ANALYSIS OF CLOUD AUTHENTICATIONS AND DATA SECURITY USING PHECC

S.Hendry Leo Kanickam[1], Dr.L. Jayasimman[2],
[1]Research Scholar, Department of Computer Science, Bishop Heber College Trichy,Tamilnadu,India.
[2]Assistant Professor, Department of Computer Applications, Bishop Heber College, Trichy,Tamilnadu,India.

Abstract: Secure data communication is of a key concern in today's rapidly growing world. Various security mechanisms are developed in order to achieve the data security. Cryptography is one among them. It is the study of mathematical techniques that are related to the aspects of information security such as confidentiality, data integrity, authentication, and availability. The paper presents a polynomial based novel model which uses a combination of modified ECC algorithm for data confidentiality, hashing for data integrity. Performance of this system is evaluated on different configurations on the basis of key, encryption and decryption time, throughput and memory usage for different data formats like text file, image file, audio file and video file.

Key Words: Secure Hash Algorithm -2, Elliptic Curve Cryptography, Elliptic-Curve Diffie–Hellman, Elliptic Curve Digital Signature Algorithm, Polynomial Based Hashing and Elliptic Curve Cryptography.

I. INTRODUCTION

Security was become the main issue that face everybody that using internet in their daily work, The security of Cloud Computing can be accomplished by meeting the security objectives i.e. Availability, Authorization, Authentication, Confidentiality, Integrity, Non repudiation and Freshness4,5. These security objectives are achieved by some techniques; the confidentiality is achieved by cryptography, integrity is achieved by hashing, and availability is achieved by access control and etc. The integrity and confidentiality is an important requirement in information security.

For integrity, Client authentication needs security for remote login while the client's program tries to communicate with the server's program over insecure networks like Internet. The identity and a secret password of a client are used for mutual authentication and access control; password can be compromised during transmission, many authentication mechanisms were introduced to control the accessing to the cloud services depending on different authentications protocols, if an efficient scheme is not followed. So, to protect the data from unauthorized access and to ensure that our data are intact proposed a polynomial based hashing scheme, which solves the problem of integrity, unauthorized access, privacy and consistency.

For confidentiality, Elliptical curve cryptography (ECC) is a public key encryption technique based on the mathematics of elliptic curves theory that can be used to create faster, smaller, and more efficient cryptographic keys. In ECC, main operations such as key agreement, signature generation, signing and verification involve scalar multiplication.

The speed of scalar multiplication plays a significant role in the efficiency of whole system. So the Fast multiplication is particularly more fundamental for some environments such as central servers, where large numbers of key agreements or signature generations occur, and in handheld devices with low computational power. Because of such importance of scalar multiplication, numerous methods have been developed, such as binary method, signed binary method, sliding window method. In this paper, use Lattice multiplication method that relies on the simple binary method. This method not solely reduces the total number of point addition, but also reduces the number of point doubling. The method is straightforward and particularly appropriate for low time delay. The objective of this paper, proposed a polynomial based novel algorithm that combines the Polynomial based Hashing for authentication and modified polynomial based ECC operations for the security mechanism.

II. METHODOLOGY

Today's, various hybrid algorithm are developed for security purpose in cloud computing such as Blowfish-MD5, ECC-Md5, ECC-SHA, AES-MD5 etc. These old methods are not useful last years because of their security level and the MAC has been broken easily by crackers [4, 5]. In this research aimed at designing a new security method by using a hybrid cryptosystem, for data security in the cloud. The hybrid algorithm that has been implemented is a combination of two popular and most widely used cryptographic Hashing and ECC. The new proposed polynomial based hybrid algorithm combines the PH (Polynomial based Hashing) for authentication and modified ECC operations for the security mechanism. The performance of the proposed algorithm is comparable with other hybrid algorithm ECC-SHA.

2.1 ECC-SHA

Elliptic Curve Cryptography (ECC) with SHA-512: An elliptic curve is given by an equation in the form of $y^2 = x^2+ax+b$, where $4a^3+27b^2 \neq 0$ The finite fields those are

commonly used over primes (FP) and binary field (F2n). The security of ECC is based on the elliptic curve discrete logarithm problem (ECDCP). This problem is defined as: Given point X, Y on elliptic curve, find z such that X=zY. The following steps describe how ECC works with SHA-512.

ECC key generation: To generate a public and private key pair for use in ECC communication the steps followed are:

- Find an elliptic curve E(K), where K is a finite field such as Fp or F2n, and a find point Q on E(K). n is the order of Q.
- Select a pseudo random number x whose value lies as $1 \leq x \leq (n - 1)$.
- Compute point P = xQ.
- ECC key pair is (P, x), where P is public key, and x is private key.

Signature Generation: To create a signature S for message m, using ECC key pair (P, K) over E(k), the following steps followed:

- Generate a random number k such that $1 \leq k \leq (n - 1)$.
- Compute point kQ = (x1, y1).
- Compute r = x1 (mod n). If r = 0, go to step 1.
- Compute k-1 (mod n).
- Compute SHA-512(m), and convert this to an integer e.
- Compute s = k-1(e + xr) (mod n). If s = 0, go to step 1.
- The signature for message m is S = (r, s)

Signature Verification: This part verify a signature s=(r, s) for message m over a curve E(k) using the public key P performing steps:

- Verify r and s are integers over the interval [1, n - 1].
- Compute SHA-512(m) and convert this to an integer e.
- Compute w = s-1(mod n).
- Compute u1 = ew (mod n) and u2 = rw (mod n).
- Compute X = u1Q + u2P
- If X = 0, reject S. Otherwise, compute v = x1 (mod n).
- Accept if and only if v = r.

Encryption: Let 'm' be the message that we are sending. We have to represent this message on the curve. These have in-depth implementation details. Consider 'm' has the point 'M' on the curve 'E'. Randomly select 'k' from [1 – (n-1)]. Two cipher texts will be generated let it be C1 and C2.

$$C1=k*P,\ c2=M + k*Q$$

Decryption: Have to get back the message 'm' that was send

$$M=c2-d*c1$$

M is the original message that we have send.

III.PROPOSED SCHEME (PHECC)

This proposal is aimed to provide more secure method to secure users' data protection, reduce the complexity of key generation, confidentiality and integrity by using modified ECC algorithm. The proposed scheme consists of five phases— Initialization phase, Registration phase, Authentication phase, Encryption phase and Decryption phase. Now each of these phases is discussed below.

Initialization phase

To work with ECC, all the users must agree upon the elements defining the curve. These are called as "Domain Parameters". They are 6 distinct values: (p, a, b, G, n, h). The parameters are the base point (P), the field ($\mathbb{F}_p$), the prime number (p), the order (n), the curve (E), co-factor of curve (h), and the curve's parameters (a, b). Authority ensures that all the users agree on the domain parameters. The definition of an elliptic curve hides a lot of details: An elliptic curve is a smooth degree-3 plane curve. Basically, the elliptic curve lives in the projective plane or usual xy-plane. Every curve defined by a polynomial equation in the usual plane can be written as If f(x, y) = 0 is the equation of the curve in the xy-plane.

In this phase, several security parameters used for authentication and other processes are calculated. First, an elliptic curve equation $E_p(a, b)$: $y^2 = x^3 + ax + b \pmod p$ over a prime finite field $\mathbb{F}_p$, is selected. After agreeing on the domain parameters, The EC is generated by constructing certain polynomials based procedure and fined their roots. The N is a positive integer; N represents the degree of the polynomials at most N-1 and q is the large modulus while p is the small modulus, both of which should be relatively prime to each other. They do not have common divisor. The small polynomials f and g are secret polynomials which are selected by uniform sampling from a set of binary or ternary polynomials whereby a predetermined number of -1, 1 and 0 coefficients have been set. L_f And L_g represent private key spaces which are given by a set of small polynomials from which the private keys will be selected. The polynomials f and g are randomly generated in L_f and L_g respectively. L_m Represents the plain text space, which is given by a set of polynomials which represent the encrypt able message. The message m is expressed in the form of a polynomial whose coefficients are modulo p so that its coefficients lie between $\frac{-p}{2}$ and $\frac{p}{2}$.

Registration phase

Initially, a client A must register to the server S with his own identity ID_A and password-verifier U_A and collects the server's public key U_S , then server stores each legal client's identity, password-verifier, and a status-bit in a write protected file as depicted in the Table 3, where the status-bit indicates the status of the client, i.e., when the

client is logged-in to the server the status-bit is set to one, otherwise it is set to zero.

Table 3
The verifier table.

Identity	Password-verifier	Status-bit
ID_A	$U_A = pw_A \cdot G$	0/1
ID_B	$U_B = pw_B \cdot G$	0/1
ID_C	$U_C = pw_C \cdot G$	0/1
...	...	...

Authentication Phase

The integrity is checked by comparing the hash code which is generated before and after storing data in cloud. The overall steps of the proposed model are as follows,

Authentication steps:

- User calculates the hash value for his password using a highly secure **PH (Polynomial based Hashing)** algorithm. This hash value will be used after that as a reference when it compared with the stored hash code in the server to verify user authentication attempts. Since this hash value produced from a one way function so it can't be regenerated. It is only compared with the hash value that will be generated from the server with the function.
- The user can assure that the message from authorized party. The user then sends his hash value of his password encrypted with polynomial based ECC to the server. The encryption of these parameters over the communicating channel will protect against modification or alteration from unauthorized opponent.
- The server receives the parameters from the user and starts the checking process by verifying the ID of the user from the pre stored database in the server. This assures the identity of the sender and that the message comes from authorized user.
- Server calculates the hash value and compares it with the received value from the user. Since the creation of the hash value in the server from a specific function must be matched with the created value in the server using the same function.
- Server checks the previous parameters (Checking the ID of the user and compare the received hash with the computed one using password verifier) and if the check succeeded.
- Server will generate hashing value using PH algorithm. The server will send the generated hash code encrypted with ECC to the user. The sending hash value to the user will give assurance to the user that the message is originated from the server by comparing it with the generated one.
- The user will check the received message from the server by comparing the received value with the stored one. After the check succeeded, user generates by multiplying the hash by the generating point of the curve and send to the server.
- The server also generates by multiplying the hash with the same generating point of the curve and send to the user. After exchanging the hash value between the user and the server a mutual authentication between the server and user is achieved.

Encryption and Decryption phase

The main parameters of proposed cryptosystem are integer's n, p and q. These values are used to define the following polynomial rings; the notation for the ring is given as:

$$R = Z[X]/(X^N - 1)$$

Where Z represents the set of integers and N is 1 more than the degree of the polynomial. The ring is defined such that the multiplication operation of the polynomials is wrapped around the degree ("size") of the polynomial rather than expanding the degree of the polynomial. Further, each coefficient of the polynomial is an integer that is reduced modulo certain parameters, after every math operation.

Key Generation: To create a public key, one chooses two polynomials such that $f \in L_f$ and $f \in L_g$. Here, polynomials in L_f and L_g have small widths. Also, the polynomial f should have inverses modulo p and q. In other words, one should be able to calculate f_p^{-1} and f_q^{-1} such that

$$f * f_p^{-1} = 1 \pmod{p} \text{ and } f * f_q^{-1} = 1 \pmod{q}$$

Private key is composed of the polynomials f and f_p^{-1} . After choosing the polynomials appropriately, public key can be computed as

$$h = p\, f_q^{-1} * g \pmod{q}$$

Encryption: Now describe how Alice wraps and sends a message to Bob using Bob's public key h. Alice chooses her plaintext m from the set. The proposed system digital envelope depends on the choice of the functions G and H, and on an integer k. The probability of forging a valid ciphertext will be P^{-k}.

$$m \in P_p\,(n - k)$$

Also chooses a random polynomial $r \in \mathcal{L}_r$. Computes

$$e = r \circledast h + [m + H(m, [r \circledast h]_p)X^{n-k} + G([r \circledast h]_p)]_p \pmod{q}$$

Alice then sends e to bob.

Decryption: Suppose that Bob has received the message e from Alice and
wants to decrypt it using his private key f. To do this efficiently, Bob should have precomputed the polynomial f_p^{-1} . In order to decrypt e, Bob first computes the temporary polynomial a by

$$a \equiv f \circledast e \bmod q$$

Where he chooses the coefficients of a in the interval from –q/2 to q/2. Now treating a as a polynomial with integer coefficients, bob computes the temporary polynomial $t \in \mathbb{Z}[X]/(p, X^n - 1)$ by

$$t = f_p^{-1} \circledast a \pmod p$$

Further computes the two temporary quantities

$$b \equiv e - t \pmod p \text{ and } c \equiv t - G(b) \pmod p$$

And then writes c in the form

$$c = c' + c''X^{n-k} \text{ with } \deg(c') < n - k \text{ and } \deg(c'') < k$$

Finally he compares the quantities

$$c'' \text{ and } H(c', b)$$

If they are the same, he accepts c' as a valid decryption. Otherwise he rejects the message as invalid.

IV.EXPERIMENTAL RESULT

This section compares the performances of the scheme designed in this paper with other ECC-based authentication schemes. The implementations are done in a system having dual core Intel i7-6600U processor @ 2.60GHz. Operation system was Linux Ubuntu 16.04.The encryption and decryption algorithm are executed in the console of Google Cloud Platform using java9 and the following results are obtained. The following metrics are used to measure the performance of proposed algorithm.

Execution Time

In Figure 1, the performance evaluation of authentication time using ECC-SHA and polynomial based hashing algorithm (PHECC) is shown. By using PHECC and ECC-SHA algorithms in this research work the authentication time value is 0.457µsec and 0.675µsec. So, the proposed algorithm betters then other algorithms.

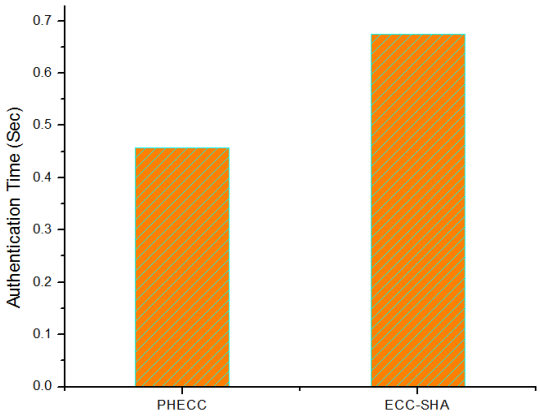

Figure 1: Comparison graph of Authentication Time with ECC-SHA and PHECC

Key Generation Time

A Key pair is being generated. For each and every user a separate private and public key is being generated. The key generation time is different even though the key length is same. It takes the very less time for generation of key having the smaller key size. The graph of PHECC cryptosystem versus the ECC-SHA cryptosystem relating to the key size versus key generation is shown in Figure 2. From the figures, we can clearly conclude that the performance of PHECC is better than ECC-SHA in terms of key generation.

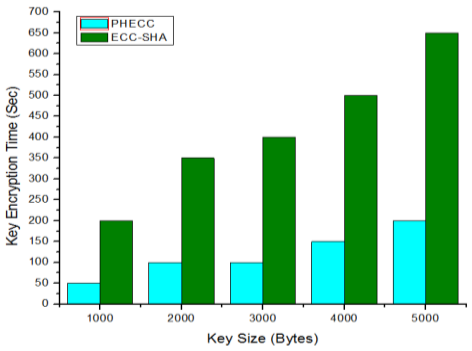

Figure 2: Comparison by Key Generation Time

Encryption / Decryption Time

Runtime measured in milliseconds. Graphical representation of encryption runtime is shown in Figure 3 and decryption runtime show in Figure 4. Encryption time is defined as the amount of time taken by the data owner to encrypt the original data into an encrypted data. Decryption time is defined as the amount of time taken by the data owner to decrypt the encrypted data into plain data. Figure 3 and figure 4 shows the encryption and decryption time of the proposed algorithm with respect to varying data size.

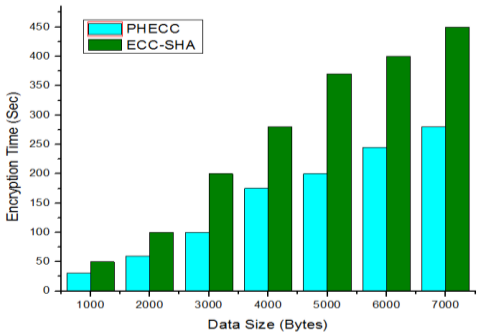

Figure 3: Graph for Encryption Time

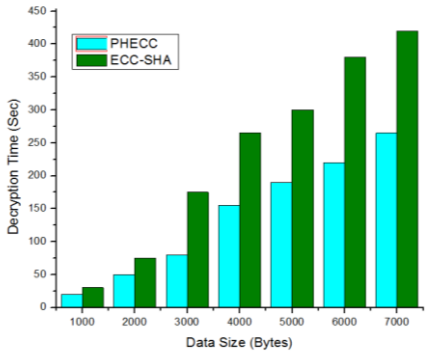

Figure 4: Graph for Decryption Time

Throughput

Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated as:

$$\text{Throughput of encryption} = T_p \text{ (Bytes)} / E_t \text{ (Sec)}$$

Where T_p is the total plain text (bytes) and E_t is the encryption time (second). As the throughput value is increased, the power consumption of this encryption technique is decreased. Figure 5 shows the throughput of PHECC compared with the existing ECC-SHA algorithm for different sizes of plain text. It is shown that PHECC have the same results and they achieve the largest values.

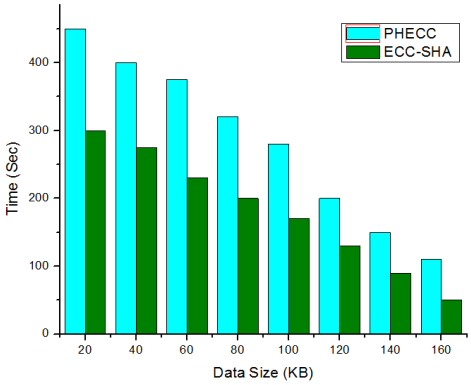

Figure 5: Throughput of ECC-SHA and PHECC

Memory Consumption

Memory consumption is defined as the amount of memory that is utilized for data storage in cloud and also, it is defined as an occupied capacity of CPU. In proposed algorithm we use the lightweight hash function which converts the message into bits which require less amount of memory storage. Also encryption process makes the size of message and key small. Figure 6 shows the memory consumption of the proposed algorithm PHECC with respect to the file size and execution time of the algorithm. From this illustration it is evaluated that the proposed algorithm requires the reduced memory space for storing the data into the cloud server.

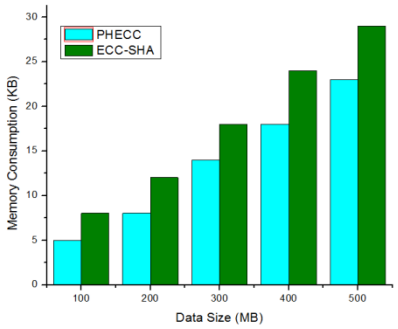

Figure 6: Comparison using Memory Consumption

Power Consumption

Figure 7 shows the power consumption of proposed algorithm for encryption and decryption respectively, and makes a comparison between the proposed protocol and the existing protocols at different sizes of plain text. The results show the superiority of PHECC algorithm over other algorithms in terms of the encryption and decryption power processing (when we encrypt the same data by using ECC-SHA, PHECC algorithms, we found that PHECC requires approximately 30 % of the power which is consumed for the least of other algorithms).

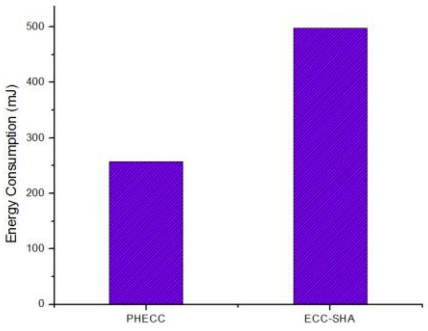

Figure 7: Power Consumption

From the aforementioned analysis, we found that our scheme prevents all possible attacks. However, the present scheme provides the trade-off among storage cost, computational cost, communication cost, execution time and security than existing schemes. Therefore, it can be said that the present scheme delivers efficiency and more security than other schemes. Thus, the present scheme is acceptable for real world applications.

V.CONCLUSION

Now-a-days security has become one of the most important aspects in every field. Each information should be secured as any changes in information leads to very serious problem. Data should be secured from malicious attacks and unauthorized access. In this paper, an efficient algorithm is proposed to secure communication between sender and receiver. The algorithm first uses the lightweight polynomial based hash function for secure authentication process. Then the modified ECC algorithm is used for message and digital signature encryption and utilized for secret key encryption and decryption process. By using the hybrid cryptography algorithm the cloud computing security increases double times in terms of authentication, confidentiality and integrity. This paper conclude that PHECC cryptosystem is fastest public key cryptosystems to provide different security levels at high speed. PHECC provides high security solutions even on constrained devices where bandwidth, storage and computing power are limited. This work would promote confidence in both large and small scale organization in Cloud investment.

VI.REFERENCES

[1]. Zhou Y, Fang Y, Zhang Y, "Securing wireless sensor networks: a survey", Journal of Communications Surveys and Tutorials, IEEE.2008 Sep; 10(3):6-28.

[2]. Ahmad Salehi S, Razzaque MA, Naraei P, Farrokhtala A,"Security in Wireless Sensor Networks: Issues and Challenges", Proceeding of the 2013 IEEE International Conference on Space Science and Communication (IconSpace), Malaysia. 2013 Jul, 356-360.

[3]. Kumar D.Sravana, Suneetha CH., Chandrasekhar A., "Encryption Of Data Using Elliptic Curve Over Finite Fields", International Journal of Distributed and Parallel Systems (IJDPS), Volume 3, No.1, January 2012.

[4]. He D, Kumar N, Chilamkurti N, Lee JH., "Lightweight ECC based RFID authentication integrated with an ID verifier transfer protocol", Journal of Medical Systems 2014; 38(10): 116.

[5]. Leon Groot Bruinderink, Andreas Hlsing, Tanja Lange, and Yuval Yarom, "Flush, gauss, and reload a cache attack on the bliss lattice-based signature scheme", Cryptology ePrint Archive, Report 2016/300, 2016. http://eprint.iacr.org/ 2016/300.

[6]. BinuSumitra, "Secure and Usable Authentication Framework For Cloud Environment", Advances in Intelligent Systems and Computing (springer) 183-202 10.1007/978-81-322-2650-5_12 Publication date 3 Nov 2015.

[7]. SnehaK.Khodke, "Multi Level Authentication Technique for Access Organization in Cloud Data", International Journal of Advance Foundation and Research in Computer (IJAFRC) Volume 1(2348 – 4853) Publication date 12 December 2014.

[8]. Jasleen Kaur, Dr. Sushil Garg, "Security in Cloud Computing using Hybrid of Algorithms", pp. 300-305, 2015.

[9]. Manorama Chauhan, "An implemented of hybrid cryptography using elliptic curve cryptosystem (ECC) and MD5", Inventive Computation Technologies (ICICT), International Conference, 26-27 Aug. 2016

[10]. Akanksha Tomar, Jamvant Singh Kumare, "Survey on Cloud Security by Data Encryption Using Elliptic Curve Cryptography", International Journal of Engineering Sciences & Research Technology, December, 2016