Key-Enforced Access Control and Performance Analysis of DES and RSA Cryptography in Cloud Computing

Y. Kiran Kumar, R. Mahammad Shafi

Abstract*: Cloud computing has been one of the most popular technologies in Information and Communication Technology the last few years. One of the main obstacles for its adoption is the sensitivity of insecurity and privacy violation. Protecting sensitive and high-value data within a cloud system necessitates the use of cryptographic techniques and cryptographic keys. Management of these keys is especially challenging in cloud environments due to the expanded exposure to various insider and outsider threat agents. However, it makes it difficult for data owners to resort to the cloud provider for updating the access control policy when the cooperative relationship changes. Key-enforced cloud access control guarantees the cloud users will outsource their data without outsourcing the control, since the user possesses the key rather than the cloud provider. In this manuscript we analyze the encryption and decryption time for DES, 2DES, RSA & Modified RSA (MDRSA) algorithms.

Keywords: Access control, Cloud Security, Encryption Management, Key-Enforced Access Control, Key Management, Multilevel Authentication.

I. INTRODUCTION

Cloud Computing has been effectively into various fields of computing with a rapidly growing marketplace for cloud-based services. With its opportune pay-as-you-go service, low-cost computing offers, and flexible but unlimited infrastructure resources, cloud computing is exceedingly to be one of the main computing paradigms in the upcoming years. Within the information security domain, certification and accreditation represents a two step process for determining the security posture of an information system [1]. Government sectors, which were moderately unwilling to adopt cloud-based decision due to security concerns, are also attractive concerned and are project to security protest, are also becoming interested and are predict to switch to the cloud [2]. Security is a key concern for distributed systems and services. Cloud Computing has transmissible all these security issues from its predecessors. Moreover, the new-fangled concepts introduced by cloud computing, such as resource sharing, computation outsourcing and external data warehousing, increased the confidentiality concerns and made cloud computing platforms flat to newer security issues and threats. Therefore, security in cloud-based solutions is extremely critical and may be considered as one of the most considerable barriers to widespread implementation and acceptance. Cloud computing not only establishes additional threats and challenges but also includes various difficulties to deploying and maintaining the active security standards. Well-known mobile device access and on-demand services accessible by cloud providers raise the security concerns and threats even further.

Hackers create an individual threat to cloud servers. One method to mitigate this is for remote login to need a two-step authentication. There are many services that are setting up systems where we enter our user ID and password, and then the system sends a code to your cell phone to authenticate you as the user. This additional step could prevent a person from gaining access to our system even if they have your username and password. There are also secure token systems that change our password every 30 seconds. RSA Secure ID [3] tokens can be integrated into most cloud systems. We enter our username and then press a button on the token and put in the random number generated by the token as our password, in addition some extra digits that only know us. Sometimes this type of system is deployed at the edge of the network and once past the first security, we enter a second username and password to get final access to the server. This is often referred to as two-factor authentication. Having big targets like government, financial institutions, or retailers at our data center may attract hackers to the site. Most cloud service companies will not disclose all of their clients, but many will have long lists on their website to make us feel comfortable in our choice of service providers.

Organizational access and commands toward the cloud managing interfaces and services must always be encrypted and authenticated using strong cryptographic protocols. If an organizational web-based management tool is used, it must be set up using encrypted HTTPS connections and at least basic authentication with a username and complex password or better, a strong authentication using a private/public key-based login. The same applies for all web-service requests or API calls that may be initiated by the cloud administrator or by other servers/tools under the control of the administrator. Some cloud providers decided to use RSA keys for all API requests and administrative commands.

Revised Manuscript Received on October 28, 2019.

Y. Kiran Kumar, Research Scholar, Department of Computer Science, Bharathiar University, Coimbatore, T.N., INDIA
E-mail: ykkumar83@gmail.com

Dr. R. Mahammad Shafi, Research Supervisor, Department of Computer Science, Bharathiar University, Coimbatore, T.N., INDIA.
E-mail: rmdshafi@gmail.com

In this case, the cloud administrator uses a unique RSA public/private key to manage the cloud server instances or to request/configure the cloud services. The cryptographic RSA key pair is linked to the cloud account used and should be at least 1024 bits long. The Public key is stored in the cloud and known to the cloud provider. The private key is only known by the cloud server administrator and kept as a secret. This key can be used to login to the administrative console, to authenticate API calls, or to register and create cloud VM images.

II. MULTILEVEL AUTHENTICATION

There are various forms of multilevel authentication that an administrator or provider can make use of to prevent unnecessary access to sensitive cloud data. At present, there is a huge range in authentication standards to access any particular cloud-based server. A few apply simple text passwords; others require added complicated passwords that include numbers and special characters. Some have images that go along with your password, as a second authentication. We need to know the text password and the image that is associated with your account. Using call back verification, biometrics and random generator token passwords is becoming more familiar. These authentication development work under most settings, but fail substantially under attack from DoS (Denial of Service) or brute force attack.

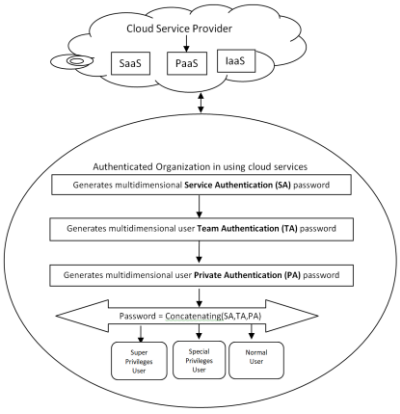

Figure I. Multilevel Key Access

A solution to this complexity would be to apply multilevel authentication with 3+ layers of compound passwords. As we can notice from multilevel key access, First level of authentication is organizational level password authentication/generation. Second level of verification is a team level password. Finally the last level will be the user level password authorization/generation, which ensures end users have particular permissions [4]. Going through the authentication procedure can sometimes be cumbersome so of course there are applications that assist control passwords. For some applications, there is a common password to get into a system, but in order to revise or overwrite existing data a second password is required. This type of managing level authentication is seen in retail. For Example an employee can create a new transaction but is not able to clear an item or give a refund without the manager level password. For some applications, there is a common password to get into a system, but in order to make alter or overwrite active data a second password is required.

Many systems merely require a simple user-generated password to gain access, while others are more robust. The requirements of our application, what laws relating to data breaches may be applicable to us and try to mitigate our risk through good security practices. SNMP, encryption, firewall, anti-virus, and strong passwords are needed to effectively monitor and protect any cloud platform from attack. Human negligence of security is arguably the largest contributor to cloud and network invasion. According to the Online Trust Alliance, a full 90% of data breaches could have been prevented if businesses had better internal controls. The Online Trust Alliance otalliance.org has more information about data breach protection. Poor password selection, stolen laptops, sharing the same password among different websites, and leaving computers on and unlocked for easy access for physical use are all in the top threats.

III. ENCRYPTION KEY MANAGEMENTAND AUTHENTICATION FOR ADMINISTRATIVE ACCESS

If a user makes use of a single machine to access the cloud, the keys for end-to-end encryption can be held by an application on that machine. With users able to access the cloud on multiple devices like Portable Digital Assistant (PDA), tablet computers and smart phones, it can be difficult to share these keys securely between devices. Proper encryption systems rely on the availability of entropy for generating true random numbers and therefore strong and unique encryption key material. The use of virtualization and VMs takes away much of this entropy and can initiate vulnerabilities if keys and seeds can be predicted by an attacker, other cloud customers, or the cloud provider. The use of crypto hardware instead of software-based encryption can be a way out of this problem.

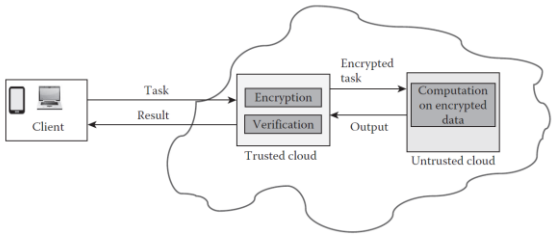

Figure II. Trusted cloud computing platform using TwinCloud architecture.

The client communicates to the trusted cloud over a secure sockets layer/transport layer security (SSL/TLS) using a well defined Representational State Transfer Application Programming Interface, which authorize the client to manage the outsourced data, programs, and queries. The untrusted public cloud computes the operations on the encrypted data and the trusted cloud verifies the results. Figure II illustrate the architecture of TwinCloud.

Administrative access and commands toward the cloud management interfaces and services must always be encrypted and authenticated using strong cryptographic protocols. If an administrative web-based management tool is used it must be set up using encrypted HTTPS connections and at least basic authentication with a username and complex password or better, a strong authentication using a public/private key-based login. The same applies for all web-service requests or API calls that may be initiated by the cloud administrator or by other servers/tools under the control of the administrator.

IV. PASSWORD MANAGEMENT

There are diverse ways a user can use password managing to their advantage in the cloud. One of these ways would be to apply password managing software that auto-syncs to the cloud. Programs like Safe-In-Cloud use services like Dropbox, SkyDrive and Google Drive to sync data held within databases to offsite data centers. These programs also offer well-built encryption algorithms that can encrypt up to 256-bit. The attractiveness of the cloud is the ability to recover passwords remotely anywhere in the world. Password managing software can have downsides though. Managing software is easily susceptible to local machine infiltration and successive theft of backup files. The result to this would be to host the password administrator in the cloud itself using the same Internet encryption standards. LastPass, AgileBits and mSeven are a few whose function in this manner. The solution is to have a great password and prevent any thief at the entrance. This will make sure security of your password crypt. Another option is to use the altering tokens that have biased passwords from the user and the respite of it is a random number generated on the token. The token is usually the size of a small USB drive. One company that recommends the RSA secureID tokens is EMC2. In addition to the substantial token, they also have software token that can run on your smart phone. The benefit is that we do not need to carry an extra device around.

V. EFFICIENT KEY-ENFORCED ACCESS CONTROL

Table I: Implementation of Access Control Policy

Secret Keys			Public Tokens	
Resources	Accessing User Sets	Encryption Keys	Labels	Tokens
r_1,r_9,r_{10}	A,B	$h_d(K_{AB})$	l_{AB}	$t_{A,AB}=K_{AB}\oplus h_a(K_A,l_{AB})$
r_3,r_4,r_5	A,B,C	$h_d(K_{ABC})$	l_{AB}	$T_{B,AB}=K_{AB}\oplus h_a(K_B,l_{AB})$
r_2,r_6	C	$h_d(K_C)$	l_{ABC}	$t_{AB,ABC}=K_{ABC}\oplus h_a(K_{AB},l_{ABC})$
r_7,r_8	D	$h_d(K_D)$	l_{ABC}	$t_{c,ABC}=K_{ABC}\oplus h_a(K_C,l_{ABC})$

Table: I Represents working of access control policy, where h_d and ha is a secure hash function. In this functioning, resources {r1, r9, r10} can be accessed by A and B; resources {r1, r9, r10} can be accessed by A, B, and C; resources {r2,r6} can be accessed by C; and resources {r7,r8} can only be accessed by D. In order to reduce keys for users to maintain, the key KABC can be derived by KAB and KC, and then the structure of a key derivation is constructed in Figure III.

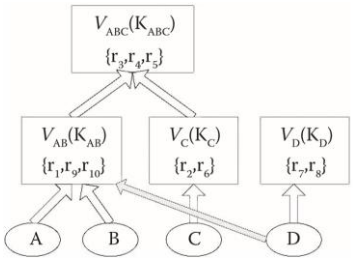

Figure III: Structure of a Key Derivation

We Consider three data resources r_1, r_2, and r_3 and the accessing user set of r_1 is {A,B} with related key K_{AB}; We Consider another three data resources $\{r_1,r_9,r_{10}\}$ can be accessed by A and B; resources $\{r_1,r_9,r_{10}\}$ can be accessed by A, B, and C; resources $\{r_2,r_6\}$ can be accessed by C; and resources $\{r_7,r_8\}$ can only be accessed by D. In order to reduce keys for users to maintain, the key K_{ABC} can be derived by K_{AB} and K_C.

VI. SECURE KEY MANAGEMENT IN THE CLOUD

Proper encryption systems rely on the availability of entropy for generating true random numbers and therefore strong and unique encryption key material. A Secure Key Management System is an integrated is an integrated approach for secure generating, distributing and managing cryptographic keys for all kinds of devices, cloud services or even customer applications.

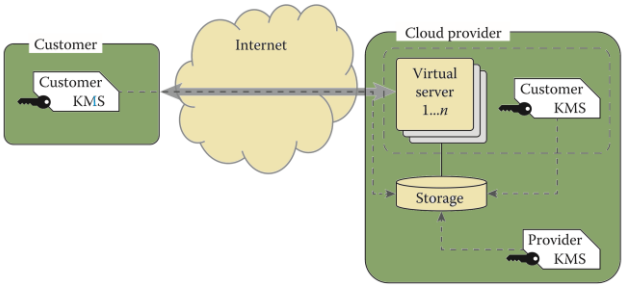

Figure IV: Secure Key Management Solution Framework in the cloud.

Figure IV describes the Secure Key Management Solution (KMS) Framework. It is a cloud provider's encryption service and provides crypto processing if needed. The cloud service provider may offer a cloud-based KMS that allows the customers to store and manage their own encryption keys, which are not known to the provider and cannot be accessed or extracted by the cloud provider or any other entity. The main challenge is Key Management is to configure, clone, or run an appliance that does not embed and reuse sensitive information such as security keys and passwords. Otherwise, the cloud administrator faces the risk that, for example, the same encryption keys or cryptographic initialization vectors are used by other cloud users or available to potential hackers to break the used encryption.

A KMS on the other hand has a broader scope and is an integrated approach for secure generating, distributing, and managing cryptographic keys for all kinds of devices, cloud services, or even customer applications. The cloud service provider may offer a cloud-based KMS service that allows the customers to store and manage their own encryption keys, which are not known to the provider and cannot be accessed or extracted by the cloud provider or any other entity. Most cloud providers decided to use RSA keys for all API requests and administrative commands. In this approach, the cloud administrator uses a unique RSA private/public key to manage the cloud server instances or to request/configure the cloud services. The cryptographic RSA key pair is linked to the cloud account used and should be at least 1024 bits long. The public key is stored in the cloud and known to the cloud provider. The private key is only known by the cloud server administrator and kept as a secret. This key can be used to login to the administrative console, to authenticate API calls, or to register and create cloud VM images [5].

VII. PROPOSED METHODOLOGY

A. Block Diagram

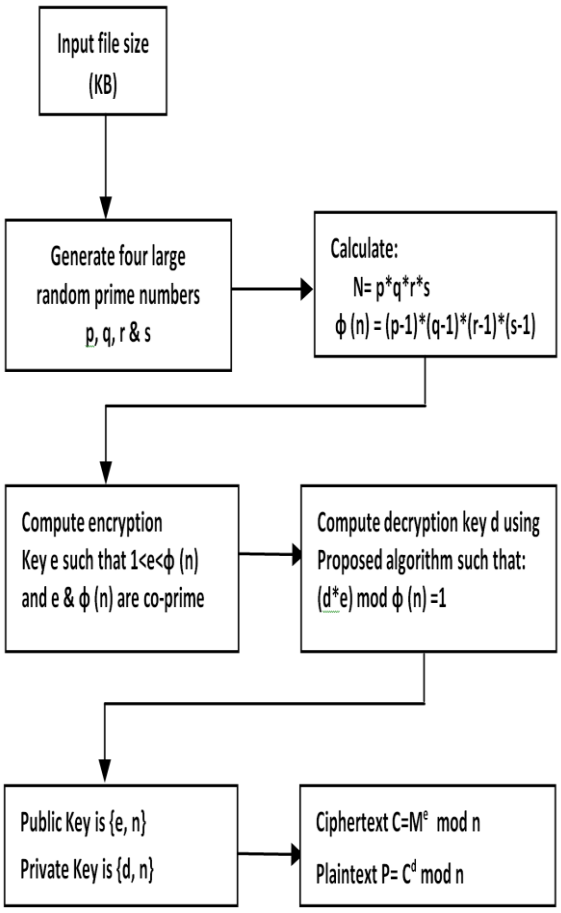

Figure V: Block diagram of Proposed Algorithm (MDRSA)

B. Modified RSA (MDRSA) Algorithm

1. Enter input file size.
2. Generate four large random and distinct prime numbers p, q, r & s.
3. If p, q, r & s are not distinct prime numbers then go to step 2.
4. Compute n ← p * q * r * s
5. Compute ϕ (n) ← (p- 1)*(q- 1)*(r-1)*(s-1)
6. Selects a random integer e (encryption key) such that: 1 < e < ϕ (n) & gcd (ϕ (n), e) = 1
   - 6.1 Set s ← 1
   - 6.2 Set e ← 1
   - 6.3 While s > 1 do
     - 6.3.1 Set e = e + 1
     - 6.3.2 Set s = gcd (ϕ (n), e)
   - 6.4 end while
7. Select the private key (i.e. decryption key) d such that the following equation is true: (d * e) mod ϕ (n) = 1
   - 7.1 Set x ← 1
   - 7.2 Set d ← 1
   - 7.3 while x ≠ 0 do
     - 7.3.1 Compute p ← ϕ (n)*d+1
     - 7.3.2 If mod (p, e) = 0 then
       - 7.3.2.1 Set d ← p/e
       - 7.3.2.2 Set x ← 0
     - 7.3.3 else
       - 7.3.3.1 Set d ← d+1
     - 7.3.4 end if
   - 7.4 end while
8. The public key is (n, e) and the private key is (n, d).
9. Input M
10. If M doesn't lie in between 0 & (n-1) (i.e., 0 < M < (n – 1)), then go to step 9 else go to step 10.
11. Compute cipher text C = M^e mod n
12. Send the cipher text C to the receiver
13. Use the decryption exponent d to recover plain text using M = C^d mod n.

C. Flow Chart

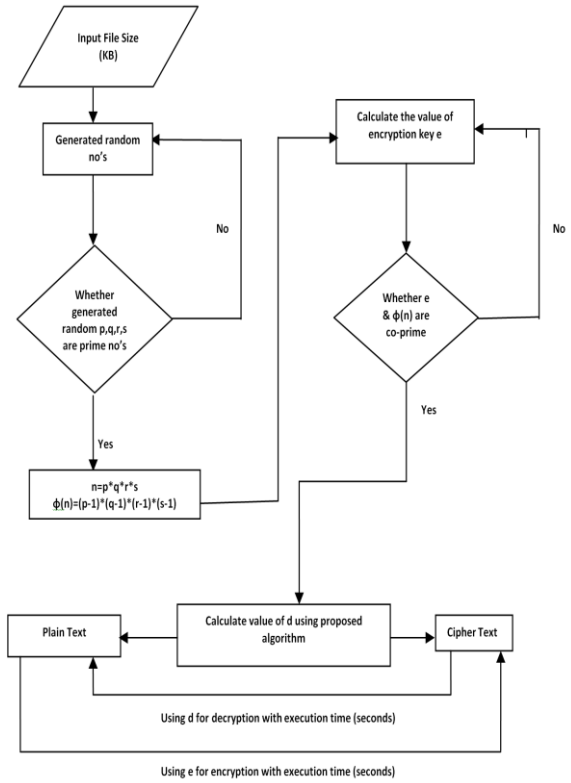

Figure VI: Flowchart of Proposed Algorithm (MDRSA)

D. Result Analysis

Table II: Execution Time of Encryption

Input File Size (KB)	Encryption Execution Time (Seconds)			
	DES	2 DES	RSA	MDRSA
15	4.55	9.1	5.64	7.58
30	9.09	18.18	11.28	13.88
45	13.64	27.27	16.92	19.21
60	18.18	36.36	22.54	25.97
75	22.72	45.44	28.21	31.84
85	27.85	55.12	34.89	39.68
100	32.98	67.98	40.86	46.96

Table III: Execution Time of Decryption

Input File Size (KB)	Decryption Execution Time (Seconds)			
	DES	2 DES	RSA	MDRSA
15	4.55	9.1	5.64	7.58
30	9.09	18.18	11.28	13.88
45	13.64	27.27	16.92	19.21
60	18.18	36.36	22.54	25.97
75	22.72	45.44	28.21	31.84
85	27.85	55.12	34.89	39.68
100	32.98	67.98	40.86	46.96

The Table II and Table III represents the encryption and decryption time for the above four algorithms was compared by varying file sizes. By analyzing the above four algorithms DES takes less time and 2 DES takes highest time [10][11]. Encryption and decryption time almost linearly increases with file size increases. Among these algorithms, MDRSA takes moderate time and more secure to encrypt and decrypt files. Hence MDRSA algorithm is an optimal solution for secure data storage in cloud computing [12][13].

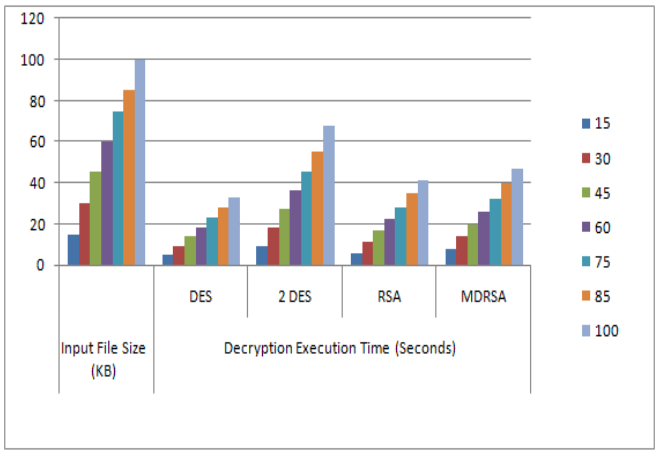

Figure VII: Execution time comparison of Encryption

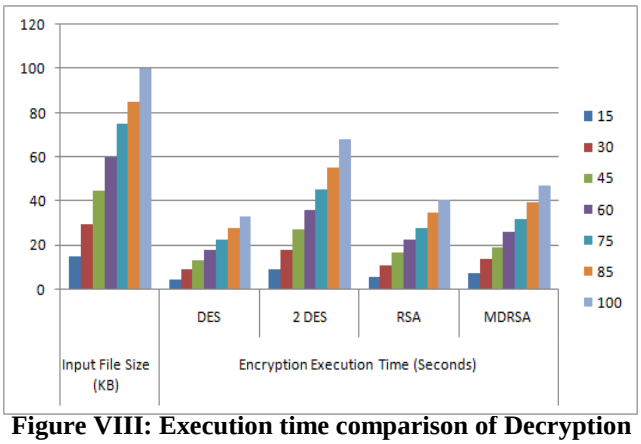

Figure VIII: Execution time comparison of Decryption

Figure VII and VIII shows the encryption and decryption time comparison between above four algorithms. We recommended that MDRSA will be a optimize solution which make balance between speed and security from the reaming algorithms [14] [15].

VIII. CONCLUSION

Cloud Computing is one of the emerging and fast development technology, so security is one of the major problem in the development of cloud computing services. The proposed paper deals the practices of an effective access control techniques and analysis of encryption algorithms. Hence protecting the sensitive data in cloud servers will use an effective key enforced access control techniques and secure encryption algorithms for improved data storage.

REFERENCES

1. Joint Task Force Transformation Initiative, NIST Special Publication 800-37 Revision 1 *Guide for Applying the Risk Management Framework to federal Information Systems*, 2010.
2. Market Research Media. (2014). *U.S. federal cloud computing market forecast* 2015-2020.
3. *RSA Secure ID*. Copyright © 2015 All rights reserved, EMC2, 176 South Street, Hopkinton, MA, 01748-9103.
4. Dinesha, H. A. "*Formal Modeling for Multi-level Authentication in Sensor-Cloud Integration System"*. International Journal of Applied Information Systems 2012;
5. Gemalto N.V. SafeNet: "*Hardware Security Modules (HSMs*)". Amsterdam, The Netherlands. 2015. Available at http://www.safenet-inc.com/data-encryption/hardware-security-modules-hsms/
6. S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "*Encryption policies for regulating access to outsourced data*", ACM Transactions on Database Systems (TODS), vol. 35, no. 2, pp. 1–46, 2010.
7. S.D.C. Di Vimercati, S.Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "*Over-encryption: Management of access control evolution on outsourced data"*, in Proceedings of the 33rd International Conference on Very Large Data Bases, pp. 123-134, VLDB Endowment, 2007.
8. Wuling Ren. "*A Hybrid Encryption Algorithm Based on DES and RSA in Bluetooth Communication"*. Second International Conference on Modeling, Simulation and Visualization Methods (WMSVM), 2010.
9. Charels Connell, "*An Analysis of New DES: A Modified Version of DES"*, Locust Street Burlington, USA, Boston MA 02215 USA.
10. Sombir Singh, Sunil K Maakar and Dr. Sudesh Kumar , "*A Performance Analysis of DES and RSA Cryptography"*, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), ISSN 2278-6856 , Volume 2, Issue 3, May – June 2013.

11. Abdullah Al Hasib, Abul Ahsan Md. Mahmudul Haque," *A Comparative Study of the Performance and Security Issues of AES and RSA Cryptography*", Third International Conference on Convergence and Hybrid Information Technology,2008.
12. Sung-Jo Han, Heang-Soo Oh, Jongan Park, "*The improved Data Encryption Standard (DES) Algorithm*", Department of Electronic Engineering, Chosun University. South Korea. IEEE,1996.
13. D. Coppersmith, "*The Data Encryption Standard (DES) and Its strength Against attacks"*, IBM J. RES. Develop. VOL.38 NO.3 MAY 1994.
14. Shivlal Mewada1, Arti Sharivastava, Pradeep Sharma, S.S. Gautam4 and N Purohit, "*Performance Analysis of Encryption Algorithm in Cloud Computing"*, International Journal of Computer Sciences and Engineering, Volume-3, Issue-2, E-ISSN: 2347-2693, 2015.
15. Asma Khatoon and Dr. Ataul Aziz Ikram, "*Performance Evaluation of RSA Algorithm in Cloud Computing Security"*, International Journal of Innovation and Scientific Research, ISSN 2351-8014 Vol. 12 No. 1 Nov. 2014, pp. 336-345.

AUTHORS PROFILE

Mr.Y.Kiran Kumar, M.C.A. He received his Master of Computer Applications from Sri Venkatesra University, Tirupati. He is Pursuing Ph.D from Bharathiar University, Coimbatore. He is having more than 11 years of teaching experience, currently he is working as a Assistant Professor in the department of M.C.A in Sree Vidyanikethan Engineering College, Affiliated by JNTUA, Ananthapuramu, India. His areas of research interests include Web Technologies, Information Security, Service Oriented Architecture and Cloud Computing.

Dr. R. Mahammad Shafi, M.C.A, M.Tech, Ph.D. He received his Ph.D from University of Allahabad, Allahabad. He is having more than 20 years of teaching experience. His areas of research interests include Software Engineering, Software Testing and Quality Assurance. He has published papers in refereed journals and conference proceedings in these areas. He has been involved in conferences and workshops as a Committee member, organizer and Session Chair. His areas of research interests include Software Engineering, Software Testing and Quality Assurance.