A Secure Single Key-based Authentication Server using Face-Recognition in Cloud Computing Services

Shilpa D
M.Tech Scholar
Department of Computer Science & Engineering
Don Bosco Institute of Technology

Madhusudan M I
Assistant Professor
Department of Computer Science & Engineering
Don Bosco Institute of Technology

Abstract— **In this paper, an efficient authentication scheme for distributed cloud computing services is proposed. The proposed scheme provides security and convenience for users to access multiple cloud computing services from multiple service providers using only a single private key. The security strength of the proposed scheme is based on bilinear pairing cryptosystem and dynamic nonce generation. In addition, the scheme supports mutual authentication, key exchange, user anonymity, and user untraceability. From system implementation point of view, verification tables are not required for (AS) service and cloud computing service providers when adopting the proposed scheme. In consequence, this scheme reduces the usage of memory spaces on these corresponding service providers. In one user authentication session, only the targeted cloud service provider needs to interact with the service requestor (user). The trusted Authentication server serves as the secure key distributor for distributed cloud service providers and clients.**

Keywords—Cloud, Key Exchange, User Anonymity, User Untraceability

I. INTRODUCTION

Cloud computing is a model that provide on time and on-demand computational services with suitable price for users. As an evolution of service-oriented architecture and virtualization, the cloud computing potentially provides infinite computational capabilities for its customers, with a method of pay-as-you-consume, which is far more different from the traditional way of network computing. This is an exciting combination for many researchers. Some of them even predict that it might inspire our research in computing over the next decade and beyond. mobile and handheld devices are constrained due to resource limitations primarily caused by limited battery life requiring recharging, constrained size of memory or limited power of the processor especially during roaming and challenge of being seamlessly connected throughout mobility or even limited size of physical persistent storage. Execution of high computational tasks in a mobile device may also drain the battery power very quickly. To address these limitations of mobile devices, cloud computing can be an obvious choice, This means that mobile users offload processing intensive and storage demanding portion of mobile application from resource constraint mobile device to resource enriched cloud. The offloading of the processing intensive and storage demanding portion(s) of mobile application enhances the capabilities of mobile devices in term of processing, storage, and battery Cloud computing with its pay-per-use mode alleviates the in-house computing cost and thereby stands as an obvious choice for global enterprises targeting cost optimization yet having flexible, secured and efficient use of IT resources. The main objectives of the mobile cloud computing are to increase the processing/storage capabilities of the mobile device and reduce the energy consumption while executing the computationally intensive jobs [1].

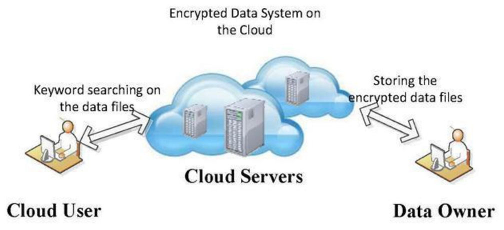
Encrypted Data System on the Cloud
Keyword searching on the data files
Storing the encrypted data files
Cloud Servers
Cloud User
Data Owner

Figure 1: Cloud Computing

The development of mobile cloud computing has become an important research field in mobile-oriented world, providing new supplements, consumption, and delivery models for IT services. As reported by ABI Research, more than 240 million business customers will be leveraging cloud computing services through mobile devices by 2015, driving revenues of $5.2 billion. In mobile cloud computing, mobile users can access computation results, resources, applications, and services that are stored, implemented, and deployed in cloud computing environments by using mobile devices through an insecure wireless local area network (WLAN) or 3G/4G telecommunication networks. When a user intends to access a mobile cloud computing service, he/she activates the Manuscript received April 24, 2013; revised January 15, 2014; accepted April 2, 2014. This work was supported in part by the Taiwan Information Security Center and in part by the National Science Council of Taiwan under Grants service through a Web browser or a cloud service application (i.e., App) installed on his/her mobile device. TheWeb browser or the cloud service application will then mutually authenticate both the cloud service provider and the user. After authentication, the user can access the resources and available services from the cloud service provider. In order to prevent illegal access, cloud providers should support a secure authentication scheme for users using mobile devices. However, there are three concerns to be resolved along with the authentication scheme. First of all, computing efficiency of the scheme should be seriously considered, since mobile devices have only relatively limited computing capability in comparison with laptop computers. Second, sufficient security strength should be supported; since all messages are transmitted via an insecure WLAN or telecommunication networks, an adversary can easily obtain, interrupt, or modify transmitting messages before they reach the desired recipient. In addition, privacy protection on user accounts is a rising issue as identity masquerade and identity tracing have become common attacks in wireless mobile environments. As mobile users generally access different types of mobile cloud computing services from a variety of service providers, it is extremely tedious for users to register different user accounts on each service provider and maintain corresponding private keys or passwords for authentication usage. In other words, key management issue for users has emerged for distributed mobile cloud environment. In consequence, mobile users will likely be interested in how to access various services from distinct mobile cloud service providers by using only one single private key or password.

II. RELATED STUDY

Smart phones and other mobile devices are heavily used in today's world and still get even more important since the usage of mobile internet. The growth of the number of applications available for those devices in the last years has shown that there is a high demand for mobile applications [1]. However one common problem that all those devices share, still needs to be addressed: the limited capabilities of the devices regarding available resources like processor power, available memory and especially energy consumption. A technology recently emerged in the IT industry offers an opportunity to solve those problems: Cloud computing (CC) gives its users the possibility to host and deliver services over the internet by dynamically providing computing resources [13].Cloud computing eliminates the requirement for users to plan ahead for acquiring different resources, such as storage and computing power, and therefore, is attractive to business owners. Moreover, enterprises can provide resources depending on service demand. Whereas, Cloud computing is emerged as the modern technology which developed in last few years, and considered as the next big thing, in the years to come.

Since it is new, so it require new security issues and face new challenges as well [1]. In last few years it is grown up from just being a concept to a major part of IT industry. Cloud computing widely accepted as the adoption of virtualization, SOA and utility computing, it generally works on three type of architecture and these are: SAAS, PAAS, and IAAS. There are different issue and challenges with each cloud computing technology. The various security concerns and upcoming challenges are addressed in [2], [4], and also reviewed in terms of standards such as PCI-DSS, ITIL, and ISO-27001/27002. Until now there is no such standard is available regarding service or operational functioning, and its security is a major concern.

There are also the architectural security issues which are changing according to various architectural designs functioning over cloud computing [3]. Various surveys are in the market depicting the current scenarios such as the leading US research firm Gartner released a report "Assessing the security risk of cloud computing" in June 2008, this report raises the concern about risk in data storage, data recovery, data privacy, and data integrity [5]. Cloud computing providing services in layered medium, so there must be some SLA (Service Level Agreement) or service management, must be applied over the layers, which eventually increase the confidence of the user. Data security over the cloud also a major concern and various methodologies are proposed [6], also privacy preserving auditing for the data storage security in cloud computing [7], raising the concern over the privacy related issues in data storage [8], such that no critical information can be intercepted as recently a case happened with Wikileaks, over the security of the data. Apart from data security, security management, and security risk management frameworks are also proposed [9], [10] addressing risk associated with cloud computing, and activities are planned in such a way ensuring that information is available and protected by applying Deming model or PDCA to curb the risk related to cloud computing security. Cloud computing works in layers as applying policies on these layers provide better security approach to manage the security concerns [11]. Cloud computing has given a new horizon to the data hosting and deploying services. The most important thing of cloud computing is that it enables customers a new way to increase capacity and add capability to their machines on the go. In last to utilize the cloud computing in Mobile and to take full advantage of it, users required optimized security solutions and assurance regarding the security issues and risk involved in information flow.

Authentication scheme is a basic security mechanism for all network-based services to prevent illegal access from unauthorized users or adversaries. Traditional authentication schemes are usually based on traditional public key cryptosystem. Traditional public key cryptosystems such as RSA require lengthy key size and consume computation resources heavily. Hence, most of traditional authentication schemes are unsuitable for mobile devices, which have limited computing resources. Elliptic curve cryptosystem (ECC), which was first introduced by Koblitz and Miller , offers the smallest key size per equivalent strength of any traditional public key cryptosystem, including RSA and Discrete Logarithm Problem (DLP). For example, a 256- bit ECC public key has the same security level as a 3072- bit RSA public key . Such computational efficiency is beneficial for mobile devices. Recently, bilinear pairing in an elliptic curve has been used in developing an ID-based cryptosystem . Since then, several ID-based cryptosystems have been proposed. An IDbased cryptosystem is one kind of public key cryptosystems that can solve the high cost issue of public key management and authentication derived from traditional public key cryptosystems. In an ID-based cryptosystem, the identity of a user is used as the public key of this user; a user therefore does not spend extra computational cost to verify public keys of others, and no extra storage space in the user's device is required to store public keys of others and their corresponding certificates. Several studies have applied ID-based cryptosystems in cloud and grid computing environments. Lim and Robshaw, first applied an ID-based cryptosystem to grid security in 2004, whereas in the same year, Mao proposed an identity-based. noninteractive authentication framework for grids. In 2009,Li et al. [32] developed a new ID-based authentication for cloud computing environment. However, the authentication protocol of Lin et al. does not provide user anonymity and untraceability.

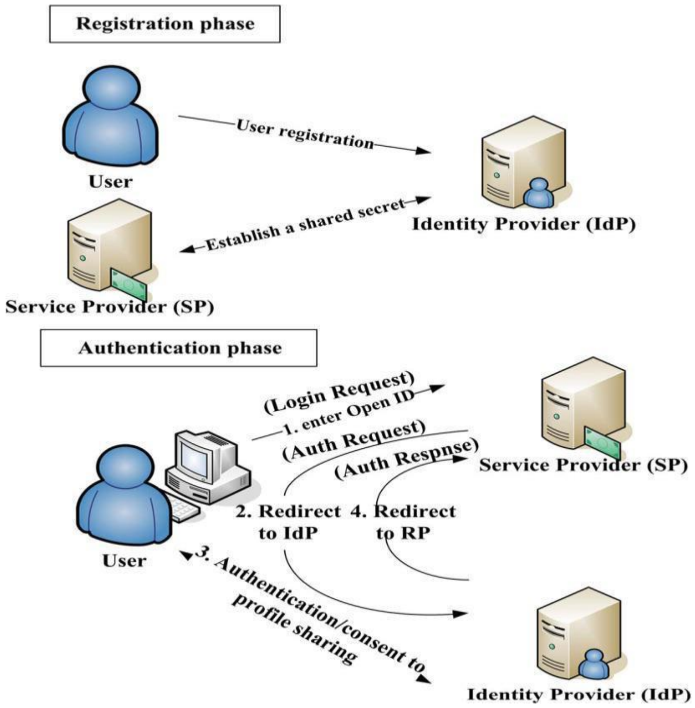

Figure 2: User Authentication Process

III. PROBLEM STATEMENT

The Various Issues and the Problems related to Cloud Computing us as summarized below points:

1. Passwords are great, but they are hard to enter on a small form factor (i.e., mobile) device. Also, harder passwords are more secure, but they are also very difficult to input, and more likely to be "written down" somewhere.
2. Signature tokens are a great way to protect communication from a secure computing device to the cloud, but smart mobile devices are always in an uncontrolled environment. Because of this, token leakage is possible.
3. Any increase in security usually leads to a corresponding decrease in usability; since the smart mobile platforms lack the computational capabilities of a traditional laptop, our options are further limited.
4. I can't presently firewall my iPhone, monitor its network activity in-situ, or run intrusion detection on the device without a significant degradation in overall performance.
5. Unlike my credit card, if my data-plan is exceeded due to a compromise, or my cloud service is compromised due to a leaked signature key, there is no comfort in having a maximum liability amount. Both assume you will secure the asset and the responsibility lies solely and wholly with the customer

IV. PROPOSED METHODOLOGY

This paper assumes that the distributed cloud service environment is supported by authentication server. Three roles take part in the proposed scheme: users, distinct cloud service providers, and a authentication service. Notice that the trusted third party used in the proposed scheme is named as the authentication server rather than the IdP service. We assume that there are many users and service providers within distributed cloud services environment, and a small portion of etc. these users and service providers are malicious. users, service providers, and the trusted AS are denoted. The proposed scheme includes three phases: system set up, registration, and authentication. During the system set up phase, the AS first selects a random number as its master private key, computes the corresponding public key, and generates all public parameters. Then, the AS publishes its public key and public parameters. After accomplishing the system set up phase, the registration phase is executed between the AS and each one of the users (or service providers) who wishes to join and utilize the authentication service.

Basic Authorization has been around for thousands of years - think "open sesame." It employs the authentication concept of a shared secret. If there is something only I and my communicating endpoint know, and we verify the knowledge of that secret, we have a substantial basis for authentication. There are a number of human factor issues with Basic Auth. The biggest comes from the fact that a password that is sufficiently complex to be secure is usually hard to remember. There is a reason for that. Consider that the mechanisms we use for associativity are easily interrogated and analysed by computers trying to compromise an account. Combine this with the fact that sufficiently hard passwords are difficult to enter on a small form factor device and we have an authentication challenge for mobile devices using basic auth.

Signature tokens allow me to simultaneously authenticate and verify the integrity of the message. This combination is key to the nonrepudiation aspect of securing cloud and "X"aaS services where X is (P)latform, (S)oftware, (H)adoop, etc. In this signature token-based auth/auth model the user is responsible for protecting the secrecy of the signature token. Encrypting the tokens, utilizing them for only the duration of your cloud interaction, then completely destroying them by overwriting their memory location.

Open ID has some attractive features, namely it and OAUTH both allow an application to authenticate once and authorize across cloud servers or other web-enabled services. This is a nice feature for mobile-cloud computing, and its biggest challenge is its token dependence. Its advantage over just a plain signature token is it ostensibly has a shorter lifetime of validity. Having an authorization or signature token is a legacy artifact of not having any objective information about a user that would identify that user uniquely. My laptop contains very little contextual information to form a context-aware authentication mechanism against, though there are technologies to enable this.

A signature token in either system is a state management mechanism that lets the application know that this access has been previously authenticated. The storage of the authorization token is as essential in the open ID model as it is in the storage of the signature token in Signature Token auth/auth environments.

V. SYSTEM MODEL

The system chooses a large prime integer to form a Diffie-Hellman group, and generator # of group $% &, i.e., # is a primitive root modulo. Normally is a Sophie Germain prime where is also prime, so that the group $%& maximizes its resilient against square root attack to discrete logarithm problem. A certificate authority (CA) as in PKI is still needed in our security framework so that communicating parties can identify each other through exchanging verifiable certificates and !" , as the certificates contain public keys which can be used to verify the session partners' signatures, thereby their identities. Certificates are relatively long termed data which are issued to all participants of communication before the commencing of communication, and CA won't be participating itself unless re-verification of identities and revocation and re-issuing certificates for participants are needed. As these should be done in a much lower frequency (e.g. once a day) than key exchanging (e.g. reexchanging key in every new session), they won't affect the efficiency of a key exchange scheme for scheduling in general. Therefore, we will ignore all communications involving CA in our scheme and won't be discussing further details on issuing and revoking certificates.

MCKE Initial exchange:

Initial exchange is used when a new task is to be executed, because that is when CLC need to decide how to distribute this new task to be executed on existing computation infrastructure, i.e., which of the server instances are involved. CLC picks a secret value x < p,computes its public keying material gx in Zp, and broadcast the following message to the domain of server instances S which contain n instances S1,...S:Round1, c-> S: HDRc, SAc1, gx,Ncwhere HDR and SA for algorithm negotiation, gx for Diffie-Hellman key exchange, and for freshness verification.

The initiator of a normal IKE scheme will generate n secret x1, x2, x3... xn, then compute and send out gx1 , gx2....gxn ,either through multicast or one by one, to establish separated security channels with each receiver. In our scheme, although we still establish one for each server instance where i=1,2,..n, we are using only one single secret value0' for CLC in all * messages in order to reduce cost. We will further analyse security and cost reduction for this variation in section 4 and 5, respectively.

The session keys are now shared between CLC and each server instance for the use of encryption of later communications. Although the Diffie- Hellman key exchange is completed, the MCKE initial exchange is not finished as the participants have to authenticate each other in order to prevent man-in-the-middle

(MITM) attacks. Similar as in IKE, CLC generates signatures " which are the signatures for these * messages, using its secret keyfrom the key pair issued by CA and broadcast the following message to S:

Round 2, *C* ->
S HDR:
HDRc, {IDc, SAc2, Certc, CerReqs1
Sig} gxy1 for i =1,2....n

The server instances can then verify the identity of the initiator of this conversation by using its session key #)7" to decrypt its own part of this message. Signatures can be verified through the public key contained in the certificate.

Similarly, server instances will send out their own encrypted ID, signature and certificate to CLC for verification:

Round 3, *S* -> *C*: DRst{IDc,SAc2,Certc,Sig}gxy
for i=1,2,.....,n

where, similar to round 2 but only signed separately, Sigst is signatures by Si to messages:

Mst = prf(prf(Nc||
Nst||gxy)||gyi||gx||IDsi) for i = 1, 2, ...n

Note that this round involves * messages as well. After the identities of both CLC and server instances are authenticated through round 3 and 4, CLC will send to S1, S2,Sn, the split task data which are encrypted with session keys gxy1,..gxyn using symmetric encryption such as AES. After task execution, returns to CLC the results which are encrypted using S1, S2, Sn as well. The prf function is often implemented as an HMAC function such as SHA-1 or MD5, which outputs a fixed-length short message (commonly 128 bits) and has high efficiency (around 200MB/s on today's desktop PCs) itself.

Rekeying:

In a multi-step task data need to be transferred back and forth in a multi-step task. In this situation it is not necessary to re-authenticate because of the high data dependency in a same task. Therefore, only rounds 1 and 2 are needed to be performed, with new keying materials and minor changes to *SA* and *HDR* fields and. As rounds 3 and 4 only contains fast operations such as signature and verification over short messages as well as symmetric-key encryption/decryption and HMAC functions, the computational overhead of rekeying process is almost identical to the initial exchange from an efficiency prospect of view. We'll further analyse this in the section of performance evaluation.

ANALYSIS

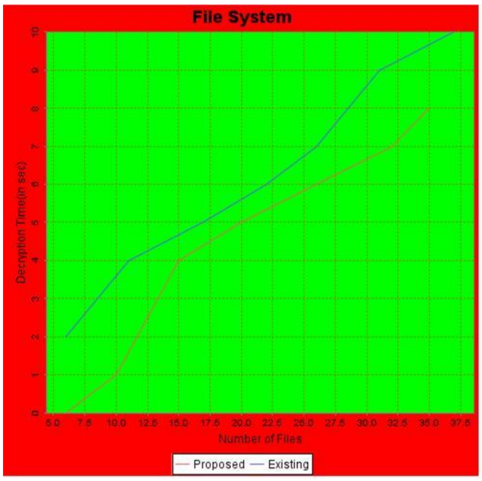

Figure 1 :Decryption time with number of files

Uploading file :

User will login providing credentials authentication server captures users image and particular user image is sent to verification after successful verification. Signature tokens allow me to simultaneously authenticate and verify the integrity of the message. Computes the corresponding public key, and generates all public parameters. User can upload file to the cloud once the mutual authentication is done. The files are encrypted and sent later user can decrypt to use the content.

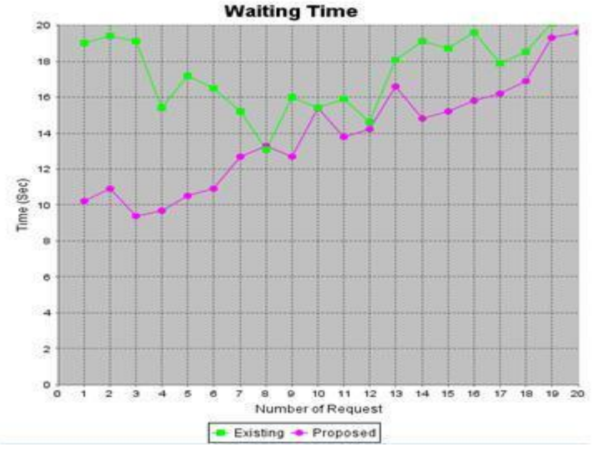

Figure 2:Waiting time with number of request

VI. CONCLUSION

This paper has proposed a new anonymous authentication scheme for distributed cloud services environment. The proposed scheme allows a user to access multiple services from different cloud service providers using only one single private key. The proposed scheme supports mutual authentication, key exchange, user anonymity, and user untraceability. Security analyses have shown that the proposed authentication scheme withstands all major security threats and meets general security requirements. In addition, no verification table is required to be implemented at service providers or the trusted AS service. In the proposed scheme, the trusted AS service is not involved in individual user authentication process. With this design, our scheme reduces authentication processing time required by communication and computation between cloud service providers and traditional trusted third party service. As security strength of the proposed scheme is based on nonce and bilinear pairing, the scheme itself is not subject to time synchronization problem and can be easily implemented in distributed cloud computing environment

REFERENCES

[1] G. S. E. Deelman, M. Livny, B. Berriman, J. Good, , in: , "The cost of doing science on the cloud: the montage example," in *ACM/IEEE Conference on Supercomputing (SC '08)*, Austin, Texas, 2008, pp. 1– 12.

[2] Balachandran reddy etal Meiko Jensen etal On technical security issues in cloud computing, , 2009

[3] Cong Wang, Ensuring Data Storage security in cloud computing etal, 2010

[4] John harauz, Data security in the world of cloud computing, etal, 2010

[5] JASON CHRISTENSEN *Part 1: Mobile Cloud authentication and authorization,Cloud Computing Journal*

[6] Kemp, R., N. Palmer, T. Kielmann, and H.Bal, Cuckoo: a Computation Offloading Framework for Smartphones, in Proceedings of the Sixteenth annual conference of the Advanced School for Computing and Imaging 2010. 2010: Veldhoven, the Netherlands. p. 70-77.

[7] Mei, C., J. Shimek, C. Wang, A. Chandra, and J. Weissman, Dynamic Outsourcing Mobile Computation to the Cloud. 2011, Department of Computer Science and Engineering, University of Minnesota: Twin Cities.

[8] Mell, P. and T. Grance, NIST SP 800-145. The NIST Definition of Cloud Computing (Draft). Recommendations of the National Institute of Standards and Technology. 2011.

[9] K.-W. Park, S. S. Lim, and K. H. Park, "Computationally Efficient PKIBased Single Sign-On Protocol, PKASSO for Mobile Devices," *IEEE Transactions on Computers*, vol. 57, pp. 821 - 834, 2008.

[10] L. Wang, J. Tao, M. Kunze, A. C. Castellanos, D. Kramer, and W. Karl, "Scientific Cloud Computing: Early Definition and Experience," in*High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on*, Dalian, China, 2008, pp. 825 - 830.

[11] J. Yao, S. Chen, S. Nepal, D. Levy, and J. Zic, "TrustStore: Making Amazon S3 Trustworthy with Services Composition," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGRID '08)*, Melbourne, Australia, 2010, pp. 600-605.

[12] D. Yuan, Y. Yang, X. Liu, and J. Chen, "On-demand minimum cost benchmarking for intermediate dataset storage in scientific cloud workflow systems," *Journal of Parallel and Distributed Computing*, vol. 71, pp. 316-332, 2011.

[13] J. Zhao and D. Gu, "Provably secure authenticated key exchange protocol under the CDH assumption," *Journal of Systems and Software*, vol. 83, pp. 2297-2304, 2010.

[14] Zhang, Q., L. Cheng, and R. Boutaba, Cloud computing: state-of the-art and research challenges. Journal of Internet Services and Applications, 2012, p. 7-10

[15] http://www.cioedge.com/content/statecloud- computing-security