International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-7, Issue-6, March 2019

A Cryptographic Mechanism using Prime Number and Character Stuffing to Prevent Hijacking of Cloud Data

Narander Kumar, Jitendra Kumar Samriya

Abstract*: Cloud computing is emerging trend with scalable space availability feature for both public and private domain. Currently, cloud security becomes challenging task to accomplish objectives of all security requirements. These issues attracts the attention of researchers because this area more scope of research and prevent the private and important information of peoples and organizations and to minimize cyber fraud even cyber crime as well as traffic hijacking. In this paper we proposed a cryptographic approach named RSA with character stuffing (RSA-CS) using prime numbers also. RSA algorithm is modified for better outcomes in perspectives of cloud environment and comparing existing stuffing approach, used for network security. The proposed mechanism helps to provide better security of cloud data and prevent hijacking as well as unauthenticated access. The implementation of proposed mechanism has been done using Eclipse IDE software. The results show better performance of proposed modified RSA with character stuffing using prime numbers than existing mechanism.

***Index Terms*: Character Stuffing, Cloud Computing, Cryptography, RSA-CS.**

I. INTRODUCTION

Cloud computing is a high performance computational environment with great availability of resources, convenient to end user, providing services from remote located server with large network access feature. It's easy to maintenance and on-demand self-device model based technology. Therefore, many international companies interested to adopt cloud as a consumer and some are in competition to provide large storage capacity as provider to consumers. Cloud security also a feature of cloud computing, however most of research is still going on to enhance security policies of data on cloud. Traffic hijacking is the most important problem found which act as a phobia in large organization as well as cloud users. Here we describe a data security policy to handle hijacking problem of data with stuffing technique.

Revised Manuscript Received on 30 March 2019.
* Correspondence Author
Narander Kumar*, Department of Computer Science, BBA University (A Central University), Lucknow, India.
Jitendra Kumar Samriya, Department of Computer Science, BBA University (A Central University), Lucknow, India.

Input data is taken as a character and perform stuffing approach with RSA algorithm. Generally stuffing refers the mechanism where data is break/partitioned along with relative cryptographic mechanism. In this research paper, stuffing approach is used with modified RSA algorithm.

II. REVIEW OF WORK

Debnath, Somen et al. [1] described a survey on Signcryption which is based on attribute that finds the essential access control and suitable of cloud data. Here the comparison is also discussed in existing research works schemes based on attributed-based signcryption working and analysis on efficiency as well as outcome performance. Islam Thouhedul et al. [2] explained and proposed a mechanism which prevents unauthorized access of files using hash function labelling protection and auto-detectable approach. Here data binding allows by labelling header in two ways. PHP is used two functions for encryption to detect injected object. Total 1600 several types of file used which results that 87% detection is correct of injected objects. Lad, Mohit et al. [3] strengthen on prefix hijacks held at random location in the internet topology. This case study results using hijacks incidents, occurred on network. Here direct customers are most resilent of tier-1. Liu, Yujing et al. [4] demonstrated solution of attacking and detection policy over the network using genetic algorithm, and proposed an effective mode for best outcomes/results. Result of above described model is most useful to eliminate IP prefix hijacking to secure cloud environment. Zhang, Daojuan et al. [5] explained several approaches to solve attack on android application. A shadow system approach with an example is used with email login by user in mobile application. Evaluation and accuracy analysis is also done in proposed attack. The outcomes also explained top developer of android application not able to hold this attack. Casas, Pedro et al. [6] introduced a technique to detect and resolve network attacks automatically using machine learning algorithms with minimal training and compare it with other relevant supervised learning detectors. In proposed outcomes real network is used coming from the WIDE backbone network, For attack labeling, evaluation is done using the most famous MAWILab data. Baitha, anuj kumar and smitha vinod [7] explained the network security issue and verifies it's very essential to resolve problem of session hijacking. A mechanism is also proposed which helps client to resolve attacking problem.

It provides the detailed information about session hijacking and discussed the prevention mechanism. Annie Christina [8] focused on the security issues and given some proactive measure to prevent the security breach on cloud system. It deals with different types of problem like Data Loss, Denial of Service (DoS), Data Breaches, Account Hijacking, Insecure API's, Malicious Insider, Abuse of Cloud Services. Aymen Mudheher Badr et. al. [9] developed dual authentication based protocol for medical data in cloud computing. Their architecture deals with attribute based encryption system for cloud sever and trusted decryption scheme that makes the architecture imperative. It also also discussed about to reduces the complexity of encryption and decryption. B. Ravi Kumar et al. [10] observed the problem related to the bit shifting and stuffing and suggested a new idea of cryptography to improve security. Here BSS method stuffing is done replacing unused bit which shifting by other character. Here encryption and cipher text generation is done using 8 bytes. Liu, Jingwei et al. [11] demonstrates IOT based authentication and key agreement policy, which motive to shift key between public infrastructure (PKI) and cryptography environment without certificate (CLC). This approach solves three problems mainly which are legal authentication access, faithful non-repudiation and key agreement, resolve denial of service (DOS) attacks. Liu, Zhen et al. [12] gives potential mitigation methods to solve security or attacking issues. Here an analysis is also explained that decryption error increases due to re-encryption, so reaction attack may occurs. It helps to get private key of the delegator by proxy user. Mumme, Dean C et al. [13] explained return oriented programming (ROP) attacks protection. User also protected from reverse engineering. The "Code obfuscation Engine" (CobE) accomplish system calls and out-of-band space utilization and code stirring. The protection is done due to hijacking specifically flooding of buffer and reture oriented programming problems. Cheema, Rupinder and Aayush Gulati [14] demonstrated improved RSA algorithm using bit stuffing technique on SSL, which provide better communication. To enhance security they applied cryptographic approaches to prevent attack of proxy user or account hijacking or data hijacking. Elrawy et al. [15] demonstrated the discussion on intrusion detection system in IoT-field. There contribution leads to develop robust intrusion detection system in a crucial environment. Their idea efficiently implemented on logistics, transportation, manufacturing, passing through the hurdle of privacy problem, integrity problem, confidentiality problem. Shereek ,Balkees Mohamed [16] described a technique using cryptography approach RSA with Fermat's rule in cloud computing to provide secure data transmission as well as communication. We use fermat rule to speed up the encryption process of RSA. Mansour Alsaleh et al. [17] proposed a protocol to prevent the dictionary based and brute force attack. The discussed protocol named bounds the login attempts from unauthorized user. It uses the ATT (Automated Turing Test) approach to provide convenience to authorized attempts. Elgendy et al [18] suggested a framework for dynamic decision making on Smartphone devices. It takes different parameters like cpu utilization, execution time, memory usage and energy consumption for offloading decision making. Their optimization model protects the data from any hazard. Hui Zhu et.al. [19] worked for biometric security threat. They proposed a new e-Finga scheme for secure authentication service that uses user's fingerprint.

III. FORMULATION OF RSA MECHANISM

Assumption 1: In this phase we take any positive prime no is selected shown as Xi = Pn1, Pn2,....., Pnn that is Pn1 > Pn2 > Pn3 > Pn4 >.....> Pnn, where all Pn denotes prime no. entity, Where value of Xi must be greater than 0 (fundamental theorem of mathematics).

Assumption 2: n1 and n2 are two factor value integers as a greatest common factor gcf named df, where integer n1 and n2 proceed with coefficient along with de defined in linear combination. The coefficient can define as m1, t belongs to Z, where df satisfy following Euclid method as:

$$df = m1int1 + t\ int2.$$

Assumption 3: let pn1 is a prime number, for all positive integer value where prime no. should also follow Fermat rule as pn1, $x\ (pn1-1) \equiv 1 \pmod{pn1}$.

Assumption 4: let pn1 and pn2 are all relates to prime number and $pn1 \neq pn2$, then

$$\phi\ (pn1\ pn2) = \phi\ (pn1)\ \phi\ (pn2) = (pn1 - 1)(pn2 - 1).$$

Assumption 5 : If the taken value of x not relates to prime no or is a co prime among n values, then we follow Euler rule

$$x^{\phi}(n) = (\text{mod } m).$$

A. Mechanism for Key Generation:

1. Calculate product of random numbers pn1, pn2 and ϕ as:
   y = (pn1-1)(pn2-1) (as according to Assmp. 4)
2. Input an integer E, $1<E<\phi$ where gcd (E, ϕ)=1 (according in Assmp. 2)
3. Evaluate encryption exponent $E*df = 1 \pmod{\phi}$ where $1<df<\phi$. (according in Assmp. 6)
4. Public key is (y, E) and private key is (y, df)
   where
   df, pn1& pn2 and ϕ are the secret values.
   E = encryption exponent
   df = decryption exponent

B. Encryption Algorithm:

At sender side:

1. User find public key (y, E)
2. Present text in positive integer in variable z
3. Encryption proceeds as $fi = z^E \text{ mod } y$
4. Send encrypted data to receiver side

C. Decryption Algorithm:

At server side:

1. Plaintext evaluation with private key (y, df)
2. as : $z = fi^{df} \text{ mod } y$
3. User received plaintext/original input data

IV. PROPOSED (RSA-CS) ALGORITHM:

1. Let four prime number are pn1, pn2, pn3 and pn4.
2. An integer Ec use as a encryption key ;
   $1<Ec<(\phi\ (pn1-1)(pn2-1)\ (pn3-1)\ (pn4-1))$;
   gcd(Ec, ϕ production of (pn1-1)(pn2-1) (pn3-1) and (pn4-1))=1, Where Ec and ϕ(n) are co-prime.
3. Find df, Ec*df=1 mod ϕ(production of (pn1-1)(pn2-1) (pn3-1) and (pn4-1));
   $0 \leq df \leq$ (production of (pn1-1)(pn2-1) (pn3-1) and (pn4-1)).
4. We use public key to send every data (D) or message as :
   $$Cipher = D^{Ec} \pmod n$$
5. Encrypted data message stored and used for stuffing named as (Cstuff), form as a character and add stuffing Cstuff in Cipher text.
   for e.g. Cipher = Cipher + Cstuff (if more than one digit have in Cstuff than we add these digit and regenerate as one step digit).
6. Now, we retrieve Data (D) by remove Cstuff at receiver end as
   Cipher = Cipher - Cstuff and original data transform as following:
   $D = Cipher^{Df} \pmod n$

V. WORKING EXAMPLE:

Here we've to choose four prime numbers and retrieve public and private keys

Let prime numbers are

Pn1 = 3, pn2 = 5, pn3 =17, pn4 = 2

Calculate n = pn1*pn2*pn3*pn4;

So n = 3*5*17*2 = 510

ϕ(n) = (pn1-1)(pn2-1) (pn3-1) (pn4-1)

ϕ(510) = (3-1)(5-1)(17-1)(2-1)

=2*4*16*1

=128

The range of E is 1<Ec<128

ϕ(n) should not be divide by E

Let Ec=3

Select Ec(mod ϕ(n)) to calculate private key as df = Public key (n = 128, Ec = 3)

Private key (n = 1995, df = 43)

Given Character Data D = 11;

Encryption:-

$Cipher = 11^3 \bmod 510$

$= 311$

Now we perform character stuffing as

Cstuff = 3+1+1=5

now cipher = 3115

Decryption:

Remove last stuffed we find original data as plain text/data.

Data (D) = $311^{43} \bmod 510 = 11$

The receive side get the original data.

VI. EXISTING RSA WITH STUFFING VS MODIFIED RSA WITH CHARACTER STUFFING (RSA-CS)

Here we used character stuffing with RSA algorithm using n prime numbers to improve cloud security and reduce account hijacking by theft of information. This can be use to resolve phishing, identity theft problem.

Table 1: Comparison between existing RSA with stuffing and Modified RSA with Character Stuffing (RSA-CS)

Sr. No.	Standard RSA	Improved RSA with stuffing
1	Access of data is fast	Accessing is slow
2	Security level is low as compare to stuffed data	High security level
3	Less overhead during transformation of data	It proposed High overhead in data transmission
4	It can use for phishing and identity theft problem with limitation of resources	Along with account hijack which leads traffic hijack problem in cloud, we can use it to resolve phishing as well as identity theft problem adding stuffing techniques accurately
5	It needed more time to execute security policy e.g. encryption and decryption	Less time required for execution of security policy e.g. encryption and decryption

VII. RESULTS AND DISCUSSION:

The above security policy to resolve account hijacking on cloud which leads traffic hijacking and identity theft problem over the network is done using Eclipse IDE software.

Step1: we implement standard RSA algorithm by using two prime numbers.

Step2: We use four prime numbers following above implementation for generation of public and private keys. We use encryption and decryption of input relevant user information data.

Step3: we use character stuffing mechanism adding in standard RSA algorithm based on process mentioned above. Following are the implementation part of proposed RSA-CS algorithm using stuffing with randomly selected input and then we use encryption and decryption polices on input data. Finally we add stuffing technique to provide more security and improve the complexity of data in perspective of hacker or unauthorized user to resolve traffic hijacking problem of any organisation/user/group in cloud environment.

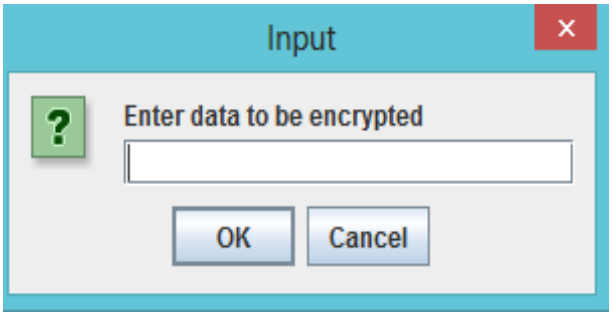

Fig 1: Blank Layout of screen during Execution.

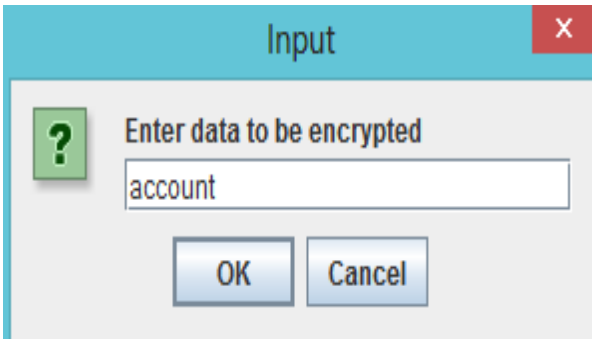

Fig 2: Data input on screen.

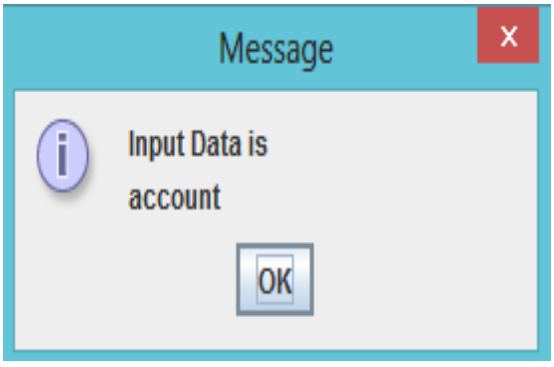

Fig 3: Confirmation of Data input.

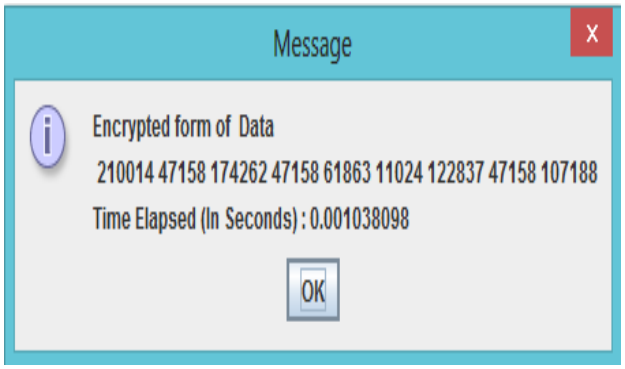

Fig 4: Encrypted form of data input.

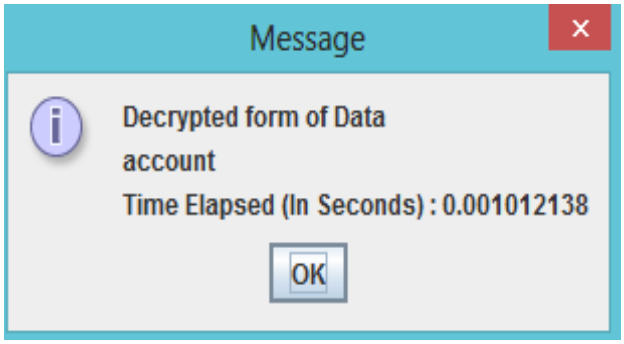

Fig 5: Decrypted form of data input.

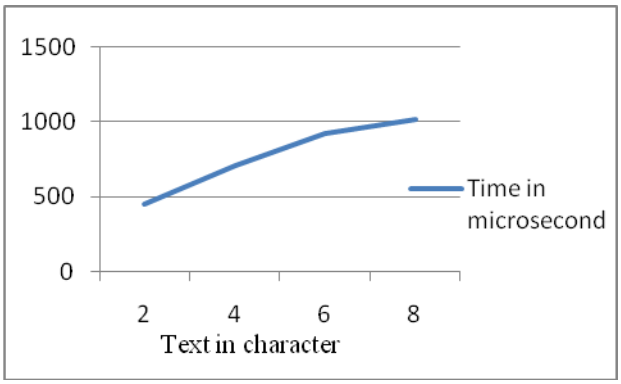

Fig 6: Taken time for Decryption

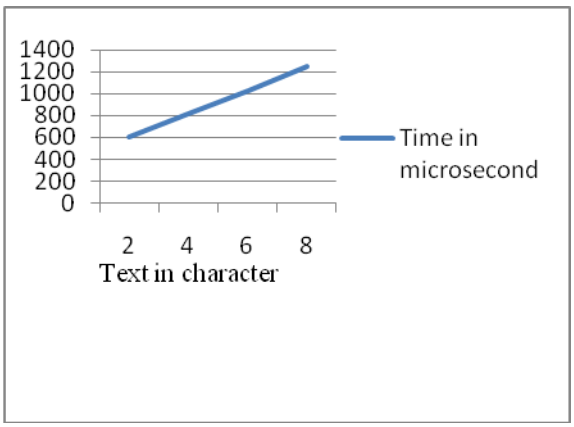

Fig 7: Time taken for Encryption

Table 2: Comparison of execution time between existing and proposed RSA-CS technique

Input data	Execution time (in Microsecond)	Proposed execution time (in Microsecond)
2	2990	1126
4	3960	1545
6	5160	1986
8	5949	2113

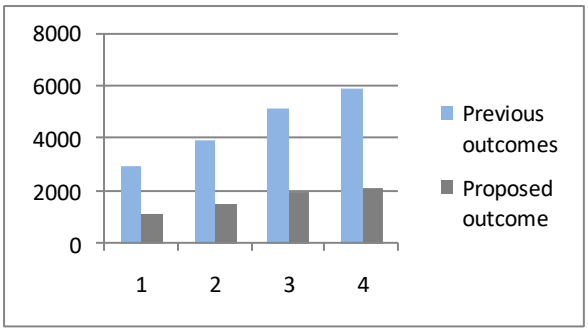

Fig 8: Comparison of encryption/decryption time with existing [14] and proposed mechanism.

Evaluation of throughput is also done after encryption and decryption process as follows:

$$Throughput = \frac{\text{Encrypted text size in MB}}{\text{Encryption time in second}}$$

Table 3: Throughput of modified algorithm

Plaintext	Data size (MB)	Encryption time (seconds)	Throughput
2	0.0009530	0.0012	0.7941
4	0.0009450	0.002	0.4725
6	0.0009589	0.00232	0.4133

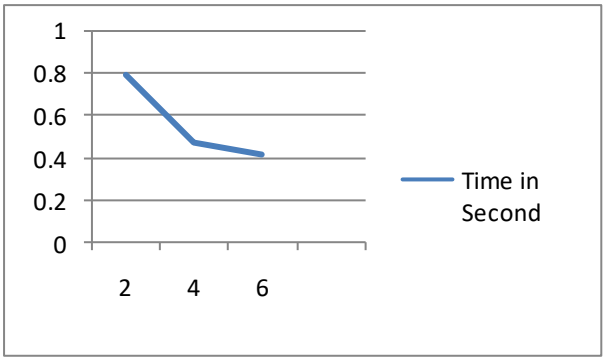

Fig 9: Throughput of proposed RSA-CS technique

VIII. CONCLUSIONS AND FUTURE PERSPECTIVES

Cloud security like traffic hijacking being attracts the attention of researchers because attackers capture user's private details like password, OTP etc. After extensive review of literature as discussed in this research paper, findings are, there are require such mechanism which provides the security of private details of cloud users. If data is in the cloud then encryption/decryption will play very important role for securing the cloud data. In this research paper, after implementation and above shows that the proposed mechanism named modified RSA-CS is increases the security and speed. The proposed mechanism uses n prime numbers and not easily factorized, if used large numbers and extra layer security provided by character stuffing over the cloud index. Thus, it will provide more secure data as compare to previous ones. As future perspectives other mathematical concepts can be applied in existing RSA mechanism to provide more security and efficiency in cloud environment.

REFERENCES

1. Debnath, S., Nunsanga, M. V., & Bhuyan, B. (2019). Study and Scope of Signcryption for Cloud Data Access Control. In Advances in Computer, Communication and Control Springer, Singapore, pp. 113-126.
2. Islam, T., Olanrewaju, R. F., & Khalifa, O. O. (2017, November). MotionSure: A cloud-based algorithm for detection of injected object in data in motion. In 2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA) IEEE, pp. 1-6.
3. Lad, M., Oliveira, R., Zhang, B., & Zhang, L. (2007, June). Understanding resiliency of internet topology against prefix hijack attacks. In 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07), pp. 368-377.
4. Liu, Y., Peng, W., & Su, J. (2011, November). Study on IP Prefix Hijacking in Cloud Computing Networks Based on Attack Planning. 10th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 922-926.
5. Zhang, D., Guo, Y., Guo, D., & Yu, G. (2017). Privacy Leaks through Data Hijacking Attack on Mobile Systems. In ITM Web of Conferences, EDP Sciences. Vol. 12, pp. 04011.
6. Casas, P., D'Alconzo, A., Settanni, G., Fiadino, P., & Skopik, F. (2016, October). POSTER: (Semi)-Supervised Machine Learning Approaches for Network Security in High-Dimensional Network Data. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security ACM, pp. 1805-1807.
7. Baitha, A. K., & Vinod, S. (2018). Session Hijacking and Prevention Technique. International Journal of Engineering & Technology. 7(2.6). pp. 193-198.
8. Christina, A. A. (2015). Proactive measures on account hijacking in cloud computing network. Asian Journal of Computer Science and Technology. 4(2). pp. 31-34.
9. Badr, A. M., Zhang, Y., & Umar, H. G. A. (2019). Dual Authentication-Based Encryption with a Delegation System to Protect Medical Data in Cloud Computing. Electronics. 8(2). pp. 171.
10. Kumar, B. R., & Murti, P. R. K. (2011). Data Encryption and Decryption process Using Bit Shifting and Stuffing(BSS) Methodology. International Journal on Computer Science and Engineering. 3(7). pp. 2818-2827.
11. Liu, J., Ren, A., Zhang, L., Sun, R., Du, X., & Guizani, M. (2019). A Novel Secure Authentication Scheme for Heterogeneous Internet of Thing. arXiv preprint arXiv:1902.03562.
12. Liu, Z., Pan, Y., & Zhang, Z. Cryptanalysis of an NTRU-based Proxy Encryption Scheme from ASIACCS'15.
13. Mumme, D. C., Wallace, B., & McGraw, R. (2017, June). Cloud Security via Virtualized Out-of-Band Execution and Obfuscation. IEEE 10th International Conference on Cloud Computing (CLOUD), pp. 286-293.
14. Cheema, R., & Gulati, A. (2012). Improving the Secure Socket Layer by modifying the RSA algorithm. International Journal of Computer Science, Engineering and Applications. 2(3). pp.79.
15. Elrawy, M. F., Awad, A. I., & Hamed, H. F. (2018, february). Intrusion detection systems for IoT-based smart environments: a survey. Journal of Cloud Computing.7(1).pp.21.
16. Shereek, B. M. (2014). Improve Cloud Computing Security Using RSA Encryption With Fermat's Little Theorem. IOSR Journal of Engineering.4.pp.1.
17. Alsaleh, M., Mannan, M., & Van Oorschot, P. C. (2012). Revisiting defenses against large-scale online password guessing attacks. IEEE Transactions on dependable and secure computing. 9(1). pp. 128-141.
18. Elgendy, I., Zhang, W., Liu, C., & Hsu, C. H. (2018). An efficient and secured framework for mobile cloud computing. IEEE Transactions on Cloud Computing, pp. 1-1.

19. Zhu, H., Wei, Q., Yang, X., Lu, R., & Li, H. (2018, august). Efficient and Privacy-preserving Online Fingerprint Authentication Scheme Over Outsourced Data. IEEE Transactions on Cloud Computing.

AUTHORS PROFILE

Narander Kumar received his Post Graduate Degree and Ph. D. in CS & IT, from the Department of Computer Science and Information Technology, Faculty of Engineering and Technology, M. J. P. Rohilkhand University, Bareilly, Uttar Pradesh, INDIA in 2002 and 2009, respectively. His current research interest includes Cloud Computing, Performance Evaluation, Quality of Service (QoS), Software Engineering, Computer Networks, Resource Management Mechanism, in the networks for Multimedia Applications. Presently he is working as Assistant Professor, in the Department of Computer Science, Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow, INDIA.

Jitendra Kumar Samriya, received his Bachelor of Technology from RTU, Kota, Rajasthan and he has completed Master of Technology in Computer science from University Institute of Engineering and Technology, BBA University (A Central University), Lucknow. Currently he is research scholar in The Department of Computer Science, BBA University (A Central University), Lucknow.