ERSA: Enhanced RSA Cryptography Algorithm to Guarantee High Security Level for Data in Cloud Environment

*[1]Mrs. K. Sivakami and [2]Dr. V. Umadevi
[1]Research Scholar, [2]Director of Research
[1,2]Jairams Arts and science College, Karur.
*[1]sivakamikathiresan84@gmail.com, [2]yazh1999@gmail.com

*Corresponding author – Mrs. K. Sivakami

Abstract- Cloud computing includes ubiquitous advantages and applications that probe many enterprises towards it. However, data security and user security are still major trepidation in cloud environment where an efficient cryptography scheme is required. To tackle this security problem, this paper presents a novel enhanced RivestShamir Adleman (ERSA) algorithm to ensure high security level in cloud environment. Here three major phases are performed in ERSA algorithm to resolve security threats in cloud. Security enhancement is realized by following phases: (i) authentication phase in which ERSA based digital signature is utilized for user authentication, (ii) evaluation phase exploits fuzzy inference system (FIS) for ensuring required security level, and (iii) encryption phase involves with ERSA algorithm for encrypting data to be stored in cloud. ERSA algorithm improves security in two ways. One is by generating prime numbers accordance to required security level using Sieve of Atkins (SoA) algorithm. Another one is by enhancing security of RSA key with the help of non-prime factor. Involvement of non-prime factor improves security level while involvement of SoA minimizes time consumed for key generation. Thus RSA algorithm isimproved in terms of security level and time consumption through simple computations in ERSA algorithm. Extensive simulation results ensure better performance in encryption time (minimized by 10s), decryption time (minimized by 140ms), key generation time (minimized by 11s), and security level (improved by 17.5%).

Keywords: Enhanced-RSA, Sieve of Atkins, Fuzzy inference system, Non-prime factor, Authentication, Cloud

I. Introduction

Cloud computing is an emerging paradigm for this technology era [1], [2]. Cloud computing offers distributed resource utilization among users through internet. Perhaps, cloud computing strikes with many advantages it is also necessary to consider another perspective of cloud [3]. Security for both user and data is major threat in cloud. Cryptography schemes are utilized to ensure data security in cloud computing over years [4]. Hash functions, RSA based homomorphic verifiable tags, pseudo random functions, homomorphic verifiable responses, hash index hierarchy, etc. are used for data security. In addition, hybrid cryptography schemes are also contributed in data security [5]. Proxy-re encryption scheme, elliptic curve key cryptography (ECC) with identity based cryptography (IBC), advance encryption standard (AES) with Diffie-Hellman key exchange are utilized in data security. In general, cryptography schemes are broadly classified into two major categories such as symmetric key cryptography and asymmetric key cryptography [6], [28]. Symmetric key cryptography schemes utilize single key for encryption and decryption while asymmetric key cryptography schemes utilize separate keys for encryption and decryption. Blowfish algorithm, data encryption standard (DES), AES, two-fish algorithm are categorized under symmetric key cryptography category [7]. RSA algorithm, Diffie-Hellman algorithm, ECC algorithm, digital signal algorithm and so on are come under asymmetric key cryptography schemes [8]. Here it is concluded that asymmetric key cryptography techniques provide high data security and confidentiality over symmetric key cryptography techniques.

Many researchers focused on improving RSA algorithm in terms of security level and time consumption [9]. Modified RSA algorithm with multiple public keys, personal information protection approach based on RSA, i-RSA, modified RSA based on offline storage and prime number, etc. are some of the variants of RSA algorithm. In [10], secure execution of RSA cryptosystem is presented with two different random numbers and two different prime values. Optimal asymmetric encryption padding (OAEP) is often combined with RSA algorithm in order to improve security level [11]. RSA is utilized for generating digital signatures as well as for authentication. Zero knowledge proof and RSA (Z-RSA) algorithm is presented to provide user authentication [12]. Hybrid RSA algorithm with blowfish algorithm is introduced for enhance data security [13].

The major contributions of this paper is listed as follows,

- A novel ERSA algorithm is proposed to improve traditional RSA algorithm in terms of security level and time consumption in cloud computing environment.
- Proposed ERSA algorithm generates prime numbers based on required security level usingSoA algorithm. Non-prime factor that is unbreakable by attackers is contributed in enhancing RSA algorithm.

- Efficient user authentication is realized by utilizing ERSA based digital signature algorithm. Authentication process is involved with random ID which is derived from ECC algorithm and unpredictable by attackers.
- Security level provided by ERSA algorithm is also evaluated by FIS in cloud server. Here number of generated prime numbers, key size, and security level required by user are considered as input parameters in FIS.

The rest of this paper is organized as follows: Section II, surveys related works on cryptography techniques in cloud environment. In section III, we define the problems presented in existing research works. Section IV, details the proposed ERSA algorithm based security system in cloud environment. In section V, performance of proposed ERSA algorithm is evaluated based on performance metrics. In section VI, we conclude our contributions.

II. Related Works

In this section, we review existing research works held on cloud environment in the perspective of data security. Through this survey, requirement of efficient security system in cloud environment is identified.

2.1 Authentication in cloud environment

Multi-biometric based authentication scheme was presented in cloud computing with three modalities [14]. Here face, iris, and fingerprint of user were consideredaccordance to user's performance strength metrics (i.e. false acceptance and false injection rates) for authentication. The multimodal result was obtained by a user-specific weighted score level fusion strategy. However, authentication based on biometric increases complexity and error rate in authentication process.

Shared authority based privacy-preserving authentication (SAPA) protocol was introduced in cloud computing [15]. Anonymous access request matching mechanism was utilized to attain shared access authority. Access control was realized by attribute based access control mechanism. Data sharing distribution among multiple users was performed in secure manner using proxy re-encryption algorithm. But in this method, complexity and overhead is increased for user authentication which makes the system slow in the involvement of numerous users.

Authentication and access control were enabled in cloud environment in order to defend against replay attacks [16]. Authentication was performed with the help of attribute based encryption (ABE) scheme and attribute based signature scheme (ABS). Several key distribution centers were deployed in environment in order to provide decentralized authentication. Access policies were employed for users to perform creation, deletion, and modification on outsourced data. Involvement of multiple key distribution centers increases space complexity and limits the efficient network management. Again, in this method all user attributes and access policies are recorded by cloud server.

2.2 Security in cloud computing

Data security in cloud computing was ensured with the support of fog computing [17]. Here, three-layer cloud storage framework was designed and Hash-Solomon code algorithm was utilized to divide data into different parts. Then the divided data parts were stored in fog storage, local machine and cloud storage to improve security and privacy of data. The distribution proportion of stored data in fog, local machine, and cloud was computed by computational intelligence algorithm.

To ensure data security in an efficient manner, data classification was performed [18]. Here, data was classified into three categories as follows: (i) access control, (ii) content, and (iii) storage. Access control was defined by restrictions such as frequency of access, frequency of update, visibility, accessibility, and retention. Content type was defined based on precision, reliability, degree of completeness, consistency, and auditability. Data was categorized under storage category based on storage encryption, communication encryption, integrity, and backup plan. Classification of all data stored in cloud storage becomes complex and time-consuming process.

SecCloud was security protocol developed to provide both storage and computation security in cloud storage [19]. SecCloud protocol was relied on identity-based cryptography. In this protocol, security was ensured by performing four steps as follows: (i) system initialization phase, which includes system setup, and user registration, (ii) secure cloud storage phase, in which data signing, data encapsulation, and data receiving steps were involved, (iii) secure cloud computation phase was involved with computation request and commitment generation steps, (iv) computation result auditing step in which Merkle hash tree construction was included. Perhaps, this protocol improves data security level, this protocol was not able to provide efficient authentication.

MetaCloudDataStorage was a security architecture that attempts to ensure high-level security for sensitive data in cloud storage [20]. For this purpose, initially user data was classified into sensitive, critical, and normal data. Based on data type, the data was stored in correspondence datacenters instead of same datacenter. HereMetaCloudDataStorage interface was able to redirect the user data to corresponding datacenter. In different datacenters, different level security was maintained. In this architecture, if the user data relies on same category then load and space complexity on particular datacenter is increased.

2.3 Improvements in RSA algorithm

Four prime numbers were generated in fast cloud-RSA algorithm in order to improve security level [21]. Here evaluation key that was shared by userwas computed by

multiplying four prime numbers and private key was generated using four prime numbers and other four numbers. These numbers were derived from four prime numbers through mathematical computations.Here security level is low due to simple computation for private key generation. If the four prime numbers are obtained from evaluation key, then the private key can be easily derived

RSA algorithm and ABE scheme were integrated to enable high-level security in cloud computing [22]. In this method, public and private keys were generated based upon security parameters of ABE scheme.Key generation was realized by RSA algorithm. In this method, user was allowed to choose subset of attributes to generate digital signatures while the sign verifier was need not to know the identity of user. One major limitation of this work is, if the number of attributes selected by user is large then the signature length generated by RSA is relatively large. In general, increasing number of attributes is able to improve security with high time consumption and complexity.

K-nearest neighbor (KNN) approach was combined with RSA algorithm in cloud environment [23]. In this scheme, initially four prime numbers were generated in RSA algorithm for generating private key and public key. Message to be encrypted was converted into ASCII value before encryption. Here, KNN algorithm was utilized to generating prime numbers in the case of initially generated prime numbers and ASCII value of message were same. Perhaps this method improves security level it also increases time consumption by generating four large prime numbers for two times. This method is also not able handle large size data since ASCII value conversion become complex.

An enhanced and secured RSA key generation scheme (ESRKGS) was focused on increasing security level of RSA algorithm [24]. In this scheme, four large prime numbers were multiplied to produce N value. Then based on N value public and private keys were generated by RSA algorithm. Here public key generation was realized by using multiple computations on N value while private key generation was realized by using multiple computations on N value as well as public key. Involvement of multiple mathematical computations on both public key and private key generation increases overhead in the system.

III. Problem Definition

A digital finger printing technique was involved with RSA and message digest5 (MD5) algorithm [25]. Here, registered users files were encrypted by RSA algorithm and corresponding message digest was generated by MD5 algorithm. In this technique, attacker is able to crack MD5 algorithm, which is relatively insecure. Once encrypted file is obtained, it is easy to find prime numbers used in RSA algorithm by performing factorization techniques. Hence, security level of file is lower in this method.Authentication process is also not effective due to insufficient metrics such as user name and password.

In RSA, a chaos system based random number generation (RNG) model [26] carried out prime number generation. In RNG, binary series were obtained then that series was converted into decimal number in order to obtain prime numbers. In this method, the major shortcoming is that obtained binary series is not always prime number. If it is not prime number then this process is repeated until prime number is obtained. Thus, this method increases complexity and time consumption in key generation process.

Dual modulus RSA based on Jordan-Totient (DMRJT) algorithm was concentrated on security level improvement [27]. Dual encryption and dual modulus process with four large prime numbers was performed to ensure high-level security. Encryption process in DMRJT was performed as follows,

$$E(m) = ((m^{e1} mod\ n_1)^{e2} mod\ n_2) \quad (1)$$

$$\text{Where, } n_1 = p_1 \times p_2 \quad (2)$$

$$n_2 = p_3 \times p_4 \quad (3)$$

Here p_1, p_2, p_3, and p_4 are prime numbers generated. From above equation, it is clear that generation of four large prime numbers and dual encryption with dual modulus introduces high overhead and high time consumption in the RSA algorithm.

Thus in most of the previous research works, security level in RSA is strengthened in the cost of high time consumption and overhead. It is necessary to improve RSA algorithm in the perspective of security as well as time consumption to ensure high-level data security in cloud environment.

IV. Proposed ERSA algorithm

To improve data security in cloud environment, a novel ERSA algorithm is presented in this work. Data security is ensured by following three phases as follows: (i) authentication phase, (ii) evaluation phase, and (iii) encryption phase. Authentication phase allows only authorized users to access the data in cloud environment. Here ERSA based digital signature is utilized to authenticate users in cloud environment. In evaluation phase, the security level of ERSA algorithm is evaluated by FIS in server. If the required security level is obtained then the data is encrypted in encryption phase using ERSA algorithm. To attain this objective, our proposed cloud environment is comprised with k users as $U = \{U_1, U_2, \dots, U_k\}$, cloud server (CS), and trusted authority (TA). Here TA is responsible to generate private and public keys accordance to security level using ERSA algorithm. CS is responsible to evaluate the public and private keys generated by TA. Overall process of proposed ERSA based security system is illustrated in figure.1. Each significant phase is detailed in following sections. The notations used throughout the paper are listed as follows

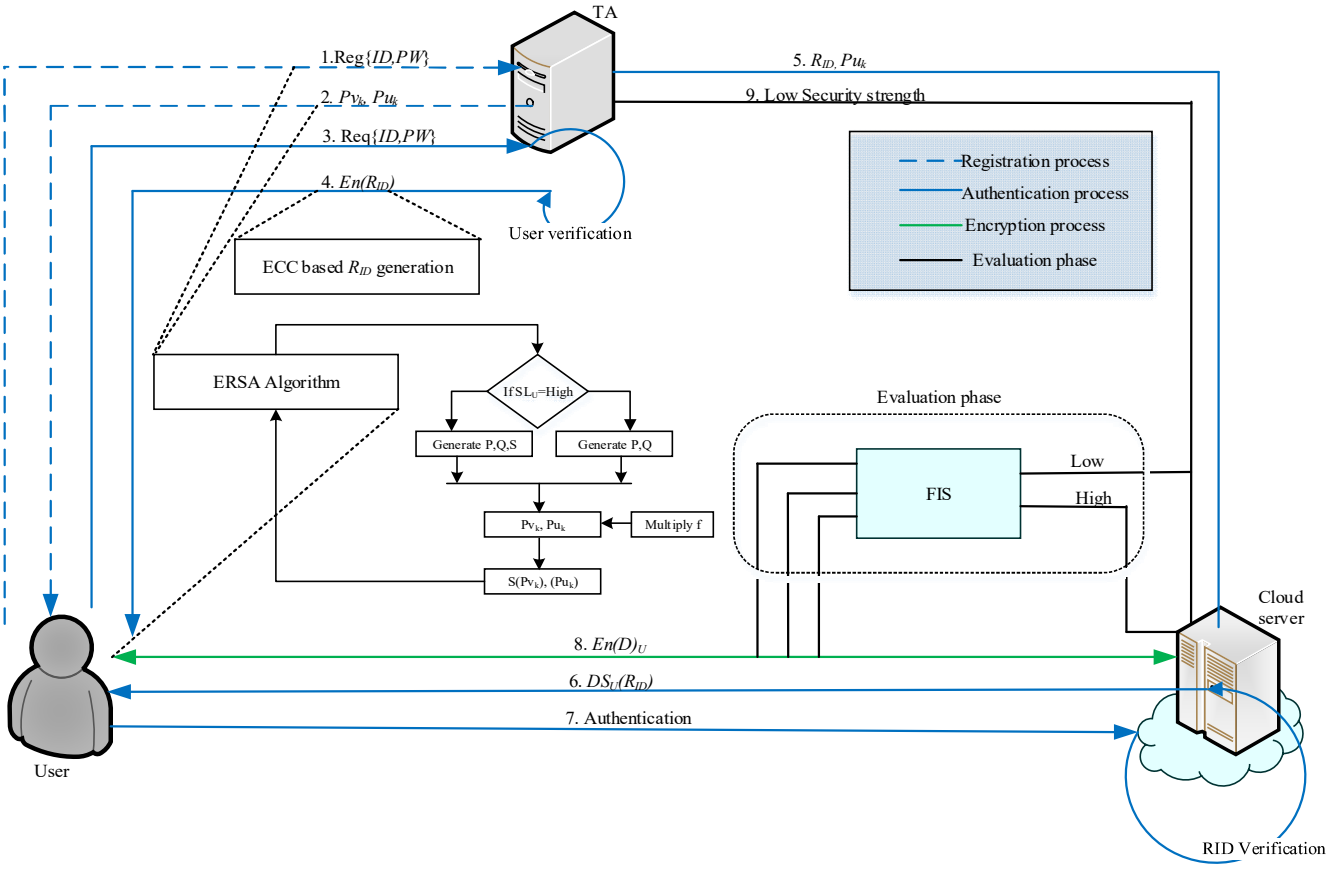

Figure.1 ERSA based secure cloud environment

Notation	*Description*
U	User
U_i	ith user
ID	User ID
PW	Password
R_{ID}	Random ID
$En(R_{ID})$	Encrypted random ID
P, Q, R	Prime numbers
N	Number of prime numbers
f	Non-prime factor
n	Modulus
e	Public exponent
d	Private exponent
D_U	User data
$En(D)_U$	Encrypted user data
Pv_k	User private key
$S(Pv_k)$	Secured private key
$S(Pu_k)$	Secured public key
Pu_k	User public key
$Size(k)$	Key size
SL_U	Security level requested by user
DS_U	Digital signature of user
$\{R_{ID}\|DS_U\}$	Signed Random ID
$H\{R_{ID}\}$	Hashed random ID

4.1 Proposed ERSA algorithm

ERSA algorithm is an improved version of traditional RSA algorithm in which security level is improved without increase in time consumption and complexity. Pseudo code of traditional RSA is given in algorithm.1.

Algorithm.1 Pseudo code for traditional RSA

1. Begin
2. Select P, Q
2. Compute, $n = P * Q$
3. Compute, $\varphi(n) = (P-1)(Q-1)$
4. Select, E as $\gcd(\varphi(n), e) = 1$
5. Compute, d as $d.e \equiv 1(mod(\varphi(n))$
6. Public key e, n}
7. Private key {d, n}

In ERSA, number of prime number generation entirely depends upon security level requested by user. IfSL_U is high, then three prime numbers are generated in ERSA algorithm. Otherwise, two prime numbers are generated in ERSA algorithm. Main security aspect in RSA algorithm is that an attacker can easily crack the private key from public key. Factorization techniques are often used by attackers to derive prime numbers from public key, which can be used to derive

private key. One reliable solution for this problem is to secure public and private key from attackers. Thus, ERSA attempts to secure secret keys from attackers using a non-prime factor. ERSA is involved with small computations and high effectiveness in security. In ERSA, key generation process follows traditional RSA algorithm. After key generation completed, the generated Pv_k andPu_k are then multiplied by f in order to secure Pv_k andPu_k. Selection of f considers following assumption,

- Should be non-prime number
- Lesser than generated keys

In algorithm.1, key generation process in proposed ERSA algorithm is illustrated. Here, Pu_kPv_k are generated as intermediate keys and secured by non-prime factor f. Finally, from algorithm.2 secured private and secured public key are obtained. These secured keys are used for both authentication as well as encryption. Here, SoA algorithm, which is fast and efficient algorithm, is incorporated for prime numbers generation. SoA algorithm generates all prime numbers below an integer, which is given as input. If the integer value is 1000, then all prime numbers below 1000 are generated by SoA.

Algorithm.2 Key generation in ERSA algorithm

Input: SL_U, P, Q, R, f
Output: $S(Pu_k), S(Pv_k)$

Begin
For all $U_i \epsilon U$
If$SL_U == High$
  Generate P, Q, S by SoA
  Compute n
$$n = P * Q * S$$
  Compute $\varphi(n)$
$$\varphi(n) = (P-1) * (Q-1) * (S-1)$$
Else
  Generate P, Q by SoA
  Compute n
$$n = P * Q$$
  Compute $\varphi(n)$
$$\varphi(n) = (P-1) * (Q-1)$$
End if
Choose e as
Satisfying $e \rightarrow 1 < e < \varphi(n)$
$$\gcd(e, \varphi(n)) = 1$$
Choose d as
$$d \leftarrow e^{-1} mod(\varphi(n))$$
$$Pu_k = (e, n)$$
$$Pv_k = (d, n)$$
Select f
$S(Pu_k) = (Pu_k) * f // n \rightarrow n * f,$
$$S(Pv_k) = (Pv_k) * f$$
End for
End

However, smaller prime numbers are not suitable for our work since high security level is attained by using large prime number. To tackle this problem, we modified the traditional SoA algorithm by setting up a range in which the prime numbers to be generated. In modified SoA algorithm, prime numbers are presented in specific range are detected by SoA algorithm. From this set of prime numbers, required number of prime numbers is chosen by ERSA algorithm in random manner. Here, key generation process is involved with simple computations in order to prevent high complexity. Perhaps, private and public key are vulnerable to attacks, secured private key and secured public key are highly secured. In following subsections, ERSA based authentication and encryption process are detailed.

4.2Authentication phase

In cloud environment, unauthorized user access is major threat. To resolve unauthorized user access problem, an efficient authentication scheme that utilizes digital signature of user is proposed. Here digital signature is generated by novel ERSA algorithm. Authentication process is involved with two steps as follows: (i) user registration, and (ii) user authentication.

User Registration

In authentication phase, user registration is initial process. In proposed cloud environment, all legitimate users must register with TA. User registration process considers user ID and password for registration. After registration, TA generates private and public key for user using ERSA algorithm. In ERSA algorithm, security level required by user also play vital role. Based on user's required security level, private and public key are generated by ERSA algorithm as depicted in algorithm.1. For all registered legitimate users, secured public and privates keys are generated by TA in this step. In authentication process, secured keys are utilized for digital signature generation while in encryption phase secured keys are utilized to secure data. Steps involved in authentication phase is illustrated as follows,

Steps involved in user authentication phase

Begin
User $Reg\{ID, PW, SL_U\} \rightarrow TA$
If$(SL_U == High)$
  Generate $S(Pu_k), S(Pv_k)$ using P, Q, S
Else
  Generate $S(Pu_k), S(Pv_k)$ using P, Q
End if
TA $S(Pu_k), S(Pv_k) \rightarrow U$
U $Req\{ID, PW\}$
TA verify (Req_U)
If$(ID \&\& PW = Matched)$
  TA$En(R_{ID}) \rightarrow U$
TA $\{R_{ID}, S(Pu)_k\} \rightarrow CS$
U $Req\{R_{ID} | DS_U\} \rightarrow CS$

```
        CS verify R_ID
        If(R_ID matched)
            Allow U
        Else
            Access denied
        End if
    Else
        Access denied
    End if
    End
```

User Authentication

Authentication process is initialized by users. When, a user wants to access the data the user must be authenticated before enter into the cloud environment. Here, user must submit $\{ID, PW\}$ to TA. Then TA verifies the user ID with password, which are already stored in TA. If verification of user is successful, then the TA generates R_{ID} for that user. The R_{ID} generation is involved with ECC algorithm in which points are detected on curve equation. Here ECC P-256 curve is used to generate the prime numbers for user authentication. ECC is not used for key generation, but it is sued for random ID generation. In ECC, curve equation is formulated as follows,

$$y^2 = x^3 + ax + b \qquad (4)$$

Random points (G, H) that satisfies above curve equation is generated as R_{ID} as follows,

$$R_{ID} = \{G, H\} \qquad (5)$$

Since ECC algorithm is relatively fast, R_{ID} generation process also minimizes time consumption. Again, from a curve equation, it is possible to generate multiple R_{ID}, which supports multiple users at same time. After generating R_{ID} for user, TA encrypts the R_{ID} using user's $S(Pu_k)$ as follows,

$$En(R_{ID}) = \{R_{ID} | S(Pu_k)\} \qquad (6)$$

Then, $En(R_{ID})$ is given to the users and original R_{ID} of particular user is given to CS. TA provides R_{ID} as well as $S(Pu_k)$ to CS for providing user authentication. This R_{ID} is only valid for particular time to ensure the freshness of authentication. For each user, R_{ID} is generated at each time access. Thus the user is not able to access the data using previous R_{ID}. Here it is worth to mentioning that, CS does not require user ID, PW and so on to preserve user privacy. In this work, TA is highly trusted and secure enough to maintain all user detail.

Upon receiving $En(R_{ID})$, the user decrypts the encrypted ID to obtain original R_{ID}. Here, only legitimate user has the corresponding $S(Pv_k)$ and able to obtain the original ID. After recovering R_{ID} from $En(R_{ID})$, the user generates DS_U and submit the R_{ID} with DS_U to CS as follows,

$$U\{R_{ID} | DS_U\} \rightarrow CS \qquad (7)$$

Signed R_{ID} is verified by CS in order to authenticate user. Here involvement of digital signature ensures high-level security in authentication since the secured ERSA keys are only known by user itself.

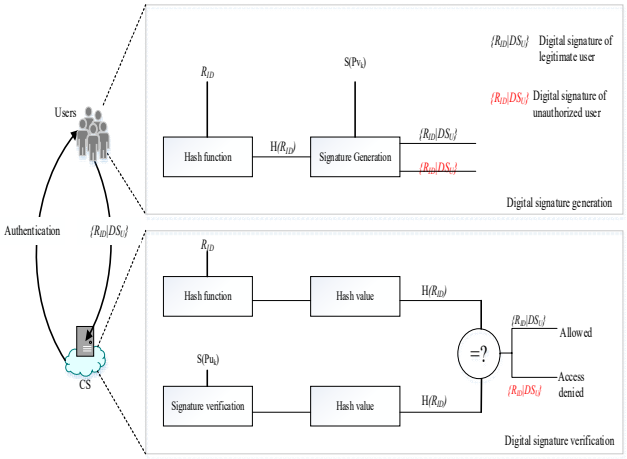

Figure.2 Digital signature based authentication

In figure.2, process of digital signature based authentication is illustrated. If signature is verified and random ID is matched by CS, then the user is authenticated. Otherwise, the user is not allowed to access the data.

4.3 Evaluation phase

Evaluation phase proposed in this work measures the strength of proposed ERSA algorithm by using FIS. Here, following metrics are considered to evaluate ERSA algorithm: generated key size, number of prime numbers, and requested security level. Based on three parameters security level provided by ERSA algorithm is evaluated in FIS. If security level provided by ERSA algorithm is low then CS alert TA to generate another secure key for particular user. Otherwise, the generated key is used for encryption and decryption. Generally, fuzzy logic is performed by applying fuzzy rules deployed in rule base. The rule base consist of 'M' rules and l^{th} rule is given as follows,

$$R^l: IF\, x_1 is \widetilde{F_1^l} and \ldots and x_p is \widetilde{F_2^l}\ THEN\ y\ is\ \widetilde{G^l} \qquad (8)$$

Here, $x = (x_1, x_2, .., x_p)$ represents input membership functions, and $\widetilde{F^l} = (\widetilde{F_1^l}, \widetilde{F_2^l}, \ldots, \widetilde{F_p^l})$ represents fuzzy set corresponding to x and $\widetilde{G^l}$ represents crisp output.

Table.1 Rules deployed in FIS

Key Size	Number of prime numbers	Requested security level	Output
Small	Two	Low	High
Small	Two	High	Low

Small	Three	Low	High
Small	Three	High	Medium
Large	Two	Low	High
Large	Two	High	Medium
Large	Three	Low	High
Large	Three	High	High

In table.2, rules deployed in FIS rule base are illustrated. Based on these rules, ERSA algorithm is evaluated and output is generated as low, medium, and high.

Steps involved in evaluation phase

Begin
For $\forall U_i \in U$
  Find $N, Size(k), SL_U$
  If(SL_U=High)
    If($N = Two \&\& Size(k) = Small$)
      $OP = Low$
    If($N = Three \&\& Size(k) = Small$)
      $OP = Medium$
    If($N = Two \&\& Size(k) = Large$)
      $OP = Medium$
    If$N = Three \&\& Size(k) = Large$)
      $OP = High$
    Else
    Else
    Else
  Else
    $OP = High$
  End If
End for
End

If user's requested security level is high, then ERSA algorithm must generates large size key and must generates three prime numbers. If key size and number of generated prime numbers are small then the security provided by ERSA algorithm is relatively weak. In this case, CS alerts TA to generate new keys for users, which is strong enough in evaluation phase. Following steps are involved in FIS based evaluation phase performed by CS, Involvement of evaluation phase enables the high-level security in cloud environment. After evaluation phase, user encrypts their data to store the file in cloud server.

4.4 Encryption phase

In this phase, legitimate users store their data in the form of encrypted data. Encryption of data ensures the data security in cloud environment. In this work, security is improved in twice by providing efficient authentication and to by providing efficient encryption through ERSA algorithm. Encryption process is involved with $S(Pu_k)$ while decryption process is involved with $S(Pv_k)$. In ERSA encryption is performed as follows,

$$En(D) = D^e mod\ n \quad (9)$$

Where, $S(Pu_k) = (e, n)$ is public key of the user. Here encryption process provides high security level for data stored in CS since ERSA algorithm is unbreakable by attackers. Secured key is generated by multiplying public and private keys with a non-prime factor in ERSA algorithm. Encryption is performed before outsourcing data to CS and while retrieving data original data is acquired by decryption process. In ERSA algorithm, decryption process is performed as follows,

$$De(D) = (En(D))^d mod\ n \quad (10)$$

Here, $S(Pr_k) = (d, n)$ is private key of user. Thus, decryption process is involved with user's secured private key.

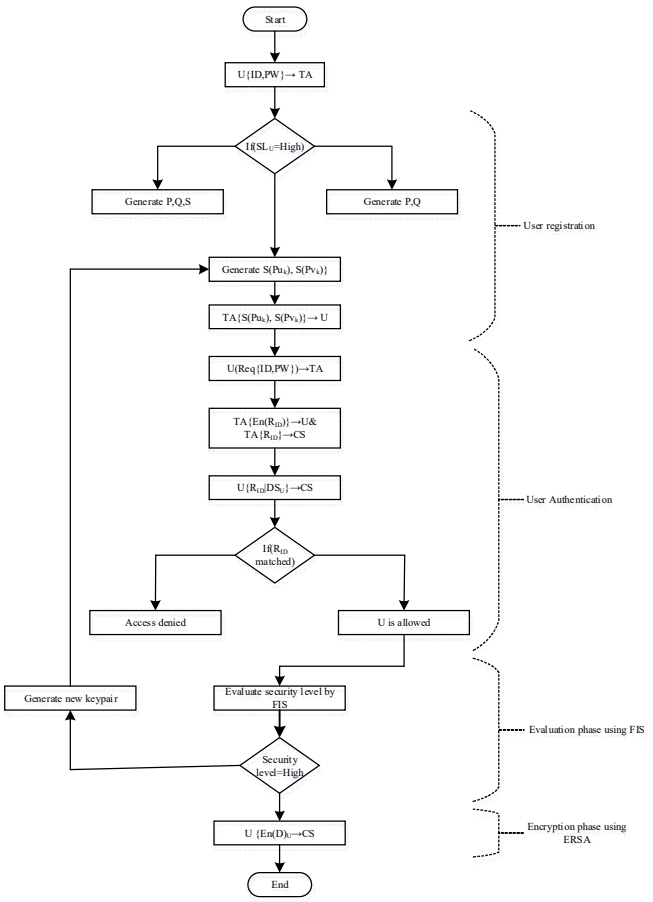

Figure.3 Process in ERSA algorithm

In figure.3, overall process involved in proposed ERSA based security scheme is depicted.

Example:Encryption and decryption using ERSA
Prime numbers: P=3, Q=11 Non-prime factor: f=10

Data: D_U=5
Then, n=3*11=33
$\varphi(n)$=(3-1)*(11-1)=20
e=3, d=7
Pu_k=(3,33); Pv_k=(7,33)
$S(Pu_k)$=(3,(10*33))=(30,330)
$S(Pv_k)$=(7,(10*33))=(70,330)
En(D_U)=$(5)^3 mod\ 330$=125
De(D)=$(125)^7 mod\ 330$=5

In above example, user data is 5 and it is stored as 125 in CS. When, decrypting 125, the user is able to obtain 5 which is original message. Thus involvement of three efficient phases such as authentication phase, evaluation phase, and encryption phase in ERSA based cloud environment ensures the high-level data security in cloud storage.

V. Performance Evaluation

In this section, proposed ERSA algorithm is evaluated in terms of performance metrics. This section is comprised with two subsections such as simulation setup, and comparative analysis.

5.1 Simulation Setup

Our proposed ERSA algorithm based security scheme in cloud environment is implemented in JAVA environment with single TA, single CS, and 'n' number of users. So that JAVA including Java run time environment as well as Java tools and JDK-1.8 kit are installed on PC. CS is designed by Wamp server by utilizing msql-5.3.16. Development of java is supported by NetBeans-8.0 Integrated Development Environment (IDE). Overall process is supported in Widows-7 Ultimate operating system.

Table.2 Simulation parameters

Parameter		Value
Number of users		100 and above
Number of key pairs	High security level	50 and above
	Low security level	50 and above
Key size	Private key	1024 bits, 2048 bits
	Public key	1024 bits, 2048 bits
Prime number range		50 to 100000
Number of generated non-prime factors		100
Algorithm used for hash generation in digital signature		SHA 1
File types supported		.Docx, .PDF, JPEG, PNG, .txt,

Table.2 illustrates significant specifications of ERSA algorithm considered in simulation.

Table.3 Obtained result in key generation

File name	File type	Size (MB)	Key
File 1	JPEG	0.59	53820
File 2	PDF	0.38	1153
File 3	PDF	0.67	1109

In table.3, uploaded file details in CS of three users is depicted with file size. Regardless file size, key generation process considers requested security level for key generation. Thus, generated key for file1 is larger than other keys.

5.2 Comparative analysis

In this subsection, we compare our proposed ERSA algorithm with existing RSA-KNN algorithm [23]. Comparisons are made in terms of performance metrics such as key generation time, encryption time, decryption time, and security level.

Table.4 Comparison between RSA, RSA-KNN, ERSA

Parameter	RSA	RSA-KNN	ERSA
Number of prime numbers	Two	Four	Two or Three
Randomness in ciphertext	Less	Medium	More
Complexity	Less	More	Medium
Time consumption	Medium	High	Low
Security	Low	Medium	High
Non-prime factor	Nil	Nil	One

In table.4, detailed comparative analysis among traditional RSA, RSA-KNN, and ERSA algorithms is provided. Hereby, ERSA algorithm is secure and minimizes time consumption.

Effectiveness of time consumption

In an efficient cryptography technique, time consumed for key generation, encryption, and decryption should be low as possible. These three time constraints have main concern on security as well as performance.

Key generation time: Key generation is the process of generating keys (both public key and private key) in cryptography.Key generation time is defined as time taken by a cryptography algorithm to perform key generation process. It includes time taken for generating both public and private keys.

In figure.4, key generation time in ERSA algorithm is compared with existing RSA-KNN algorithm. This comparative analysis shows that proposed ERSA algorithm consumes small amount of time for key generation compared with RSA-KNN algorithm.In ERSA algorithm, number of prime numbers to be generated is depends upon requested security level which resolves the problem of generating large number of prime numbers for lower requested security level. In addition, involvement of SoA algorithm in prime number

generation minimizes prime number generation time. In contrast, RSA-KNN algorithm generates four large prime numbers for each user's key, which results in higher key generation time

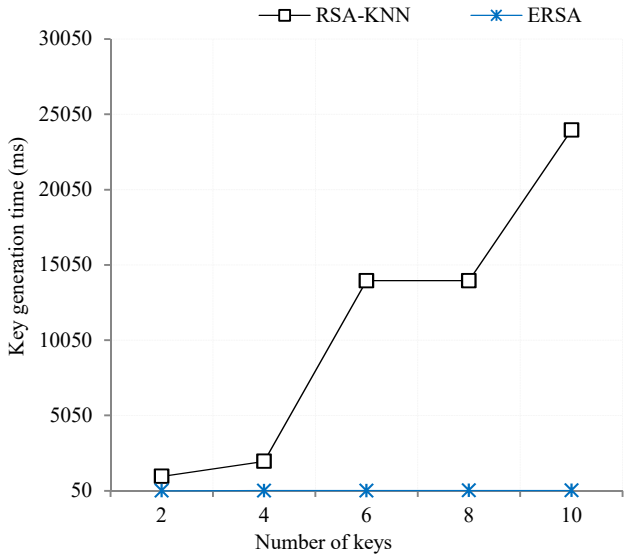

Figure.4 Analysis on key generation time

Thus, ERSA algorithm minimizes key generation time compared to RSA-KNN scheme.

Encryption time: Encryption is a process of converting plaintext into unreadable ciphertext. Encryption time is defined as time consumed by a cryptography algorithm to convert plain text into cipher text.

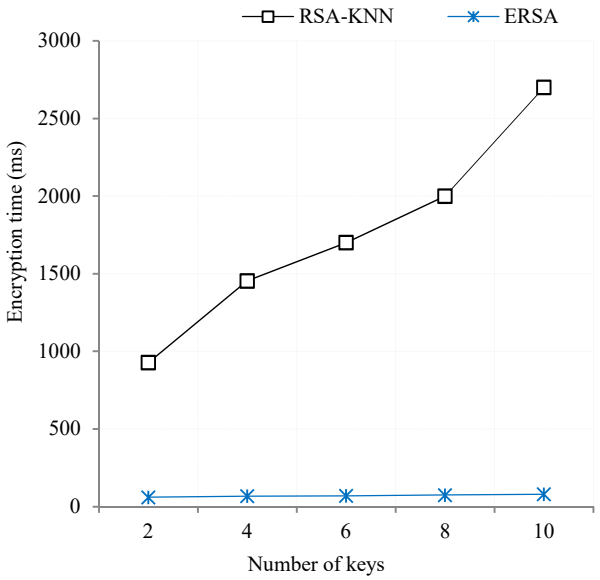

Figure.5 Analysis on encryption time

Comparative analysis on encryption time is illustrated in figure.4. ERSA algorithm relies on minimized encryption time since only small and simple computations are made on traditional algorithm. In RSA-KNN algorithm, encryption process is involved with ASCII conversion and then encryption. However, when large size is to be encrypted using RSA-KNN algorithm it have to perform ASCII conversion and encryption. This process increases the encryption time in RSA-KNN algorithm significantly. For instance, in the presence of two users encryption time in RSA-KNN algorithm is 927ms while in ERSA algorithm is only 61ms. This huge difference in encryption time is realized due to involvement ASCII conversion process in RSA-KNN algorithm.

Decryption time: Decryption is a process of converting encrypted data (i.e.) ciphertext into original format. Decryption time is referred to time taken for recovering original data from encrypted data in cryptography technique.

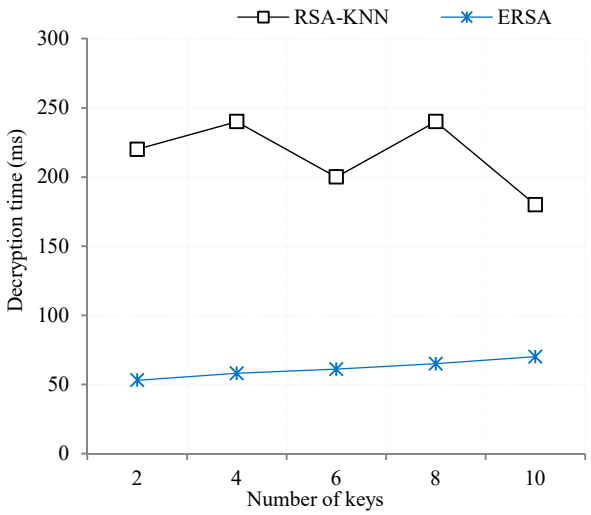

Figure.6 Analysis on decryption time

In figure.6, we compare decryption time in proposed ERSA algorithm and existing RSA-KNN algorithm. The graphical comparative analysis shows that ERSA algorithm reduces time required for decryption. In RSA-KNN algorithm decryption time is large since creates large size key even for small requested security level. In addition, the original data is recovered after ASCII code conversion. These limitations increase decryption time in RSA-KNN algorithm rapidly. For two users, ERSA algorithm provides 220ms while ERSA algorithm provides 53ms for the same number of users.

Table.5 Comparison between ERSA and RSA-KNN

Algorithm	Key generation time (ms)	Encryption time (ms)	Decryption time (ms)
RSA-KNN	11000	10811	216

ERSA	60.6	70.8	61.4

Overall, proposed ERSA algorithm provides better performance in major time constraints such as key generation time, encryption time, and decryption time. In table.5, comparison between ERSA algorithm and RSA-KNN algorithm is depicted.

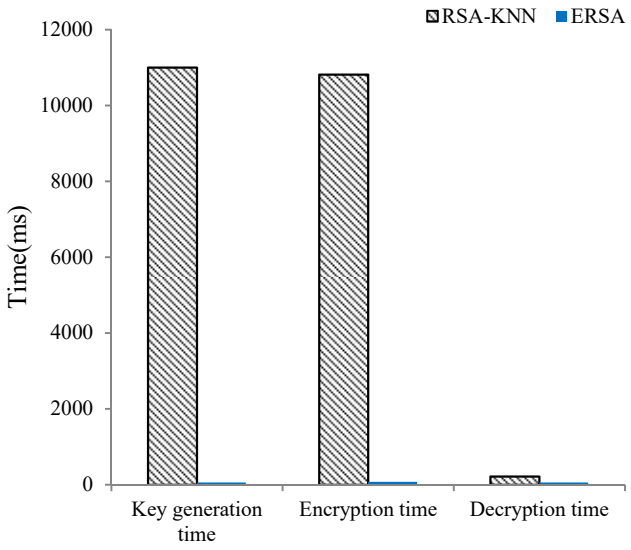

Figure.7 Effectiveness on time consumption

In figure.7, overall analysis on time consumption is depicted. Here it is obvious that proposed ERSA algorithm minimizes time consumption in each process such as key generation time, encryption time, and decryption time. Involvement of SoA algorithm in key generation process and simple computations in encryption and decryption process supports minimized time consumption in ERSA algorithm.

Effectiveness of security level

Perhaps, ERSA algorithm minimizes time consumption it is also necessary to evaluate the security level provided by ERSA algorithm. An efficient cryptography algorithm must attain lower time consumption without loss in security level.

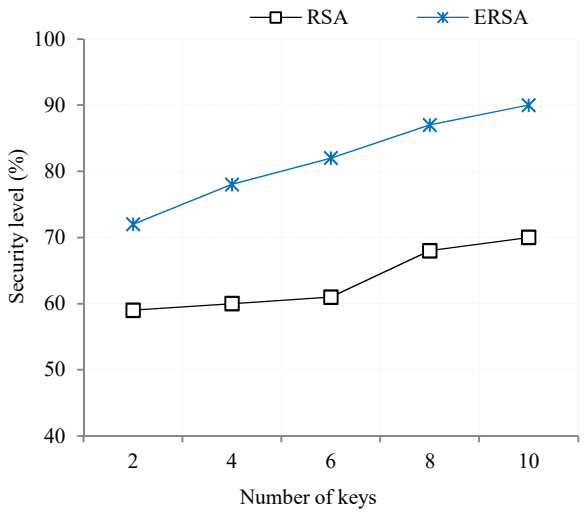

Figure.8 Analysis on security level

In figure.8, comparative analysis on security level in proposed ERSA algorithm and traditional RSA algorithm is depicted. The analysis shows that proposed ERSA algorithm achieves better security level compared to RSA algorithm. When number of users increases security level also gradually increased. For two users, security level provided by RSA-KNN algorithm is 59% and for same users security level provided by ERSA algorithm is 72%, which is 13% higher than existing algorithm.

In proposed ERSA algorithm, security is ensured with the help of efficient authentication process, evaluation process, and encryption process. Authentication process allows only legitimate users to access the system while evaluation process improves the security by evaluating proposed algorithm. Encryption process enables high security level for data in cloud environment. Average security level achieved by RSA algorithm is 64.3% whereas proposed ERSA algorithm achieves 81.8% of average security level.

Security analysis:

We analyze the security strength of our proposed ERSA algorithm in terms of time taken to launch Brute-Force attack. It is well-known attack that defines the time taken by attacker to crack the secret key of cryptography technique.

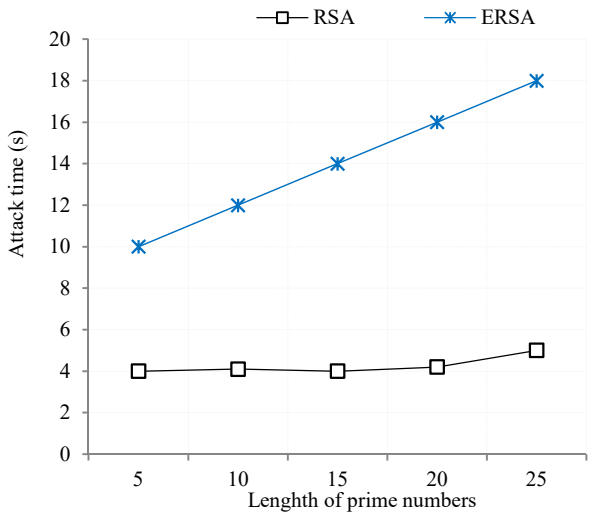

Figure.9 Brute-force attack comparsion

In figure.9, we compare time taken by attacker to crack the key of traditional RSA algorithm and proposed ERSA algorithm. Here we can see that our proposed ERSA algorithm requires more than 15s to crack the secret key since cracking the non-prime factor in our work is not possible for atatckers. However, an attacker can cracy RSA key within 5s which is 50% lesser than ERSA algorithm. Thus we can conclude that our proposed ERSA algorithm secure the data in cloud environment.

Thus, our proposed ERSA algorithm achieves better performance in security level without increase in time consumption. In cloud environment, proposed ERSA algorithm is able to secure data in terms of data security and integrity.

VI. Conclusion

In this paper, a novel ERSA algorithm is proposed to enhance security level for data in cloud environment. In ERSA algorithm, traditional RSA algorithm is improved by considering security level and non-prime factor. Proposed ERSA based security system is involved with three efficient phases such as authentication phase, evaluation phase, and encryption phase. ERSA based digital signature is utilized for user authentication in authentication phase. Hereby, random ID generation is realized by ECC algorithm, which is significantly fast. Overall authentication phase is performed between user and CS with the support of TA. In evaluation phase, FIS is employed at CS to measure the strength of ERSA algorithm. Finally, encryption phase allows user to store their data in secure manner by utilizing ERSA based encryption process in CS. Simulation results show that ERSA algorithm is effective in both security and time consumption. In future, we have planned to analyze ERSA algorithm against specific attacks such as brute-force attack, timing attack, and so on.

REFERENCES

[1] Matthew N.O. Sadiku,Sarhan M. Musa, andOmonowo D. Momoh, “Cloud computing:Opportunitiesand challenges”, IEEE Potentials, Vol. 33, Issue. , pp. 34-36, 2014.

[2] Manuel Diaz, Cristian Martin, and Bartolome Rubio, “State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing”, Journal of Network and Computer Application, Elsevier, Vol. 67, pp. 9-117, 2016.

[3] Issa M. Khalil, Abdallah Khreishah and Muhammad Azeem, “Cloud Computing Security: A Survey”, Computers, Vol. 3, pp. 1-35, 2014.

[4] Faheem Zafar, Abid Khan, Saif Ur Rehman Malik, Mansoor Ahmed,Adeel Anjum, Majid Iqbal Khan, Nadeem Javed, Masoom Alam, and Fuzel Jamil, “A survey of cloud computing data integrityschemes: Design challenges, taxonomy andfuture trends”, Computers and Security, Elsevier, Vol.5, pp. 29-49, 2017.

[5] SaurabhSingh, Young-Sik Jeong, and JongHyukPark, “A survey on cloud computing security: Issues, threats, and solutions”, Journal of Network and Computer Application, Elsevier, Vol. 75, pp. 200-222, 2016.

[6] Mansoor Ebrahim, Shujaat Khan, and Umer Bin Khalid, “Symmetric Algorithm Survey: A Comparative Analysis”, International Journal of Computer Applications, Vol. 61, Issue. 20, pp. 12-19, 2013.

[7] Roshan M. Pandey, and Vijay Kumar Verma, “Data Security Using VariousCryptography Techniques: A RecentSurvey”, International Journal for Research in Engineering Application & Management, Vol. 1, Issue. 9, pp. 1-4, 2015.

[8] Sourabh Chandra, Smita Paira, Sk Safikul Alam, and Goutam Sanyal, “A comparative survey of symmetric and asymmetric keycryptography”, International Conference on Electronics, Communication and Computational Engineering, 2014.

[9] Sarika Khatarkar, and Rachana Kamble, “A Survey and Performance Analysis of Various RSA based Encryption Techniques”, International Journal of Computer Applications, Vol. 114, Issue. 7, pp. 30-33, 2015.

[10] Rana M Pir, “Security improvement and Speed Monitoring of RSA Algorithm”, International Journal of Engineering Development and Research, Vol. 4, Issue. 1, pp. 195-200, 2016.

[11] M. Preetha, and M. Nithya, “A Study and PerformanceAnalysis of RSA Algorithm”, International Journal of Computer Science and Mobile Computing Vol.2 Issue. 6, pp. 126-139, 2013.

[12] Vikash Mainanwal, Mansi Gupta, and Shravan Kumar Upadhayay, “Zero knowledge protocol with RSA Cryptography Algorithm forAuthentication in Web Browser Login System (Z-RSA)”, Fifth International Conference on Communication Systems and Network Technologies, 2015.

[13] Viney Pal Bansal, and Sandeep Singh, “A Hybrid Data Encryption Technique using RSAand Blowfish for Cloud

Computing on FPGAs", IEEE International conference on Recent Advances in Engineering and Computational Sciences, India, 2016.
[14] Christina-Angeliki Toli, Abdelrahaman Aly, and Bart Preneel, "Privacy-Preserving Multibiometric Authentication in Cloud with Untrusted Database Providers", IACR Cryptology ePrint, 2018.
[15] Hong Liu,Huansheng Ning,Qingxu Xiong,and Laurence T. Yang, "Shared Authority Based Privacy-preserving Authentication Protocol in Cloud Computing", IEEE Transactions on Parallel and Distributed Systems,Vol. 26,Issue. 1, pp. 241-251, 2015.
[16] Sushmita Ruj,Milos Stojmenovic,andAmiya Nayak, "Decentralized Access Control with AnonymousAuthentication of Data Stored in Clouds", IEEE Transactions on Parallel and Distributed Systems, Vol. 25, Issue. 2, pp. 384-394, 2014.
[17] Tian Wang, Jiyuan Zhou, Xinlei Chen , Guojun Wang , Anfeng Liu , and Yang Liu, "A Three-Layer Privacy Preserving Cloud StorageScheme Based on Computational Intelligencein Fog Computing",IEEE Transactions On Emerging Topics In Computational Intelligence, Vol. 2, Issue. 1, pp. 3-12, 2018.
[18] Rizwana Shaikh, and M. Sasikumar, "Data Classification for achieving Security in cloud computing", Procedia computer Science, Elsevier, Vol. 45, pp. 493-498, 2015.
[19] Lifei Wei, Haojin Zhu, Zhenfu Cao,Xiaolei Dong, Weiwei Jia, Yunlu Chen, and Athanasios V. Vasilakos, "Security and privacy for storage and computation in cloud computing", Information Sciences, Elsevier, Vol. 258, pp. 371-386, 2014.
[20] Gunasekaran Manogaran, Chandu Thota, and M. Vijay Kumar, "MetaCloudDataStorage Architecture for Big Data Security inCloud Computing", International Conference on Recent Trends in Computer Science & Engineering, Elsevier, Vol. 87, pp. 128-133, 2016.
[21] Khalid El Makkaoui, Abderrahim Beni-Hssane, Abdellah Ezzati, and Anas El-Ansari , "Fast Cloud-RSA Scheme for Promoting Data Confidentiality in the Cloud Computing", Procedia Computer Science, Elsevier, Vol. 113, pp. 33-40, 2017.
[22] JavierHerranz, "Attribute-basedsignaturesfromRSA", TheoreticalComputerScience, Elsevier, Vol. 527, pp. 73-82, 2014.
[23] Shikha Mathur, Deepika Gupta, Vishal Goar and Sunita Choudhary, "Implementation of Modified RSA Approach for Encrypting and Decrypting Text Using Multi-power and K-Nearest Neighbor Algorithm", Networking Communication and Data Knowledge Engineering, Springer, pp. 229-237, 2017.
[24] M. Thangavel, P. Varalakshmi, Mukund Murrali, and K. Nithya, "An Enhanced and Secured RSA Key Generation Scheme (ESRKGS)", Journal of Information Security and Applications, Vol. 20, pp. 3-10, 2015.
[25] Nithya Chidambaram, Pethuru Raj, K. Thenmozhi,and Rengarajan Amirtharajan, "Enhancing the Security of Customer Data in CloudEnvironments Using a Novel Digital Fingerprinting Technique", International Journal of Digital Multimedia Broadcasting, 2016.
[26] Unal Çavusoglu, Akif Akgül, Ahmet Zengin, and Ihsan Pehlivan, "The design and implementation of hybrid RSA algorithm using a novel chaos based RNG", Chaos, Solitons and Fractals Nonlinear Science, and Non-equilibrium and Complex Phenomena, Elsevier, Vol. 104, pp. 655-667, 2017.
[27] Balram Swamia, Ravindar Singh, and Sanjay Choudhary, "Dual Modulus RSA based on Jordan-Totient function, Procedia Technology, Elsevier, Vol. 24, pp. 1581-1586, 2016.
[28] Ritu Tripathi, and Sanjay Agrawal, "Comparative Study of Symmetric and AsymmetricCryptography Techniques", International Journal of Advance Foundation and Research in Computer, Vol. 1, Issue. 6, pp. 68-76, 2014.