http://dx.doi.org/10.5755/j01.eie.26.1.25312 ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 26, NO. 1, 2020

Storage and Communication Security in Cloud Computing Using a Homomorphic Encryption Scheme Based Weil Pairing

Demet Cidem Dogan*, Huseyin Altindis
Department of Mathematic, Faculty of Science, Erciyes University,
38039 Kayseri, Turkey
demetcidem@erciyes.edu.tr

***Abstract*—With introduction of smart things into our lives, cloud computing is used in many different areas and changes the communication method. However, cloud computing should guarantee the complete security assurance in terms of privacy protection, confidentiality, and integrity. In this paper, a Homomorphic Encryption Scheme based on Elliptic Curve Cryptography (HES-ECC) is proposed for secure data transfer and storage. The scheme stores the data in the cloud after encrypting them. While calculations, such as addition or multiplication, are applied to encrypted data on cloud, these calculations are transmitted to the original data without any decryption process. Thus, the cloud server has only ability of accessing the encrypted data for performing the required computations and for fulfilling requested actions by the user. Hence, storage and transmission security of data are ensured. The proposed public key HES-ECC is designed using modified Weil-pairing for encryption and additional homomorphic property. HES-ECC also uses bilinear pairing for multiplicative homomorphic property. Security of encryption scheme and its homomorphic aspects are based on the hardness of Elliptic Curve Discrete Logarithm Problem (ECDLP), Weil Diffie-Hellman Problem (WDHP), and Bilinear Diffie-Helman Problem (BDHP).**

***Index Terms*—Cryptography; Homomorphic encryption; ECC; Cloud security; Modified Weil pairing.**

I. INTRODUCTION

Cloud computing can be defined as an internet based information processing system that provides easy and customizable services. Cloud computing allows backup/storage of data and management of various applications on central servers. Cloud storage is a data storage structure in cloud computing, where digital data is stored in logical repositories. Stored data can be updated by different users and necessary computations on the stored data may be performed by cloud servers in cloud storage. Also, the data are transferred to the devices when they are requested. Transportation, healthcare, smart city, smart mobility, smart metering, smart grid, etc. are some areas where cloud computing is used. Security and privacy are the major challenges for cloud computing, which prevent the cloud computing from being widely accepted in practice.

Wei *et al.* [1] aim to solve the computation security. Actually, maintaining cloud storage privacy protection and ensuring data transmission security have become important issues for the improvement of cloud computing. Recently, many researches have been conducted on those issues [2]–[4]. In [5], authors propose a method that authorizes only data owner to store and access data in the cloud. In [6] and [7], secure and efficient data forwarding is achieved by key transmission. In [8], encryption scheme has an intuitive key distribution mechanism to enable data access. In [9], there is a masked part of the secret key between the data user and attribute authorities. In [10], two separate cloud systems that communicate with each other are required and the data user must be authenticated.

Encrypted data to be stored in cloud is managed to ensure transmission security. However, the disadvantage is that cloud server first needs to perform decryption when any kind of computation is required. If a user has ability to achieve computation on encrypted data, then the same user can utilize from power of the cloud in a more secure way. Potey *et al.* presented an encryption scheme that provides cloud security using homomorphic encryption (HES) in [11]. Homomorphic encryption technique, which was first suggested by Rivest *et al.* [12], allows calculation on encrypted data. Afterwards, a fully homomorphic encryption proposed in 2009 by Gentry [13], the desired operations (addition/multiplication) can be executed on the encrypted data. So, homomorphic encryption provides better security level in cloud storage. However, there are some disadvantages of full homomorphic encryption schemes, which are the greatness of public key, large expansion rate of the ciphertext, and long consuming time for calculating the ciphertext. A full homomorphic encryption scheme is presented for cloud security in [14]. Nevertheless, the scheme is vulnerable for attacks due to the easy setup of encryption-decryption algorithm. Problems mentioned above are solved by using ECC (Elliptic Curve Cryptography) as in [15]. A new method that balances the load of storage servers and effectively utilizes the server capabilitiesis suggested in [16]. Gupta and Biswas in [17], [18] offer homomorphic encryption scheme for cloud security with use of ECC. ECC is a public key cryptosystem based on elliptic curve's group structure. The essential advantage of ECC becomes to execute same security level by using smaller keys than the

Manuscript received 30 February, 2019; accepted 28 July, 2019.

conventional asymmetric cryptographic schemes based on a factoring module or a discrete logarithm. As mentioned in [19] and [20], the security of ECC is much better than the Rivest, Shamir, Adleman (RSA) cryptosystem and ECC is faster than the RSA cryptosystem.

We use cryptographic pairings in our proposed Homomorphic Encryption Scheme based on Elliptic Curve Cryptography (HES-ECC). Cryptographic pairings have been widely used after an identification based encryption scheme was suggested in [21]. Morales-Sandoval *et al*. [22] offer a pairing based cryptographic scheme that requires a secure hash function. However, the scheme does not have homomorphic properties. Although the cryptographic pairings have been widely used after the identification based encryption scheme was suggested in [21], in literature, there are too few encryption scheme with homomorphic properties using only algebraic structures. The main strength of the Weil pairing in cryptography is its bilinearity and non-degeneracy. However, the Weil paring is trivial when applied to two dependent points. Weil pairing with distortion map is called modified Weil pairing, which does not allow two dependent points as input [23].

In this paper, modified Weil pairing is used for encryption. The security of encryption method we propose is based on the difficulty of Elliptic Curve Discrete Logarithm Problem (ECDLP) and Weil Diffie-Hellman Problem (WDHP). Since our master goal is to assure cloud storage security, we use the homomorphic encryption techniques in our encryption scheme. Modified Weil pairing and bilinear pairing are used for homomorphic property in the paper as well. Hence, the security of homomorphic property is based on ECDPL, WDHP and Bilinear Diffie-Helman Problem (BDHP). The proposed HES-ECC consists of a scheme that uses only algebraic structure of elliptic curves and pairings. Except of them, there is no need for calculations anything like xor operation, hash function, secure key distributor, trusted third party, etc. Open messages cannot be seen along the way or in the public cloud. Since plaintext is not used along the way, safe transmission is provided.

II. DEFINITIONS AND BASIC PROPERTIES

A. Elliptic Curve

Let q is a prime power. Let the finite field with containing q element is denoted by F_q. An elliptic curve $E(F_q)$ consists of the point at infinity O and the set of all solutions (x, y) over F_q to an equation

$$y^2 + a_1xy + a_3y = \\ = x^3 + a_2x^2 + a_4x + a_6, \tag{1}$$

where $a_j \in F_q$ for all j. Elliptic curves have two operations, which are a point addition and a scalar multiplication [24].

1) Weil Pairing

Let l be a positive integer, which is prime to the characteristic of F_q $(char(F_q))$, where $char(F_q) = p$. Let $\overline{F_q}$ be an algebraic closure of F_q. Let set of points of order l is $E[l] = \{P \in E(\overline{F_q}) \mid lP = O\}$. The Weil pairing of order l is the map

$$e_l : E[l] \times E[l] \longrightarrow \mu_l, \tag{2}$$

where μ_l is the set of l th roots of unity in $\overline{F_q}$. The Weil pairing has some properties, such as bilinearity, non-degenerate in each variable $e_l(T, T) = 1$ for all $T \in E[l]$ [25].

2) Modified Weil Pairing

If the point P and $Q \in E[l]$ are linearly dependent, then $e_l(P, Q) = 1$, since the Weil pairing has bilinearity property. This causes some trouble in many cryptographical applications. The trouble can be avoided using distortion maps. A distortion map on $E(F_q)$ is an endomorphism σ of $E(F_q)$, such that $\sigma(P) \notin \langle P \rangle$, where $\langle P \rangle$ is a group of points generated by P. We can define a modified Weil pairing e_l as follows. Let $P \in E(F_q)$ be a point of order n and let G_1 be the subgroup of points generated by P. Let G_2 be the subgroup of $F^*_{q^k}$ of order n for some k [26]

$$e_l : G_1 \times G_1 \longrightarrow G_2,\ e_l(P, Q) = e_l(P, \sigma(Q)). \tag{3}$$

The properties of modified Weil pairing are given as follows.

Bilinear: For all $P, Q \in G_1$ and for all $a, b \in \mathbb{Z}$ we have

$$e_l(aP, bQ) = e_l(P, Q)^{ab}. \tag{4}$$

Non-degenerate: If P is a generator of G_1, then $e_l(P, P)$ is a primitive n th root of unity.

Computable: Given $P, Q \in G_1$, there is an efficient algorithm to compute $e_l(P, Q) \in G_2$ [21].

B. Bilinear Pairing

G and G_2 are two multiplicative groups of some order n. A bilinear map $\hat{e} : G_2 \times G_2 \longrightarrow G$ is defined by following three properties:

Bilinear: $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$ for all $u, v \in G_2$ and all $a, b \in \mathbb{Z}$.

Non-degenerate: There exists an u in G_2, such that $\hat{e}(u, u) \neq 1$.

Computable: Given $u_1, v_1 \in G_2$, there is an efficient algorithm to compute $\hat{e}(u_1, v_1) \in G$ [17].

C. Homomorphic Encryption

For an encryption scheme, if deciphering the encrypted results after certain mathematical operations applied on ciphertext is equal to results after certain mathematical operations applied on the plain text, then the scheme is called homomorphic encryption. Namely, let *Enc* is encryption function, *Dec* is decryption function, and m_a, m_b are any two plaintext. Let $\oplus$ is the addition operation and $\odot$ is the multiplication operation defined on cipher text

$$m_a + m_b = Dec\left(Enc\left(m_a\right) \oplus Enc\left(m_b\right)\right). \quad (5)$$

This property is called additional homomorphic

$$m_a * m_b = Dec\left(Enc\left(m_a\right) \odot Enc\left(m_b\right)\right). \quad (6)$$

Also, this property is called multiplicational homomorphic. If an encryption scheme provides these properties, then it is called homomorphic encryption scheme.

D. Hard Assumption Problems

Some of the assumptions, which are hard to solve, are used in the proposed work and are given below.

1) Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given $P, Q \in E\left(F_q\right)$ and $Q = kP$, it is hard to find $k \in \mathbb{Z}$ [24].

2) Weil Diffie-Hellman Problem (WDHP)

Let G_1 and G_2 are defined as in Section II-A2. Given $P, aP, bP, cP \in G_1$ for random $a, b, c \in \mathbb{Z}_q^*$, it is difficult to calculate $e_l\left(P, P\right)^{abc} \in G_2$ [27].

3) Bilinear Diffie-Hellman Problem (BDHP)

Let G_2 and G are two multiplicative groups defined as in Section II-A3 and let $u \in G_2$ is a fix generator point. Given $u, u^a, u^b, u^c \in G_2$ for random $a, b, c \in \mathbb{Z}_q^*$, it is difficult to calculate $\hat{e}\left(u, u\right)^{abc} \in G$ [28].

III. Proposed HES-ECC

The purpose of the proposed encryption scheme is to ensure cloud security. It aims to prevent the leakage of data from original file by processing on the encrypted text without applying any decryption process. General progress of HES-ECC in the cloud is demonstrated in Fig. 1, where Enc is encryption function and Dec is decryption function.

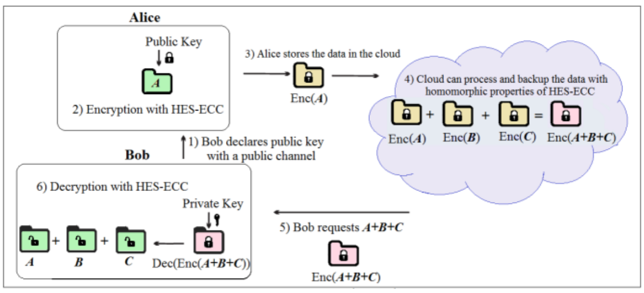

Fig. 1. The using of HES-ECC in the cloud.

Firstly, a public key is generated and it is declared by a public channel. Then, the open data are encrypted with user's public key and the data are stored as encrypted in the cloud. Finally, the user can decrypt the changed encrypted message with the own private key. The encrypted data are not understandable for remote server due to remote server can see only the encrypted message. Thanks to the homomorphic property, even if the cloud server performs some operations (addition/ multiplication) to the encrypted messages, the user can decrypt it.

The proposed scheme is made of the following steps: key generation, encryption, decryption, and evaluation. Firstly, Bob declares his public keys. Then, Alice encrypts the open messages with these public keys and stores them in public cloud. Finally, Bob can decrypt these processed messages with his private key. Cloud server can process these messages, but cannot see plaintexts as well. Hereby, the cloud computing security and communication safety is ensured.

A. Step 1: Key Generation

Let we choose two different big primes without loss of generality; say the greater one is $n.$ Let G_1 is defined as in Section II-A2 with P be a base point of order $n.$ Let q_1 is the other great prime. Let g_2 and g are primitive roots of G_2 and $G,$ which are described as in Section II-B, respectively. Let $n - q_1 = q_2.$ Calculate $Q = q_2 P$ and $R = -q_1 P.$ The public key and the private key are described in (7) and (8), respectively:

$$PK = \left(n, G_1, G_2, G, e_l, \hat{e}, P, Q\right), \quad (7)$$

$$PR = q_1. \quad (8)$$

Receiver Bob generates his PK and PR with this key generation procedure. While Bob declares his PK with a public channel, he keeps secret his PR .

B. Step 2: Encryption

$M = \{0, 1, 2, ..., I\}$ is the message space, where $I < n.$ A sender Alice wants to send an open message $m \in M$ to receiver Bob. Firstly, Alice gets Bob\`s PK and calculates $E_1 = e_l\left(mP - Q, P\right) \in G_2,$ where e_l is defined in Section II-A2 and E_1 is the encrypted message of m . Then, she sends E_1 to Bob and stores E_1 on the cloud storage.

C. Step 3: Decryption

Bob computes $E_2 = e_l\left(R, P\right)$ with his private key q_1 after receiving the encrypted message E_1 from the cloud. Bob multiplies E_1 and E_2 in order to decrypt the open message. He obtains $g_2{}^m$ and he can compute discrete logarithm of $g_2{}^m$ base g_2 . So, he gets Alice's open message $m.$

The verification of decryption process is as follows:

$$\begin{aligned} E_1 * E_2 &= e_l\left(mP - Q, P\right) e_l\left(R, P\right) = \\ &= e_l\left(mP - Q, P\right) e_l\left(-q_1 P, P\right) = \\ &= e_l\left(mP - q_2 P, P\right) e_l\left(-q_1 P, P\right) = \\ &= e_l\left((m - q_2) P, P\right) e_l\left(-q_1 P, P\right), \end{aligned} \quad (9)$$

Since $m-q_2<n$ and e_l bilinear

$$E_1*E_2=e_l\left(P,P\right)^{m-q_2-q_1}. \quad (10)$$

Since $e_l\left(P,P\right)=g_2$ and $q_1+q_2=n$

$$E_1*E_2=g_2{}^{m-n}. \quad (11)$$

Since $g_2{}^n=1$

$$E_1*E_2=g_2{}^m. \quad (12)$$

D. Step 4: Evaluation

Evaluation step explains the homomorphic properties of presented HES-ECC. Alice stores encrypted data into the cloud. As additional homomorphic properties, Bob requests the encrypted data added two open messages m_a, m_b corresponding to the encrypted data E_{1a}, E_{1b} from cloud server. We define $E_{1a}\oplus E_{1b}=E_{1a}*E_{1b}*A$, where $A=e_l\left(rQ,P\right)$, such as $r+1$, is the number of E_i to be added $i\in\{1a,1b,1c,...\}$. The cloud computes $E_{1a}\oplus E_{1b}$ and sends to Bob. Bob can decrypt $E_{1a}\oplus E_{1b}$ using decryption algorithm and obtains $\left(m_a+m_b\right)\bmod n$. Verification of additional homomorphic property can be explained as follows. Here, $r+1=2$ due to addition of two messages. Let say $E_{1(a+b)}$ is encrypted data for m_a+m_b

$$\begin{aligned}&E_{1a}\oplus E_{1b}=E_{1a}*E_{1b}*A=E_{1a}*E_{1b}*e_l\left(rQ,P\right)=\\&=e_l\left(m_aP-Q,P\right)e_l\left(m_bP-Q,P\right)e_l\left((2-1)Q,P\right)=\\&=e_l\left(P,P\right)^{m_a+m_b-q_2}=e_l((m_a+m_b)P-Q,P)=E_{1(a+b)}.\end{aligned} \quad (13)$$

As multiplicational homomorphic properties, Bob requests encrypted data that multiplied two open message m_a, m_b corresponding to encrypted data E_{1a}, E_{1b} from cloud server.

We define $E_{1a}\odot E_{1b}=\hat{e}\left(E_{1a},E_{1b}\right)$ where $\hat{e}$ is defined at Section II-B. The cloud computes $E_{1a}\odot E_{1b}$ and sends the result to Bob. Before applying decryption algorithm, Bob computes $\left(m_a+m_b-q_2\right)^{-1}\equiv t(\bmod n)$ with $m_a+m_b(\bmod n)$ and computes $\left(E_{1a}\odot E_{1b}\right)^t$. Bob can decrypt $\left(E_{1a}\odot E_{1b}\right)^t$ via multiplying it to $E_2\odot g_2$. He obtains $m_am_b(\bmod n)$ with discrete logarithm of $g^{m_am_bt}$ base g^t. This calculation consume time of $O\sqrt{I}$ with lambda method of Polard.

Verification of multiplicative homomorphic property is as follows:

$$\begin{aligned}&\left(E_{1a}\odot E_{1b}\right)^t=\left(\hat{e}\left(E_{1a},E_{1b}\right)\right)^t=\left(\hat{e}\left(g_2{}^{m_a-q_2},g_2{}^{m_b-q_2}\right)\right)^t=\\&=\hat{e}\left(g_2,g_2\right)^{m_am_bt-q_2(m_a+m_b-q_2)t},\end{aligned} \quad (14)$$

Since $\hat{e}\left(g_2,g_2\right)=g$:

$$\left(E_{1a}\odot E_{1b}\right)^t=g^{m_am_bt-q_2}, \quad (15)$$

$$\left(E_{1a}\odot E_{1b}\right)^t*E_2\odot g_2=g^{m_am_bt-q_2}*g^{-q_1}=g^{m_am_bt}. \quad (16)$$

The implementation of the proposed scheme is based just on elliptic curve point operations and calculation of modified Weil pairing and bilinear pairing. These operations can be computed using [29]–[31].

IV. SECURITY ANALYSIS

Proposed HES-ECC fairly assures against an eavesdropper due to the properties of ECDLP, WDHP, and BDHP. The following theorem shows that eavesdropper's attack to the secret key or open messages is smaller than a negligible function under hard assumption problems.

Theorem 1: If each polynomial time randomly generates eavesdropper, which can be neglected for attacking secret and original data, then it is said that the proposed HES-ECC is probably secure against the attack of eavesdropper.

Proof: Let an eavesdropper with his Ψ views (x,y) that information exchanges between Alice and Bob on insecure channel. Suppose that the eavesdropper can process his achieving data with A. The proposed HES-ECC can be expressed as

$$\begin{aligned}&\left\langle Alice(m,m_a,m_b),Bob(q_1)\right\rangle\left(n,G_1,G_2,G,e_l,\hat{e},P,Q\right)=\\&=\left\langle E_1,E_{1a}\oplus E_{1b},E_{1a}\odot E_{1b}\right\rangle.\end{aligned} \quad (17)$$

Where Alice's inputs are open data and Bob's input is his secret key under public key the output of encryption process is $\left\langle E_1,E_{1a}\oplus E_{1b},E_{1a}\odot E_{1b}\right\rangle$.

Eavesdropper can just reach to $E_1, E_{1a}\oplus E_{1b}, E_{1a}\odot E_{1b}$ and PK. Possible adversary attacks on $PR=q_1$ of Bob is calculated as follows:

$$\begin{aligned}&\Pr\left[\mathrm{A}(PK,\Psi\left(\left\langle Alice(m,m_a,m_b),Bob(q_1)\right\rangle\left(n,G_1,G_2,G,e_l,\hat{e},P,Q\right)\right)=q_1\right]=\\&=\Pr\left[\mathrm{A}(PK,(x,y))=q_1\right]=\Pr\left[\mathrm{A}(PK,\left\langle E_1,E_{1a}\oplus E_{1b},E_{1a}\odot E_{1b}\right\rangle)=q_1\right]=\\&=\Pr\left[\mathrm{A}(PK,\left\langle e_l\left(mP-Q,P\right),e_l\left(P,P\right)^{\beta_1},\hat{e}\left(g_2,g_2\right)^{\beta_2}\right\rangle=q_1\right]<negl.\end{aligned} \quad (18)$$

Possible adversary attacks on open message is calculated as follows:

$$\Pr\left[\mathrm{A}(PK, \Psi\left(\langle Alice(m, m_a, m_b), Bob(q_1)\rangle\left(n, G_1, G_2, G, e_l, \hat{e}, P, Q\right)\right) = m\right] =$$
$$= \Pr\left[\mathrm{A}(PK, (x, y)) = m\right] = \Pr\left[\mathrm{A}(PK, \langle E_{1,} E_{1a} \oplus E_{1b}, E_{1a} \odot E_{1b}\rangle) = m\right] =$$
$$= \Pr\left[\mathrm{A}(PK, \left\langle e_l\left(mP - Q, P\right), e_l\left(P, P\right)^{\beta_1}, \hat{e}\left(g_2, g_2\right)^{\beta_2}\right\rangle = m\right] < negl, \tag{19}$$

where *negl* is defined as negligible function $\beta_1 = m_a + m_b - q_2$, $PK = \left(n, G_1, G_2, G, e_l, \hat{e}, P, Q\right)$, and $\beta_2 = m_a m_b - q_2(m_a + m_b - q_2)$. This inequalities hold under WDHP, BDHP, and ECDLP assumptions.

TABLE I. COMPARISON OF PROPOSED HES-ECC WITH OTHER METHOD IN THE LITERATURE.

Method	Homomorphism	ECC	Hash Function	Constraints
Proposed HES-ECC	Additional and Multiplicational	Yes	No	
Ref. [5]	No	No	Yes	Targets encryption with large key
Ref. [6]	No	Yes	Yes	Uses key aggregate
Ref. [9]	No	Yes	Yes	Uses solving key escrow problem
Ref. [10]	Additional	No	No	Needs two separate cloud system
Ref. [14]	Additional and Multiplicational	No	No	
Ref. [17]	Additional	Yes	No	
Ref. [18]	Additional and Multiplicational	Yes	No	

If an encryption scheme for cloud storage has homomorphic properties, then cloud computing can provide complete security assurance in terms of privacy protection, confidentiality and integrity. Because computation and transmissions are provided using encrypted data, key transfer and decryption are not required on cloud. Also, ECC is faster than other known encryption algorithms, such as RSA and ECC is more secure with a smaller key. HES-ECC does not need hash function. We cannot say that the hash function is a disadvantage, but secure hash is required to avoid collisions, otherwise the encryption scheme is vulnerable. However, a secure hash function is complicated to implement and computationally expensive to compute. In the light of this information, HES-ECC proposes secure communication and data storage.

V. CONCLUSIONS

In this paper, we propose a public key encryption scheme, which is a homomorphic encryption scheme based on elliptic curve cryptography (HES-ECC) in order to provide security of cloud storage. This scheme offers a design that gives addition or multiplication of original data by processing on the encrypted data stored in cloud. The cloud only displays encrypted data. HES-ECC allows to process this data by the cloud server. After taking processed encrypted data, the receiver can decipher them and obtain open message with these operations processed on it. The plaintext cannot be seen, except sender and receiver. Therefore, safe communication is provided. The implementation of HES-ECC is based on modified Weil pairing, bilinear pairing, and elliptic curve operations. Hence, the scheme is simple for practice. Also, a security analysis is given in the paper. So, it is shown that the HES-ECC is secure by dint of the hard assumption problems.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos, “Security and privacy for storage and computation in cloud computing”, *Information Sciences*, vol. 258, pp. 371–386, 2014. DOI: 10.1016/j.ins.2013.04.028.

[2] P. Awasthi, S. Mittal, S. Mukherjee, and T. Limbasiya, “A protected cloud computing algorithm using homomorphic encryption for preserving data integrity”, *Advances in Intelligent Systems and Computing*, vol. 707, pp. 509–517, 2019. DOI: 10.1007/978-981-10-8639-7_53.

[3] L. V. Silva, P. Barbosa, R. Marinho, and A. Brito, “Security and privacy aware data aggregation on cloud computing”, *Journal of Internet Services and Applications*, vol. 9, pp. 1–13, 2018. DOI: 10.1186/s13174-018-0078-3.

[4] A. Viejo and D. Sanches, “Secure and privacy-preserving orchestration and delivery of fog-enabled loT services”, *Ad Hoc Networks*, vol. 82, pp. 113–125, 2019. DOI: 10.1016/j.adhoc.2018.08.002.

[5] S. Sirinat, A. Pal, and G. Deepak, “A novel iris based data storage and retrieval in cloud environment (ibds)”, in *Proc. of 2017 International Conference on Innovations in information Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 2017, pp. 1–5. DOI: 10.1109/ICIIECS.2017.8276123.

[6] B. Cui, Z. Liu, and L. Wang, “Key aggregate searchable encryption (kase) for group data sharing via cloud storage”, *IEEE Transactions on Computers*, vol. 65, pp. 2374–2385, 2016. DOI: 10.1109/TC.2015.2389959.

[7] B. Ramachandran and K. Subramaniam, “Secure and efficient data forwarding in untrusted cloud environment”, *Cluster Computing*, vol. 1, pp. 1–9, 2018. DOI: 10.1007/s10586-018-2240-x.

[8] A. Paverd, S. Tamrakar, H. L. Nguyen, P. Pendyala, D. T. Nguyen, and E. Stobert, “Omnishare: Encrypted cloud storage for the multi-device era”, *IEEE Internet Computing* (to be published). DOI: 10.1109/MIC.2018.182130646.

[9] V. K. A. Sandor, Y. Lin, X. Li, F. Lin, and S. Zhang, “Efficient decentralized multiauthority attribute based encryption for mobile cloud data storage”, *Journal of Network and Computer Applications*, vol. 129, pp. 25–36, 2019. DOI: 10.1016/j.jnca.2019.01.003.

[10] Y. Liu, Y. Luo, Y. Zhu, Y. Liu, and X. Li, “Secure multilabel data classifications in cloud by additional homomorphic encryption”, *Information Sciences*, vol. 468, pp. 89–102, 2018. DOI: 10.1016/j.ins.2018.07.054.

[11] M. M. Potey, C. A. Dhote, and D. H. Sharma, “Homomorphic encryption for security of cloud data”, *Procedia Computer Science*, vol. 79, pp. 175–181, 2016. DOI: 10.1016/j.procs.2016.03.023.

[12] R. L. Rivest, L. Adleman, and M. L. Dertouzos, “On data banks and privacy homomorphism”, *Foundations of secure computation*, vol. 4, pp. 169–180, 1978.

[13] C. Gentry, “A fully homomorphic encryption scheme”, Ph.D. dissertation, Dept. Computer Science, Stanford Univ., Standford,

ABD, 2009.

[14] F. Zhao, C. Li, and C. F. Liu, "A cloud computing security solutions based on fully homomorphic encryption", in *Proc. of 16th International Conference on Advanced Communication*, Pyeongchang, South Korea, 2014, pp. 485–488. DOI: 10.1109/ICACT.2014.6779008.

[15] M. Q. Hong, P. Y. Wang, and W. B. Zhao, "Homomorphic encryption scheme based on elliptic curve cryptography for privacy protection of cloud computing", in *Proc. of 2016 IEEE 2nd International Conference on Big Data Security on Cloud*, New York, 2016, pp. 152–157. DOI: 10.1109/BigDataSecurity-HPSC-IDS.2016.51.

[16] R. Singh, P. K. Gupta, , P. Gupta, R. Malekian, B. T. Maharaj,D. Andriukaitis, A. Valinevicius, D. C. Bogatinoska, A. Karadimce, "Load Balancing of Distributed Servers in Distributed File Systems", in *Proc. of ICT Innovations 2015: Emerging Technologies for Better Living*, Ohrid, Macedonia, 2015, pp. 29–37, 2015. DOI: 10.1007/978-3-319-25733-4_4.

[17] D. S. Gupta and G. P. Biswas, "A secure cloud storage using ECC-based homomorphic encryption", *International Journal of Information Security and Privacy*, vol. 11, pp. 54–62, 2017. DOI: 10.4018/IJISP.2017070105.

[18] D. S. Gupta and G. P. Biswas, "Secure computation on cloud storage", *Journal of Cases on Information Technology*, vol. 17, pp. 1–8, Jul. 2015. DOI: 10.4018/JCIT.2015070103.

[19] M. Suarez-Albela, T. M. Fernandez-Caram, P. Fraga-Lamas, and L. Castedo, "A Practical performance comparison of ECC and RSA for resource- constrained loT devices", in *Proc. of 2018 Global Internet of Things Summit (GIoTS)*, Bilbao, Spain, Jun. 2018, pp. 1–6. DOI: 10.1109/GIOTS.2018.8534575.

[20] H. Eberle, N. Gura, S. C. Shantz, V. Gupta, L. Rarick, and S. Sundaram, "A public key processor for RSA and ECC", in *Proc. of IEEE International Conference on Application-Specific Systems Architectures and Processors*, Galveston, TX, USA, 2004, pp. 98–110. DOI: 10.1109/ASAP.2004.1342462.

[21] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing", *SIAM Journal on Computing*, vol. 32, pp. 586–615, 2003. DOI: 10.1007/3-540-44647-8_13.

[22] M. Morales-Sandoval, J. L. Gonzalez-Compean, A. Diaz-Perez, and V. J. Sosa, "A pairing based cryptographic approach for data security in the cloud", *International Journal of Information Security*, vol. 17, pp. 441–461, 2018. DOI: 10.1007/s10207-017-0375-z.

[23] E. R. Verheul, "Evidence that XTR is more secure than supersingular elliptic curve cryptosystems", *Journal of Cryptology*, vol. 17, pp. 277–296, 2004. DOI: 10.1007/s00145-004-0313-x.

[24] L. C. Washington, *Elliptic Curves Number Theory and Cryptography*. USA, Chapman and Hall, 2008, pp. 9–173. DOI: 10.1201/9781420071474.

[25] J. Hoffstein, J. Pipher, and J. H. Silverman, *An introduction to Mathematical Cryptography*. USA, Springer Science+Business, 2008, Ch. 5.

[26] S. Chatterjee, A. Menezes, and F. R. Henríquez, "On instantiating pairing-based protocols with elliptic curves of embedding degree one", *IEEE Transactions on Computers*, vol. 66, no. 6, pp. 1061–1070, Jun. 2017. DOI: 10.1109/TC.2016.2633340.

[27] A. Joux, "A one round protocol for tripartite Diffie Hellman", *Journal of Cryptography*, vol. 17, no. 4, pp. 263–276, Jun. 2004. DOI: 10.1007/s00145-004-0312-y.

[28] D. Boneh, G. D. Crescenzo, R. Ostravsky, and G. Persiano, "Public key encryption with keyword search", in *Proc. of Advance in Cryptography- EUROCRYPT 2004*, Interlaken, 2004, pp. 506–522. DOI: 10.1007/978-3-540-24676-3_30.

[29] C. M. Park, M. H. Kim, and M. Yung, "A remark on implementing the Weil pairing", *Information Security and Cryptology CISC*, vol. 3822, pp. 313–323, 2005. DOI: 10.1007/11599548_27.

[30] V. S. Miller, "The Weil pairing and its efficient calculations", *Journal of Cryptology*, vol. 17, no. 4, pp. 235–261, 2004. DOI: 10.1007/s00145-004-0315-8.

[31] J. Ding, S. Li, and Z. Gu, "High speed ECC processor over NIST prime fields applied with Toom cook multiplication", *Transactions on Circuits and Systems I*, vol. 66, pp. 1003–1016, 2019. DOI: 10.1109/TCSI.2018.2878598.