Research Article

International Journal of Advanced Computer Research, Vol 8(37)
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
http://dx.doi.org/10.19101/IJACR.2018.837010

A novel multi-user fingerprint minutiae based encryption and integrity verification for cloud data

Ruth Ramya Kalangi[1*] and M. V. P. Chandra Sekhara Rao[2]
Research Scholar, Department of Computer Science and Engineering, Acharya Nagarjuna University, Nagarjuna Nagar, Guntur, Andhra Pradesh, India[1]
Professor, Department of Computer Science and Engineering, RVR & JC College of Engineering, Chowdavaram, Guntur, Andhra Pradesh, India[2]

Received: 05-May-2018; Revised: 09-July-2018; Accepted: 11-July-2018

Abstract
Data confidentiality and integrity are two major aspects that the cloud users need to consider while deploying data in the cloud. Traditional integrity techniques use cryptographic hash algorithms, but most of these hash algorithms are vulnerable to third party attacks. Traditional encryption algorithms such as advanced encryption standard (AES), fully homomorphic attribute based encryption (FHABE) and key policy attribute based encryption (KP-ABE) are failed to generate biometric based attributes and policies due to limited computing resources and memory. So, novel multi-user fingerprint minutiae with ciphertext-policy attribute based encryption for integrity verification and encryption (MFM-CP-ABE) model is proposed. MFM-CP-ABE model considers fingerprints of multiple users as attributes for encryption and also calculates integrity value. This model is the combination of multi-user fingerprint minutiae (MFM) extraction policy integrity method and improved ciphertext policy attribute based encryption (ICP-ABE) algorithm. This model is efficient in comparison to the traditional models in terms of encryption and decryption time and data size.

Keywords
MFM-CP-ABE, Fingerprint biometry, Hash algorithm, Cloud computing.

1.Introduction
Biometrics are implemented over decades for data authentication. They are being used in medical databases, cloud databases, etc., for data security. As the size of the data increases, it becomes difficult to the traditional data security algorithms used in cloud to encrypt the data before uploading to the cloud environment due to security and memory issues. Hence, it is essential to improve the cloud security algorithms using the user's biometric identities such as fingerprint, iris, palm vein and face recognition against non-trusted authorities. Among all biometric approaches, fingerprints are considered to be the best authentication techniques for cloud data. Fingerprint patterns remain same for a person throughout his/her life. The other advantages associated with fingerprints include accuracy, uniqueness, economical and requires less storage space.

There exist many issues and challenges in the traditional single biometric-based cryptographic systems [1].

*Author for correspondence

If the cryptographic keys are compromised, then those keys can't be used in the process of encryption and decryption. In the proposed model, multiple biometric patterns are evaluated dynamically in the integrity verification phase and decryption phase to check the user's identity for data access [2].

Identity based encryption (IBE) and attribute based encryption (ABE) are the two different models that have been used for cloud data security with different attribute list and policies [3]. In IBE [4], a unique random string is used as user identification for data security. Most of the IBE based models use a private key generator (PKG) for unique identity key generation. Also, IBE model is used to verify the data integrity verification and encryption on limited data size and limited number of identities. A small change in the cloud data produces significant changes in the integrity values. The most frequently used traditional integrity verification algorithms on cloud data include message digest (MD) and secure hash algorithm (SHA). Most of the traditional encryption algorithms such as AES, DES, elliptic curve cryptography, KP-ABE etc., is used to encrypt the

cloud data using user attributes with plain text format as shown in *Figure 1*. The data is protected in such a way that, no unauthorized person can get access to secure data. Traditional encryption algorithms are failed due to statistical and dictionary type of attacks. User's sensitive data is transferred in different modes of communications to the remote cloud server. It is essential to protect the data prior to deployment in the cloud sever using attribute based encryption.

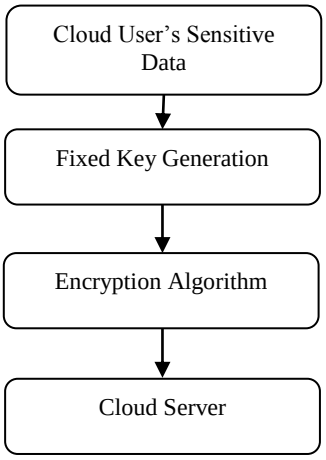

Figure 1 Traditional cloud data security using keys

Erkin et al. [5] proposed privacy preserving model using the face recognition procedure. This model computes eigenvalue of the faces for data encryption and decryption process. This model is applicable when eigenvalue of a single user are considered. The key problem with this model is that it is not optimal due to static initialization and randomization parameters.

Wazid et al. [6] implemented a novel user authentication and key exchange scheme using biometric computation and key agreement. This model is tested in a multi-server environment for user anonymity identification on cloud data. It also requires high computational resources such as memory and time as the size of the cloud data increases.

Atallah and Hopper [7] developed an advanced biometric authentication technique. This model is suitable for small dataset with static initialization parameters. In this model, a Chebyshev chaotic map in multi-server systems that include the advantages of light-weight multi-server authentication techniques are designed and implemented for data security. This model is also implemented using the random oracle model along with Burrows-Abadi-Needham (BAN) logic for security analysis.

Torres et al. [8] proposed a secured biometric authentication protocol using fully homomorphic encryption (FHE) technique. FHE approach is used to encrypt small data with static key initialization parameters. This model is applicable to real world applications such as small enterprise cloud services, e-voting, medical web services etc.

Tarif et al. [9] proposed a survey on biometric data security and chaotic encryption techniques on a small dataset. This model uses a traditional chaotic encryption technique along with Berboulli's mapping.

Huang et al. [10] proposed a new and advanced biometric-based encryption technique. It is an asymmetric authentication technique key initialization process. Reference tokens are generated to save the user identity in the biometric verification process. Saved tokens will not be completely matched along with any newly provided biometric token.

The main contributions of this paper are as follows:

i. Presented MFM-CP model for providing security to the cloud data. In this model, fingerprint patterns are considered from multiple users and minutiae points are extracted using a minutiae extraction algorithm for policy construction.
ii. Presented ICP-ABE algorithm for encrypting and decrypting data that is to be stored in the cloud. In this algorithm, a complex fingerprint pattern based polynomial function is used to provide more security for sensitive data against vulnerability attacks.

2.Proposed model

A MFM-CP-ABE model has been proposed and implemented on cloud data. The overall framework is shown in *Figure 2*. Initially, multiple user's fingerprints are taken as input for policy extraction. Fingerprint policy patterns are extracted using binarization, thinning and pattern extraction procedure. The extracted features are used for integrity value computation using an ICP-ABE encryption algorithm.

The overall framework is divided into 3 phases:

1. Multi-user fingerprint pattern extraction.
2. Biometric integrity MFM-CP-ABE algorithm.
3. Biometric integrity based improved ICP-ABE algorithm.

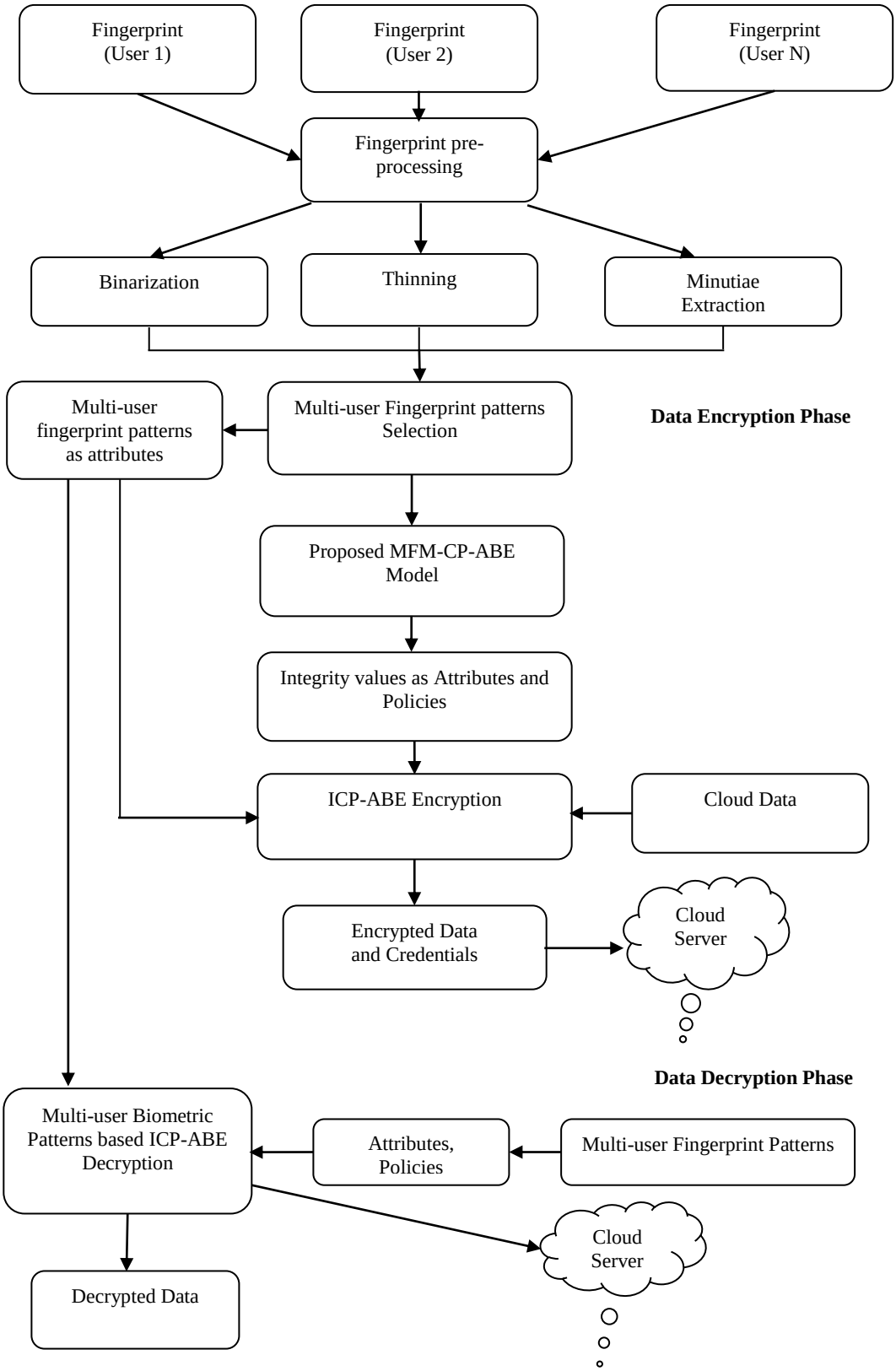

Figure 2 A novel MFM-CP-ABE model

2.1Multi-user fingerprint pattern extraction

In this phase, each fingerprint image from multiple users is pre-processed using three steps as binarization, thinning and minutiae extraction. In binarization steps, images is converted into a binary image in terms of 1 (white pixels) and 0 (dark pixels). Thinning is used to find the skeleton of an image by eliminating redundant and noisy pixels from the fingerprint image. Binarization, thinning and minutiae extraction processes are applied to gray scale fingerprint images. The proposed model optimizes the minutiae extraction process for pattern evaluation. Four different types of fingerprint patterns are extracted using equation 1.

$$M_p(i,j) = \sum_{i=1}^{g} (\left| P_i - P_{i+1} \right|) \quad (1)$$

Where
if $M_p(i, j) = 0$; Isolated Pattern Point
if $M_p(i, j) = 1$; Ending Pattern Point
if $M_p(i, j) = 2$; Connection Pattern Point
if $M_p(i, j) = 3$; Bifurcation Pattern Point

Equations 2-5 represent the four fingerprint patterns that are uniquely identified for data security based on the computational values of M.

Isolated Pattern Point (IPP) = { I(i,j).x I(i,j),y}; ∀ i, j ϵ M_p (i,j)=0 (2)
Ending Pattern Point (EPP) = { I(i,j).x I(i,j),y}; ∀i, j ϵ M_p (i,j)=1 (3)
Connective Pattern Point (CPP) = { I(i,j).x I(i,j),y}; ∀ i, j ϵ M_p (i,j)=2 (4)
Bifurcation Pattern Point (BPP) = { I(i,j).x I(i,j),y}; ∀ i, j ϵ M_p (i,j)=3 (5)

Basic 8 bit block of the fingerprint image is shown in *Figure 3* that illustrates an 8-bit block with central pixel P(c) and its neighbours.

P(i+3)	P(i+2)	P(i+1)
P(i+4)	P(c)	P(i)
P(i+5)	P(i+6)	P(i+7)

Figure 3 Block of fingerprint image after binarization

Algorithm 1 explains about the extraction of fingerprints from multiple users. For each user's input image, a series of image processing operations are performed such as grayscale conversion, binarization, thinning, minutiae extraction and finally fingerprint extraction procedure.

Algorithm1: Multi-user fingerprint pattern extraction
Input: Multiple fingerprint images as MI [], B [] binary images, T [] thinning CP [] Combined pattern.
Output: Integrity Computation.

1. Input multiple fingerprint images.
2. For each input image MI[i]
3. Do
    Perform grayscale to binary image conversion as binarization B[i]
    Perform thinning on binary image B[i] as T[i]
    Perform minutiae extraction Min[i] on thinning T[i] using Equation 1−5
    Fingerprint Patterns FP[i] = {Min[i]}
4. Add {MI[i], CP[i] =FP [0] ^ FP [1] ^ FP [2]..... FP[n]}; // CP: Combined Pattern
5. Done
6. Compute integrity computation for each fingerprint pattern {MI[i], CP[i]}.

2.2Biometric integrity MFM-CP-ABE algorithm

In this model, fingerprint biometric patterns and its integrity values are used as attributes and policies in the improved CP-ABE algorithm. Integrity value of the four fingerprint patterns of an image is used as policy for data encryption and decryption. The steps used in the biometric integrity MFM-CP-ABE algorithm for integrity value computation are shown below in algorithm 2. A novel hash function is proposed to find a unique integrity value for every fingerprint image pattern. In the proposed multi-user based integrity computation, an Eigenvalue based transformation is used to permute the message in each round.

Input:Initialization parameter block size (BS), number of rounds (NR), block bits, h [] round hash vector, data size (DS), input data (CP), x and K are permutation matrices.
Output: Biometric integrity value.

Algorithm 2: Biometric integrity algorithm

1. For each user's finger print pattern CP[i]
2. Do
   Initialization of input parameters and hash vector.
   BlockBits=CP[i];
   h[blockbits/8]← 0 // Initialize hash vector to zero.
3. While (DS>blockbits/8)
4. Do
   BlockD←D(0,blocklength)
   RestD←D(blocklength, DS-blocklength)
5. If(BlockD!=8==0) Then
   Print("Block mismatch");
6. End if
7. Else
   // Extract block data into sub blocks (PBlock) of 4 bytes each or 32 bits each
   P← First 4 bytes of sub block
   CP← Last 4 bytes of sub block
8. For each partition block PBlock
9. Do
   For(r=0 to NR)
   Do
   x1=(K^T.x)mod1
   E=compute Eigen value vector to x1
   y=E.Knonproduct(x1).scale(256)
   C_i= P[i]+c[max(0,i-1)]
   $C_i=C_i \oplus y_i$
   $C_{sum}=\sum C_i$
   R= Max(NR)–Min(NR)+1
   C_{sum}= Min(NR)+($C_{sum\%R}$)
10. Done
   // reverse ordering and shift 3 positions to the left of block P
   P[i]← CyclicReverse(P[i])
   P[i]← LeftShift(P[i],3)
11. If(r+1< NR) Then
   P[i]←RightShift(P[i],3)
   P[i]← Reverse(P[i])
12. End if
13. Done
   $H[i]=h_0 + h_1 +\ldots\ldots\ldots\ldots h_{NR}$
14. Done
   U[i]={CP[i],H[i]};
15. Done

2.3Biometric integrity based ICP-ABE algorithm

ICP-ABE performs the encryption with the help of public key and decryption would be successful if the decryptor attribute meets the access tree structure. This algorithm is used to encrypt cloud user's data before deploying into the remote cloud server as shown in algorithm 3. Bilinear pairing is used as a mathematical group operation for efficient data processing in this algorithm. This bilinear pairing depends on the cyclic group elements in prime order to evaluate the bilinear pairing conditions.

2.3.1Bilinear pairing

Let us consider G_0 and G_1 are two separate multiplicative cyclic groups of prime order q. Here, <g> is the producer of G_0. The bilinear map can be represented by e. This implies G_0 X G_1→ G_1.There are two important conditions which are required to be satisfied for occurrence of bilinear pairing

- For every individual a,b ∈ Z_q^+ and <g> in G_0. Here, $e(g^a,g^b) = e(g,g)^{ab} = e(g^b, g^a)$
- $e(g,g)\neq 1$ (in case of non-degenerated)

If and only if the above two conditions will be satisfied, then the only bilinear map can be evaluated. This technique is effectively integrated with Bernoulli's mapping in order to enhance the overall randomness and performance.

2.3.2Improved ciphertext policy attribute based encryption (ICP-ABE)

There are four phases in ICP-ABE algorithm as shown below. They are:
A. Setup phase
B. Encrypt phase
C. Key generation phase (KEYGEN)
D. Decrypt phase

Algorithm 3: ICP-ABE algorithm

A. Setup scheme

It generates the master key (MK) and public key (PK) parameters using the user's credentials. Public Key and master key are generated as follows:

PublicKey(PK)={(g ϵ $G_{1,}$ (H[i]), g_p ϵ G_2 (H[i])/G_1,$G_2\epsilon$ Integrityvalues(H[i]), secret key

CP[I])),), $e(g,g_p)^{\theta}{}_1$,h ϵZ_r , g=e (g,g_p)}

MasterKey(MK) ={α ϵ PK(g_p), β ϵ Hash(secret key(CP[0]^CP[1]^……CP[n])),$Z_{r,}$ e (α, β) $^{\theta}{}_2$} (6)

B. Encrypt scheme (PK, Attributes and Policies)

The encryption algorithm takes the cloud data (M) as input and generates ciphertext. In this phase, attributes and policies are defined as integrity values of the each user's finger print image H[i]. The encryption algorithm encrypts the message M using the polynomial access tree structure T. Starting with the root node , this technique selects a random number r in p–integer modulo Z with the polynomial function and sets q(R,0)=r. For the intermediate nodes x, it sets q(x, 0) = q(parentnode(x,index)). Let L be the set of leaf nodes in access tree structure, then the ciphertext is generated based on the given access tree structure T as:

Ciphertext(C)={Fill_AccessTree(IntegrityPolicies $\in$ H[i], s $\in$ Z_r, PublicKey), forall x $\in$ X:$C_x=k^{q(x,0)}$, $C^1{}_x$= ATree(H[i](x))$^{q(x,0)}$, m.g,h $\in$ PK} (7)

C. KeyGen scheme (Attributes, Public Key, Master Key)

The KeyGen algorithm generates private key (PrK) using the attributes' set (A). The KeyGen algorithm takes set of attributes A, H(sharedkey) as input and generate secret key as output. This algorithm selects a random number r and $rand_j$ for each attribute A_j and these random numbers are selected as the factor of Integrityvalue(sharedkey) and holds in Z_p.

SecretKey(SK) = {r_j ϵ Zr, hs ϵ IntegrityValues(H[i]), hs. $r_{j,}D'_=$ g_p ϵ PK} (8)

D. Decrypt scheme

It takes user's credentials (Private Key, User's fingerprint integrity value set (C[i])), Ciphertext (C, embedded with the access structure (T)), and public key (PK) as input. Decryption process is executed recursively. A recursive procedure is executed with three parameters ciphertext, secret key, fingerprint integrity value set as attributes A and the node x from access tree.

3.Results and discussion

Amazon cloud storage and computing services are used for data encryption and decryption process. The experimental results of the proposed MFM-CP-ABE model are computed on the real-time Amazon elastic compute cloud (EC2) cloud server with multiple cloud instances. Access key and secret key are used to connect the proposed model to the Amazon web services (AWS) server. In MFM-CP-ABE model, secret key and the ciphertext are stored in the cloud user's storage. While decrypting, each authorized user can decrypt the message using the secret key, ciphertext and fingerprint images. *Table 1*, illustrates the performance of the proposed ICP-ABE model with traditional attribute based encryption algorithms used in cloud computing. From the table, it is observed that ICP-ABE model has less computational time compared to traditional models. Different integrity algorithms are evaluated on cloud data and their bit change rate is compared with the proposed model. ICP-ABE algorithm is considered to be

stronger as the number of bits affected when the single bit cloud data are altered for various data sizes.

Table 2 illustrates the performance of the ICP-ABE model with the traditional attribute based encryption algorithms used in cloud computing. As the size of the cloud data increases ICP-ABE model has less computational time compared to the traditional models. In this table, different cloud data sizes are used as attributes and policies in the encryption model. As the size of the data increases, proposed model gives less computing time compared to the traditional models.

Table 3 describes the comparison of different properties of MFM-CP-ABE model to the traditional models in terms of fixed length, execution time, dynamic key generation, key size, data size, communication overhead, etc. From the table, it is clear that the MFM-CP-ABE model satisfies the maximum possible properties in a cloud environment.

Table 4 describes the comparison of different hash based models and its supporting properties on large data. Here, L_1 represents the operators in the access policy structures, $|G_1|$ represents the length of one element in G_1, E (G_1) is the exponentiation in G_1, P is pairing $|Z_p|$ length of one random number in Z_p, N is total number of attributes in the system, FP is the total number of fingerprint patterns. PP is the public parameter, CT is ciphertext, and SK is secret key parameters

Table 1 Performance analysis of integrity value computation time of ICP-ABE compared with traditional encryption algorithms

Number of Fingerprint Attributes	KPABE (ms)	FHABE (ms)	HOMOMORPHIC (ms)	ICP-ABE (ms)
Two	843.34	813.76	801.97	783.53
Three	972.73	893.87	863.87	769.35
Four	1284.24	1168.24	1109.35	1003.24
Five	1473.12	1398.24	1329.43	1178.35

Table 2 Comparison of ICP-ABE with traditional models

Data size (bytes)	KPABE (secs)	FHABE (secs)	Homomorphic (secs)	ICP-ABE (secs)
500	15.35	13.44	12.53	10.14
1000	27.98	25.87	25.02	23.87
1500	43.97	41.97	40.98	39.86
2000	58.95	55.87	54.57	51.67
2500	69.67	69.13	68.23	67.89
3000	89.23	87.57	86.87	84.68

Table 3 Comparison of MFM-CP-ABE model with traditional models in terms of various security parameters

Parameters	MD5	SHA-256	SHA-512	SHA-1024	Whirlpool	MFM-CP-ABE
Fixed length key	Yes	High	Yes	Yes	Yes	Yes
Execution time (Large data)	High	High	High	High	High	Low
Dynamic key generation	No	No	No	No	No	Yes
Keysize	Fixed	Fixed	Fixed	Fixed	Fixed	Variable
Large data	No	No	No	No	No	Yes
Communication	High	High	High	High	High	Low

Table 4 Complexity analysis of MFM-CP-ABE model with traditional models in terms of encryption, decryption and integrity computations

Model	Encryption	Decryption	Security parameters
IBE-ET[11]	$6*E(G_1)$	$2(P+E(G_1))$	$PP=2*\|G_1\|$ $CT=4*\|G_1\|+\|Z_p\|$ $SK=2*\|G_1\|$
ABE-KS[12]	$2(N+2)E(G_1)$	$NP+E(G_1)$	$PP=8*\|G_1\|$ $CT=4*\|G_1\|$ $SK=3*\| G_1\|$

Model	Encryption	Decryption	Security parameters
CP-ABE-ET[13]	$(2N+11)E(G_1)$	$(8*L_1+1)\mid G_1\mid+4*\mid G_2\mid+12*P$	$PP=(N+7)*\mid G_1\mid+6*\mid G_2\mid$ $CT=8*\mid G_1\mid+\mid Z_p\mid$ $SK=6*\mid G_1\mid$
MFM-CP-ABE	$2(\mid FP\mid)*E(G_1)$	$(\mid FP\mid+1)\mid G_1\mid+\mid G_2\mid+2*P$	$PP=\mid G_1\mid+\mid G_2\mid$ $CT=2*\mid G_1\mid$ $SK=\mid FP\mid*\mid G_1\mid$

Figure 4 describes the comparison of MFM-CP-ABE with existing models in terms of change bit of integrity value and its variation. In the table, it is clear that the proposed model has better changed bit hash value as compared to the traditional models. *Figure 5* illustrates the performance of the MFM-CP-ABE model to the traditional attribute based encryption algorithms used in cloud computing.

It is observed that MFM-CP-ABE has less computational time compared to the traditional models. In this figure, different fingerprint patterns are used as attributes and policies in the encryption model. As the size of the fingerprint patterns increases, proposed model gives less computing time compared to the traditional models.

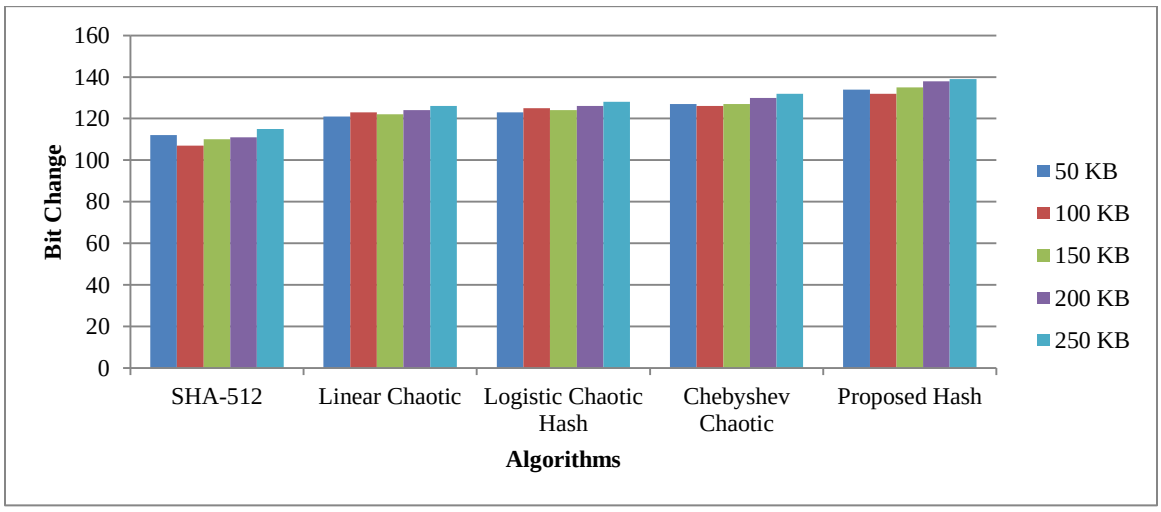

Figure 4 Comparative analysis of MFM-CP-ABE model with existing models with respect to integrity bit change

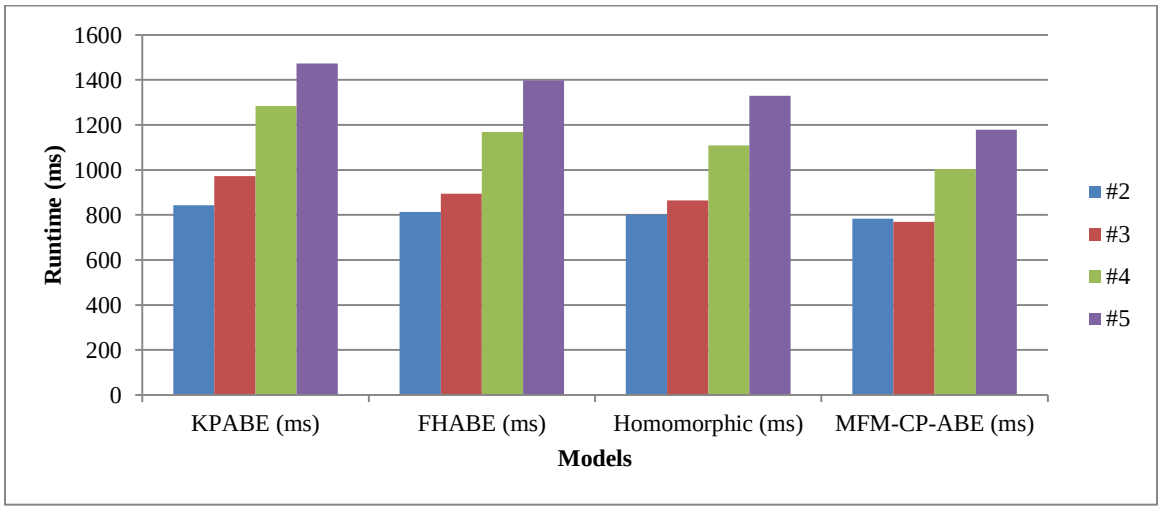

Figure 5 Performance analysis of average integrity value computation time of ICP-ABE for the traditional encryption models

Figure 6 illustrates the performance of the MFM-CP-ABE model to the traditional attribute based encryption algorithms used in cloud computing. As the size of the data increases MFM-CP-ABE model has less computational time as compared to the traditional models. *Figure 7* illustrates the statistical performance of the proposed MFM-CP-ABE algorithm in a cloud environment. It is clear that the MFM-CP-ABE algorithm takes less time compared to the existing techniques.

Statistical T-Test is applied on the encryption and decryption runtime and P-statistic value is compared with the computed statistical T-Test value. In this statistical analysis, proposed MFM-CP-ABE model runtime is less than the existing algorithms. It is null hypothesis. In this experimental study the null hypothesis is accepted in all the cases. It is inferred from this that the MFM-CP-ABE model is better than existing algorithms in terms of running time.

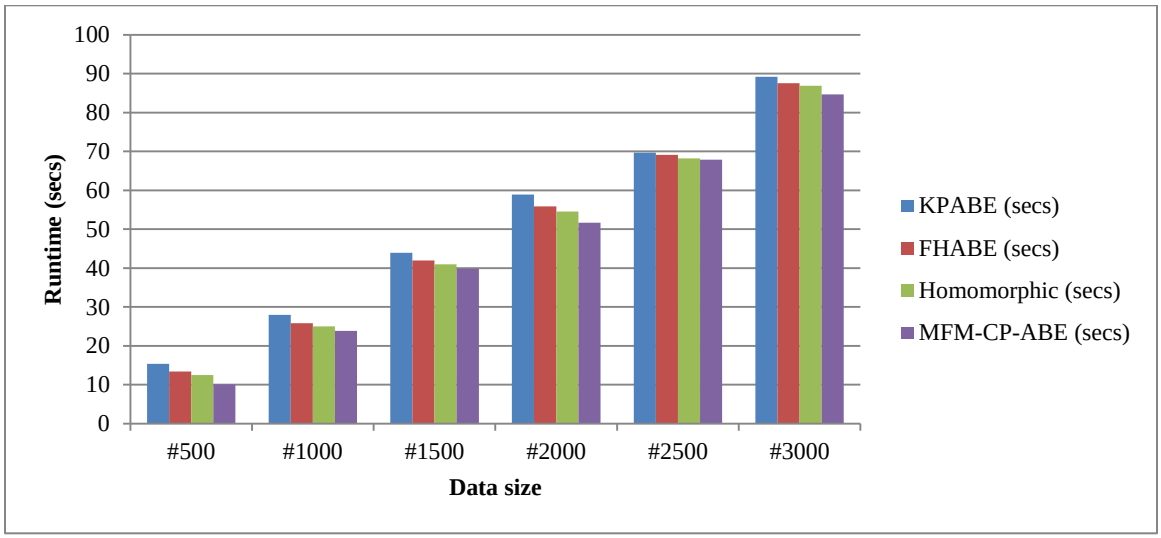

Figure 6 Comparison of MFM-CP-ABE model with existing models

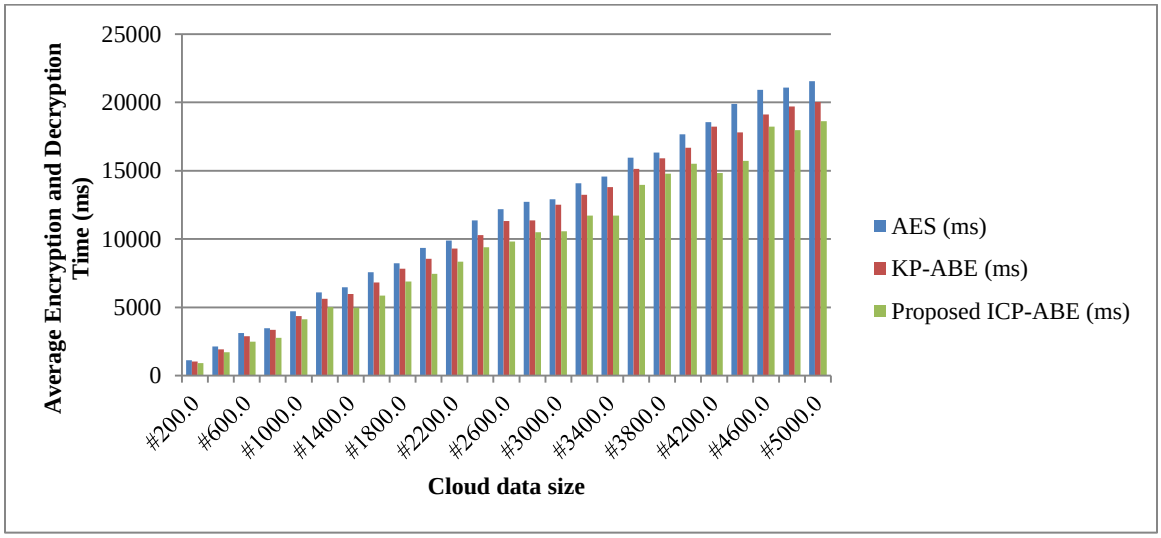

Figure 7 Statistical comparisons of the proposed MFM-CP-ABE model to the existing algorithms in terms of average data encryption time and decryption time

4.Conclusion and future work

In this paper, an efficient MFM-CP-ABE model has been proposed. It is efficient in handling large data size considering various security parameters. Multiple users' fingerprint patterns are used for integrity value computation.

These integrity values are computed using binarization, thinning and pattern extraction methods. Computed integrity values are used to generate public key, master key, and secrete key and used in the initialization process for ICP-ABE model. Experimental results are simulated by considering various sizes of data with different attribute sets. Experimental results proved that the MFM-CP-ABE model has a high computational efficiency compared to the traditional cloud security models in terms of integrity bit change, encryption and decryption time. This work can be extended to implement integrity computation using CP-ABE algorithm on other biometrics like iris and facial recognition.

Acknowledgment
None.

Conflicts of interest
The authors have no conflicts of interest to declare.

References
[1] Lan C, Li H, Caifen WA. Analysis of the comments on "Identity-based distributed provable data possession in multicloud storage". IEEE Transactions on Services Computing. 2017:1-1.
[2] Roy S, Das AK, Chatterjee S, Kumar N, Chattopadhyay S, Rodrigues JJ. Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications. IEEE Transactions on Industrial Informatics. 2018:1-1.
[3] Nayak SK, Tripathy S. SEPDP: secure and efficient privacy preserving provable data possession in cloud storage. IEEE Transactions on Services Computing. 2018:1-1.
[4] Guo F, Mu Y, Susilo W, Hsing H, Wong DS, Varadharajan V. Optimized identity-based encryption from bilinear pairing for lightweight devices. IEEE Transactions on Dependable and Secure Computing. 2017; 14(2):211-20.
[5] Erkin Z, Franz M, Guajardo J, Katzenbeisser S, Lagendijk I, Toft T. Privacy-preserving face recognition. In international symposium on privacy enhancing technologies symposium 2009 (pp. 235-53). Springer, Berlin, Heidelberg.
[6] Wazid M, Das AK, Odelu V, Kumar N, Conti M, Jo M. Design of secure user authenticated key management protocol for generic IOT networks. IEEE Internet of Things Journal. 2018; 5(1):269-82.
[7] Atallah MJ, Hopper NJ. Privacy enhancing technologies. In international symposium on privacy enhancing technologies 2010 (pp. 21-3). Springer, Berlin, Heidelberg.
[8] Torres WA, Bhattacharjee N, Srinivasan B. Effectiveness of fully homomorphic encryption to preserve the privacy of biometric data. In proceedings of the international conference on information integration and web-based applications & services 2014 (pp. 152-8). ACM.
[9] Tarif EB, Wibowo S, Wasimi S, Tareef A. A hybrid encryption/hiding method for secure transmission of biometric data in multimodal authentication system. Multimedia Tools and Applications. 2018; 77(2):2485-503.
[10] Huang K, Shi J, Xian M, Liu J. Achieving robust biometric based access control mechanism for cloud computing. International conference on information and network security 2013 (pp. 1-7). IEEE.
[11] Ma S. Identity-based encryption with outsourced equality test in cloud computing. Information Sciences. 2016; 328:389-402.
[12] Zheng Q, Xu S, Ateniese G. VABKS: verifiable attribute-based keyword search over outsourced encrypted data. In proceedings of international conference on computer communications 2014 (pp. 522-30). IEEE.
[13] Wang Q, Peng L, Xiong H, Sun J, Qin Z. Ciphertext-policy attribute-based encryption with delegated equality test in cloud computing. IEEE Access. 2018; 6:760-71.

Mrs. Ruth Ramya Kalangi is a Research Scholar in the department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. She is currently Assistant Professor in the department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India. Her research interests are Biometrics, Network Security and Wireless Sensor Networks.
Email: ramya_cse@kluniversity.in

Dr. M.V.P. ChandraSekhara Rao received his Ph.D. degree in Computer Science and Engineering from the Jawaharlal Nehru Technological University, Hyderabad. He is currently Professor in the Department of Computer Science and Engineering, RVR & JC College of Engineering, Guntur, Andhra Pradesh, India. His research interests are Data Mining, Big Data Analytics and Privacy Preserving in Data Mining.
Email:manukondach@gmail.com