Survey on Efficient Secure Storage Authentication in Cloud Storage System

Amaresh Patil
M.Tech, Computer Science and Engineering
Sahyadri College of Engineering and Management, Mangalore-575007

Valluripalli Srinath
M.Tech, Computer Science and Engineering
Sahyadri College of Engineering and Management, Mangalore-575007

Sudheer Shetty
Associate Professor and HOD, Computer Science and Engineering Sahyadri College of Engineering and Management, Mangalore-575007

***Abstract*— Cloud computing has become one of the migrating technology in the field of Information Technology(IT).In which user can access Infrastructure, Platform, Software, as Services anywhere from the Internet on as needed basis. The biggest challenge is to provide efficient confidentiality for the data to be stored in cloud due to this many organizations are not coming forward to store their confidential data in cloud. The typical technology used to provide data confidentiality is Cryptography system so that the data stored in third party is encrypted. In which the confidentiality is depends on the security level of the cryptography algorithm which is used. Meanwhile we should consider the efficiency (computation and storage cost) of the algorithm which increases the efficiency of cloud service. In this survey paper we look at various current researches being done to solve these issues, the current trends in storing, provide confidentiality and availability for the data in cloud and also discuss a public key cryptography scheme called Elliptic Curve Cryptography scheme which provide efficient security level by using small key, the challenges that are required to implement secure cloud data storage. This survey help to identify the future research areas.**

Keywords*— *Cloud storage system, RSA algorithm, Elliptic Curve Cryptography (ECC).

I. INTRODUCTION

Cloud computing is a computing environment, where resources such as computing power, storage, network and software are integrated and provided as services through the Internet as user demands in a remotely accessible fashion. User should pay as use the resources in cloud environment Billing models (pay as use) is based on the policy which is setup by service provider and agreed by both user and provider. On-demand availability, ease of provisioning, dynamic and virtually infinite scalability is some of the key attributes of cloud computing [6].

The main concept of cloud computing is providing the services. It provides the various types of services. Some of important services are Software-as-a-Service (SaaS), Infrastructure-as-a Service (IaaS), and Platform-as-a-Service (PaaS). SaaS is a model of software deployment where a provider licenses an application to customers for use as a service user can access any application/softwares as a service on demand from anywhere through internet. Infrastructure as-a-service (IaaS) delivers the infrastructures on demand in the form of virtual hardware, storage, and networking. Platform as-a-Service (PaaS) Solution which delivers scalable and elastic runtime time environments for execution of user application i.e. the application developer can utilizes platforms to develop and execute his application as service [9].

Storage is the one of infrastructure provided by cloud as Storage-as-service is one of facility provided by cloud computing among various services. The need of this service was increasing rapidly because this service provide the cloud user a needed storage space and reduces the maintenance cost, many organization moves their data to the cloud storage and extending the storage capacity later according to their needs[1][2]. This method used to focus on designing a cloud storage system for robustness, privacy, and functionality. A cloud storage system is considered as a large scale distributed storage system that consists of many self-governing storage servers among them the data is distributed but the user unaware of these stuffs and the data computation is abstracted to user i.e. they no need to worry about how data is stored and managed. In this system the responsibility of the provider is virtualizes and monitoring the storage servers.

II. CLOUD STORAGE SYSTEM

The most important things should be fulfilled by cloud storage system are authentication, robustness and data confidentiality. Authentication can provided by many password technology, and the robustness (recovering the data in case of failures) of storage server achieved by RAID technologies. In which distributed data can recover by redundant data which is stored along with data in storage server, one of the new and most trusted method is called eraser code method. Confidentiality of data is still challenging one in cloud storage system. Data confidentiality which

- Make sure that no one can intercept your data as it moves from point A to point B in the cloud.
- Make sure that no data leaks (malicious or otherwise) from any storage in the cloud [11].

There are many proposals of storing data over storage servers. One way to present data robustness is to duplicate a message such that each storage server stores a copy of the

message. It is robust because the message can be retrieved as long as one storage server survives but it is expensive [2].

Hsiao-Ying Lin and Wen-Guey Tzeng have proposed the integrated cloud storage system which integrates robustness, confidentiality and additional functionality called data forwarding from storage server by using proxy re- encryption, the architecture explained in [1] gives the detail idea about the storage system with some algorithms. As shown in below Fig 1, system model consists of users, n storage servers $SS_1, SS_2, ...SS_n$ and m key servers $KS_1, KS_2, ...KS_m$ Storage servers provide storage services and key servers provide key management services.

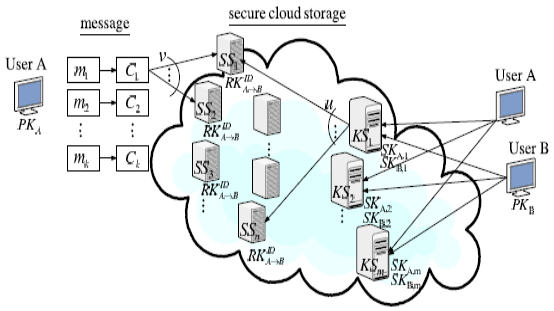

Fig 1: General system model [1]

Distributed storage system consists of four phases: system setup, data storage, data forwarding, and data retrieval. These four phases are described as follows. In the system setup phase, the system manager chooses system parameters and publishes them. Each user A is assigned a public- secret key pair (PK_A, SK_A). User *A* distributes his secret key SK_A to key servers such that each key server KSi holds a key share SK_{A_i}, $1 \le i \le m$. The key is shared with a threshold t. In the data storage phase, user A encrypts his message *M* and dispatches it to storage servers. A message *M* is decomposed into k blocks $m_1, m_2, ...m_k$ and has an identifier *ID*. User A encrypts each block mi into a cipher text *Ci* and sends it to randomly chosen storage servers. Upon receiving cipher text from a user, each storage server linearly combines them with randomly chosen coefficients into a code word symbol and stores it. Note that a storage server may receive less than k message blocks and assume that all storage servers know the value k in advance. In the data forwarding phase, user A forwards his encrypted message with an identifier ID stored in storage servers to user B such that B can decrypt the forwarded message by his secret key. To do so, A uses his secret key SK_A and B's public key *PKB* to compute a re- encryption key $RK_{ID,A\to B}$ and then sends $RK_{ID,A\to B}$ to all storage servers. Each storage server uses the re-encryption key to re-encrypt its code word symbol for later retrieval requests by *B*. The re-encrypted code word symbol is the combination of cipher text under B's public key. In order to distinguish re-encrypted code word symbols from intact ones, which are called as original code word symbols and re-encrypted code word symbols, respectively. In the data retrieval phase, user A requests to retrieve a message from storage servers. The message is either stored by him or forwarded to him. User *A* sends a retrieval request to key servers. Upon receiving the retrieval request and executing a proper authentication process with user A, each key server KS_i requests u randomly chosen storage servers to get code word symbols and does partial decryption on the received code word symbols by using the key share SK_{Ai}. Finally, user A combines the partially decrypted code word symbols to obtain the original message *M*. System recovering: When a storage server fails, a new one is added. The new storage server queries k available storage servers, linearly combines the received code word symbols as a new one and stores it then it is recovered [1][2].

Another consideration is efficient algorithm usage for encryption to strengthen the security and to reduce time needed this leads to think about which cryptography system is suitable. The two encryptions are used one for storing data in distributed server and one as proxy re-encryption, we should choose best algorithm for encryption which satisfies following two [4].

- Strength of the security is more so that the cryptanalysis hard to break the security
- Speed of the algorithm so that it should take less time to generate keys , encrypt and decryption

III. CHOOSING EFFICIENT ENCRYPTION ALGORITHM

There are mainly two kind s of cryptographic systems

1. Symmetric key cryptography
2. Public key cryptography.

Symmetric key cryptography: In this, a single key is used for both encryption and decryption. The sender uses the key (or some set of rules) to encrypt the plaintext and sends the cipher text to the receiver. The receiver applies the same key to decrypt the message and recover the plaintext. Because a single key is used for both functions, secret key cryptography is also called symmetric encryption. With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver. The disadvantage of this approach is it is most vulnerable to attack since the same key is using as secrete key for both. But this approach is having speed of execution than other method, because of its security strength it is not suitable to for system [3].

Public key cryptography: Public or asymmetric key cryptography involves the use of key pairs: one private key and one public key. Both are required to encrypt and decrypt a message or transmission. It is not to be shared with anyone. The owner of the key is responsible for securing it in such a manner that it will not be lost or compromised. Public key cryptography intends for public keys to be accessible to all users. In fact, this is what makes the system strong. If a person can access anyone public key easily usually via some form of directory service then the two parties can communicate securely and with little effort, i.e. without a prior key distribution arrangement [3]. RSA algorithm is the first relatively complete public key algorithm it can be used for data encryption, also can be used for digital signature algorithms. RSA cryptosystem is based on the difficulty of integer

factorization in the group Zn, and its security establishes in the assumption that constructed by almost all the important mathematician it is still a theorem that does not permit which is lack of proof, but Mathematicians believe it is existent [3].

When we consider the storage in cloud system this will included both security and key storage CPU usage and all. The commonly using cryptography in cloud storage to provide data confidentiality is RSA. But in RSA algorithm the strength of the security is depends on the key size as key size increases the security also increases and storage capacity required to store key in key server will be large. In this paper we discuss that use the algorithm which will provide high security with less key size. The cryptosystem called elliptic curve cryptosystem which is public key cryptography and provide high security with less key size.

IV. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic curves in public-key cryptography was independently proposed by Koblitz and Miller in 1985 and, since then, an enormous amount of work has been done on elliptic curve cryptography (ECC) [4] [7]. The attractiveness of using elliptic curves arises from the fact that similar level of security can be achieved with considerably shorter keys than in methods based on the difficulties of solving discrete logarithms over integers or integer factorizations $y^2 = x^3 + ax + b$ Where $a, b \in R$

A. *Elliptic curves*

An elliptic curve (EC) is a smooth, projective algebraic curve of genus one, on which there is a specified point *O*. An elliptic curve is in fact an abelian variety – that is, it has a multiplication defined algebraically, with respect to which it is a (necessarily commutative) group – and *O* serves as the identity element. Often the curve itself, without *O* specified, is called an elliptic curve. Any elliptic curve can be written as a plane algebraic curve defined by an equation of the form:

$y^2 = x^3 + ax + b$ Where $a, b \in R$

The different curves which are satisfied the above equation by different values of *a* (along y axis) and *b* (along x axis) are as below.

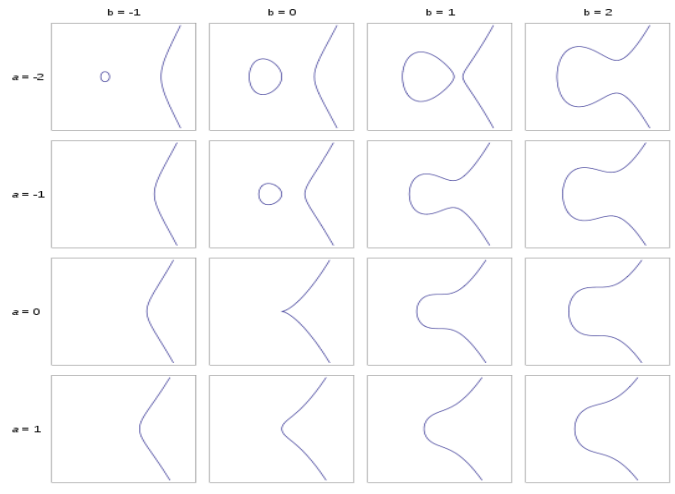

Fig 2: Elliptic curves [12]

But the entire curve should have the following characteristics:

- Forms an abelian group
- Symmetric about the x-axis
- Point at Infinity acting as the identity element

B. *Elliptic Curve cryptographic Schemes*

Several discrete logarithm-based protocols have been adapted to elliptic curves, replacing the group $(Z_p)^{\times}$ with an elliptic curve:

- The Elliptic Curve Diffie–Hellman (ECDH) key agreement scheme is based on the Diffie–Hellman scheme,
- The Elliptic Curve Integrated Encryption Scheme (ECIES), also known as Elliptic Curve Augmented Encryption Scheme or simply the Elliptic Curve Encryption Scheme,
- The Elliptic Curve Digital Signature Algorithm (ECDSA) is based on the Digital Signature Algorithm,
- The ECMQV key agreement scheme is based on the MQV key agreement scheme.
- The ECQV implicit certificate scheme.

There are many vendors providing these schemes as API's one that vendor is **Bouncy castle** with reference http://www.bouncycastle.org [8].

C. *Parameter Considerations*

To use ECC all parties must agree on all the elements defining the elliptic curve, that is, the *domain parameters* of the scheme. The field is defined by *p* in the prime case and the pair of *m* and *f* in the binary case. The elliptic curve is defined by the constants *a* and *b* used in its defining equation. Finally, the cyclic subgroup is defined by its *generator* (aka. *base point*) *G*. For cryptographic application the order of *G*, that is the smallest non-negative number *n* such that $nG = \infty$ is normally prime. Since *n* is the size of a subgroup of $E(F_p)$ it follows from Lagrange's theorem that the number $h = \frac{|E(F_p)|}{n}$ is an integer. In cryptographic applications this number *h*, called the *cofactor*, must be small $(h \le 4)$ and preferably $(h = 1)$. Let us summarize: in the prime case the domain parameters are (P, a, b, G, n, h) and in the binary case they are (m, f, a, b, G, n, h, s).

Unless there is an assurance that domain parameters were generated by a party trusted with respect to their use, the domain parameters *must* be validated before use. The generation of domain parameters is not usually done by each participant since this involves counting the number of points on a curve which is time-consuming and troublesome to implement. As a result several standard bodies published domain parameters of elliptic curves for several common field sizes. Such domain parameters are commonly known as "standard curves" or "named curves"; the National institute of

Standards and Technology (NIST) has defined 15 standard curves[4][5][7][13].

The main operations of this cryptography are point addition and point multiplication. Adding two points lies on curve E result in third point which lies on curve i.e. Where point P and Q and R=P+Q where R is point on curve E. The Point multiplication is repeated addition If P is a known point on the curve (aka Base point; part of domain parameters) and it is multiplied by a scalar k, Q=kP is the operation of adding P + P + P + P… +P (k times). In these point addition and point multiplication explanations we have chosen random Elliptic curve

Addition procedure:

- Choose point P and Q on curve E
- Draw line L through P and Q.
- The line *L* intersects the cubic curve *E* in a third point. Call that third point *R*.
- Draw the vertical line through R which hits the curve E at another point the point is resultant point which denoted as $P \oplus Q$ or $P + Q$

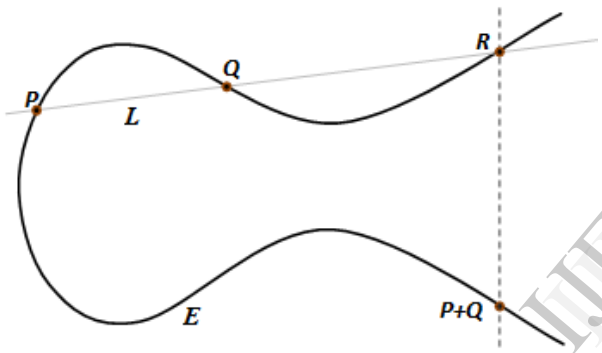

Fig 2: Point Addition on Elliptic Curve

Addition procedure:

- Choose point *P* on curve E
- Draw tangent to E at point P which intersect some another point on E called R.
- Draw vertical line from R which hits the curve in another point that is resultant point denoted as *P+P or 2P*

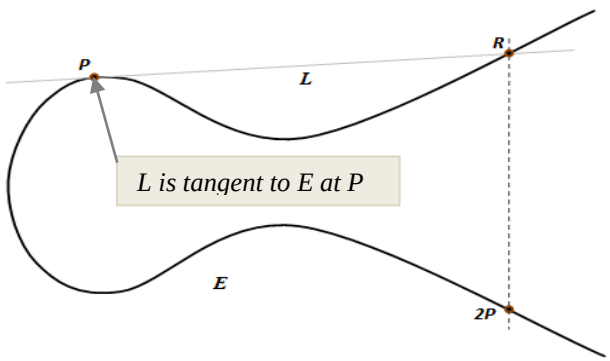

Fig 3: Point multiplication on Elliptic Curve

The multiplication takes place repeatedly integer *d* times the result is new point $Q = dP$ now the point *Q* will become a public key and *d* become private key.

D. Elliptic curve Diffie–Hellman key exchange algorithm

Suppose Alice wants to establish a shared key with Bob, but the only channel available for them may be eavesdropped by a third party. Initially, the domain parameters (that is, (P, a, b, G, n, h, s) in the prime case) must be agreed upon. Also, each party must have a key pair suitable for elliptic curve cryptography, consisting of a private key d (a randomly selected integer in the interval [1, n-1]) and a public key $Q = dP$. Let Alice's key pair be (d_A, Q_A) and Bob's key pair be (d_B, Q_B). Each party must have the other party's public key (an exchange must occur).

Alice computes $(x_k, y_k) = d_A Q_B$. Bob computes $(x_B, y_B) = d_B Q_A$. The shared secret is x_k (the *x* coordinate of the point). Most standardized protocols based on ECDH derived a symmetric key from x_k using some hash-based key derivation function. The shared secret calculated by both parties is equal, because $d_A Q_B = d_A d_B P = d_B d_A P = d_B Q_A$.

The only information about her private key that Alice initially exposes is her public key. So, no party other than Alice can determine Alice's private key, unless that party can solve the elliptic curve discrete logarithm problem. Bob's private key is similarly secure. No party other than Alice or Bob can compute the shared secret, unless that party can solve the elliptic curve Diffie–Hellman problem. The public keys are either static (and trusted, say via a certificate) or ephemeral. Authentication is necessary to avoid man-in-the-middle attacks. If one of Alice or Bob's public key is static then man-in-the-middle attacks are thwarted. Static public keys provide neither forward secrecy nor key-compromise impersonation resilience, among other advanced security properties. Holders of static private keys should validate the other public key, and should apply a secure key derivation function to the raw Diffie–Hellman shared secret to avoid leaking information about the static private key.

V. RSA V/S ECC ALGORITHM

RSA was conceived by Rivest, Shamir and Adleman in 1977 and Koblitz and Miller independently published work on ECC in 1985. The fundamental operation underlying RSA is modular exponentiation in integer rings and its security stems from the difficulty of factoring large integers. ECC operates on groups of points over elliptic curves and derives its security from the hardness of the elliptic curve discrete logarithm problem (ECDLP) [5].

Advantages of elliptic curve cryptography

- Shorter key length
- Lesser Computational Complexity
- Low Power Requirement
- More Secure

Shorter key length:

ECC operates on groups of points over elliptic curves and derives its security from the complex of the elliptic curve discrete logarithm problem (ECDLP). While sub-exponential algorithms can solve the integer factorization problem only exponential algorithms are known for the ECDLP. This allows ECC to achieve the same level of security with smaller key sizes and higher computational efficiency ECC-160 provides comparable security to RSA-1024 and ECC-224 provides comparable security to RSA-2048. [5]. the below table [5] shows the key length comparison of different algorithms with the same level security.

Symmetric Encryption (Key Size in bits)	RSA and Diffie-Hellman (modulus size in bits)	ECC Key Size in bits
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Lesser time and power consumption

ECC encryption with Koblitz's method of encoding is one of the best encryption algorithms that provide reliable security. Such algorithms consume additional power for encoding and encryption, whereas our process does not consume so much power.

More Security:

In RSA Algorithms the attacker can compute the prime factors of the number $n = pq$. If he able to derive the primes p and q from n then $\Phi = (p-1)(q-1)$ can be computed it enables the determination of the private key $d \equiv e^{-1} (\mathrm{mod}\, \Phi)$. In RSA the key size of the n should be larger to provide high level of security. Where as in ECC P is a point on the curve $Q=dP$ *is* Public key where d is private key, one who wants to find private key d he should solve the discrete logarithm problem (ECDLP), since it is very complex and takes more time so small key size can provide high level of security.

VI. CONCLUSION

Cloud storage is one of the challenging issues in terms of upload and store confidential data to cloud system. In this paper, we discussed various concepts for cloud storage with traditionally used algorithms and also discussed how Elliptic curve algorithm is better than RSA in terms of key size and security level about confidential data. Elliptic Curve Cryptography system which provides sufficient security with less key size. So which provide security to storage system in cloud by utilizing less power and less computation time and also less storage to store keys in key server. Future work is to implement and analyze Elliptic Curve Cryptography over the traditional algorithms like RSA for cloud storage system.

REFERENCES

[1] Hsiao-Ying Lin, Member, IEEE, and Wen-Guey Tzeng, Member,,"A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding" IEEE transactions on parallel and distributed systems, vol. 23, no. 6, june 2012

[2] Qian Wang, *Student Member, IEEE,* Cong Wang, *Student Member, IEEE,* Kui Ren, *Member, IEEE,* Wenjing Lou, *Senior Member, IEEE,* and Jin Li, Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing, IEEE transactions on parallel and distributed systems, vol. 23, no. 6, Jan 2008

[3] Principles of Information Security, Fourth Edition, Michael E. Whitman, *Ph.D., CISM, CISSP Herbert* J. Mattord, *CISM, CISSP Kennesaw State University*

[4] Prof. Renu Vig, Ravi Tandon, Performance Analysis of Elliptic Curve Cryptography on Reconfigurable Hardware. Proceedings of the World Congress on Engineering 2008 Vol I WCE 2008, July 2 - 4, 2008, London, U.K.

[5] Nils Gura, Arun Patel, Arvinderpal Wander,Hans Eberle, Sheueling Chang Shantz, Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs, Sun Microsystems Laboratories {Nils.Gura, Arun.Patel, Arvinderpal.Wander, Hans.Eberle, Sheueling.Changg}@sun.com http://www.research.sun.com/projects/crypto

[6] Johnson D, Kiran Murari, Murthy Raju, Suseendran RB, Yogesh Girikumar (2010), Eucalyptus Beginner's Guide - UEC Edition, CSS Open Source Services, UEC Guide.v1.0. (Ubuntu Server 10.04 - Lucid Lynx).

[7] Joppe W. Bos, J. Alex Halderman, Nadia Heninger, Jonathan Moore, Michael Naehrig, and Eric Wustrow, Elliptic Curve Cryptography in Practice, Microsoft Research, University of Michigan, University of Pennsylvania

[8] Kimmo Järvinen, *Student Member, IEEE,* and Jorma Skytta, On Parallelization of High-Speed Processors for Elliptic Curve Cryptography, IEEE Transactions on very large scale integration (vlsi) systems, vol. 16, no. 9, September 2008

[9] Certicom, Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, September 2000, Available at http://www.secg.org/download/aid-386/sec2_final.pdf

[10] Mastering Cloud Computing Foundations and Applications Programming, Rajkumar Buyya, Christian Vecchila,S. Thamarai Selvi.

[11] Cloud computing for dummies by Jutith Hurwit,Robin Bloor.

[12] http://en.wikipedia.org/wiki/File:EllipticCurveCatalog.svg

[13] http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf