Secure Key Management using Mutual Acknowledgement in Cloud Computing

1. Anjali Maurya

(Research Scholar, Bhagwant Institute of Technology, Muzaffarnagar, UP, India)

2. Dr. Pushpneel Verma

(Assistant Director, Bhagwant Group of Institutions)

3. Dr. Ajay Singh

(HOD-CSE, Bhagwant Institute of Technology, Muzaffarnagar, UP, India)

ABSTRACT- Recent times have seen a rise in interest for the three-factor Mutual Authentication and Key Agreement (MAKA) protocols for multi-server systems. These protocols were developed by Microsoft. These protocols make it simpler to access the services and protect the confidentiality of communication that takes place over a public network. Additionally, they make it possible to use the services securely. The great majority of the three-factor MAKA protocols that are now available, on the other hand, do not offer a formal security proof, which leads in a number of attacks on the related protocols. In addition, the computation and transmission expenses associated with using these protocols are rather high. In addition, the vast majority of three-factor MAKA protocols do not have a dynamic revocation mechanism in their design. As a consequence of this, it is impossible to terminate the access of malicious users in a timely way. As a remedy to these limitations, we provide an MAKA protocol that is shown to be dynamic, revocable, and three-factor in nature.

Keywords- Key, Management, Protocol, Security, Mutual Authentication

Introduction:

In the most recent ten years, the whole computing power of the cloud has been made accessible for usage in commercial environments [1,2]. Only very recently was it made feasible for anyone to have access to this. It is not out of the question that this may result in a drop in the amount of money that is spent on the service [3], in addition to an improvement in the efficacy of the service as a consequence of the increased efficiency. The platform that is known as cloud computing is being used by an ever-increasing number of companies to host their services for the purposes of software development, administration, and maintenance. Not only does this lessen the burden of local maintenance that these companies are required to deal with, but it also provides uniform security and operation management for all of the services that are housed on the cloud platform that is offered by a third party [4,5]. This not only makes the burden of local maintenance that these companies are required to deal with more manageable, but it also makes it possible for these businesses to save money. These advantages are dual in nature: First, it lessens the burden of local maintenance that these companies are required to deal with. Second, it provides uniform security and operation management for all of the Even though third-party cloud platforms are outfitted with more robust technologies and a greater number of standard technical specifications, which help to ensure that the servers operate in a relatively secure environment, communication between users and servers still takes place in the public net. Despite the fact that third-party cloud platforms are outfitted with more robust technologies and a greater number of standard technical specifications, which This is due to the fact that even though third-party cloud platforms are outfitted with more robust technologies and a greater number of standard technical specifications, which help to ensure that the servers operate in a relatively secure environment, there is still a risk that sensitive data could be compromised. This is because users still have access to the information that is stored on the servers, despite the fact that the servers operate in an environment that is generally secure. The reason for this is due to the fact that users still have access to the information that is held on the servers. As a consequence of this, authentication and key agreement are very important factors that play a part in determining the level of security that is provided to the connection [10]. The utilisation of mutual authentication and key agreement (MAKA) protocols not only prevents hostile attackers from abusing server resources, but it also prevents hostile attackers from posing as the server in order to steal the user's information and impersonate the server. This is because hostile attackers cannot impersonate the server if they are unable to steal the user's information. This is achieved by stopping hostile attackers from misusing server resources and utilising them for their own benefit. The implementation of MAKA is what makes all of this possible in the first place. The implementation of

MAKA is what makes all of this workable. Since Lamport suggested a method for password-based authentication, the MAKA protocols have been the focus of a large amount of study ever since. Since that time, this line of inquiry has been continuously pursued [11-13].

The act of creating a shared secret key among the members of a network, also known as a session key, is referred to as "key setup." In this procedure, the session key is also known. [14] Participating nodes in the network will need to make use of an interactive protocol in order to carry out this operation. The term "key setup" is used in the vernacular of the business world to refer to this procedure. After that, you will be able to use this session key to achieve some cryptographic goal, such as creating a secure communication channel between entities or ensuring the integrity of the data [15]. You will be able to do this because you will be allowed to utilise it. You may do this by making sure that the data's integrity is maintained in some way. One entity generates a key, which is then securely sent to another entity via key transport protocols when the steps have been completed. On the other hand, in the case of key agreement protocols, both parties submit information that is then utilised to construct the shared key in a collaborative manner [16]. Some examples of key establishment protocols are key transit protocols and key agreement protocols. There are a wide variety of various forms of key establishment protocols. When an entity A has the assurance that no other entity besides a specifically identified second entity B is able to possibly learn the value of a particular secret key, it is said that a key agreement protocol offers implicit key authentication. This is because the value of the secret key cannot be discovered by any other entity. This is the sense in which the term "has the certainty that no other entity" should be understood. This is as a result of the fact that the value of the secret key can never be uncovered by any other organisation or person. This is due to the fact that, in addition to the second entity B that has been specifically identified, there is not a single other entity that is able to determine the value of a secret key. The reason for this is because there is no other organisation or person that is competent to do these tasks. One of the many different forms of key agreement protocols that are now accessible is referred to as an authenticated key agreement protocol. Both of the parties who are participating in the key agreement protocol will have access to an unspecified type of key authentication if they use this specific variation of the protocol. There aren't too many instances of authenticated key agreement mechanisms being used, to be honest. The question of whether or not the key setup protocol ought to allow for explicit key authentication in the event that users are given access to both implicit key authentication and key confirmation is one that is currently being discussed. One kind of key agreement protocol is referred to as an authenticated key agreement with key confirmation. The use of key confirmation is part of this sort of protocol. When using this kind of communication, explicit key authentication is provided to both of the parties who are participating in the protocol for key agreement [17].

Protocols have recently been included into the smart energy HAN, allowing for the authentication of users as well as the generation of key pairs. These techniques are predicated on the idea that every entity has secure access to a Certificate Authority (CA), from whom it may get a certificate. This is necessary for the methods to work properly. This is essential for the systems to function in the correct manner. It is essential that this be done in order to guarantee that the mechanisms will function in the appropriate way. It is of the highest significance to do study into the topic of whether or not it is feasible to offer access to at least one CA to all HANs. This topic is of the utmost importance. This approach is based on the Diffie-Hellman protocol for creating keys, and it works on the assumption that each node already has their own unique set of public and private keys that they may use throughout the process. The Diffie-Hellman protocol is used to generate keys. A one-of-a-kind signature was established for each and every communication by using a hash-based message authentication code that also contained a time stamp. This allowed for the messages to be uniquely identified. This was done with the intention of thwarting any successful attempts at replaying previous attacks. The communication that takes place between SG nodes within the ZigBee network may now be made more secure than it was previously possible as a direct consequence of the deployment of an encryption architecture that makes use of 256-bit AES.

This paper gives a method for secure authentication in multi cloud environment using mutual acknowledgement with encryption.

Implementation

The cloud computing platform is built using Java Server Pages, the Apache Tomcat Server, and the MySQL Server; it also employs three-way mutual authentication and key agreement (MAKA).

Java Server Pages, sometimes known by its abbreviation JSP, is a programming tool that is used on the application server side to develop web-based applications. It is also occasionally known by its full name, Java Server Pages. JSP provides dynamic methods that function in a manner that is not contingent on the platform on which they are run [18].

When it comes to the creation of web applications, the JSP technique may be used in precisely the same way as Servlet technology can be employed. It is possible to consider it an extension of Servlet due to the fact that it provides more capability than that provided by Servlet. This is because it provides a greater variety of options than Servlet offers, which is the reason for its popularity. It is simpler to maintain the JSP pages up to date than it is to keep the Servlet ones current due to the fact that design and development may be done independently in JSP. This is due of the hierarchical organisation of the JSP pages. In addition to being made up of JSP components, the structure of Java Server Pages may also be made up of HTML tags [19].

This open-source Java servlet container implements a number of Java Enterprise Specifications, some of which include the Websites API, Java-Server Pages, and, last but not least, the Java Servlet. Among the other Java Enterprise Specifications that are implemented by this container is the Java Servlet. Tomcat's complete moniker is "Apache Tomcat," and the year 1998 marks the year that it was introduced to the general public for the very first time. The creation of it took place in an atmosphere that openly encouraged participation from the general public and kept the lines of communication open throughout the whole procedure. When it was originally made available, it served as the benchmark implementation for the very first Java Server Pages and the Java Servlet Application Programming Interface (API).

In spite of the fact that it is no longer used as the reference implementation for any of these technologies, consumers continue to have the idea that it is the better alternative. Because it has a number of desirable properties, such as high degrees of extensibility, a tried-and-true core engine, thorough testing, and long-term reliability [20], it continues to be one of the Java servers that is used the most often.

Servlets are a kind of software that, when combined with HTTP protocols, makes it possible for a web server to properly manage dynamic content. This is accomplished via the usage of servlets. Information that is written in Java is what is meant when referring to this kind of material.

Oracle is the company that created the relational database management system (RDBMS) known as MySQL. It uses structured query language as its fundamental underlying technology, and its foundation is built on it. MySQL was first created by the business known as Oracle (SQL).

A collection of diverse types of data that has been structured in an orderly form is what we mean when we talk about a database. It might be anything from a simple shopping list to a photo gallery or even a location inside a corporate network to store the huge quantities of information that are already present there. It could even be a place to display pictures. For example, a relational database is a kind of digital storage that collects data and organises it in line with the relational paradigm.

This form of database is called a relational storage system. In this approach, the tables are made up of rows and columns, and the interactions between the various data components adhere to a very specific logical structure. The tables in this architecture serve as an illustration of a paradigm that is known as relational databases. An RDBMS, also known as a Relational Database Management System, is nothing more than a collection of software tools that are used to actually develop such a database, as well as to manage it, and to query it. It is also sometimes referred to as an RDBMS.

In the field of cryptography, the term "key-agreement protocol" refers to a protocol that enables two or more parties to come to an agreement on a key in such a way that both parties have some degree of influence on the result.

This can be accomplished in several different ways, but the end result is always that both parties have some degree of influence. Additionally, this kind of protocol may make it possible for the parties participating in the discussion to exchange a key with one another. If done correctly, this will prevent unwelcome third parties from imposing a significant decision on the parties to the agreement that they would rather not have to make. Protocols that are useful in practise ensure that the key that has been agreed upon will not be revealed to any party that has the capacity to overhear the conversation and listen in on it.

One side is often tasked with the responsibility of manufacturing the key, while the other party is just responsible for receiving it after it has been sent to them in many key exchange systems. The recipient of the key does not have any say in the procedure through which the key is generated, hence that party does not have any influence on the process. If users implement a key-agreement protocol, they are able to circumvent a number of the key distribution issues that are common to these kinds of systems.

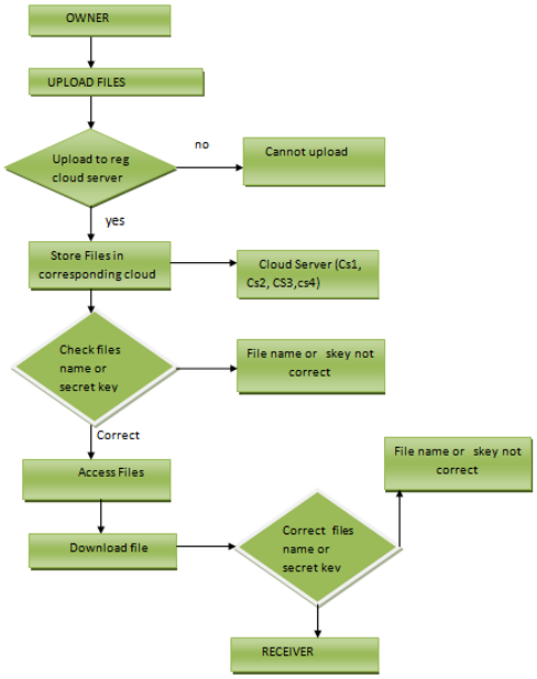

Figure 1: Flow Chart

Figure 1 shows the flow chart of our cloud enabled system.

Results

Fig. 2: Home Page

Step 3: Go to Register tab and enter some credentials as in Fig. 3 to put in some new users.

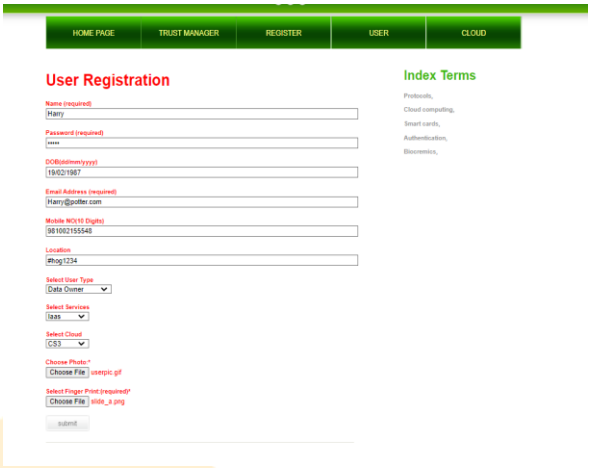

Fig. 3: User Registration

After registration Fig. 4 will be shown

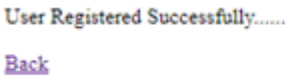

Fig. 4: User Registered Successfully

Similarly, there is cloud login and Trust Admin Login, the login of user, trust and cloud are shown below in Fig. 5, 6, 7 and 8.

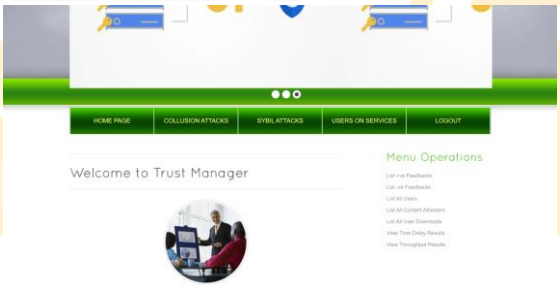

Fig. 5: Trust Admin Login

The above login consists of attacks information, users information and positive and negative feedbacks, attackers.

Fig. 6: User (Data Owner)

The above data owner user can purchase VM, upload files and verify, and also can check trustworthiness of cloud, finding costs of memory and viewing cloud files.

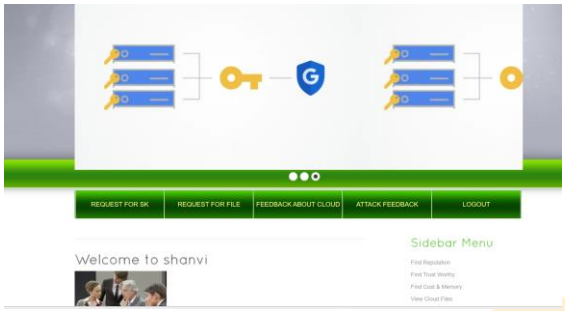

Fig. 7: User (Cloud Consumer)

The cloud consumer can request for secret key, request to download files, give feedback and attack to other feedback reviews.

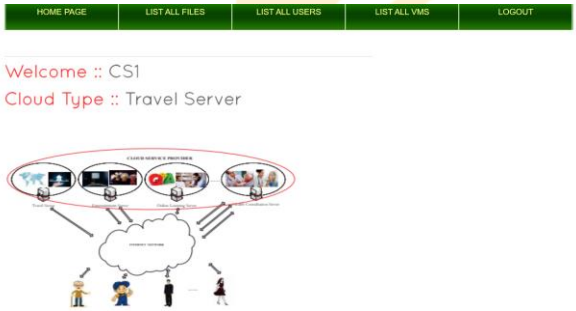

Fig. 8: Cloud Login

The above cloud login can list all files, all users and all virtual memories.

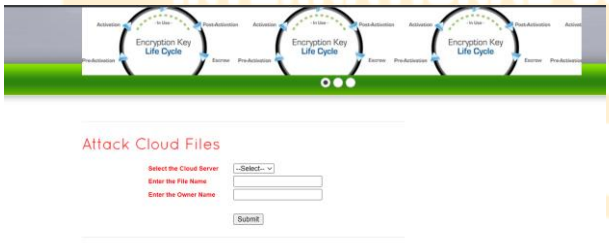

Fig. 9: Attack Cloud File Screen

The above screen Fig. 16 attacks data inside a file in cloud, which can be recovered from cloud.

Trust Admin Login:

The below figure shows collusion attack view in trust admin login.

Fig. 10: Collusion Attack

Fig. 11: View User on Services

Fig. 12: Content Attack

3 User Data Owner:

This section shows features in Data Owner user. Fig. 20 shows Virtual Memory purchase.

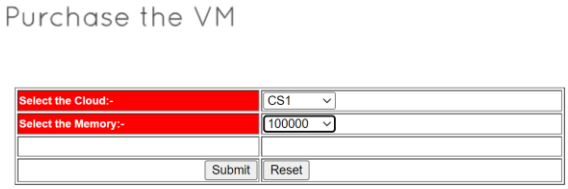

Fig. 13: Memory Purchase

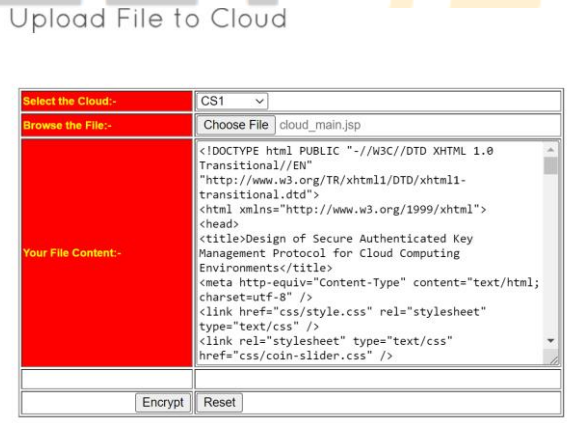

Fig. 14: Upload File

Fig. 14 to 17 shows upload and encryption of file.

Fig. 15: Encrypt File

The secure key authenticated system also uses AES algorithm to encrypt data and generates the secret key.

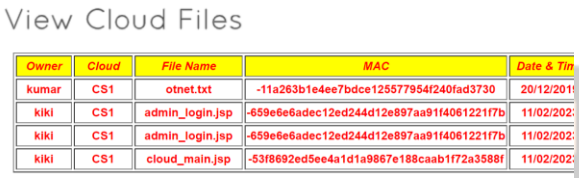

Fig. 16: Encrypt File

Fig. 17: Trustworthy and reputation of Cloud Server View

The above shows the number of attacks in cloud server.

Cloud Consumer User:

Fig. 18: Secret key Generation

Above figure 18 shows secret key generation of an uploaded file in cloud, which can be downloaded as shown in below Fig. 19.

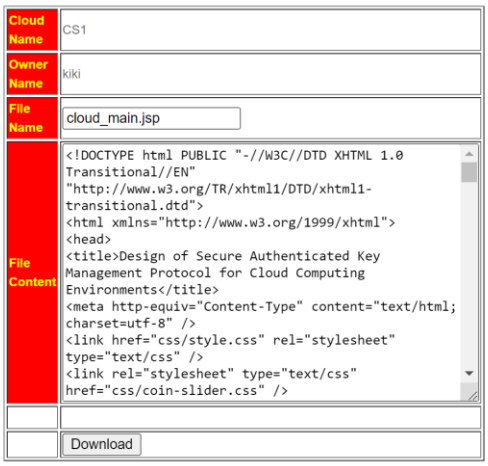

Fig. 19: Download

This shows high encryption using mutual authentication and acknowledgement.

Conclusion

The management of keys is the fundamental component of any data security system. Encryption keys are used to encrypt and decode data, which implies that if any encryption key were lost or compromised, the data security measures that had been put into place would be rendered useless. In addition, keys guarantee the integrity of data while it is being sent through an Internet connection. A method has been developed as part of the scope of this research project for the secure and efficient keeping and sharing of information via the use of public cloud storage. This framework can be found here. In order to achieve this objective, two unique strategies are used in terms of storing the data and delivering it to the appropriate parties. The user does not need the authorization of a third-party auditor in order to be able to save their data in a public cloud storage service; this authorization is not required and a mutual authentication using secret key is required to download the file.

References

[1] W. Li, X. Li, J. Gao and H. Wang, "Design of Secure Authenticated Key Management Protocol for

Cloud Computing Environments," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 3, pp. 1276-1290, 1 May-June 2021, doi: 10.1109/TDSC.2019.2909890.

[2] P. E. Abi-Char, A. Mhamed and B. El-Hassan, "A Secure Authenticated Key Agreement Protocol For Wireless Security," Third International Symposium on Information Assurance and Security, 2007, pp. 33-38, doi: 10.1109/IAS.2007.56.

[3] H. Nicanfar, P. Jokar and V. C. M. Leung, "Efficient authentication and key management for the Home Area Network," 2012 IEEE International Conference on Communications (ICC), 2012, pp. 878-882, doi: 10.1109/ICC.2012.6364549.

[4] H. Tan and I. Chung, "Secure Authentication and Key Management With Blockchain in VANETs," in IEEE Access, vol. 8, pp. 2482-2498, 2020, doi: 10.1109/ACCESS.2019.2962387.

[5] C. Popescu, "A secure authenticated key agreement protocol," Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (IEEE Cat. No.04CH37521), 2004, pp. 783-786 Vol.2, doi: 10.1109/MELCON.2004.1347048.

[6] A. J. Prakash and V. R. Uthariaraj, "Secure Authenticated Key Establishment Protocol for Ad Hoc Networks," 2009 Third International Conference on Network and System Security, 2009, pp. 87-94, doi: 10.1109/NSS.2009.61.

[7] Q. Cheng, G. Han and C. Ma, "A New Efficient and Strongly Secure Authenticated Key Exchange Protocol," 2009 Fifth International Conference on Information Assurance and Security, 2009, pp. 499-502, doi: 10.1109/IAS.2009.122.

[8] M. Ma, D. He, H. Wang, N. Kumar and K. -K. R. Choo, "An Efficient and Provably Secure Authenticated Key Agreement Protocol for Fog-Based Vehicular Ad-Hoc Networks," in IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8065-8075, Oct. 2019, doi: 10.1109/JIOT.2019.2902840.

[9] Lijiang Zhang, "A provably secure authenticated key exchange protocol," 2010 IEEE International Conference on Wireless Communications, Networking and Information Security, 2010, pp. 292-297, doi: 10.1109/WCINS.2010.5541786.

[10] R. Vinoth, L. J. Deborah, P. Vijayakumar and N. Kumar, "Secure Multifactor Authenticated Key Agreement Scheme for Industrial IoT," in IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3801-3811, 1 March1, 2021, doi: 10.1109/JIOT.2020.3024703.

[11] V. Odelu, A. K. Das, M. Wazid and M. Conti, "Provably Secure Authenticated Key Agreement Scheme for Smart Grid," in IEEE Transactions on Smart Grid, vol. 9, no. 3, pp. 1900-1910, May 2018, doi: 10.1109/TSG.2016.2602282.

[12] H. M. Elkamchouchi, Y. A. Saleh and A. M. Sary, "New authenticated key agreement protocols," The 2011 International Conference on Computer Engineering & Systems, 2011, pp. 58-63, doi: 10.1109/ICCES.2011.6141012.

[13] Ai-fen Sui et al., "An improved authenticated key agreement protocol with perfect forward secrecy for wireless mobile communication," IEEE Wireless Communications and Networking Conference, 2005, 2005, pp. 2088-2093 Vol. 4, doi: 10.1109/WCNC.2005.1424840.

[14] C. D. de Saint Guilhem, M. Fischlin and B. Warinschi, "Authentication in Key-Exchange: Definitions, Relations and Composition," 2020 IEEE 33rd Computer Security Foundations Symposium (CSF), 2020, pp. 288-303, doi: 10.1109/CSF49147.2020.00028.

[15] P. H. Griffin, "Thought-Based Authenticated Key Exchange," 2019 ITU Kaleidoscope: ICT for Health: Networks, Standards and Innovation (ITU K), 2019, pp. 1-8, doi: 10.23919/ITUK48006.2019.8996150.

[16] M. Chakraborty, B. Jana and T. Mandal, "A Secure Cloud Computing Authentication Using Cryptography," 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), 2018, pp. 1-4, doi: 10.1109/ICETIETR.2018.8529100.

[17] S. R. Moharana, V. K. Jha, A. Satpathy, S. K. Addya, A. K. Turuk and B. Majhi, "Secure key-distribution in IoT cloud networks," 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS), 2017, pp. 197-202, doi: 10.1109/SSPS.2017.8071591.

[18] S. Graf, P. Lang, S. A. Hohenadel and M. Waldvogel, "Versatile Key Management for Secure Cloud Storage," 2012 IEEE 31st Symposium on Reliable Distributed Systems, 2012, pp. 469-474, doi: 10.1109/SRDS.2012.80.

[19] B. Celiktas, I. Celikbilek and E. Ozdemir, "A Higher-Level Security Scheme for Key Access on Cloud Computing," in IEEE Access, vol. 9, pp. 107347-107359, 2021, doi: 10.1109/ACCESS.2021.3101048.

[20] A. Kumari, M. Y. Abbasi and M. Alam, "A smartcard-based key agreement framework for cloud computing using ECC," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021, pp. 43-48, doi: 10.1109/ICICV50876.2021.9388629.