords in its database, which must be available at the time of operation to guarantee successful client authentication. If the server fails, a single point of failure occurs that threatens the entire authentication system. To recover the system in this case, a redundant offline server with an updated database of customer passwords can be used. Two servers, a front- and a back-end authentication server, were used in this architecture. The former acts as a public server, while the latter is responsible for maintaining an updated dictionary of client passwords. The client's authentication requests are sent only to the front-end authentication server. The functional behaviors of the system were represented as Generalized Stochastic Petri Nets (GSPNs) to visualize the impact of the authentication service, considering scaled-up/down versions of the authentication servers by exploiting a diversity of backup approaches. Hot and cold standby were used to improve the reliability of the system. Platform-Independent Petri Nets (PIPE) were used to generate a system reachability graph to derive the equivalent Continuous-Time Markov Chain (CTMC). The availability and reliability of the system were estimated by applying traditional Markovian state-based analysis, and the performance-related attributes of the authentication framework were monitored. Experimental results showed that the cold standby strategy can be used to improve the availability of the two-server PAS in case of attacks to a greater extent than the single-server PAS, while the reliability of the single-server PAS is higher than that of the two-server.

A multi-level authentication method was proposed in [8] that allowed access by authenticating the user's password at three levels. Organizational authentication was first conducted by verifying that the user has the privileges to access the services provided, where a failure terminated the session. If organizational authentication succeeded, team-level authentication followed, where access was provided to specific cloud services. The third level, user authentication, was then performed, where specific permissions and privileges were granted to the user. The user can access cloud services only if the first two levels of authentication have been satisfied. This method can be used within an organization's cloud storage.

C. Image-based Authentication Frameworks

A three-step data security model was proposed in [9] using cryptography and steganography during data storage and sharing in the cloud. The model was composed of three levels. The first level employed cryptography using RSA to prevent unauthorized data access. The second level used steganography, where the data were hidden in an image in the form of ciphertext using StegoTools by creating a symmetric key that was sent to the receiver. In the third authentication level, the data were accessed in the image and decrypted by the RSA. The receiver decrypted the image through StegoTools by specifying the symmetric key created during image steganography. The ciphertext was decrypted using the RSA algorithm to obtain the original plaintext. This model can be applied to audio, video, and text data.

D. Digital Certificate-based Authentication Frameworks

Data anonymization was proposed in [10] to prevent the misuse of user data by the cloud provider. This study focused on the trust issue between users and cloud providers based on the anonymity of shared data in the cloud. The idea was to conserve the capability of the cloud provider to charge for the use of reliable cloud services by cutting off the semantic linkage between the data and their owners. The required operations can then be performed by the cloud service provider on an anonymized dataset, following which they can convey the outcomes to the users and charge for the use of services while keeping their identity private. Cloud users can thus safely avail themselves of the services without being identified. This approach guarantees privacy in CC by using ring and group signatures. The study claimed that the anonymity of clients can be preserved using signatures. Typically, this means that each client has a signature that can be verified by the cloud provider by referring to a list of users while preserving their privacy, as

it cannot identify the member of the list to whom the signature in question belongs. This approach requires preparing a set of legitimate users willing to use the cloud resources. However, it is useful only if flat-rate accounting is allowed in the relevant business model. According to it, the user has to pay in advance for access (for a fixed period) to cloud services. When group signatures are used, the group manager is an actor in addition to the verifier and signer. Users can register and obtain credentials to prove that they belong to the set of registered clients. In case of a conflict, the group manager can revoke the anonymity of the group signature.

In [11], a model was proposed based on Elliptical Curve Cryptography (ECC), using a digital signature-based identification protocol for authentication. The study claimed that it does not require modular inversion, which improves its efficiency. However, the system required an algorithm to find the index by using a multiplicative group of finite fields.

E. N-factor-based Authentication Frameworks

1) Two-factor Authentication

An Advanced Encryption Standard–Cipher-text-Identity And Attribute-Based Encryption" (AES–CP–IDABE) was proposed in [12]. This system guaranteed the secure sharing of documents for temporally confidential data using two files: the original and a metadata file. The document's time of release and its attribute were stored on the server in a separate file. The secret keys of the metadata and the original document depended on each other. If the file needed to be deleted or updated, the user can do so without having to decrypt it. Furthermore, two-factor authentication was applied to prevent shoulder surfing and keylogger attacks and improve security. The first factor was a unique user ID and password and the second was mobile authentication. Following password authentication, a QR code was generated and sent to the user, which could be read by the built-in application in Android mobile phones. Then, the phone automatically sent the International Mobile Equipment Identity (IMEI) number to the server and an OTP was generated once the IMEI number was verified. The access policy file needed to be decrypted only if the data owner wished to change the data attributes, which saved time.

2) Multi-factor Authentication

A model of a three-layer authentication mechanism was proposed in [13] to ensure data security, encrypting the data at many levels using several security-related techniques. Unauthorized access was prevented in the first layer through authentication. This layer used various methods, including Fast Identity Online Alliance, multi-factor authentication, one-time password, and the short message service. The second layer was connected to the first to ensure that only legitimate customers could communicate. This layer used different cryptographic encryption techniques, such as homomorphic and proxy-based encryption. The third layer was in charge of the interactions of the second layer and ensured that users who requested access to the data were verified. The study claimed that this model can minimize data security issues in all three layers of cloud service models. This model provided login access to the end user to prevent malicious access to the stored data. To protect the user's confidential information, it ensured the fast retrieval of data while using advanced data protection security and intelligence. However, multi-factor authentication can cause a computational overhead at the user's end.

In light of the drawbacks of multi-factor authentication, an architecture was proposed in [14] to improve it. This architecture used a combination of explicit and implicit authentication factors to provide users with secure access to cloud services at different levels. Implicit authentication is convenient for the user but cannot deliver the required security because it considers user behavior for authentication and has a high false-positive rate [14]. Explicit authentication is more secure, as the user directly provides the authentication factor, but is not user-friendly. An algorithm was proposed to reduce the perceived authentication difficulty for users at different access levels. The algorithm presented a difficult authentication factor if the user failed implicit authentication. In other words, the user needed to be validated through implicit authentication to be easily explicitly authenticated. The architecture involved sandboxing, implicit and explicit authentication, a metalearner, and component F. User access control was performed by sandboxing, whereby the user's access to different levels of the cloud service was controlled. Implicit and explicit authentication mechanisms were used to authenticate the user and obtain access to different service levels in the cloud. Each explicit authentication factor was assigned an authentication score. The metalearner (machine learning engine) provided the weight of the authentication based on the implicit authentication factor. The F component calculated the user authentication score and then identified a set of the best explicit authentication factors to be used by the user to obtain a higher access level. Each cloud service was assigned an operational sensitivity using sandboxing, which specified its security level. If the service was highly sensitive, only highly trusted users could gain access to it. If the authentication score of the user was high, he was highly trusted. The implicit authentication factor was used to assign weights to the scores of the explicit authentication factors. The weight of an explicit factor decreased (increased) if its corresponding implicit factor was invalidated (validated). The user needed a high authentication score to gain access to sensitive services, equal to or higher than the required operational sensitivity. This method reduced the difficulty of user authentication by 29% compared to other methods and can adapt the difficulty of authentication based on the user's situation. Progressive multi-factor authentication results in user fatigue if it is frequently applied because it requires user involvement, and multi-factor authentication can also excessively burden the user.

In [15], a multifactor authentication framework was proposed, which was essentially an extended version of the one proposed in [12]. This framework had three phases, including registration, multi-factor authentication, and encryption phases. The multi-factor authentication, unlike [12], was implemented through Captcha and OTP (One Time Password), in addition to the traditional password authentication process. The Captcha process verified if the user was a machine or a human, and OTP verified if the user was genuine or not. The problem with this authentication mechanism was the redundant use of Captcha, as the OTP process can prove both.

F. Identity-based Authentication Frameworks

In [16], a lightweight mutual authentication system comprising three phases was proposed. In the first phase, the cloud server and the customer generate a random number and an identity request and send it to a third party known as agency. The agency verifies the identity of the customer and the server and responds with a timestamp in the second phase. In the last phase, the user and the server forward identity and timestamp, along with the random number, to each other for mutual authentication. This verification process uses three operations: string concatenation, hashing, and the logical exclusive OR function. The disadvantages of this scheme are that it requires a third party as an agency and transferring computations and identities to the third party makes it vulnerable to attacks.

G. Location-based Authentication Frameworks

Message Digest and Location-based Authentication (MDLA) is a mutual authentication scheme proposed in [17] and involves symmetric key encryption. The goal is to authenticate a mobile user and the cloud server. The scheme consists of registration, authentication, and update phases. It starts only if the user is registered to the cloud service provider, where their credentials are stored on a server. The first phase involves the generation of a random number as a secret key and a message digest along with a primary key. The secret key is used to validate the cloud server in the authentication phase. The primary key is obtained using the user's hashed ID, password, and the previous or first location of the mobile device. If the user is registered, he can send an authentication request using an authentication key, which is composed of the user's current location and timestamp, to the cloud server to access resources. When the authentication request is received on the server side, the information on the mobile user is located to obtain the associated expiration period and the primary key. If the expiry period is zero, the authentication request is rejected by the server, which asks the user to re-register or updates the notification to the mobile user. In case of successful registration, the message is decrypted by the cloud server using the primary key to obtain the current location and timestamp that form the authentication key. The encrypted message is decrypted using the authentication key to obtain the digest of the message. The user is considered legitimate if it matches the stored message digest. Following the authentication of the mobile client, an authentication reply is sent to the mobile user containing the encrypted message digest through the previously shared secret key. Sending this response authenticates the cloud server. A random number is selected as the new secret key by the cloud server and is sent to the mobile user for the next authentication request. If the stored message digest matches the received message digest, the cloud server is considered legitimate. There are three types of update phases: key updates, client registration or re-registration if the expiry period is zero, and updates to authentication or re-authentication in case of a connection loss.

H. Service Model-based Authentication Frameworks

In [18], a personal CC framework based on a service model was proposed, providing users with a conveniently exposed Open API and a secure connection for accessing the cloud. The environment of "cloud orchestration and single sign-on token" was considered so that the user could have a smooth experience. The study described various details of its various components based on the service model and then presented an application of the required security technologies.

Optimizing customer security in an Infrastructure as a Service (IaaS)-based cloud and the use of cloud resources were the objectives of [19], where a framework based on Virtual Machines (VMs) was proposed for the secure usage of sensitive data. Usage Management (UM) was applied to control the resources in the cloud and satisfy the performance and security-related requirements. UM involves constantly monitoring the policies related to the resources. The study presented Randomized Algorithms (RAs) and a policy model to specify the security policies. The RAs compute the optimal distribution of cloud resources of the IaaS between users to ensure security. The architecture consists of a central UM framework that manages cloud resources and Amazon S3 data. It ensures that the resources are moved to the VMs only if the security-related requirements are met, otherwise they are rejected. The UM framework also provides cloud VMs. A local Usage Management Module (UMM) was developed, where any VM provided by the central UM framework can be injected to enforce policies during the lifetime of the resource. If the context of the public cloud does not satisfy the policy requirements regarding the resources, the UMM retracts the resources. The architecture starts when the user requests a policy-protected resource. The user submits contextual information (such as credentials) to the UM framework and is authenticated and granted access based on access control policies for the relevant resources. Depending on the level of security and policies, the resources are transferred to the VM. The RA then receives information on the resource utilization of the VMs to calculate the cost function. Following this, the UMM receives the RA results and distributes the resources to multiple VMs based on security parameters and performance.

I. Cryptography-based Authentication Frameworks

In [20], security schemes that can guarantee the integrity, confidentiality, and authenticity of data were investigated, proposing a framework that uses symmetric and asymmetric cryptographic techniques to enhance data security, including the AES, RSA, and SHA algorithms. The proposed framework used a two-step authentication process: one for user authentication based on a login password for access to data on the server, and the other used RSA to generate a fingerprint verification mechanism for enhanced authentication. The process is carried out at the sender and receiver ends to make the system invulnerable to man-in-the-middle and hijacking attacks. AES was used to encrypt the key and the message using an asymmetric key algorithm, like the RSA. The simulation results showed that the proposed strategy was scalable, efficient, and cost-effective for simple data sharing/access.

In [21], a framework was proposed for secure data sharing in the cloud using hashed message authentication codes, index building, and data classification. An algorithm was developed to protect the data against several issues, e.g., malicious insiders, traffic hijacking, denial of service, and shared technology vulnerabilities, using public and private key

encryption (RSA, AES) along with a digital signature. An error localization algorithm was also proposed to specify the location of errors during an operation, such as deletion and appending while storing the data. Data are classified into private, public, and restricted access modes, by the data owner, to provide access control according to the attributes of confidentiality, integrity, and availability. Index-building is used to search through the encrypted data as the files are retrieved. A file index was proposed to be generated, and both the index and the files were encrypted. The index was composed of keywords that are encrypted as useless data. The encryption process involves the creation of a private/public key using RSA, and Symantec's encryption desktop tool is then applied to generate the user's authentication certificate and verify his identity. This certificate is a unique identifier that can be used to prevent malicious access by attaching it to the message. The Symantec tool generates the customer's private and public keys, and a request for a passphrase when they enter their username and email. The user is allowed to view the toolbar with the identifier, public key, and subkeys once they have provided the passphrase. A secure shell (SSH) is used to produce the keys, as it can be entered anywhere. A hash function is then used to generate the hash of the message, which is encrypted using the private key of the sender and joining the original message with the encrypted message. The sender and receiver share a symmetric key, which is a session key. Using this shared key, the sender re-encrypts the encrypted message along with the original message. The shared key is then encrypted by the receiver's public key and sent with the re-encrypted message. On the receiver side, the message is decrypted using the private key to obtain the session key, which is checked to determine if it matches the shared key. If they match, the receiver uses the sender's public key and the session key for decryption to obtain the original message. The HMAC is applied as a checksum to detect errors during data transmission. The checksum can determine if errors have occurred in the data but cannot correct them. The user can retrieve the data from the cloud by registering with the relevant organization to obtain his access credentials, and his username is stored in the cloud directory. The username is compared with the one saved in the cloud; if the user requests a segment of the data available for public access, no authentication is needed. However, if he requests access to private or restricted data, authentication is carried out by comparing the provided username with the one stored in the database. Authentication is carried out as the user sends his password to the owner and answers some security questions. Upon the user's request, the owner sends them using a digital signature and a keyword, and then sends the user's digital signature to the cloud along with his ID for subsequent use. When the user applies the received digital signature along with the keyword, the cloud confirms whether their ID is valid. The encrypted keywords help the cloud specify the user's location, which can be decrypted by the owner's sent key. The user then submits a file download request to the cloud, which sends the HMAC along with the encrypted file. This is in turn decrypted using the received key. Fully homomorphic encryption can be performed on the encrypted data if there are multiple sub-customers to whom the data need to be sent. The experimental study using Eclipse ID proved that the proposed algorithm was secure. The hybrid encryption-based scheme was faster than the RSA but suffered from memory and time limitations such that it cannot be used in the cloud.

A security framework was proposed in [22] combining the processes used in Kerberos and Pretty Good Privacy (PGP). Kerberos provides mutual authentication, confidentiality, and integrity, and prevents eavesdropping and reply attacks. This system used the concept of a trusted third party (TTP). Kerberos has three logical components: the Authentication Server (AS), the Real server (RS), and the Ticket Grating Serer (TGS). The authentication server acts as a Key Distribution Center (KDC) and issues a session key after verifying the user credentials. The session key is used for communication between the AS and the TGS. The TGS issues a ticket for the Real Server (RS) and provides a session key for communication between a user and the RS, which provides services to the user. The PGP provides cryptographic and authentication services. In this scheme, the user is registered with the Kerberos KDC, which provides a ticket to communicate with the cloud Service Provider (SP). KDC also sends the user credentials and the ticket to the SP, which stores them and acknowledges their receipt. The user encrypts the data before transmission to the cloud. PGP authenticates the user and sends user-encrypted data to the SP. The SP in turn sends the data, as requested by the user, to PGP, which decrypts them with the user information, and sends decrypted data to the user after authentication. The problem with this framework is that Kerberos does not support non-repudiation.

In [23], a hybrid encryption algorithm was proposed combining RSA and AES cryptography to secure data during cloud upload and download. The algorithm uses three separate keys for encryption and decryption. The first is a public key shared between the parties, the second is a private key only on the customer side, and the third is a secret key for decryption. Three keys are generated as the data are uploaded: an AES secret key, an RSA private key-d, an RSA public key-n, and an RSA public key-e. These keys are generated based on system time. The user is expected to save the AES secret key and the RSA private key, as they are needed to upload the data to the cloud (this key is unknown to the cloud administrator). Upload begins through user authentication (username and password). The data are uploaded in an encrypted form to the cloud and stored in a temporary directory. The RSA and AES algorithms are then called. The user is subsequently required to provide the AES secret key, so the data are permanently kept in the cloud and their temporary directory is deleted. The user has to identify the filename to download for the download process and use the AES secret key and the RSA private key to decrypt the data. This study claimed that this hybrid mode of encryption can prevent unauthorized access to user data, as a secret private key is required for this. This access control method can protect the data even if the private key has been compromised, as the original data are not given (unreadable data are downloaded). This also applies to the cloud administrator if he tries to view the data.

In [24], a mechanism based on the RSA cryptographic system was proposed, however, its key generation process is complex and unreliable for certain processes. In [25], two-layer attribute-based hybrid encryption was proposed. The first layer

used HMAC on sensitive attributes. A substitution mechanism was applied in the second layer to re-encrypt the data, and access control policies were also used. The overall mechanism is inefficient as several copies of the encrypted data are stored using the same key and the computational cost is high. Users are supposed to manage encryption keys, which is a significant drawback of the proposed system.

To simultaneously ensure the security, authenticity, and verification of the data, a three-way mechanism was proposed in [26] using Diffie–Hellman key exchange merged into AES and a digital signature. The Diffie–Hellman algorithm is responsible for key exchange through the generation of keys. Once the encryption algorithm has been executed, the digital signature is applied for user authentication. Two servers are involved in this system: one for storing user data files and the other for the encryption process (it is referred to as the trusted computing platform). The flow starts when Diffie–Hellman key exchange is used to exchange keys once the user attempts to upload data to the cloud server. The user is authenticated through the digital signature. The AES encryption algorithm is used to encrypt user data, and the file is uploaded to the storage server. The same steps are carried out if the user wants to download data.

The method proposed in [27] used the hash function, RSA, and DES to ensure data security in the mobile cloud. The system consists of a data owner, a cloud service provider, and a third-party auditor. On behalf of the data owner, the third-party auditor verifies the integrity of the data stored in the mobile cloud. The data in the cloud are encrypted twice: once by the private key of the data owner and again by the third-party auditor's private key. RSA is applied for message authentication. Each data owner and the third-party auditor uses RSA to generate public and private keys for each key generation phase. Only the third-party auditor's public key is then exchanged with the data owner over a secure channel (key-sharing phase). The owner encrypts the data using their public key and then generates a hash message of the encrypted file. The encrypted file is then re-encrypted using the third-party auditor's public key, as is the hash message. The two sets are joined and sent to the third-party auditor (encryption phase). The latter stores the encrypted hash function to ensure data integrity and generates a random key to encrypt the message (using DES). The received package is decrypted using the third-party auditor's private key. The produced random key is stored for later decryption. To verify the correctness of the data after performing DES, the encrypted package is uploaded to the cloud and submitted to the third-party auditor. A random key generated by the third-party auditor is used to decrypt the message. The third-party auditor then produces a hash of the encrypted file and compares it with the decrypted and stored hash value. The result is used to send the appropriate file to the owner along with the requested file. The latter is encrypted by the data owner using their public key so that only they can decrypt it.

In [28], a framework was proposed based on classifying the data into three segments, private, public, and limited access, and accordingly assigning sensitivity ratings to them. The owner classifies the data into segments. The framework consists of two phases. The first phase is further divided into three components: index-building, classification, and message authentication. The index builder is used to sort the encrypted data, and the index itself is encrypted to provide security. Encryption is performed using the Secure Socket Layer (SSL) with a single key. A Message Authentication Code (MAC) is generated after encryption to perform an integrity check. Finally, the encrypted data are stored in the cloud based on a sensitivity rating according to the classification. In the second phase, the data retrieval process requires the registration of the user. The user credentials are stored in the cloud, and access to cloud data requires the submission of these credentials along with an access request. The cloud then takes action based on the request and the sensitivity rating of the data. Authentication is not required if the data are classified as public but is needed if the requested segments are private or have limited access. A digital signature mechanism is used for authentication and verification. Finally, an integrity check of the data is performed using MAC. This scheme emphasizes a trade-off of key size, where a large key increases system complexity and leads to throughput problems.

In the context of mobile cloud computing, an authentication scheme called Message Digest Authentication (MDA) was proposed [29]. This scheme does not involve any additional hardware infrastructure, like the IMSI chip, and consists of a registration and an authentication phase. In the former, the user needs to register by creating an account consisting of a user ID and password, as well as some unique identifying information. Following this, the cloud stores the user's ID and the hashed password along with information on the mobile device used. A message digest of the user, which consists of a user certificate and policy, is then generated by the cloud server. A message digest of the cloud, consisting of a cloud certificate and policy, is also generated. The cloud sends the registered device an encrypted message in the first phase that includes the user's message digest, the cloud's message digest, the cloud's public key, and the relevant column reference in the authentication database. A key is generated by XORing the user ID and the hashed password to encrypt the message. The authentication process starts once the user's message digest, the cloud's digest, and the encryption key have been received by the user's device and consists of two operations: the authentication of the cloud by the mobile device and the authentication of the mobile device by the cloud. The key generated by XORing the user ID and hashed password is used when the mobile device sends an authentication request to the cloud. This key is also used to produce an authentication key to encrypt the message digest. The encrypted message along with the column reference is sent to the cloud, which performs decryption after receiving the message. The cloud searches for the specific user ID and the hashed password. If they are found, it generates a key to decrypt and obtain the message digest. The message digest is decrypted using the authentication key. The user is considered legitimate if the stored message digest matches the obtained message digest. The cloud server submits its digital signature, including the encrypted message digest, using its private key when the mobile device has been authenticated. In return, the mobile device decrypts the message by using the public key of the cloud. If the stored message digest matches the obtained

message digest, the cloud server is considered legitimate. This step establishes data transmission. However, the claimed hypothesis fails if the vulnerability of certain parameters is high.

III. CONCLUSIONS AND FUTURE WORK

This study reviewed several authentication frameworks and their models in cloud computing, reported the characteristics and features of various frameworks, outlined the outstanding threats to them, and proposed solutions to counter these threats. All proposed solutions share a common problem: they are excessively geared toward the client or server side of the cloud architecture. The requirements for developing an efficient cloud computing authentication framework are outlined as follows:

- Client-based solutions are either based on the client's or the cloud provider's features. A common framework should be designed and implemented for both parties. Such solutions should consider the requirements of the clients and the service vendors to avoid interoperability issues within the cloud environment.
- Greater focus is needed on data safety, confidentiality, and privacy, and less on the underlying architecture and processes running within the clients' systems.
- •A low computational overhead should be a major desideratum in designing cloud-based services or devices as the scope of application has expanded to low-power systems.
- The vendor-specific solutions in some proposals imply hardware incompatibility in the cloud environment and thus need to be addressed.

REFERENCES

[1] M. Alsaif, N. Aljaafari, and A. R. Khan, "Information Security Management in Saudi Arabian Organizations," *Procedia Computer Science*, vol. 56, pp. 213–216, Jan. 2015, https://doi.org/10.1016/j.procs.2015.07.201.

[2] B. Heydari and M. Aajami, "Providing a New Model for Discovering Cloud Services Based on Ontology," *Engineering, Technology & Applied Science Research*, vol. 7, no. 6, pp. 2268–2272, Dec. 2017, https://doi.org/10.48084/etasr.1577.

[3] M. Ramzan, M. S. Farooq, A. Zamir, W. Akhtar, M. Ilyas, and H. U. Khan, "An Analysis of Issues for Adoption of Cloud Computing in Telecom Industries," *Engineering, Technology & Applied Science Research*, vol. 8, no. 4, pp. 3157–3161, Aug. 2018, https://doi.org/10.48084/etasr.2101.

[4] M. F. Hyder, S. Tooba, and Waseemullah, "Performance Evaluation of RSA-based Secure Cloud Storage Protocol using OpenStack," *Engineering, Technology & Applied Science Research*, vol. 11, no. 4, pp. 7321–7325, Aug. 2021, https://doi.org/10.48084/etasr.4220.

[5] G. J. W. Kathrine, "A secure framework for enhancing user authentication in cloud environment using biometrics," in *2017 International Conference on Signal Processing and Communication (ICSPC)*, Coimbatore, India, Jul. 2017, pp. 283–287, https://doi.org/10.1109/CSPC.2017.8305854.

[6] S. K. Khatri, Monica, and V. R. Vadi, "Biometrie based authentication and access control techniques to secure mobile cloud computing," in *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, Dec. 2017, pp. 1–7, https://doi.org/10.1109/TEL-NET.2017.8343558.

[7] D. Chattaraj and M. Sarma, "Dependability Quantification of Cloud-Centric Authentication Frameworks," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, Jul. 2018, pp. 840–844, https://doi.org/10.1109/CLOUD.2018.00117.

[8] H. A. Dinesha and V. K. Agrawal, "Multi-level authentication technique for accessing cloud services," in *2012 International Conference on Computing, Communication and Applications*, Dindigul, India, Oct. 2012, pp. 1–4, https://doi.org/10.1109/ICCCA.2012.6179130.

[9] V. K. Pant, J. Prakash, and A. Asthana, "Three step data security model for cloud computing based on RSA and steganography," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, Greater Noida, India, Jul. 2015, pp. 490–494, https://doi.org/10.1109/ICGCIoT.2015.7380514.

[10] M. Jensen, S. Schäge, and J. Schwenk, "Towards an Anonymous Access Control and Accountability Scheme for Cloud Computing," in *2010 IEEE 3rd International Conference on Cloud Computing*, Miami, FL, USA, Jul. 2010, pp. 540–541, https://doi.org/10.1109/CLOUD.2010.61.

[11] L. Wang and T. Song, "An Improved Digital Signature Algorithm and Authentication Protocols in Cloud Platform," in *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, New York, NY, USA, Aug. 2016, pp. 319–324, https://doi.org/10.1109/SmartCloud.2016.46.

[12] K. S. Gajghate and R. V. Mante, "Secure Document Sharing and Access Control on Cloud for Corporate User," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, Apr. 2018, pp. 135–138, https://doi.org/10.1109/ICICCT.2018.8473095.

[13] P. Sirohi and A. Agarwal, "Cloud computing data storage security framework relating to data integrity, privacy and trust," in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, Dehradun, India, Sep. 2015, pp. 115–118, https://doi.org/10.1109/NGCT.2015.7375094.

[14] R. Fathi, M. A. Salehi, and E. L. Leiss, "User-Friendly and Secure Architecture (UFSA) for Authentication of Cloud Services," in *2015 IEEE 8th International Conference on Cloud Computing*, New York, NY, USA, Jun. 2015, pp. 516–523, https://doi.org/10.1109/CLOUD.2015.75.

[15] L. K. ALnwihel and A. R. Khan, "A Novel Cloud Authentication Framework," in *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, Sep. 2020, pp. 1302–1308, https://doi.org/10.1109/CSCI51800.2020.00243.

[16] J. Shen, D. Liu, S. Chang, J. Shen, and D. He, "A Lightweight Mutual Authentication Scheme for User and Server in Cloud," in *2015 First International Conference on Computational Intelligence Theory, Systems and Applications (CCITSA)*, Ilan, Taiwan, Sep. 2015, pp. 183–186, https://doi.org/10.1109/CCITSA.2015.47.

[17] S. Dey, S. Sampalli, and Q. Ye, "A light-weight authentication scheme based on message digest and location for mobile cloud computing," in *2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC)*, Austin, TX, USA, Sep. 2014, pp. 1–2, https://doi.org/10.1109/PCCC.2014.7017041.

[18] S. H. Na, J. Y. Park, and E. N. Huh, "Personal Cloud Computing Security Framework," in *2010 IEEE Asia-Pacific Services Computing Conference*, Hangzhou, China, Sep. 2010, pp. 671–675, https://doi.org/10.1109/APSCC.2010.117.

[19] V. Nandina, J. M. Luna, C. C. Lamb, G. L. Heileman, and C. T. Abdallah, "Provisioning Security and Performance Optimization for Dynamic Cloud Environments," in *2014 IEEE 7th International Conference on Cloud Computing*, Anchorage, AK, USA, Jun. 2014, pp. 979–981, https://doi.org/10.1109/CLOUD.2014.150.

[20] . S. K. M., "Enhanced Security Framework to Ensure Data Security in Cloud Using Security Blanket Algorithm," *International Journal of Research in Engineering and Technology*, vol. 02, no. 10, pp. 225–229, Oct. 2013, https://doi.org/10.15623/ijret.2013.0210033.

[21] A. Bhandari, A. Gupta, and D. Das, "A framework for data security and storage in Cloud Computing," in *2016 International Conference on Computational Techniques in Information and Communication*

Technologies (ICCTICT), New Delhi, India, Mar. 2016, pp. 1–7, https://doi.org/10.1109/ICCTICT.2016.7514542.

[22] S. C. Patel, R. S. Singh, and S. Jaiswal, "Secure and privacy enhanced authentication framework for cloud computing," in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, Coimbatore, India, Oct. 2015, pp. 1631–1634, https://doi.org/10.1109/ECS.2015.7124863.

[23] V. S. Mahalle and A. K. Shahade, "Enhancing the data security in Cloud by implementing hybrid (Rsa & Aes) encryption algorithm," in *2014 International Conference on Power, Automation and Communication (INPAC)*, Amravati, India, Jul. 2014, pp. 146–149, https://doi.org/10.1109/INPAC.2014.6981152.

[24] P. Yellamma, C. Narasimham, and V. Sreenivas, "Data security in cloud using RSA," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, Tiruchengode, India, Jul. 2013, pp. 1–6, https://doi.org/10.1109/ICCCNT.2013.6726471.

[25] S. Verma and S. Ahuja, "A hybrid two layer attribute based encryption for privacy preserving in public cloud," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, Dec. 2016, vol. 2, pp. 1–5, https://doi.org/10.1109/INVENTIVE.2016.7824822.

[26] P. Rewagad and Y. Pawar, "Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing," in *2013 International Conference on Communication Systems and Network Technologies*, Gwalior, India, Apr. 2013, pp. 437–439, https://doi.org/10.1109/CSNT.2013.97.

[27] P. Garg and V. Sharma, "An efficient and secure data storage in Mobile Cloud Computing through RSA and Hash function," in *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, Ghaziabad, India, Oct. 2014, pp. 334–339, https://doi.org/10.1109/ICICICT.2014.6781303.

[28] S. K. Sood, "A combined approach to ensure data security in cloud computing," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1831–1838, Nov. 2012, https://doi.org/10.1016/j.jnca.2012.07.007.

[29] S. Dey, S. Sampalli, and Q. Ye, "MDA: message digest-based authentication for mobile cloud computing," *Journal of Cloud Computing*, vol. 5, no. 1, Nov. 2016, Art. no. 18, https://doi.org/10.1186/s13677-016-0068-6.