Comparative Study Between Cryptographic and Hybrid Techniques for Implementation of Security in Cloud Computing

Sumit Chaudhary, Foram Suthar and N. K. Joshi

Abstract Cloud computing depicts the latest technology in the field of information communication technology. Cloud computing technology offers the user with an endless list of services which they can avail, and these services range from hardware and software and infrastructure to resource utilization. Cloud computing allows us to access online software application, data storage, and processing power of system from anywhere and at anytime. Cloud computing supports the organization to increase their capacity remotely without creating own infrastructure, platform, purchasing new licensed software that is required for automation of various processes. There are a number of dominant parameters like energy dissipation, resource allocation, virtualization, and security that a user needs to keep in mind while selecting cloud computing services. Of the dominant parameters, security as a parameter in cloud holds a special concern and is one of the biggest challenges in cloud computing as far as data transfer process and data storage process are a concern as it happens through the Internet. In this research work, the authors have presented a comparative study between cryptographic and hybrid techniques for security concerns of the cloud computing paradigm. Through the results obtained from the comparative study of these techniques, it is observed that the effectiveness of the hybrid techniques is better in terms of execution of the algorithm than the cryptographic techniques (common public- and private-key cryptography) of security implementation in cloud computing.

Keywords Cloud computing · Security · Cryptography · Cryptographic security techniques · Hybrid security techniques

S. Chaudhary (✉) · N. K. Joshi
Uttaranchal University, Dehradun, India
e-mail: iimtsumit@gmail.com

N. K. Joshi
e-mail: nkjoshi2001@yahoo.com

F. Suthar
Indrashil Institute of Science & Technology, Cadila Group, Ahmedabad, India
e-mail: foram.suthar@iist.edu.in

M. Pant et al. (eds.), *Performance Management of Integrated Systems and its Applications in Software Engineering*, Asset Analytics,
https://doi.org/10.1007/978-981-13-8253-6_12

1 Introduction

Presently, cloud computing is the latest trend in IT sector. The Internet has been graphically derived by cloud symbol since many years. Cloud shows that the data are transferred over the Internet in a computer network. Computing is processed to utilize computer network to perform our task, and it can be done by hardware device–software device. Computing is done by people every day in the life like to perform calculation automatically, transfer a data, sending a mail, swapping credit card, etc. Cloud computing allows us to access online software application, data storage, and processing power of system at anytime and anywhere. Thus, the name cloud computing is inspired by the symbol of the cloud where the computing will happen over the Internet.

The US National Institute of Standards and Technology (NIST) [1] defines cloud computing as follows: 'cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction' [2]. Cloud computing supports the organization to increase their capacity remotely without creating own infrastructure, platform, purchasing new licensed software that is required for automation of various processes [3].

Cloud is a platform where the user can store the data and use the resources virtually. Cloud allows the user to pay for whatever resources they use, and the user can expand and scale down the resources according to needs. Cloud allows client to access different services using different service delivery model: infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS) [4], and deployment model—private cloud, public cloud, and hybrid cloud [5].

Cloud support several essential characteristics like on-demand self-service, broad network access, rapid elasticity, resource pooling, and measured services to the user. One of the biggest challenges in cloud computing is data security because data transfer process and data storage process are happening through the Internet. So, it is difficult to secure our data and resource against an unauthorized person, masquerade, and eavesdropper. There are several security issues that occur in cloud computing like physical security lost because we do not know where the resource runs. Data integrity is lost because common standard to ensure data integrity does not exist [6]. There are several security algorithm available to prevent the data security while accessing Web in cloud computing. Detail description of preventing techniques is mentioned below.

2 Literature Review

Dimitrios et al. study cloud security in depth in 2012 [7]. The authors proposed security solution by using trusted third party. The solution is based on single sign-on

(SSO) mechanisms and the lightweight directory access protocol (LDAP) [8] that is cryptography specifically in public-key infrastructure. This is to make sure the security proposed solution consists of using cryptography to ensure confidentiality and integrity of involved data. However, it does not recognize which encryption algorithms used. But, this solution does not recognize the encryption algorithm to be used.

Cunsolo et al. in 2009 [9] came up with mechanism that protected data in distributed systems such as grid, cloud, and autonomic. In this paper, author implements the techniques using symmetric and asymmetric algorithm. The limitation of this technique is that the concept of sharing resources is contradicted because data access was done by owner only in the cloud environment.

Hashizume et al. [10] classified different cloud service models (SaaS, PaaS, and IaaS) to solve security issues. The relation between cloud layers and the common threats shows the main vulnerabilities in cloud computing. The solution of splitting some available countermeasures is a technical implementation, which is not covered in this study.

Rahmani et al. [11] implemented a new technique, encryption as a service (EaaS), as a solution based on XaaS concept for cryptography in cloud computing. The security risks and the inefficiency of cloud provider's encryption and of client-side encryption can be prevented by this solution, respectively. Moreover, this solution does not show a comparative study of cryptographic algorithms that can be integrated.

The performances of cryptographic algorithms in cloud platform are evaluated using symmetric and asymmetric algorithms by the Mohammad et al. [12]. Different encryption techniques which are based on key size, the performance, and the size of the output file are discussed in this paper. The distribution of encryption keys in a secure way is not proposed, but it proposed AES algorithm to encrypt data for more security.

In [7] paper, secure cloud architecture using cryptography was first proposed by this D. Zissis. For cloud storage, they proposed to use cryptographic algorithms [11, 13]. But, they do not specify which algorithm is recommended to encrypt data and how to distribute cryptographic keys while maintaining adequacy with cloud characteristics. So these solutions remain incomplete.

In [2], Belguith proposed a new lightweight cryptographic algorithm which is the combination of AES as public-key algorithm to encrypt data and RSA as public–private-key algorithm to distribute keys. During conserving the rights of users to access data by a secured and authorized way, this combination helps to benefit from the efficient security of asymmetric encryption and the rapid performance of symmetric encryption.

3 Cryptographic Techniques

(a) **Data Encryption Standard (DES)**
DES is proposed by NIST in 1977. DES is block cipher and follows 16-round Feistel cipher structure. DES is based on symmetric key where the same key is used for encryption and decryption. This algorithm support 64-bit plaintext and 56-bit key value. First 64-bit plaintext value goes through the initial permutation (IP) stage which reorders the bits and generates permuted input. After initial permutation your 64-bits input value divided into two halves of 32-bits [14].

In key generation process, the function will ignore every 8-bit from original 64-bit key value and generate 56-bit key from 64-bit key. Those 8-bit key value is further used as parity bits in DES algorithm; 56-bit key value is further divided into two halves of 28 bits, and individual value will rotate 1 or 2 bits by circular left shift register. In next step, 56-bit key value passes to the permutation that produces final 48-bit key. Now, right halves of 32-bit plaintext value pass through expansion permutation where you bits will expand from 32 bits to 48 bits. These 48-bit plaintext and 48-bit key value will be XORed and substituted into 32-bit plaintext by S-box.

Final 32-bit plaintext value passes from permutation P-box and XOR with left halves of 32-bit plaintext. This process will do till 16 rounds, and 48-bit key value will be changed at each round. The heart of this algorithm is an inner function which includes expansion, substitution, and permutation process.

DES is less secure algorithm because it supports 56-bits key that is too short. It is vulnerable to brute-force attack.

(b) **Triple DES (3DES)**
Triple DES is similar to DES, but it includes three stages of encryption and decryption with two different keys k1 and k2. First, plaintext will be encrypted by k1 key; then, plaintext will be decrypted by k2 key, and at last, it will gain encrypted by k1 key. Triple DES overcomes the security problems of DES. To crack DES through brute-force attacks, 2ˆ112 operation is required, so it is still be secure.

(c) **Advanced Encryption Standard (AES)**
AES is developed by two people Joan Daeman and Vincent Rijmen in 2000 which is also called Rijndael cipher. AES is extended of DES and published by NIST in 2001 [15]. It is block cipher and supports 128-bit block as plaintext. Except DES, AES supports three key sizes—128, 192, or 256 bits. It is a symmetric key cryptography, which supports both confusion and diffusion. On base of key value, it works on three rounds—10, 12, or 14. In symmetric key cryptography (AES), 128-bit plaintext passes through different four stages in one round as shown below.

1. **Substitution bytes**: AES uses 16×16 matric of byte values, called as S-box to perform a byte-by-byte substitution [16]. This S-box contains permutation of all possible 256 8-bit values. 128-bit plaintext value is divided into 16 blocks of 8 bits. The leftmost 4 bits are considered as row value, and

rightmost 4 bits are considered as column value [17]. Unique 8 bit output value generated by row and column value which are indexes into the S-Box.

2. **ShiftRows**: ShiftRows process is simple permutation process. In ShiftRows process, 4 × 4 matrixes are generated using unique 128 bits which are generated by substitution byte process. This stage performs process as its name shows. First row of state will remain same. The value of second row will shift left by 1 byte. For the third row, 2-byte left shift is performed. And four rows shifted left by 3 bytes.
3. **MixCloumns**: MixCloumns process is simple substitution process. Final 128 bits which are generated by ShiftRows stage pass through MixCloumns process. Each byte of column is multiplied with new function value.
4. **AddRoundKey**: In AddRoundKey bits value XOR with key value (128, 192, 256 bits).

These four stage processes are repeated till the number of rounds 10, 12, and 14 depends on key values of 128, 192, and 256 bits. AES is completely secure against the brute-force attacks because it is a required multibit key to encrypt the data.

(a) **Rivest–Shamir–Adleman Algorithm (RSA)**
Three researchers Rivest, Shamir, and Adleman proposed secure public-key cryptography algorithm. RSA is block cipher where plaintext and ciphertext are integers between $[0, n - 1]$ for some n where size of n is 1024 bits. RSA supports public–private-key (asymmetric cryptography algorithm) concept where encryption will be done by public-key and decryption will be done by private-key. Key generation schema of RSA is shown below.

1. Choose two very large digit prime numbers a and b, where $a \neq b$
2. Calculate: $x = a*b$.
3. Calculate: $\phi(x) = (a-1)*(b-1)$
4. Select integer p Where, $\gcd(\phi(x), p) = 1$; $1 < p < \phi(x)$
5. Calculate n where, $q = p\text{-}1 \pmod{\phi(x)}$
6. Now, we have Public Key $PU = \{p, x\}$ Private Key $PR = \{q, x\}$
7. Encrypt plaintext M and create cipher text: $C = M*p \bmod x$
8. Decrypt cipher text C and create plain text $M = C*q \bmod x$

RSA algorithm is very popular and secure because it is very difficult to factor very large numbers. Factoring large number is not a difficult task, but today not a single algorithm exists to factor a 200-digit number in the reasonable amount of time. This algorithm can be applied to any electronic fund transmissions [18].

(b) **Diffie–Hellman Key Exchange (D-H)**
Diffie and Hellman published a new public-key algorithm which enables two users to establish secret key [19]. This is one of the first secret key transfer protocol over a public channel. When two parties want to share their data over network, first they generate secret key and transfer that secret key in secure channel. Data transfer process will happen between two users when both parties

get keys securely by D-H algorithm [20]. Key generation schema between two person M and N is shown below.

1. Choose two random number: a, b where a is prime number and b is primitive roots of a and b < a
2. Select private key for user M: XM where XM < a
3. Calculate public key for user M: YM where YM = b XM mod a
4. Select private key for user N: XN where XN < a
5. Calculate public key for user N: YN where YN = b XN mod a
6. Compute secret key by user M: K = (YN) XM mod a
7. Compute secret key by user N: K = (YM) XN mod a

In DH key exchange protocol secure because to calculate exponentials modulo of a prime is easy task, but calculation of discrete logarithms is very difficult.

4 Hybrid Techniques

The basic idea behind hybrid techniques is improving the efficiency of the existing algorithm. Many hybrid techniques are used to encrypt the data; some of them are explained below.

(a) **Dual RSA**
It is a new public-key cryptography algorithm; it is also called RSA-CRT, because it uses Chinese remainder theorem, CRT, for its decryption [21]. Dual RSA has been developed for better performance in terms of computation costs and memory storage requirements. RSA takes one block at a time to encrypt and decrypt the data. But dual RSA takes two blocks during encryption and decryption [21]. Encryption time of dual RSA is more as compared to decryption of two blocks. Thus, dual RSA increases the performance as compared to RSA.

(b) **AES–RSA**
Symmetric algorithms are faster to encrypt data as compared to asymmetric techniques. RSA (asymmetric algorithm) takes more processing time to encrypt data due to longest key size. A new symmetric and asymmetric hybrid model has been developed to increase the level of security. In AES–RSA [22] hybrid techniques, AEs first generate 256-bit key. That 256-bit key is expanded to 1024-bit key and is used in RSA as private-key.

(c) **RSA–AES–Digital Signature**
Digital signature [23] is an authentication mechanism which provides data integrity and authentication. It is public–private-key algorithm where data are encrypted by sender's public-key and decrypted by receiver's private-key. In this hybrid key generation algorithm, first 1024-bit key is generated by RSA and XOR with 1024-bit key of AES. Final hybrid key generated by RSA and AES algorithms is used in digital signature as sender's private-key (Table 1; Fig. 1).

Table 1 Efficiency of cryptographic algorithm

Algorithm	Efficiency
DES	45
3DES	20
AES	65
RSA	10
Dual RSA	30
AES–RSA	78
RSA–AES–DS	80

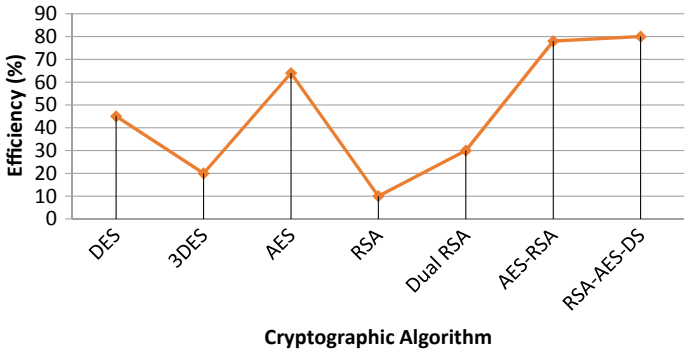

Fig. 1 Comparison of cryptographic and hybrid cryptographic techniques

5 Conclusion and Future Scope

Data security is major issue in cloud computing. Many techniques are available to make secure communication, but one of the most efficient techniques is hybrid technique. This paper explores the comparison between cryptography and hybrid cryptography techniques. Based on the result, we can conclude that the efficiency of symmetric algorithm is more as compared to asymmetric algorithm, and the efficiency of hybrid technique gives average efficiency in any procedure. Because hybrid technique is a combination of public-key cryptography and private-key cryptography, then the security and avalanche effect will be more and also with the use of hybrid technique non-linear function generated.

The work is defined in this paper in terms of efficiency, but we can extend the same work in the different parameters like security, power saving, scheduling, and complexity. Also, we can show the same work with any simulation tool for showing all the parameters, and we can analyze each hybrid technique for different parameters.

References

1. Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*. Information Technology Laboratory: National Institute of Standards and Technology.
2. Belguith, S., Abderrazak, J., & Attia, R. (2015). Enhancing data security in cloud computing using a lightweight cryptographic algorithm. In *The Eleventh International Conference on Autonomic and Systems*.
3. Pai, T., & Aithal, P. S. (2017). Cloud computing security issues-challenges and opportunities.
4. Kanday, R. (2012). A survey on cloud computing security. In *2012 International Conference on Computing Sciences (ICCS)*. IEEE.
5. Sudha, M., & Monica, M. (2012). Enhanced security framework to ensure data security in cloud computing using cryptography. *Advances in Computer Science and its Applications, 1*(1), 32–37.
6. Arora, R., Parashar, A., & Cloud Computing Is Transforming. (2013). Secure user data in cloud computing using encryption algorithms. *International Journal of Engineering Research and Applications,* 3(4), 1922–1926.
7. Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. *Future Generation Computer Systems, 28*(3), 583–592.
8. "Directories and Public–Key Infrastructure (PKI)", VeriSign, 2004.
9. Cunsolo, V. D., Distefano, S., Puliafito, A., & Scarpa, M. (2009). Achieving information security in network computing systems (pp. 71–77). In *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC'09)*.
10. Hashizume, K., Rosado, D. G., Fernández-Medina, E., & Fernandez, E. B. (2013). An analysis of security issues for cloud computing. *Journal of Internet Services and Applications, 4,* 1–13.
11. Rahmani, H., Sundararajan, E., Ali, Z. M., & Zin, A. M. (2013). Encryption as a Service (EaaS) as a solution for cryptography in cloud. *Procedia Technology, 11,* 1202–1210.
12. Mohammad, J., Omer, K., Abbas, S., El-Horbaty, E. S. M., & Salem, A. B. M. (2013). A comparative study between modern encryption algorithms based on cloud computing environment. In *8th International Conference for Internet Technology and Secured Transactions (ICITST'13)* (pp. 531–535). IEEE.
13. Singh, G. (2013). Modified Vigenere encryption algorithm and its hybrid implementation with Base64 and AES. In *2013 2nd International Conference on Advanced Computing, Networking and Security (ADCONS)*. IEEE.
14. Kumar, G., Rai, M., & Lee, G. (2011). An approach to provide security in wireless sensor network using block mode of Cipher. *Security Technology*, 101–112.
15. Elminaam, D. S. A., Abdual-Kader, H. M., & Hadhoud, M. M. (2010). Evaluating the performance of symmetric encryption algorithms. *IJ Network Security, 10*(3), 216–222.
16. Shivkumar, S., & Umamaheswari, G. (2011) Performance comparison of Advanced Encryption Standard (AES) and AES key dependent S-box-Simulation using MATLAB. In *2011 International Conference on Process Automation, Control and Computing (PACC)*. IEEE.
17. Yifeng, Bai, Xiao Jian, and Yu Long. Kernel partial least-squares regression. In *International Joint Conference on Neural Networks*. *IJCNN'06*. IEEE.
18. Milanov, E. (2009). *The RSA algorithm*. RSA Laboratories.
19. Rescorla, E. (1999). *Diffie-hellman key agreement method*.
20. Van Der Merwe, J., Dawoud, D., & McDonald, S. (2005). Fully self-organized peer-to-peer key management for mobile ad hoc networks. In *Proceedings of the 4th ACM Workshop on Wireless Security*. ACM.
21. Subasree, S., & Sakthivel, N. K. (2010). Design of a new security protocol using hybrid cryptography algorithms. *IJRRAS, 2*(2), 95–103.
22. Al Hasib, A., & Haque, A. A. M. M. (2008). A comparative study of the performance and security issues of AES and RSA cryptography (Vol. 2). In *Third International Conference on Convergence and Hybrid Information Technology*. *ICCIT'08*. IEEE.

23. Somani, U., Lakhani, K., & Mundram, M. (2010). Implementing digital signature with RSA encryption algorithm to enhance the data security of cloud in cloud computing. In *2010 1st International Conference on Parallel Distributed and Grid Computing (PDGC)*. IEEE.
24. Yifeng, B., Xiao, J., & Yu, L. (2006). Kernel partial least-squares regression. In *International Joint Conference on Neural Networks*. IJCNN'06. IEEE.
25. Ruhai, W. (2006, November). NIS05-1: Performance Analysis of Advanced Encryption Standard (AES). In *IEEE Globecom*.
26. Singh, A., & Misra, A. (2012, January) Analysis of cryptographically replay attacks and its mitigation mechanism. In *Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India*. Berlin/Heidelberg: Springer.
27. Kofuji, S. T. (2013). Performance analysis of encryption algorithms on mobile devices. In *2013 47th International Carnahan Conference on Security Technology (ICCST)*. IEEE.
28. Al-Anzi, F., Al-Enezi, M., & Soni, J. (2016). New proposed Z-transform based encryption algorithm. In *2016 6th International Conference on IT Convergence and Security (ICITCS)*. IEEE.