An efficient Access Control Protocol for cloud data security using Hyper Elliptic Curve Cryptography

Ms.S.Selvi,
Department of Computer Science
PSG College of Arts and Science
Coimbatore, India

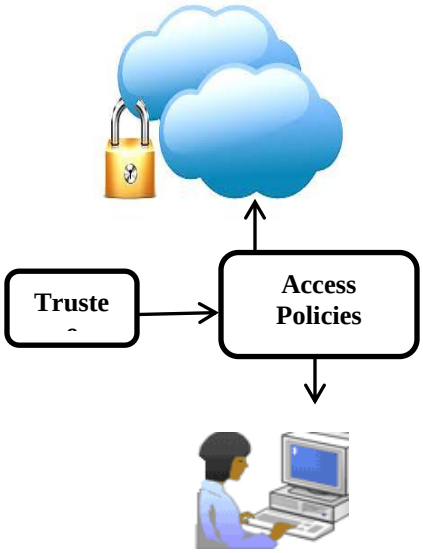
Mr.R.Ganesan
School of Computer Science
Vellore Institute of Technology
Chennai, India

Abstract:
One of the popular area in Information Technology (IT) is Cloud computing. It provides services such as Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) in on-demand basis. Clouds are capable of servicing millions of user requests. Due to huge amount of sensitive information are stored in cloud there is a need to authorize and authenticate users to prevent attacks that can be performed over the cloud. The cloud era has brought various challenges to handle malicious insider and cyber-attacks. By setting access control for cloud users only authorized users can have access valid cloud-provide service. Cryptographic techniques is be used to enforce some form of access control. The proposed HECKBE system protocol which includes the additional feature of applying hyper elliptic cryptography to exhibit a higher and secured authentication method which are used to provide access control to the valid users to gain access from cloud. A protocol HECKBE is applied to obtain privacy and authenticity. It is used to secure highly sensitive information like Health records, Financial records and various societal applications in cloud.

Key Words:Cryptography,Saas,Iaas,Paas,Cyber-attacks

I. INTRODUCTION

The cloud resources are dynamically allocated to the users to maximize effectiveness of computing resources. Clouds are classified as public, private or hybrid. Computing resources are shared among valid users by using higher security mechanisms in Cloud environment. A security service access control is discussed to set the privileges to the users to prevent unauthorized access. Information flow policies are authorized [9].Access control models applied with access control lists (ACLs) as access policies provided by the trusted and higher level authorities of a concern on cloud. The next level authorities have the right to provide tokens to the KDCs where highly secured private and public keys of the clients have been stored. The proposed primitive HECKBE(Hyper Elliptic Curve Key Based Encryption) is used to generate the keys. Access policies, attributes and Keys are compared with the client credentials to prove the validity of the client. The major types of access control models are Mandatory Access Control (MAC)[10] and Discretionary Access Control (DAC)[11].



The administrator decides the access policies in MAC where the owner in the later one. An access control matrix created in DAC to handle access permissions for the users. Access control matrix was introduced in [12]. In access control matrix users and resources are placed in rows and columns respectively. And the actions are represented in intersection of related row and column.

The following are the attribute sets to define the acess policy

i. Security Cnstraints (Sc,≤); U represents a set of users and security attributes O represents object;
ii. A policy $\lambda : U \cup O \rightarrow Sc$, where $\lambda(x)$ denotes the security construct of entity x. Then a user u is authorized to read o if and only if $\lambda(u) \in \lambda(o)$.

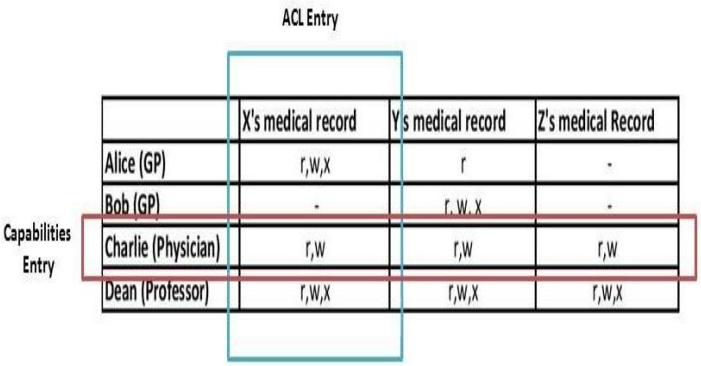
The trustee can create a public policy for a group users to set privileges for accessing cloud resources. The following statement shows the access policy of a group to gain access.

(“Age ≥50” “Country=India” “role = citizen” {policies, Senior Citizen,India},“LIC.xml”)

This access policy describes that the access rights given to the users to access the sub documents ‘policies’ , ’Senior Citizen’ if their age is of above 50 and holding user role as indian citizenship.

The user privileges can be set through access control list (ACL) as shown in Table 1.

Table1



The sample API with ACL is described in the following XML schema,

“acl”:[{“object”:“user_affg56hjmssfbg6723432567c4adf34da4 jhjhkdsestf”,”role”:”admin”,”u_id”:”34dsfc4567fsr3789hfds45 632scv567dfdfgd4434”}]

Client’s identity is compared with associated list provided by the owner’s access policies before enter into cloud. Access control in clouds has more consideration because it is important to preserve the confidential information in secret. Types of access controls are categorized as Role Based Access Control (RBAC), User Based Access Control (UBAC) and Attribute Based Access Control (ABAC).Users on their individual roles they are classified in Role Based Access Control. Cloud data can be retrieved by users by matching their roles. In User Based Access Control, the privileged users are placed in the access control list (ACL). The ACL users only considered as authorized persons to access data. It is not easy job to retrieve the vast list of users to verify among million of users. In Attribute Based Access Control (ABAC) is more extended in scope, in which users are assigned with an attributes set and access policies. The uses and limits of RBAC and ABAC are discussed in. Users those who have valid set of attributes, and matching the access policies, can access the data from cloud. The API statement shows that `user.domain_id` is attribute setting of created user.

```
"identity:create_user":[["role:admin",domain_id:
 %(target.user.domain_id)s"]]
```

ABAC in clouds are discussed in various aspects ([2],[3],[4],[5]). Two classes of ABAC are Key-policy ABE & Cipher text-policy ABE. Both types are discussed by Goyal et al[6],[7],[8]. The major weakness is a single KDC is presented for distributing the attributes to the users.

I.1 Our Contributions

i. Proposed model uses Multiple KDCs which are scattered for key distribution instead of single KDC.

ii. User credentials are created by the highly secured Hyper Elliptic Curve (HECKBE)algorithms which small in key size.

iii. Users identities are protected from the cloud.

iv. By setting privileges or access control to the users an unauthorized user cannot access the data on cloud. They are getting authorization after matching credentials like Keys , access policies and attributes and then they provided with data from the cloud.

I.2 Organization

This paper is arranged in the following order: Related work is placed in Section 2. Background of the problem is presented in Section 3. Proposed Access control protocol is arranged in Section 4. Section 5 has Conclusion and References.

II. RELATED WORKS

Securing the cloud is the major concern in the cloud computing environment. Many research works are being proposed to secure the cloud environment. Public cloud security model discussed by Kamara S and Lauter by applying cryptographic primitives for data integrity verification[6]. User generates public key to encrypt information and send encrypted data to another user. Decryption is performed by private key. Search over encrypted data is performed by employing symmetric and asymmetric encryptions.

Wang C and et al. have designed a model which employs encryption techniques for security and user should know information regarding encrypted data previously [1]. Symmetric searchable encryptions have been employed along with order preserving symmetric encryption (OPSE). Analysis of the model shows its efficiency in case of ranked keyword search. But, there is no information concerned to attacks, integrity and confidentiality. So, it may not be suitable for providing security. Incremental encryption[14] allows encrypting the data prior to its storage in cloud and whiling sharing with other authorized user the encrypted data is again re-encrypted with a different encryption key.

Security issues in cloud are discussed by Agarwal, A. and Agarwal [18].,An architecture is proposed for sharing data in cloud using RSA and for data integrity using MD5 algorithm. Data security in cloud can also be provided by employing RSA algorithm to encrypt large data files. The model works well with static data. But, in cloud data is mostly dynamic in nature. 128-bit SSL encryption[20] has been employed along with message authentication code to provide security for data in cloud.

MIT is implemented a protocol called Kerberos using symmetric key cryptography to authenticate a user in a secured manner even he is in unsecured network. Session keys are used here to secured communications in cloud.

A new system is proposed with a protocol named Hyper Elliptic Curve Key Based Encryption(HECKBE) to provide higher security for stored data in cloud. The records with sensitive information are encrypted with HECKBE after verifying the access policies stored on cloud. Clients can retrieve the decrypted information by applying the secret key by matching attributes. All the services like identification and authentication (I&A), authorization, and accountability are provided by using this protocol.

III. BACKGROUND OF THE PROBLEM

Elliptic Curve Cryptography has received a lot of attention because it offers several benefits over other public-key cryptosystems, such as RSA. With a higher security per key bit than RSA, HECC allows for a comparable level of security with a smaller key size[13]. Public-key cryptography[21] can be used to provide the services of Key establishment, digital signatures, and Encryption.
In modern applications, public-key primitives are used to provide all these three services. For the encryption and authentication of large data streams, one uses symmetric key algorithms because public-key algorithms are relatively efficient. Digital signatures have been a driving force behind the usage of public-key algorithms. They provide integrity, sender authentication as well as non-repudiation. Thus, the sender of a message cannot deny the creation of a message which can be crucial. Since 1976, three different variants of public-key cryptosystems of practical relevance have emerged namely:

i) Cryptosystems based on the difficulty of integer factorization.
ii) Solving the Discrete Logarithm (DL) problem in finite fields(e.g., Diffe Hellman key exchange or Digital Signature Algorithm) and
iii) The DL problem in the group of algebraic curves over a finite field.
(e.g., Elliptic Curve and Hyper Elliptic Curve Cryptosystems (HECC)
are the most well-known types.

HECC are a generalization of Elliptic Curve Cryptosystems (ECC) and were suggested for cryptographic applications in 1988 by Koblitz. Hyper elliptic curve cryptosystems have been extensively studied not only by the research community but also in industry.

A. $a_A \in_R N$ ***[choose a prime (***a_A***) at random in N***
B. $P_A \longleftarrow [a_A] D$
[The form of P_A is (u(x),v(x)) representation which is referred to as Mumford Representation]
C. ***return*** P_A ***and*** a_A

In step A, random prime number generation is given. one can apply the probabilistic test of Robbin-Miller or the deterministic test of AKS. However, various researches have proved that it takes exponential time to determine the given large number is prime or not using AKS algorithm.

IV. HYPER ELLIPTIC CURVE KEY BASED ENCRYOTION[HECKBE] PROTOCOL

The KDCs(Key Distribution Centers) are privacy-preserving decentralized systems which are used to protect the user's public and private keys. In the proposed architecture, all the user's private keys are tied to his credential attributes those are provided by root authority of a business. This system also avoided the collision attacks in cloud environment by checking the public and individual credentials.The message is encrypted using HECC encryption algorithm. Appropriate keys are generated and public keys and user credentials are stored on KDCs. Stored information on cloud are turning into an unreadable encrypted message by using the keys. User could not be able to obtain the original message without the associated keys, attributes and access policies.

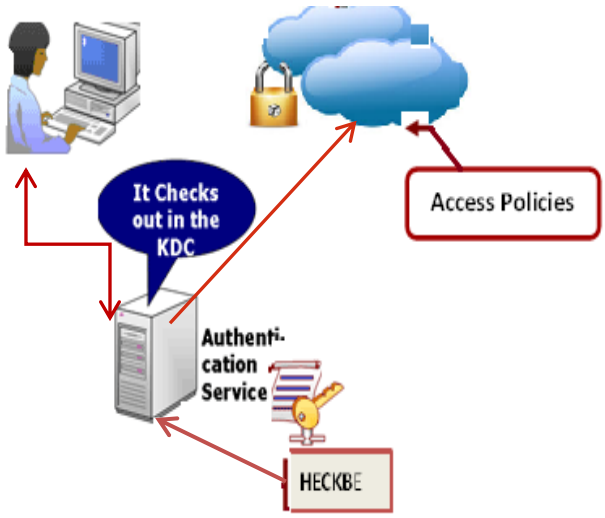


Figure 2

The Figure 2 verifies the user authentication with the generated keys by HECKBE protocol. ACL consisting the corresponding roles and attributes to the users private key. The user has to prove his authorization by his private key. Encrypted message is decrypted by decryption algorithm algorithm which usually requires a user's private key.

Before retrieving information from the cloud the proposed protocol is activated to check the authentication of a user. KDC will not provide the attributes if the user is not a valid by checking privileges in ACLs.
The proposed algorithm has the following steps:
1. A public parameter λ(Access policies) is created by the trustee and stored on cloud.
2. HECKBE protocol played a major role to create public and private keys Pk and Sk for users ,a considerable factor λ is verified before setting privileges.

3. Encryption of M essage:User can encrypt the message Msg by applying the user credentials receiver's Public key Pk,User access control(Ma,ρ) and the public parameter λ. The encoded message is referred to as Em and is created as following steps

- ***$p \in_R N$ (choose pr as a random positive prime number in N)***
- ***$Q \longleftarrow [pr]D$ (D is the Divisor of the HEC & The form of Q is (u(x),v(x)))***
- ***$P_k \longleftarrow [pr]P_B$ (P_B:(u(x),v(x)) is receiver's(B's) public key)***
- ***$C_m \longleftarrow \{ Q , P_k \}$ (C_m:(u(x),v(x)) is the Cipher Text going to be added with encoded message)***

Em $\longleftarrow$ Encryption_process(Msg,(Ma,ρ), λ,{Cm},i)

4. To decrypt the Encrypted text Em, receiver has to retrieve Cipher Message Cm by applying his credentials like user's Private key Sk, User access control(Ma,ρ) and the attributes and roles. Then he extracts the first coordinate 'Q' from Cm then it is multiplied with its Private Key and subtract the rsult from the second coordinate to get original message Em from Cm. The encryption or decryption process used ElGamal method to design HEC-ElG Algorithm (HEC-ElGA). Details on ElGamal method can be had from (Avanzi & Lange 2006).

ElGamal technique for key generation process, encryption and decryption process which is named as HEC-ElG Algorithm (HEC-ElGA).

Algorithm for Public Key & Private Key generation

Input: The public parameters are hyper elliptic curve C, prime p and divisor D

Output: The Public key pA and Private key sA

E_m +kpB –sB (Q) = E_m + k ***pB*** – sB (kD) = E_m + k ***pB*** – k(sBD) = Em +k ***pB*** – k ***pB*** = Em

In the above process, 'A' has masked the message Em by adding kpB to it. The 'A' know the value of k, so even though pB is a public key, nobody can remove the mask kpB. For an attacker to remove message, the attacker would have to compute k from the given D and [k]D i.e. Q, which is assumed very hard.

V. EXPERIMENTAL OBSERVATIONS

The budding technique in Public Key Cryptography is Hyper elliptic Curve Cryptography (HECC). This technique is applied in the proposed protocol to obtain access control key for the users on cloud. By use of them the resources in cloud are secured. By comparing key sizes HECC is better than the existing public key cryptography(19) technique such as RSA, DSA, AES and ECC. Computational time can be compromised while comparing the higher security of shared resources on cloud.

The performance of the HECC for genus 2 and genus 3 was analyzed based on the length of the prime generated and the time taken for the divisor generation and key generation. Number of members who are joining on cloud increases reflects in the computation time which also increases proportionately. A new protocol proposed by Steiner, Tsudik, and Waidner [22] based on Diffie Hellman Key exchange. Access policies not playing here to show the hierarchical access limits. This protocol is not suitable for large groups with privileges on cloud.

The key generation process is discussed as above and the output is shown in Table 2.

Table 2

	HECC FOR GENUS 2 OVER PRIME FIELD FP	HECC FOR GENUS 3 OVER PRIME FIELD FP
HECC Equation	C: C: v^2=u^5+4	C: v^2=u^7+u^5+4
Divisor Generation	D:div (div(u^2+78u+252, 28u+191) To create Divisor, it took 0.14807403701850924 Seconds	D: div (u^3+76u+49, 91u^2+251u+221) To create Divisor , it took 0.1325662831415708 Seconds
Key Gen	Privatekey A skA: 65553370776048789 Seconds Public key ApkA: div (u^2+34u+119, 36u+141) User A SecKey and PubKey generated in 0.03151575787893947 Seconds Private Key B 100013000640014200121 Public Key B div (u^2+41u+23, 196u+140) User B SecKey and PubKey generated in 0.02351175587793897 Seconds	Private key A skA: 6553370776048849 Seconds Public key A pkA : div (u^3+100u^2+205u+1, 98u^2+177u+81) User A SecKey and PubKey generated in 0.05502751375687844 Seconds Private Key B 100013000640014201121 Public Key B div (u^3+71u^2+110u+121, 203 u^2+45u+160) User B SecKey and PubKey generated in 0.054527263631815 905 Seconds

The schemes introduced in [15-17] will not support concurrent access while the proposed methodology in this paper will supports concurrent access with setting privileges to users with multiple KDCs on cloud. In the proposed approach multiple KDCs are placed to help in fault tolerance in case if a KDC fails to support. The proposed methodology is collusion resistant because user attributes are created with HECKBE protocol and compared with the client credentials before providing data from the cloud.

VI. CONCLUSION

A security protocol for resource sharing access control mechanism in cloud is discussed in this paper. The difficulties of key distribution are being reduced by using HECKBE by having multiple KDCs scattered to avail this functionality. Attribute based techniques also included to encrypt the user data so that the prominent security is availed by the proposed protocol. According to the different personal roles the cloud data can be accessed in highly secured manner. It can be developed further as access policies can be created by using this cryptographic technique to drastically increases **performance** in cloud. Groups or small sets of separate identities can be encrypted by applying this protocol. This paper deliberated current access control models, and also it confers the performance evaluation of proposed model.

REFERENCES

[1] K. Ren, C. Wang, Q. Wang, Security challenges for the public cloud, IEEE Internet Comput. 16 (1) (2012) 69–73.
[2] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in ACM ASIACCS, 2010, pp. 261–270.
[3] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in ACM CCS, 2010, pp. 735–737.
[4] F. Zhao, T. Nishide, and K. Sakurai, "Realizing fine-grained and flexible access control to outsourced data with attribute-based cryptosystems," in ISPEC, ser. Lecture Notes in Computer Science, vol. 6672. Springer, 2011, pp. 83–97.
[5] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed access control in clouds," in IEEE TrustCom, 2011.
[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in ACM Conference on Computer and Communications Security, pp.89–98, 2006.
[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in IEEE Symposium on Security and Privacy. , pp. 321–334, 2007.
[8] X. Liang, Z. Cao, H. Lin and D. Xing, "Provably Secure and Efficient Bounded Ciphertext Policy Attribute Based Encryption,"in ACM ASIACCS, pp 343–352, 2009.
[9] Denning, D.: A lattice model of secure information flow.
[10] Sandhu, R.; Munawer, Q. How to do discretionary access control using roles. In Proceedings of the 3rd ACM Workshop on Role-Based Access Control, Fairfax, VA, USA, 22–23 October 1998.
[11] Zhao, G.; Chadwick, D.W. On the modeling of bell-lapadula security policies using RBAC.
In Proceedings of the 2008 IEEE 17th Workshop on Enabling Technologies: Infrastructure for
Collaborative Enterprises (WETICE '08), Washington, DC, USA, 23–25 June 2008; pp. 257–262.
[12] Samarati, P.; Vimercati, S. Access control: Policies, models, and mechanisms. In Foundation of Security Analysis and Design; Springer: Berlin Heidelberg, Germany, 2001; Volume 2171,pp. 137–196
[13] Ramachandran Ganesan , Mohan Gobi , and Kanniappan Vivekanandan; A Novel Digital Envelope Approach for A Secure E-Commerce Channel, International Journal of Network Security, Vol.11, No.3, PP.121–127, Nov. 2010
[14] M. Bellare, O. Goldreich, and S. Goldwasser. Incremental cryptography: The case of hashing and signing. In Advances in Cryptology – CRYPTO '94, pages 216–233, 1994.
[15] Shyam Nandan Kumar, "Cryptography during Data Sharing and Accessing Over Cloud." International Transaction of Electrical and Computer Engineers System, vol. 3, no. 1 (2015): 12-18.
[16] D. X. Song, D. Wagner, A. Perrig. "Practical techniques for searches on encrypted data". Proceedings of the IEEE Symposium on Security and Privacy, 2000, pp. 44-55.
[17] D. Boneh, G. D. Crescenzo, R. Ostrovsky, G. Persiano. "Public key encryption with keyword search. Advances in Cryptology"-EUROCRYPT'04, 2004, LNCS 3027, Springer, pp. 506-522.
[18] A. Agarwal, A. and Agarwal, A. (2011). The Security Risks Associated with Cloud Computing. International Journal of Computer Applications in Engineering Sciences, 1 (Special Issue on CNS), 257-259.
[19] Janakiraman V S, Ganesan R, Gobi M "Hybrid Cryptographic Algorithm for Robust Network Security" ICGST- CNIR, Volume (7), Issue (I), July 2007
[20] Rui Zhang 1,2 and Ling Liu 1 ," Security Models and Requirements for Healthcare Application Clouds "Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference; July 2010
[21] Dr M.Gobi and D.Kannan," A Secured Public Key Cryptosystem for Biometric Encryption", International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 184-191
[22] Chun-Li Lin,"Three-party encrypted key exchange without server public-keys" Published in: Communications Letters, IEEE (Volume:5 , Issue: 12),2001;

AUTHORS PROFILE

Ms.S.Selvi,MCA.,Mphil(PhD),Assistant Professor,PSG College of Arts and Science,Coimbatore-14.

Dr.R.Ganesan, MCA.,Mphil.,PhD,Associate Professor,VIT,Chennai Campus.