An Efficient and Secured Data Storage Scheme in Cloud Computing Using ECC-based PKI

XiaoChun YIN*, ZengGuang LIU**, Hoon Jae LEE ***

*Department of Ubiquitous ITGraduate School of Dongseo University, Sasang-Gu, Busan 617-716, Korea; Weifang University of Science & Technology, Shouguang 262700, China

**IP Platform Department of MGC, Alcatel-Lucent, QingDao SongLing Road 169, China

***Division of Computer and Engineering of Dongseo University, Sasang-Gu, Busan 617-716, Korea

yinspring2012@gmail.com, hjlee@dongseo.ac.kr, sterling.liu@alcatel-lucent.com

***Abstract*— Cloud computing is set of resources and services offered through the Internet. Cloud services are delivered from data centres located throughout the world. Cloud computing facilitates its consumers by providing virtual resources via internet. The rapid growth in field of "cloud computing" also increases severe security concerns. Security has remained a constant issue for Open Systems and internet, when we are talking about security, cloud really suffers. Lack of security is the only hurdle in wide adoption of cloud computing. Cloud computing is surrounded by many security issues like securing data and examining the utilization of cloud by the cloud computing vendors. This paper proposes a scheme to securely store and access of data via internet. We have used ECC based PKI for certificate procedure because the use of ECC significantly reduces the computation cost, message size and transmission overhead over RSA based PKI as 160-bit key size in ECC provides comparable security with 1024-bit key in RSA. We have designed Secured Cloud Storage Framework (SCSF) . In this framework, users not only can securely store and access data in cloud but also can share data with multiple users through the unsecure internet in a secured way. This scheme can ensure the security and privacy of the data in the cloud.**

***Keywords*—Cloud storage, Cloud computing, ECC, PKI, Certificate**

I. INTRODUCTION

Cloud computing is the most demanded technologies used all over the world. It provides all kinds of services for the users. One of the most prominent service offered by cloud computing is cloud storage. Cloud storage is simply a term that refers to on line space that you can use to store your data. In more strict way, cloud storage is a service model in which data is maintained, managed and backed up remotely and made available to users over a network. Compared with hard disc storage, we can think cloud storage as some kind of network storage, different types of storage devices in the network work together through the cluster, grid or distributed file system functionality to provide the storage space for user.

The biggest concern about cloud storage is security. With cloud storage, users store their data to multiple third party servers. Users worry that data saved on a remote storage system is vulnerable. There's always the possibility that a hacker will find an electronic back door and access data. Hackers could also attempt to steal the physical machines on which data are stored. In another way, a disgruntled employee could alter or destroy data using his or her authenticated user name and password. Cloud storage companies invest a lot of money in security measures in order to limit the possibility of data theft or corruption. Users still aren't likely to entrust their data to the cloud provider without a guarantee that they can access their data information whenever they want and no one else is able to get it. Since all the data are in plaintext format, not only during the transferring between users and cloud servers but also during stored on the servers, the data faces security threat.

We propose a scheme to build a trusted cloud storage system, which allow the user to store and access their data securely in the cloud by encrypting the data in the client side and decrypting the data after down loading from the cloud. Since the private key is owned by the user of the data, no one can decrypt the data, even though hackers can get the data through some approaches. This scheme also allows the user to share the data with the authenticated users. If the owner of the data wants to share the data with some authenticated users, the owner only needs to save the data in the share data part and encrypt the data with his private key, and then other users can check the owner's public key from the certificate list, and decrypt the data with owner's public key. This scheme can make users assure about the security of data stored in the cloud.

The rest of this paper is organized as follows: We first provided preliminaries in section 2. Then section 3 discussed the proposed scheme. Section 4 provided the security and efficiency analysis of the proposed scheme and section 5 described the conclusion.

II. PRELIMINARIES

To facilitate of our proposed scheme, the following articles are briefly introduced.

A. Elliptic Curve Cryptography (ECC)

The elliptic curve cryptosystem was initially proposed by Koblitz and then Miller in 1985 to design public key cryptosystem and presently, it becomes an integral part of the modern cryptography. A brief introduction of ECC is given below:

Let E/F_p denotes an elliptic curve E over a prime finite field F_p, which can be defined by

$$y^2 = x^3 + ax + b \quad (1)$$

where, $a, b \in F_p$ and the discriminate $D = 4a^3 + 27b^2 \neq 0$

The points on E/F_p together with an extra point O called the point at infinity used for additive identity form an additive group A as

$$A = \{(x, y) : x, y \in F_p, E(x, y) = 0\} \cup \{0\} \quad (2)$$

Let n, the order of A, is very large and it can be defined as $n \times G \bmod q = O$, where G is the generator of A. Also A be a cyclic additive group under the point addition "+" defined as $P + O = P$, where $P \in A$.

The scalar point multiplication over A can be defined as

$$tP = P + P + \cdots + P \text{ (t times)} \quad (3)$$

If $P, Q \in A$, the addition $P + Q$ be a point $-R$ (whose inverse is R with only changing the sign of y coordinate value and lies on the curve) on the E/FP such that all the points P, Q and $-R$ lie on the straight line, i.e., the straight line cuts the curve at P, Q and $-R$ points. Note that if $P = Q$, it becomes a tangent at P or Q, which is assumed to intersect the curve at the point 0.
The security strength of the ECC lies on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) and it provides same level of security of RSA with less bit size key, which is addressed in the next sub-section.

B. Computational Problems

Similar to the DLP problem (known as discrete logarithm problem), some computational hard problems on ECC are defined below, which have not any polynomial time algorithm.

- **Elliptic Curve Discrete Logarithm Problem (ECDLP)**
  Given $Q, R \in A$, find an integer $k \in F_p^*$ such that $R=k.Q$.
- **Computational Diffie-Hellman Assumption (CDHA)**
  Given $P, xP, yP \in A$, it is hard to compute $xyP \in A$.
- **Decisional Diffie–Hellman Problem (DDHP)**
  Given $P, aP, bP, cP \in G$ for any $a, b, c \in F_p^*$, decide whether or not $cP = abP$.

C. Certificate Authority (CA) and PKI Enabled Application(PEA)

A Certification Authority (CA) which is the base of a PKI, is an entity trusted by one or more entities to create and assign certificates. The entities individually contact with a CA by providing their identity such as name, address, date of birth, public key etc. of each entity and after validation through handshake procedure.

A PK-Enabled application is able to invoke one or more of the following public key cryptography based functions: securely manage keys, trust anchors, and certificates; use one or more of the security services supported by the PKI by accepting and processing approved certificates; and obtain relevant certificate and revocation data.

III. PROPOSED SCHEME

A. Symbolical Notations and Definition

For the convenience of the description of our work, we first define in Table 1 the symbolical notations and their definition for the clarity and easy readability of our scheme.

TABLE 1. SYMBOLICAL NOTATIONS AND DEFINITION

Symbol	Definition
$h(.)$	One-way hash function
ID_{USER}	Identity of the user
ID_{CA}	Identity of the CA
E	An elliptic curve defined on Fp with prime order n;
P	A point on elliptic curve E with order n;
$(s1, V1)$	Private/public key pair of user, where $V1 = s1P$
$(s2, V2)$	Private/public key pair of CA, where $V2 = s2P$
p, n	Two large prime numbers

B. Secured Cloud Storage Framework

Cloud storage is the most prominent service in cloud computing, with cloud storage, users can store and access their data at any time/anywhere, it brings much convenience to the users, however since the data is stored over the cloud and flow through the network in plaintext format, users worry about the security of the data, although the cloud providers claim data stored in the cloud is much security.

We provide a framework, in figure 1; there are two parts for every user's data, the private data part and the shared data part. In the private data part, users can store their private and sensitive data which is used only by themselves; in the shared data part, users can share data with multiple authenticated users. User's operations are described as following:

(1)User authenticates to CA: Before users consume the service of the two parts provided by the cloud, they first need to authenticate to the CA and register for the certificate, and then CA will publish the certificate list in the cloud interface, all the registered users' public key can be found in the certificate list.

(2)User authenticates to the Cloud interface: After finishing the certificating, user can use his identity and

password to login the cloud interface , cloud will check the user's certificate according to the certificate list published by CA. If user can successfully authenticated to the cloud, then he can consume the two parts' cloud storage service.

(3)Private data part operation: In the private data part, user first encrypts the data at the client side with the help of the PEA, and then uploads it to the private data part of the cloud, when he needs the data, first downloads and then decrypts the data with his session key.

(4)Shared data part operation: In the shared data part, user can store the data which they want to share with other users. When user wants to share data with other authenticated users, he first encrypts the data with his session key and encrypts the session key with the private key of the key pair which is certificated by the CA. After finishing the encryption, user uploads the concatenation of the two parts' encrypted data to the shared data part of the cloud. From the certificate list, other authenticated users can check the public key of the user who uploads the data and use this public key decrypt the encrypted session key, after obtaining the session key, users can use it to decrypt the encrypted data.

In both private data and shared data parts, user encrypts data using symmetric encryption algorithms with different session keys, and only in shared data part, users encrypt the session key using ECC public key algorithm with their private key, and also decrypt the encrypted session key using ECC public key algorithm with corresponding user's public key. Moreover, users manage all the operations with CA and cloud interface through PEA. This scheme not only allows users store and access their data securely but also allows users share data with multiple authenticated users securely through the unsecure internet.

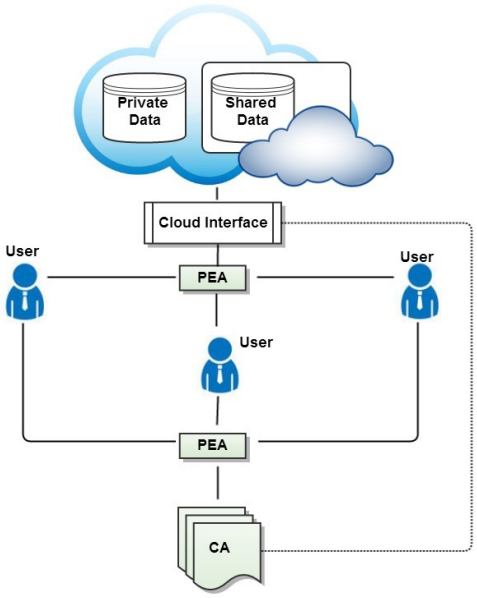


Figure 1. Secured Cloud Storage Framework (SCSF)

C. PKI Certificate Management Procedure

In our proposed ECC-based PKI scheme, initially, user should be authenticated to CA to ensure that the user possess the valid private key of corresponding public key. And once the certificate is issued by CA and used by the user, the mutual authentication of both cloud provider and user must also be established. Thus the proposed ECC-based PKI certificate procedure involves the following steps .

Step - 1: User sends request message to CA

User uses PEA to generate key pair including private key s1 and public key V1, and then generates the request message M by concatenating V1 and the identity of user. Next, it selects a random number r1 and generates R1=r1*P and also calculates an ECDH session key K=s1*V2=(K_X,K_Y).Then user concatenates the hash digest of M with R1, encrypts the concatenated message using K_X ,concatenates M with the encrypted message and then sends the concatenated message to CA as a certificate request message.

Step - 2: CA verifies user's identity and sends request message to user

After receiving the certificate request message, CA gets the identity of user and also gets the public key of the user from M for which the certificate is requested. Now it calculates the hash digest H' of received M as H' = h (M) and the ECDH session key using user's public key as K = s2*VI = (K_X, K_Y), decrypts the encrypted message using K_X, gets H and R1 and then, compares the received H with calculated H'. If both match, then CA confirms that the user has generated the private key for the corresponding public key. Now, for authentication purposes, RA selects a random number r2 and generates R2=r2*P, calculates the hash digest of R2 as h(R2), encrypts its identity and the hash digest using K_X and then sends the encrypted message along with (R1+R2) to user for authentication.

Step - 3: User authenticates to CA

CA decrypts the encrypted message using K_X and gets the identity of user and h(R2). It also retrieves R2 by subtracting R1 from (R1 + R2), calculates the hash digest of R2 and compares it with the received hash digest. If both match, CA is authenticated to user. user calculates the hash digest of R2 as h(R2), encrypts its identity and the hash digest using KX and then sends the encrypted message to CA.

Step - 4: CA sends to acknowledgement message to user

CA decrypts the message using K_X and compares the output with the hash digest of R2. If both match, user is authenticated to CA, which completes the mutual authentication procedure. Now CA sends an acknowledgement message to user and generate ECC public key certificate of the user and signed it with ECDSA signature.

Step - 5: Certificate Issuance

After finishing authentication between user and CA, CA will sign the certificate with his ECDSA signature and publishes it in its directory and also sends the certificate URL to cloud provider, authenticated users can check the certificate list from the cloud interface.

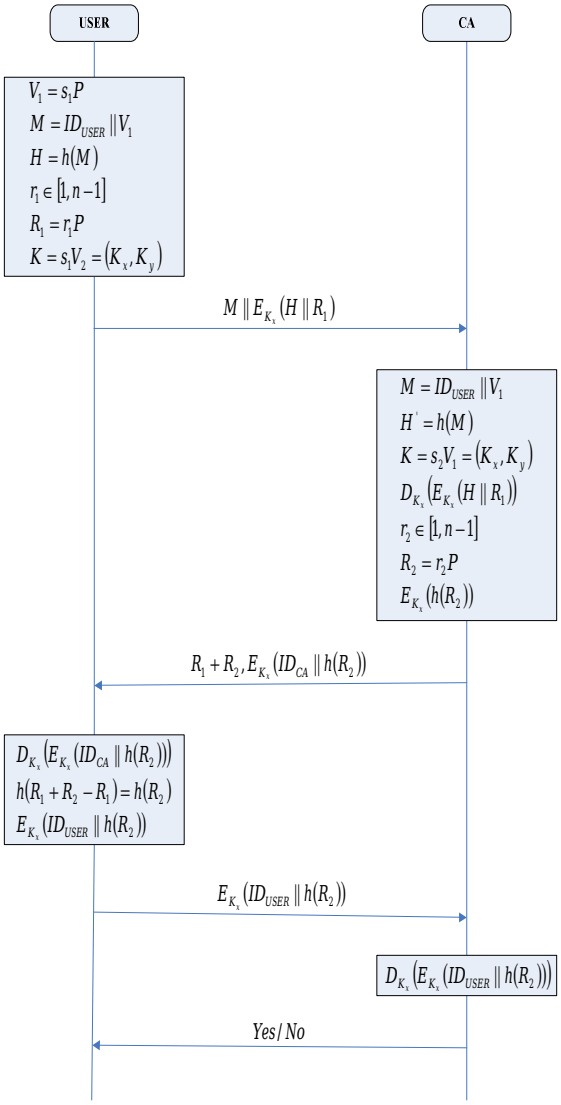


Figure 2. PKI Certificate Management Procedure

IV. SECURITY AND EFFICIENCY ANALYSIS OF PROPOSED SECURED CLOUD STORAGE SCHEME

A. Security Analysis

In our proposed secured cloud storage scheme, initially, user should be authenticated to CA to ensure that the user possess the valid private key of corresponding public key. And once the certificate is issued by CA and used by the user, the mutual authentication of both CA and user must also be established. Thus the proposed ECC-based PKI procedure involves the following cryptographic operations, where shown briefly that all are well protected. After receiving the certificate request message, CA verifies the message, completes the mutual authentication. The detail procedure of mutual authentication is discussed in step 1 to 5 of section 3_A which assure that before issuing a certificate, the CA must authenticates the user. Plus, when user authenticates to the cloud interface, the cloud also check the certificate of the user, so only the user both authenticated to CA and cloud interface can get the cloud storage service.

B. Efficiency Analysis

The proposed ECC-based PKI procedure is more efficient than the existing RSA-based schemes due to the following reasons:

(1) Provides comparable security with small key-length: In general, it is seen that 160-bit key in ECC is equivalent in security with 1024-bit key in RSA. This is because the existing RSA based PKI uses Diffie–Hellman key exchange protocol , in which the public challenges generated with key-size is at least 1024 bits, otherwise it is assumed that RSA is compromised. On the other hand, in ECC, the public challenges are of 160 bits key length, which is not easily compromised due to the unique properties of ECC.

(2) Requires less computation cost: Since the main computation carried out in ECC is the scalar point multiplication, thus it requires much lesser computation cost than RSA, which uses the most costly modular exponentiation operation. In addition, ECC uses all 160-bit operation, but RSA requires 1024-bit manipulation for comparable security. Also the proposed scheme uses cryptographic hash function, elliptic curve multiplication/addition and symmetric encryption, which further reduces the processing time over the RSA based scheme that follows public key encryption technique (as it is known that the symmetric approach is faster in processing than the public-key one). Therefore, the proposed ECC- based PKI requires less computation cost than the existing RSA based-PKI.

(3) Requires less communication cost: Due to the use of less key-size in ECC, each message-size in the proposed scheme is reduced and also due to use of certificate, the total number of messages between users is reduced to a minimum as possible. Because users don't need to exchange public key between each other. Thus the proposed scheme is communication efficient.

V. CONCLUSIONS

In this paper, we investigated the problem of data security in cloud storage. We propose a secured cloud storage scheme that allows user not only securely store and access data in the cloud, but also allow user share data with multiple users. In paper[6], we provided a scheme to ensure user share data with other users in the same group. Here, We apply ECC based-PKI in the certificate procedure, which can ensure user shares data with multiple users securely not restricted by the group. Moreover, the proposed ECC-based PKI certificate procedure provides low computation and communication cost as well as less key-size to provide same level of security as of RSA, and thus it is more efficient.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology. (Grant number: 2013-071188). And it also supported by the BB21 project of Bussan Metropolitan City.

REFERENCES

[1] [19] Hankerson, D., Menezes, A. and Vanstone, S. 2004. Guide to elliptic curve cryptography. Springer-Verlag, New York, USA (2004).

[2] [20] Koblitz, N. 1987. Elliptic Curve Cryptosystem. Journal of mathematics computation. 48, 177 (Janaury, 1987),203-

[3] 2009.

[4] [21] Miller, V. 1985. Use of elliptic curves in cryptography. In Proc. of Advances in Cryptology-CRYPTO, 85, LNCS 218 (1985), 417–426.

[5] [22] Diffe, W. and Hellman, M. 1976. New directions in cryptology. IEEE Transaction on Information Theory. 22 (1976), 644–654.

[6] XiaoChun Yin, Non Thiranant, Hoonjae Lee ,Secured Data Storage Scheme in Cloud Computing using Elliptic Curve Cryptography ,APICIST 2013

[7] Stallings, W, "Cryptography and Network Security: Principles and Practices", Prentice Hall, 4th Edition, pp 420-430, 2009.

[8] Hankerson, D, Menezes, A, Vanstone, S, "Guide to elliptic curve cryptography", Springer-Verlag, New York, USA, 2004.

[9] Koblitz, N, "Elliptic Curve Cryptosystem", Journal of mathematics computation, Vol. 48, No. 177, pp203- 209, 1987.

[10] Miller, V, "Use of elliptic curves in cryptography", Proc. of Advances in Cryptology-CRYPTO' 85, LNCS, Vol. 218, pp. 417–426, 1985.

[11] V.Miller, "Uses of elliptic curves in cryptography", Lecture Notes in Computer Science218: Advances in Cryptology- CRYPTO'85, pages417-426, Springer-Verlag, Berlin, 1986.

[12] N.Koblitz, "Elliptic curve cryptosystems", Mathematics of Computation, 48:203-209, 1987.

[13] Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS), Apr. 2002, http://www.ietf.org/rfc/rfc3278.txt

[14] M. Abdalla, M. Bellare, P. Rogaway, DHIES: An Encryption Scheme Based on the Diffie-Hellman Problem, Contribution to IEEE P1363a, 1998, http://cseweb.ucsd. edu/users/mihir/papers/dhaes.pdf.

[15] M. Abdalla, M. Bellare, P. Rogaway, The oracle Diffie-Hellman assmptions and an analysis of DHIES, Lecture Notes in Comput. Sci.2020(2001), 143–158.

[16] American National Standards Institute, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1998.

[17] V. Gayoso Martínez, L. Hernández Encinas, and C. Sánchez Ávila, Journal of Computer Science and Engineering, Volum 2, Issue 2, August

[18] Brainpool, ECC Brainpool Standard Curves and Curve Generation, 2005, http://www.ecc-brainpool.org/ download/Domain-parameters.pdf.

[19] Bundesamt f¨ur Sicherheit in der Information stechnik, Elliptic Curve Cryptography, 2009, https://www.bsi. bund.de/cln 183/EN/Home/home node.html.

[20] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Trans. Inform. Theory 22(1976), 644–654.

[21] Arjun Kumar, Byung Gook Lee, HoonJae Lee, Secure Storage and Access of Data in Cloud Computing, ICT Convergence (ICTC), 2012 International Conference on Date 15-17 Oct. 2012

[22] Institute of Electrical and Electronics Engineers, Standard Specifications for Public Key Cryptography -Amendment 1: Additional Techniques, 2004.

[23] International Organization for Standardization / International Electro technical Commission, Information Technology – Security Techniques – Encryption Algorithms – Part 2: Asymmetric Ciphers, 2006.

[24] N. Koblitz, Elliptic curve cryptosystems, Math. Comp. 48(1987), 203–209.

[25] V. Gayoso Mart´ınez, L. Hern´andez Encinas, C. S´anchez´Avila, Security and practical considerations when implementing the Elliptic Curve Integrated Encryption Scheme, preprint, 2010.

[26] A. J. Menezes, Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, Boston, MA, USA, 1993.

[27] V. S. Miller, Use of elliptic curves in cryptography, Lecture Notes in Comput.Sci.218(1986), 417–426.

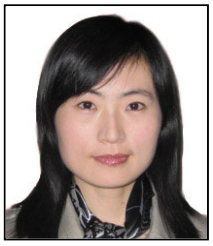
[28] National Institute of Standards and Technology, Digital Signature Standard (DSS), 2000.

[29] J. H. Silverman, The Arithmetic of Elliptic Curves, volume 106 of Graduate texts in Mathematics, Springer-Verlag, New York, NY, USA, 1986.

[30] Standards for Efficient Cryptography Group, Test Vectors for SEC 1, 1999, http://www.secg.org/download/ aid-390/gec2.pdf.

[31] Standards for Efficient Cryptography Group, Elliptic Curve Cryptography, 2000, http://www.secg.org/ download/aid-386/sec2 final.pdf.

[32] Standards for Efficient Cryptography Group, Recommended Elliptic Curve Domain Parameters, 2000, http://www.secg.org/download/aid-780/sec1-v2.pdf.



XiaoChun Yin

She received the B.S. degree in education and technology from Qufu Normal University, Qufu, China in 2004, and received the M.S. degree in education and technology from Nanjing Normal University, Nanjing, China in 2007. She had been working as a lecturer in Weifang University of Science & Technology, China from 2008 to 2012. Currently she is a doctoral candidate in cryptography and network security at Dongseo University, Korea. Her research interests include network security, cloud security, authentication protocol and real-time communication.

ZengGuang Liu

He received the B.S. and M.S. from Dept. of computer engineering, University of ShangHai for Science and Technology, China in 2005 and 2008 respectively. He is a senior software engineer at IP platform dept. of Alcatel-Lucent, QingDao, China from 2008. His research interests include Operation System and real-time communication.

Hoon Jae Lee

He received the B.S., M.S. and Ph.D. degree in Electrical Engineering from Kyungpook national university in 1985, 1987 and 1998, respectively. He had been engaged in the research on cryptography and network security at Agency for Defense Development from 1987 to 1998. Since 2002 he has been working for Department of Computer Engineering of Dongseo University as an associate professor, and now he is a full professor. His current research interests are in security communication system, side-channel attack, USN & RFID security. He is a member of the Korea institute of Information security and cryptology, IEEE Computer Society, IEEE Information Theory Society and etc.