International Journal on Applications in Information and Communication Engineering
Volume 10 : Issue 4 : December 2024 , pp 1 –8 www.aetsjournal.com ISSN (Online) : 2394 - 6237

Optimized Security-Enabled Authentication for Reduced Processing Time in Distributed Cloud Computing

D. Kaleeswaran, Chinnasamy U
Department of Computer Science and Engineering Regional Language
Rathinam Technical Campus, Coimbatore, Tamilnadu, India

***Abstract*—** advancements in cloud computing technology, users can now access data, applications, computed results, and cloud services from anywhere in the world at any time. To utilize these cloud computing services, users often rely on unsecured networks. Security is a critical factor to consider when accessing cloud services. The traditional Single Sign-On (SSO) system, commonly used for authentication, employs an OpenID. This OpenID allows users to access cloud services with a single password or secret key. However, the SSO system involves a third party in every step of the authentication process, which can significantly increase computational time. To improve user convenience and leverage the Elliptic Curve Cryptography algorithm, the Bilinear Pairing technique, and a Smart Card Generator act as trusted third parties. This a pproach also employs cryptographic systems to enable key exchange, mutual authentication, and user anonymity. One of the primary threats during the authentication process is phishing attacks. To address this risk, the Password Hash Chain Generator (PHCG) has been proposed. This generator creates a chain of hash values that securely provide passwords to both the user and the cloud server, thereby mitigating phishing attempts. The Advanced Hashing Encryption (AHE) algorithm is utilized to design the password hash chain generator.

Index Terms— Key exchange, phishing attack, user ambiguity, computational time, cloud computing services Local Binary Pattern (LBP), Neural Network (NN), Support Vector Machine (SVM), Graph Cut, Bounding box technique, Gaussian filter.

I. INTRODUCTION

The term "cloud" can be taken in its literal sense to refer to the Internet or a network that is located in a remote location. Cloud computing services can be accessed through either public or private networks. Because the cloud is accessible to anyone, security is becoming an increasingly pressing concern [1]. The data files that are stored in the cloud need to be kept in an encrypted format so that a malicious person will not be able to access the shared data files. When it comes to protecting data stored in the cloud, it is essential to have a solid understanding of the following mechanisms: access control, auditing, authentication, and authorization. The concept of authentication is the primary focus of this work [2]. When there is a concern about security, both the user and the service provider need to concentrate on providing authentication. Authentication is a standard security measure that is implemented across all network-related services in order to prevent unauthorised access. Both the client and the server, which are both participating in the transaction, need to have their identities verified. Take for example a computer network or even a portable electronic device. It is necessary for the owner to keep the password in a safe place. Even an unauthenticated user will not be able to access the system if they do not have the password. The term for this process is known as user authentication. This is primarily accomplished by a function that is referred to as the "local security authority subsystem service." Throughout the entirety of the authentication phase, there will be a number of issues to contend with, including privacy concerns, supply concerns, safety concerns, and redundant concerns. [3]-[5]. Concerns about security pose a greater risk in this context and could potentially slow down the procedure. The use of group pairings is one of the many approaches that can be taken to address this issue. Pairings is a term that is used in the field of cryptography to describe the process of creating a new cryptographic group by mapping between any two different cryptographic groups. The criteria of computability, non-degeneracy, and bi-linearity must be met by cryptographic parings in order for them to be valid. If the same group is used for pairing, then the result is symmetric. As a result, the use of pairs is essential for the transmission of messages.

The cloud's central premise is that the user and the service provider must work together in order for the user to realise any benefits from using the cloud. Therefore, there must have been an attack during the pairing process that necessitated additional security. In order to accomplish this, we will use bilinear mapping. Cryptography based on elliptic curves [6] is required for bilinear mapping. Files can be encrypted with elliptic curve cryptography in such a way that only users with the appropriate permissions can decode them. To a significant extent, it is predicated on mathematical theorems. It places an emphasis on the encryption and decryption processes while ensuring that there is no break in continuity.

When encrypting and decrypting a message, one of the primary advantages of utilising the shortest key possible is that the process is completed much more quickly and makes very little use of the resources available on the computer.

A. COMPANION WORKS

For the purpose of preventing unauthorised users from gaining illegal access to network-based services, the authentication strategy is regarded as a security technique that can be applied across the board. Public-key cryptographic systems that have been around for a long time typically serve as the basis for traditional authentication methods. The standard public key algorithm known as RSA calls for a significant key size. In addition to this, a significantly higher amount of computing power is required. As a direct consequence of this, conventional authentication procedures are not supported by systems that have a restricted amount of processing resources. Koblitz [7] and Miller [8] presented the concept of the Elliptic Curve Cryptosystem (ECC), which offers the same level of security as traditional public key cryptosystems despite having the smallest key size possible. If, for example, a public key with 3092 bits of RSA encryption offers a certain level of protection, then a public key with 256 bits of ECC encryption will also offer that level of safety. This method of calculation yields more accurate results.

Bilinear pairing [9]-[11] in an elliptic curve has been utilised in recent years for the purpose of developing ID-based cryptosystems. Since then, a significant number of ID-based cryptosystems [12] have been predicted. The issue of high costs associated with public key management can be remedied by utilising this ID-based cryptosystem, which is one of the public key cryptosystems. In a cryptosystem that is based on user IDs, the user's identity doubles as their one-of-a-kind public key. This is done so that the user does not need to bear the additional computational costs associated with checking the public keys of other users. After that, the authentication procedure, which is based on the user's ID, is used in the cloud computing environment [13]. This ID-based authentication, on the other hand, does not provide any room for user uncertainty. The vast majority of authentication methods use bilinear pairing or ECC [14–17]. [14]–[17] Because they were designed to work in a client-server architecture, these schemes are not suitable for use in a distributed environment. Because of the straightforward nature of this approach, it is not possible for multiple service providers (SP) to take part in it. Tsai et al. [18] developed a method of authentication that is based on smart cards in order to solve this problem. This system allows for multiple service providers (SP) to participate in dispersed cloud environments by utilising a smart card generator as a trusted third party.

B.SECURITY

Security is absolutely necessary for every facet of computer use. When a user wishes to make use of a cloud computing service, the user in question must first be validated before that user is granted access to the cloud service offered by the various providers. In a similar vein, in order for the data sharing to be done in a secure manner, the service provider itself needs to be checked out [19]. OpenID is used in the conventional Single Sign-On approach [20] so that customers can log into multiple cloud computing services with just one password or key. This allows customers to save time and reduces the risk of password theft. There is a significant need for computing resources due to the fact that each user authentication session that takes place using this SSO approach involves a reliable third party [21]. The three roles that this OpenID goes through are users, reliant partners (RP) or service providers (SP), and identity providers (IdP). Each Service Provider (SP) is required to share all of their data with the Identity Provider so that they can improve their ability to work together and identify one another (IdP). Before attempting to log into a website, the user needs to first sign up with an identity provider (IdP) so that they can obtain an OpenID identifier. The process of a user registering with an identity provider by providing their personal information is depicted in Figure 1.

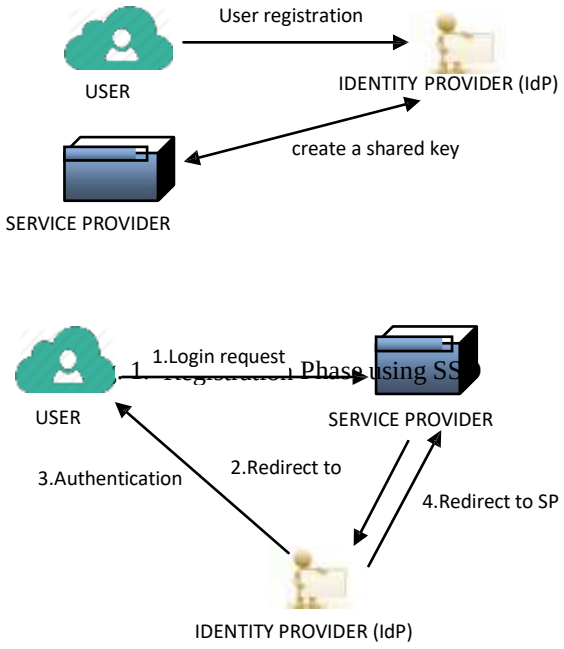

Fig. 2. Authentication Phase using SSO

After that, the user sends their OpenId identifier to the service provider through a protocol known as Secure Socket

Layer (SSL). After the SP has received the user's OpenId, it then sends a request to the IdP for user verification, as shown in Figure 2. This allows the SP to confirm the user's identity. When the identity provider is satisfied with the user's legal standing, the IdP will send the user ID along with a card or certificate back to the specific service provider. If the card is still active, the provider of the service will fulfil the user's request in order to be of assistance. An SP will not be able to afford to provide its services to a requesting user if the identity provider (IdP) is either too busy to process the incoming request or is experiencing technical difficulties.

The fact that traditional authentication methods need a regular public key cryptosystem is the primary disadvantage of using such methods. The algorithms that are used in traditional public key cryptosystems require a large key size and make substantial use of the computing power available. The vast majority of traditional authentication mechanisms are therefore not taken into account because the device has a limited computing capacity. In order to circumvent this obstacle, the security system ought to make use of more recent and developing strategies, as these will produce better results. Bilinear pairing is utilised to avoid involving a third party and to reduce the amount of time spent on processing in order to reduce customer frustration. In addition to this, it offers improved capabilities for both mutual authentication and the exchange of keys.

Because the trusted third party is not required to be involved in the regular user authentication session when using this bilinear pairing, the total processing time for authentication can be reduced, resulting in a shorter overall time. A dependable Smart Card Generator (SCG) service is used to provide support for an environment consisting of cloud services distributed across multiple locations [22]. Figure 3 depicts the steps that users and service providers need to take in order to register their information with Smart Card Generator.

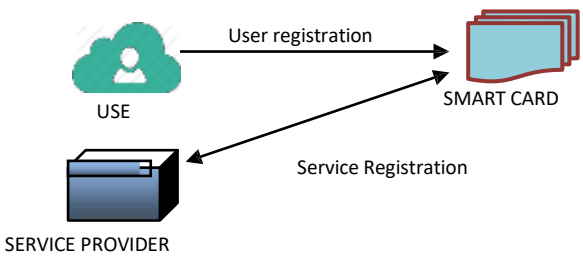

Fig. 3. Registration Phase using SCG

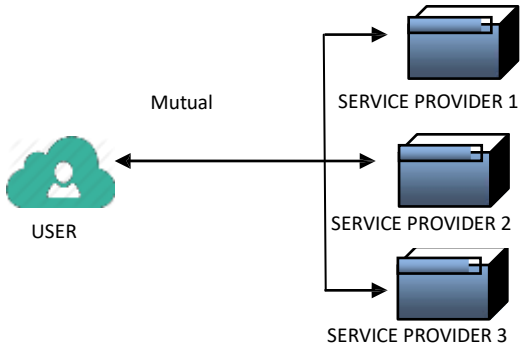

Fig. 4. Authentication Phase using SCG

Figure 4 depicts the authentication process of user with multiple service provider.

Users, companies that provide users with cloud services, and a reliable SCG service make up the three components of this SCG scheme. The SCG, also known as the Smart Card Generator, is a reliable third-party generator that can be used in place of the IdP service. It is responsible for issuing a smart card or a key in a secure manner to each and every user who registers during the registration process. This concept takes into account not one but three separate procedures: setting up the system, registering users, and authenticating users. During the phase in which the system is being configured, the SCG will select a master private key at random, compute the public key, and then produce all of the key's public parameters. After that, the SCG reveals both its public key and each and every one of its public parameters. After the initial phase of setting up the system, the next step is the registration phase, which is carried out between the SCG and the users and service providers.

Both the user and the service provider are required to enrol by presenting their own identification documents to the SCG. After receiving the identities, the SCG will begin working on it and will then securely distribute the encrypted private keys to the registered users as well as the service providers. When a user requests a service, the authentication procedure between the user and the service provider is ultimately brought to a successful conclusion. The authentication procedure is carried out between the user and the service provider of their choosing, without the participation of a third party like SCG. The session key that was obtained during the authentication phase is then used to encrypt and decrypt messages that are being transmitted between the user and the service provider after the authentication phase has been completed.

The primary advantage is that an efficient cryptosystem can support mutual authentication and user ambiguity even in the absence of SSL in the authentication process. This is a significant step forward from the current state of affairs. Instead of participating in both the registration and authentication phases of the process, the trusted third party will only take part in the registration step so that the processing can be completed more quickly.

C. CONTRIBUTIONS

Even though the Smart Card Generator shortens the amount of time needed for computation and provides security as well as user convenience, there is still a possibility that messages could be compromised due to the numerous security risks. One of the most significant dangers is phishing. If you only use one password for all of your online accounts, you leave yourself open to phishing scams. The authentication strategy should ideally include protection against phishing attacks. After successfully logging in, users are redirected to a bogus website where their passwords are stolen. The Password Hash Chain Generator (PHCG) creates a series of hash values so that the login process does not require the use of a single password. This helps to ensure that the system is secure. The hash chain is an encryption method that uses advanced hashing encryption to produce a sequence of hash values for a single password (AHE). This AHE merely adjusts the hash value; it is not necessary to use a verification table with it. If the hash chain is provided rather than the original password, the phishing attack can be thwarted.

II. FRAMEWORK FOR PROTOCOLS

The primary focus of this research is on the ecosystem supporting distributed cloud service deployments. The proposed system is comprised of three distinct roles. Cloud service providers, mobile users, and reputable third parties are included in this group. In this particular instance, the reliable third party is the smart card generator (SCG). This particular SCG covertly generates one private key or private id for each requested service provider and user while the registration step is being completed. In addition, the authentication process allows the user to access multiple computing services provided by a variety of service providers while only requiring the use of a single private key. Setup of the system, registering users, and authenticating users are the three stages that make up the proposed method.

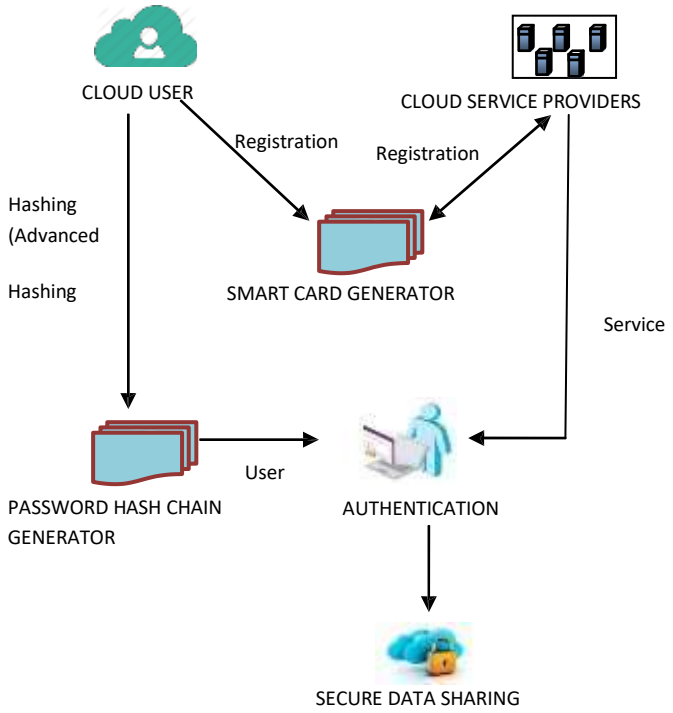

Fig. 5. Authentication scheme-framework.

During the phase in which the system is being set up, the SCG generates its master public key by selecting an integer at random. It then distributes both its public parameters and its key. Following the completion of the setup phase comes the registration phase. During this phase, any user who will be utilising the computing service as well as any service provider will be required to register with a Smart Card Generator. This smart card generator issues a private ID and key to each and every one of the company's customers as well as service providers. It is recommended that users make use of a Password Hash Chain Generator, which works to protect users from phishing attacks by producing a chain of hash values connected to passwords. During the authentication phase, it is possible for both the user and the service provider to authenticate one another without the utilisation of a smart card generator. Figure 5 illustrates the foundation that the Password Hash Chain Generator uses for the authentication phase of the process. At long last, the user and the service provider are able to trade confidential information.

Third in a series of password hash chain generators

The next section will go over the details of the authentication approach that has been suggested for use in an environment that makes use of distributed cloud services. The Password Hash Chain Generator is an excellent tool for providing protection against phishing assaults. System setup, user registration, and user authentication are the three stages that make up this process.

The following is the algorithm for the password hash chain generator, beginning with the phase that deals with setting up the system:

The system is capable of accommodating a large number of users as well as Service Providers. In addition to that, there is a Smart Card Generator that operates in the capacity of a third party. A user in this system can gain access to numerous Service Providers that provide a variety of cloud computing services by only needing to remember a single private key. This particular private key was generated by the Smart Card Generator. One of the most important capabilities of the Smart Card Generator is its capacity to generate the public key and to distribute the public parameters while the system is being configured.

Registration Phase:

First, the user must provide his or her own personal details in order to register with the Smart Card Generator (SCG). These details include the user's name, gender, email address, password, and user identification number.

Step 1 Once the Smart Card Generator has received the user's personal information, it will generate a private key and a private id for the user, both of which will be sent to the user in an encrypted format.

Step 2: The next step is for the service provider to register with the Smart Card Generator by providing their personal information. This information includes the Datacenter Name, Service Name, Email ID, and Password, as well as the Provider ID.

After the user has registered with the service provider, the Smart Card Generator will then send that user's public key to the service provider.

Authentication Phase:

Step 3: After the registration phase, if the user wants to access a specific service provider, then the user sends Name, User ID> directly to the service provider without the assistance of the Smart Card Generator, i.e. without the involvement of the third party. This step ensures that the user is the only one who has access to the information. This reduces the amount of time required for the computation.

The user's id is given to the user after it has been encrypted by the service provider in the fourth step of the process.

Step 4 requires the user to decrypt the encrypted id; if the decrypted id corresponds to the user id, the service provider is authenticated; otherwise, the service provider is not authenticated.

/*Service Provider*/

Authentication ()

User sends UserID → Service Provider

Service Provider (encrypt the UserID)

Service Provider sends encrypted UserID →User

User (decrypted the UserID)

If (decrypted ID = =UserID) **then**

Service Provider = authenticated

Else

Service Provider = not authenticated

End if

Step 4: The Elliptic Curve Cryptography (ECC) algorithm is used to carry out both the encryption and the decryption processes.

In the fifth step, the user transmits the password to the Password Hash Chain Generator (PHCG). The hash value is then sent to the service provider by this PHCG after it has been generated.

Step 5: The service provider will receive the equivalent password that was recovered from the hash value with the assistance of the PHCG and the hash value that was provided by the user.

Step 6: The user is considered authenticated if both of the hash values are the same; otherwise, the user is not authenticated.

/*User*/

Authentication ()

User sends Password → PHCG

PHCG (generates Hash Value)

PHCG sends Hash Value → Service Provider

User sends equivalent Hash Value → Service Provider

If (User Hash Value == PHCG Hash Value) **then**

User = authenticated

Else

User = not authenticated

End if

Step 5.3: Each time when the user login for a particular service provider, the hash value will be different so that it prevents phishing attack.

Step 6: This sequence of hash value generated by the PHCG is done by Advanced Hashing Encryption (AHE) algorithm.

Algorithm :- AHE

Step 6.1: This AHE algorithm provides an encrypted sequence of hash values to be used in subsequent steps.

Step 6.2: The fact that this algorithm does not require the maintenance of verification tables is the primary benefit it offers over other algorithms such as AES and hash-based password algorithms, amongst others. owing to the fact that the cost of maintaining the verification table is increasing.

This step of the algorithm only displays the number, which indicates the number of times the hashing sequence has been completed, and it also displays the result.

only the updated hash value not all the hash values.

Step 7: After the authentication phase, a secure data sharing is done between the user and the service provider.

IV. ANALYSIS OF PERFORMANCE

In the following section, we will evaluate the functionality of the Password Hash Chain Generator (PHCG). Cloud Simulator version 3.0.3 is what's being utilised for the PHCG implementation. In this scenario, each user must first register with the smart card generator, which then generates a private key and sends it to the appropriate user. This process repeats until all users have received their private keys. In a manner analogous to that which Service Provider utilises in order to register with Smart Card Generator, Smart Card Generator will only divulge specific user information to Service Provider in order to prevent user confusion. Following registration, the user will be required to authenticate themselves before being granted access to the Service Provider that does not require SCG. This process shortens the amount of time spent computing while simultaneously increasing the user's level of uncertainty. In Figure 6, a comparison is drawn between the system with and without SCG. In this illustration, the system that utilises SCG requires less time to compute but results in a higher level of user uncertainty in comparison to the system that does not utilise SCG, which requires more time to compute but results in a lower level of ambiguity

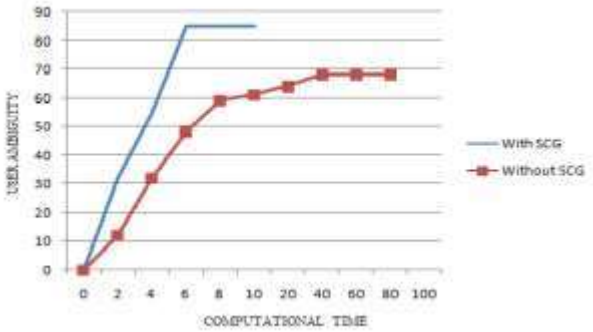

Fig. 6. Comparison of the system with and without SCG

The user communicates their ID to the provider of the service, who encrypts it before sending it back to the user. If both ids are the same, the user is able to recognise the service provider and decrypt the message. This encryption utilises an algorithm that is derived from elliptic curve cryptography. After that, PHCG will generate a string of hash values based on the password that the user has provided.

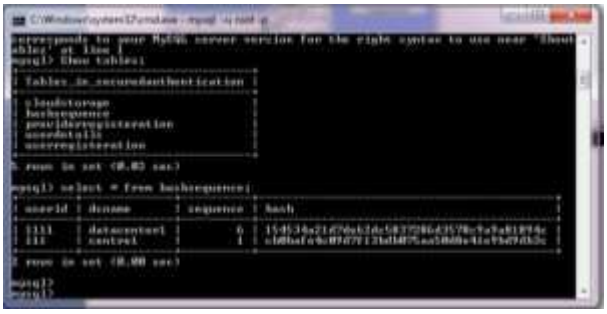

Fig. 7. The sequence number and the most recent hash value

Encryption of the data in this PHCG is accomplished with the help of the Advanced Hashing Encryption (AHE) algorithm. This AHE does not provide the verification table; all that is displayed is the current value as well as the number of times the sequence has been created. As a direct consequence of this, maintenance expenses are decreased. Figure 7 displays the updated hash value along with the sequence it generates. If the hash value generated by the PHCG is identical to the user's own hash value, then the user is considered to be authorised. If the user and the service provider can both be proven to be who they say they are, then a secure data exchange can take place between the two parties. Through the use of Cloud Simulator, the user sends an encrypted document file to the service provider after initially uploading it to the cloud. Figure 8 provides a visual representation of the user's encryption process.

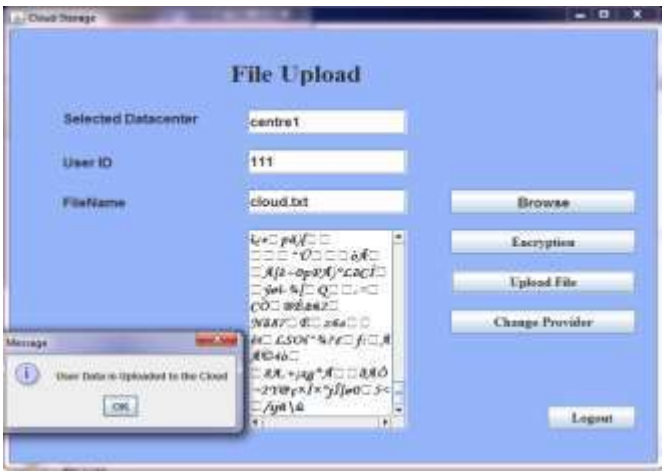

Fig. 8. Encryption process.

The Service Provider decrypts the file and reads it. This data sharing is a secured one. Figure 9 illustrates the decryption process done by the Service Provider.

Fig. 9. Decryption process

V. CONCLUSION AND WORK TO BE DONE IN THE FUTURE

When it comes to cloud computing services, security concerns almost always surface whenever users of the cloud make an attempt to access the cloud. In the course of this piece of research, a novel authentication strategy is implemented for the context of cloud services that are geographically dispersed. Cloud users will only need to remember a single private key in order to access a wide variety of cloud services provided by a number of different Service Providers if the system that is currently being proposed is implemented. It is the responsibility of the Smart Card Generator to generate this one private key, which helps to cut down on the amount of time spent computing. The Elliptic Curve Cryptography (ECC) algorithm is utilised in this situation so that the information can be encrypted.

When it comes to reducing the amount of time needed to complete a task, the performance evaluations make it abundantly clear that the Smart Card Generator is a great deal more effective than other reputable third parties. In addition, a programme known as the Password Hash Chain Generator (PHCG) has been developed in order to defend users against phishing attacks. This system employs a cutting-edge form of encryption called hashing (AHE). This PHCG will, with the assistance of AHE, generate a string of hash values that will be used for the password. Because the AHE in question does not require a verification table, the associated maintenance costs are lower than they would otherwise be. In the not too distant future, tamper proofing will be finished in order to deliver a higher level of safety to the plan that has been laid out.

REFERENCES

[1]. "Security architecture for Cloud computing," by Ali Newaz Bahar, Md. Ahsan Habib, and Md. Manowarul Islam, published in International Journal of Scientific Knowledge, Volume 3, Issue 3, ISSN 2305-1493, July 2013.

[2]. H. Ahn, H. Chang, C. Jang, and E. Choi, "User authentication platform employing provisioning in cloud computing environment," in Proceedings of the ACM Conference on Computer Communications and Information Security, Volume 199, Pages 132–138, 2011.

[3]. W. Itani, A. Kayssi, and A. Chehab, "Privacy as a service: Privacy-aware data storage and processing in cloud computing systems," published in the proceedings of the IEEE International Conference on Dependable Autonomous Secure Computation, pages 711–716, December 12, 2009.

[4]. Jaydip Sen wrote an article titled "Security and Privacy Issues in Cloud Computing" that was published in the Innovation Labs of Tata Consultancy Services Ltd. in Kolkata, India. [Online] Available:http://arxiv.org/ftg/ arxiv/papers/1303/1303.4814.pdf.

[5]. H. Takabi, J. B. D. Joshi, and G. Ahn, "Security and privacy concerns in cloud computing systems," published in IEEE Security Privacy, Volume 8, Number 6, Pages 24–31, November/December 2010.

[6]. The following article provides a straightforward explanation of elliptic curve cryptography (ECC): https://bithin.wordpress.com/2012/02/22/ simple-explain-for-elliptic-curve-cryptography/.

[7]. Elliptic curve cryptosystem by N. Koblitz, published in Math. Comput., Vol. 48, No.177, pages 203-209, 1987.

[8]. V. Miller authored an article titled "Use of Elliptic Curves in Cryptography," which was published in Proc. CRYPTO, pages 417–426, in 1986.

[9]. Don Boneh and Matthew Franklin, "Identity-based Encryption from Bilinear Pairing," pages 155-159, published in June of 2011.

[10]. An Efficient Signature Scheme from Bilinear Pairings and Its Applications by Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo was published in Springer Berlin, Volume 2947, Pages 277-290, PKC.2004.

[11]. Ben Lynn, "An Implementation of Pairing-Based Cryptosystems," a Dissertation Submitted to the Department of Computer Science and the Committee on Graduate Studies at Stanford University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, June.2007. [11] Ben Lynn, "An Implementation of Pairing-Based Cryptosystems,"

[12]. According to Dennis Meffert's "Bilinear Pairings in Cryptography," which was presented at the Radboud Universiteit Nijmegen Computing Science Department, Toernooireld 1, 6525, ED Nijmegan, on May 24th, 2009, [citation needed]

[13]. H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-based Authentication for Cloud Computing," in Proceedings of CloudCom, 2009, pages 157-166. [Citation needed]

[14]. Hongwei Li, Yuanshun Dai, and Bo Yang, "Identity-Based Cryptography for Cloud Security," in Hongwei Li, Yuanshun Dai, and Bo Yang (Eds.), [0nline]. You can access this document by going to http://eprint.iacr.org/2011/169.pdf.

[15]. T. H. Chen, H. L. Yeh, and W. K. Shih, "An enhanced ECC dynamic ID based remote mutual authentication strategy for cloud computing,"

in Proceedings of the Fifth Annual FTRA International Conference on Multimedia and Ubiquitous Engineering, pages 155–159, 2011. [Citation needed]

[16]. An improved bilinear pairing based remote user authentication technique was published by T. Goriparthia, M. L. Das, and A. Saxena in the journal Comput. Std. Interfaces in January 2009, Volume 31, Number 1, Pages 181–185.

[17]. [Online] G. Shailaja, K. Phani Kumar, and Ashutosh Saxena have written an article titled "Pairing-based Mutual Authentication Scheme Using Smart Cards." Available: http://eprint.iacr.org/2006/152.pdf.

[18]. The article "A Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services," written by Jia-Lun Tsai and Nai-Wei L, was published in September 2015 in Volume 9, Issue 3 on pages 805-815.

[19]. "Secure User Authentication in Cloud Computing Management Interfaces," by Liliana F. B. Soares, Diogo A. B. Fernandes, Mário M. Freire, and Pedro R. M. Inácio, published in IEEE 32nd International Conference on Performance Computing and Communications Conference (IPCCC), pages 1-2, December 6, 2013.

[20]. 2008 Google publication titled "SAML Single Sign-On (SSO) Service for Google Apps."[Online].You can get it at this location: https://developers.google.com/ googleapps/sso/saml reference implementation?hl=zh-tw.

[21]. An authentication issue in browser-based single sign-on protocols: Impact and remediations, by A. Armando et al., published in Computer Security, Volume 33, Pages 41–58, March 2013.

[22]. [Online].Available:http://searchsecurity.techtarget.com/definition/smart-card.