International Journal of Applied and Advanced Scientific Research (IJAASR)
Impact Factor: 5.655, ISSN (Online): 2456 - 3080
(www.dvpublication.com) Volume 3, Issue 1, 2018

CRYPTOGRAPHIC ALGORITHMS A SURVEY

K. Sundaravadivelu
Assistant Professor, Department of Computer science Madurai Kamaraj University, Madurai, Tamilnadu

Cite This Article: K. Sundaravadivelu, "Cryptographic Algorithms A Survey", International Journal of Applied and Advanced Scientific Research, Volume 3, Issue 1, Page Number 398-403, 2018.

Abstract:

Cloud computing has become predominant entity in today's world. Cloud computing acts as the fifth utility for providing computing after the other four utilities (water, gas, electricity and telephone). Even though Cloud computing has a lot of advantages, Any Cloud computing research works addresses three critical problems namely security, integrity and privacy. The privacy problem can be preserved by encryption followed by the search process. The challenging scenario in the cloud computing field is the data contribution from several data owners and search process by several users. Secret key derivation policy is common to both the integrity proofs. PDP is verified by Message Authentication Code (MAC). PoR is verified by MD5. Second integrity verification technique also includes two client authentication techniques: One Time Password (OTP) and dynamic missing number puzzle. The major advantage of the proposed approach is that the integrity verification is done by private auditing, that is the data consumer itself. Need for Third Party Auditing (TPA) which is trusted partially is eliminated.

Index Terms: Proof of Data Possession (PDP), cloud computing, data integrity, One Time Password (OTP) and dynamic missing number puzzle

1. Introduction:

Cloud computing provides several services among which "Storage as a service" is an important service. Data owners can host their data in the cloud which provides "24/7/365" data access to the consumers. The desirable properties of Cloud computing such as scalability, elasticity, fault-tolerance and pay-per-use make it as an unavoidable commercial trend. Service Oriented Architecture, autonomic and utility computing, widespread adoption of hardware virtualization, low cost computers and storage devices and availability of high capacity networks have made the enormous development in cloud computing. The entrepreneurs will face the critical consequences, if their confidential data is hacked by their business competitors or public. The confidentiality of the patient's Personal Health Records should be preserved. Current data security approaches focus only on data security in which cryptographic solutions are followed by the random key generation processes. But, the prevailing security technique suffers from minimum data integrity. Loss of key in the conventional cryptographic techniques crash the original data provided by the data owner.

Downloading the entire file for its integrity verification is not an advisable solution due to the I/O and transmission cost expensiveness across the network. It is difficult to detect the data corruption only when we access the data. The users correctness assurance cannot be given for those un-accessed data. A properly designed auditing protocol is essential to prevent data from "flowing away" towards external parties. Due to the potential exposure of decryption keys, unauthorized data leakage occurs.

Access rights are given to some users and forbidden for other users. This process is called as "access control". Secure, scalable access control in cloud computing is achieved by "Attribute Based Encryption (ABE)". "One-to-many" encryption service is provided. One encrypted file can be decrypted by multiple users whose attributes conform to the access policy. The most suitable technology for data access control is Cipher text Policy ABE, because the data owner is given more direct control on access policies and the policy checking is performed "inside the cryptography". CP-ABE becomes expensive because of attribute revocation problem. A revoked user is a person whose attribute is revoked. Two basic requirements for the attribute revocation are forward secrecy and backward secrecy. Re-keying and re-encryption operations are performed by the data owner to avoid the revoked user from accessing the future data.

A Key-Policy based ABE is based on the association of attributes and decryption keys of the user. In the KP-ABE scheme, a cipher text relates to the set of attributes. The decryption key of the user is associated with a monotonic tree access structure. The user can decrypt the cipher text, only when the user attribute related with the cipher text satisfies the tree access structure. Hierarchical Attribute Set Based Encryption (HASBE) technique increases the efficiency of user revocation in multiple value assignment environments. The key escrow problem induced in HASBE technique is considered by Multi Authority Attribute Based Encryption (MA-ABE). The shared files between the data owner and user have hierarchical structure. A number of hierarchy subgroups located at different access levels share a group of files. The storage cost of cipher text and time cost of encryption could be saved, if the files in the same hierarchical structure could be encrypted by an integrated access structure. Three security control mechanisms such as authentication, encryption and data verification technique are incorporated into a single, stand - alone system in this paper. The main drawback of the ABE technique is the increase in the computational cost for key generation and encryption. To overcome

this problem, this paper proposes a secret key derivation process and MAC and MD5 verification to ensure data integrity and authentication techniques to verify data security in the cloud services.

The novel contributions of the proposed efficient key derivation policy and integrity verification techniques are listed as

- Effective users' addition and removal are done by multi-attributes-based secret key generation process.
- Capability to solve the simultaneous multi-users/ keys handling problem.
- Time complexity is reduced by the hash=based mapping and the attributes decomposition-based secret key generation and the secure data transfer level is improved.
- Integrity is verified by MAC and MD5.

The rest of the paper is structured as follows: Section II includes the existing works related to the ABE encryption techniques, integrity verification techniques; one time password and software puzzle authentication techniques for the cloud computing applications. Section III describes the detailed description of the proposed integrity verification techniques with key derivation process and authentication techniques. Section IV illustrates the results of the proposed technique and section V presents the conclusion and future work of this paper.

2. Related Works:

This section describes the conventional encyrption techniques for the cloud computing applications. A scalab le, flexible and fine-grained access control of the outsourced data, Hierararchical Attribute-Set-Based Encryption (HASBE) is proposed by Wan et al. [1]. The access control of the outsourced data in the cloud computing was efficient and flexible in the proposed scheme. A hierarchical encryption scheme combining the identity based encryption and cipher text policybased encryption systems was proposed by Wang et al. to achieve fine-grained access control [2]. Proxy and lazy re-encryption techniques are used efficiently to revoke the access rights of the users in the proposed scheme. A revocable Identity Based Encryption (IBE) was proposed by Li et al. to deploy a hybrid private key for each user [3]. The key generation complexity was reduced in the proposed scheme, while improving the efficiency and security.

The following paragraphs describe the conventional integrity verification techniques. An efficient and secure publlic verification of data integrity scheme was proposed by Yuan Zhang et al. to protect against external adversaries and malicious auditors [7]. A random masking technique was adopted to protect against external adversaries in the proposed scheme and required users to audit auditors' behaviors to prevent malicious auditors from fabricating verification results. Bit coin was used to construct unbiased challenge messages to thwart collusion between malicious auditors and cloud servers. A theoretical framework for the design of PORs was proposed by Bowers. K. D et al [8]. PoR is a compact proof by a file system (prover) to a client (verifier) that a target file F was intact, so that a client can fully recover it.

Different variants of PoR problems such as knowledge-soundness vs. information-soundness, bounded-use vs. unbounded-use are developed by Dodis et al., and gave nearly optimal PoR schemes for each of these variants [9]. Their contributions either generalized the prior PoR constructions or gave the known PoR schemes with the required properties. An efficient Key Derivation Policy (KDP) was proposed by Senthil Kumari et al. for enhancing data security and integrity in the cloud and overcome the problems in traditional methods [10]. In the proposed method, local key was generated from the data attributes, i.e., file attributes. The combination of local key with the user attribute was used to generate the private key. Hashing of private key generated the secret key. Data owner and user shared the secret key. MAC verification process was used to validate the data integrity. A survey of techniques and tools used for cloud data integrity verification was presented by Princelly Jesu. A et al. [11]. Depending on the type of data and its size, the method selection was performed. The overall drawbacks of all the methods with various considerations were summarized in the survey finally.

An application based on Hadoop and Map Reduce framework was implemented by Rajat Saxena et al. [12]. Variant of the Paillier homomorphic cryptography system with homomorphic tag and combinatorial batch codes were the building blocks of their technique. A novel integrity auditing scheme for cloud data sharing services characterized by multi-user modification, public auditing, high error detection probability, efficient user revocation as well as practical computational / communication auditing performance was proposed by Vedire Ajayani et al. [13]. User impersonation attack can be resisted in their scheme, which was not considered in existing techniques that supported multi-user modification. In the case of the untrusted server, a scheme which made use of Merkle Hash Tree (MHT) and AES algorithm to maintain data integrity was proposed by Poonam M. Pardeshi et al. [14]. The proposed scheme used Third Party Auditor (TPA) which acted on behalf of client for data integrity checking and sent an alert to notify the status of the stored data. The proposed system also assured the recovery of data, in case of data loss or corruption, by providing a recovery system.

The following paragraphs describe the existing one time password technique for the cloud computing applications. A novel OTP algorithm was proposed by Hoyul Choi et al. and compared it with the existing algorithms [15]. The proposed scheme was secure against MITM (Man-in-the-Middle) attack and MITPC/Phone (Man-in-the-PC/Phone) attack by using a CAPTCHA image, IMSI number embedded in SIM card and limiting available time of an attack. Authentication protocols were used in mobile applications in different forms. An adversary could know a valid OTP value and be authenticated with this secret information in the presence of

those attacks. A framework model to authenticate cloud users in secured way using One Time Password (OTP) was proposed by Boopathy et al. [16]. The proposed scheme avoided fake log-in by providing OTP based authentication. The proposed scheme used Request based One Time Passkey (ROTP) which was a combination of numeric value, alphabets in small and capital letters with some permitted special characters.

A secure and accessible multimodal authentication method was proposed by Kristin S. et al. that used a one-time-password client installed on a mobile phone which allowed usage by people whose functional impairments adversely affected their ability to use existing solutions [17]. The method used a one-time-password (OTP) client installed on a mobile phone that replaced dedicated OTP generators. The client provided both visual and auditory output, and was based on an application approved for secure log-in to sensitive online Internet banking services. A "One Time Password" user authentication module along with Advanced Encryption Standard (AES) was proposed by Ramesh K et al. for encrypting the owners personal Health Record before uploading it onto the semi trusted cloud server [18]. Researches had been taken up in enhancing the security of the out sourced Personal Health Records by implementing the Attribute Based Encryption (ABE) to assure the patients control over their PHRs.

The following paragraphs describe the existing software puzzle techniques for the cloud computing applications. A software puzzle scheme was proposed by Deepu et al. to deal with Denial of Service (DoS) attacks of certain types [19]. When a client request came in, the server generated a software puzzle for the client to solve. The algorithm was such that an attacker was unable to solve the puzzle in time. A Denial of Service (DoS) attack did not steal or damage the server but blocked or prevented access to the server or website. Such DoS attacks targeted the network bandwidth or connectivity. Qiao Yan et al. discussed the new trends and characteristics of DDoS attacks in cloud computing, and provided a comprehensive survey of defense mechanisms against DDoS attacks using Software Defined Networking (SDN) [20]. Their work can help to understand how to make full use of SDN's advantages to defeat DDoS attacks in cloud computing environments.

3. Secret Key Generation Process and Integrity Techniques:

This section describes the secret key generation process which is common to both the integrity techniques. This paper mainly focuses on the key security for the outsourced data in the cloud servers along with the integrity techniques. The secret key generation algorithm provides secure access control mechanism and data access policies.

3.1 Secret Key Generation Process:

The extraction of the data and user attributes are performed initially. Then, any two file attributes are randomly selected. The AND operation is performed on the selected attributes. The resultant value of the AND operation is the local key. The exclusive OR (XOR) operation is performed with the local key and user attribute and a private key is generated. Then, the Hashing operation is performed to convert the private key into a secret key. When the users need to retrieve data, their request is transferred to the data owner. The data owner sends the secret key directly to the user. Using this secret key, the user can decrypt the cipher text obtained from the cloud, to get the original plain text.

3.2 ABE Process:

The ABE process defines an access control policy, so that the user can access the data, if the data attributes and user attributes satisfy the defined policy. The attribute-set-based encryption prevents the combination of the attributes across the multiple sets. The user access control policy is defined on the basis of corresponding attributes of the users. Initially, the users have to send the request to the key authority, to access a service. The key authority performs computation using the user attributes and private key issues. Then, the key authority issues the public parameter to the service provider, to encrypt the service response. The key security for the message transmission is computed to ensure the security and integrity.

3.3 Secret Key Generation Algorithm:

The key generation algorithm depends upon the two issues such as a secret key and the attributes of the user. The attribute authority receives the master key 'y'. Let file name (fname) and file extension (fext) be the two data attributes. Local key L_K is generated by the intersection of fname and fext. User name is used as the third attribute (uname).

The private key P_K is generated by using the Ex-or operation of the L_K and un. The secret key S_K is generated by hashing the private key P_K. The cost function is performed using the secret key S_K and the selected file. Finally, the secret key is used for encryption.

3.4 Integrity Verification Techniques:

The secret key generation process is common to both the integrity verification techniques. First, integrity is verified by MAC. Second, integrity is verified by MD5. This sub-section describes both the integrity techniques.

3.4.1 MAC Verification Technique:

MAC verifies the data integrity by using a secret key shared between the data owner and user. MAC standard defines the cryptographic checksum, which is obtained from passing the data through a message

authentication algorithm along with the user attributes. The MAC utilizes a session key to detect both accidental and intentional data modifications. The outsourced data file F consists of a finite ordered set of blocks S_1, S_2….S_m. A direct way to ensure data integrity is pre-computation of the MACs for the entire data file. The data owner pre-computes MACs of the file using a set of secret keys and stores them locally, before data outsourcing. For each time during the auditing process, the data owner reveals a secret key to the cloud server and requests for a fresh keyed MAC for verification. MAC verification process enables high data integrity, since it covers all data blocks. The formula of MAC verification is given by the following equation

$$MAC(K_E, M) = H((K_E\ xor\ pad_o)|H\big((K_E\ xor\ pad_i)\big|M\big))$$

The hash function H is used to compute the verified parameters such as secret key K_E, authenticated message M, inner pad and outer pad sequences (pad_o &pad_i).

MAC utilizes a session key and message to detect both concatenated data modifications in hash function. The data owner pre-computes MACs of the file using a set of secret keys and stores them locally, before data outsourcing. For each time during the auditing process, the data owner reveals a secret key to the cloud server and requests for a fresh keyed MAC for verification. MAC verification process enables high data integrity, since it covers all data blocks.

For a large file, downloading and calculating the MAC of the file is an overwhelming process and takes a lot of time. Also, it is not practical since it consumes more bandwidth. Therefore, there is a need for using a lighter technique, which is calculating the hashing value. The advantage of MD5 integrity verification is that it is simple and implementable.

3.4.2 MD5 Verification Technique:

The simplest form of proof of retrievability is taking the hash of block using a keyed hash function. Data Owner uses MD5 Cryptographic hash function of the private key as the secret key. The data owner encrypts the file with the secret key and sends the file to a remote server. When the data owner needs to check his data retrievability, he sends his key and asks the server to send the hash values by using his key in order to compare them with the hash values that data owner has. This method is used by the data owner for private auditing.

In the second method, Data Owner sends the secret key to Data User (client). When the client needs to check his data retrievability, he sends the secret key and compare it with the hash value stored in the data base. If they are the same, the client is authenticated to download the file which he wants from the Cloud Service Provider (CSP) which gives Storage as a Service. This method is used by the data user for private auditing for integrity verification. Data User is given with further authentication checks in order to verify the security. Client is performing One Time Password authentication technique. Overall MD5 integrity verification technique is illustrated in Figure 2.

3.4.2.1 One Time Password Authentication Technique:

Registration process should be done by both the Data Owner and Data User separately in the Cloud Service Provider's Cloud Repository. All the details of the data owner and Client including their mobile number and email address are collected in the cloud data repository. Separate login and control panel user id and password are given for both the data owner and client respectively. Data owner's user name of the control panel is taken into consideration for or operation with the local key and private key is generated. MD5 hash of the private key is considered as the secret key. It is given to the authorized client by the data owner. In the Key integrity checking, if the secret key values are the same, then the client is further checked by One Time Password CAPTCHA code which is sent to the client's mobile number. If the valid OTP Code is entered by the client, then he is further checked by the dynamic missing number puzzle authentication technique before he is given valid permission to download the file which is uploaded by the data owner.

4. Results and Discussion:

4.1 MAC Verification:

The proposed research work is implemented using Java Swing as a front end tool, IDE – Net Beans 7.1, Back end – Wamp Server in Core 2 duo system with hard disk capacity 500 GB. User registration is performed with several attributes such as name, identity number, address, e-mail id and contact mobile number. Data owner enters the file to be uploaded. Its data attributes are created as shown in Figure 3. Local key is created with the data attributes. Private key is generated by xoring local key with user attribute. Hashing of private key results in secret key. Now, Data owner encrypts the file with the secret key and uploads into the cloud.

Data owner gives the valid secret key to the authorized user. User enters the file that he wants to download. The secret key of the user is compared with the secret key of the data owner. Validation is performed. If they are the same, user can retrieve the file that he wants to download. Now, user performs MAC verification. When the user clicks "Integrity check", next page occurs. It shows the contents of the file that the user receives. When the user clicks "Mac Generation", MAC key is displayed.

5. Conclusion:

In this paper, secret key generation process and integrity techniques are proposed to ensure high data security and integrity in the cloud for secure data transmission. However, there are many disadvantages such that the data owner needs to store many keys in order to use one each time. Also, the number of checking is

limited by the number of keys since the remote server could store all keys and the hash values and use them when it is asked to prove having that file. Processing overhead depends on the data consumer either the data owner or the data user. Hence, the future work shall be extended to remove all the drawbacks.

6. References:

1. Z. Wan, J. e. Liu, and R. H. Deng, "HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing, "IEEE Transactions on Information Forensics and Security", Vol. 7, pp. 743-754, 2012.
2. G. Wang, Q. Liu, J. Wu, and M. Guo, "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers," computers & security, vol. 30, pp. 320-331, 2011.
3. J. Li, X. Chen, C. Jia, and W. Lou, "Identity-based encryption with outsourced revocation in cloud computing", IEEE Transactions on Computers, 64(2), pp. 425-37, 2013.
4. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy Attribute Based Encryption," in Proc. IEEE Symp. Security and Privacy, Oakland, CA, 2007.
5. R. Bobba, H. Khurana, and M. Prabhakaran, "Attribute-sets: A practically motivated enhancement to attribute-based encryption," in Proc. ESORICS, Saint Malo, France, 2009.
6. Jiadi Yu, Peng Lu, Yanmin Zhu, Guangtao Xue and Minglu Li,"Toward Secure MultiKeyword Top-k Retrieval over Encrypted Cloud Data", IEEE Transactions on Dependable and Secure Computing, Vol. 10, No. 4, July/August 2013, pp. 239-250.
7. Yuan Zhang, Chunxiang Xu, Hongwei Li and Xiaohui Liang," Cryptographic Public Verification of Data Integrity for Cloud Storage Systems", IEEE Cloud Computing published by the IEEE computer society, September/October 2016, pp.44-52.
8. Bowers K. D., A. Juels, and A. Oprea, "Proofs of retrievability: theory and implementation", in CCSW '09: Proceedings of the 2009 ACM workshop on Cloud computing security, (New York, NY, USA), ACM, 2009, pp. 43–54.
9. Dodis Y., S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in TCC '09: Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography, (Berlin, Heidelberg), Springer-Verlag, Mar - 2009, pp. 109–127.
10. Senthil Kumari. P and Nadira Banu Kamal.Dr. A. R, "Key Derivation Policy for Data Security and Data Integrity in Cloud Computing", Journal of Automatic Control and Computer Science, Vol. 50(3), ed : Springer,DOI:10.3103/S0146411616030032, July 2016, pp. 165–178.
11. Princelly Jesu. A and Ramesh Kumar. S, "A Survey of Techniques and Tools Used for Cloud Data Integrity Verification", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4(2), Feb 2016, pp.1923-1928.
12. Rajat Saxena and Somnath Dey, "Cloud Audit: A Data Integrity Verification Approach for Cloud Computing", Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016), Procedia Computer Science 89 (2016), pp. 142 – 151.
13. Vedire Ajayani, K. Tulasi and Dr P. Sunitha, "Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation", International Journal of Advanced Technology and Innovative Research, Vol. 8(16), Oct-2016, pp. 3146-3152.
14. Poonam M. Pardeshi and Deepali R. Borade, "Improving Data Integrity for Data Storage Security in Cloud Computing", IJCSNS International Journal of Computer Science and Network Security, Vol. 15(6), June 2015, pp.75-82.
15. Hoyul Choi, Hyunsoo Kwon and Junbeom Hur, "A Secure OTP Algorithm Using a Smartphone Application", IEEE Seventh International Conference on Ubiquitous and Future Networks ICUFN Aug - 2015, pp. 476-481.
16. Boopathy D and M. Sundaresan, " Framework Model and Algorithm of Request based One Time Passkey (ROTP) Mechanism to Authenticate Cloud Users in Secured Way", Third International Conference on Computing for Sustainable Global Development (INDIA Com ,Mar - 2016, pp.3898 - 3903.
17. Kristin S. Fuglerud and Oystein Dale, "Secure and Inclusive Authentication with a Talking Mobile One-Time-Password Client", IEEE Security & Privacy, co-published by IEEE Computer and Reliability Societies, March/April 2011, pp.27-34.
18. Ramesh K and Ramesh S, "Implementing One Time Password Based Security Mechanism for Securing Personal Health Records in Cloud", 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 968-972.
19. Deepu. S. D, Dr. Ramakrishna. M.V, "Software Puzzle Counterstrike for Denial of Service Attack ", International Journal of Innovative Research in Computer and Communication Engineering", Vol. 4(4), April 2016, pp. 8061-8065.

20. Qiao Yan, F. Richard Yu, Qingxiang Gong and Jianqiang Li, " Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges", IEEE Communications Surveys & Tutorials, Vol. 18(1), First Quarter 2016, pp. 602-622.
21. Pankaj Kumar, S. S. Ahluwalia and Tharun Kumar S. V, "Using Software Puzzle for Reducing DDos / Dos Cost on SSL / TLS ", International Journal of Computer Applications, Vol. 151 (4), Oct-2016, pp. 23 - 27.
22. Rupali Anil Suravase. Miss and Mr. Pramod Patil, " Survey on: Software Puzzle for Offsetting DoS Attack", International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 4(5), May - 2016, pp. 413 - 415.
23. Shui Yu, Yonghong Tian, Song Guo and Dapeng Oliver Wu, "Can We Beat DDoS Attacks in Clouds?", IEEE Transactions on Parallel and Distributed Systems, Vol. 25(9), Sept - 2014, pp. 2245 - 2254.
24. Yongdong Wu, Zhigang Zhao, Feng Bao, and Robert H. Deng, “Software Puzzle: A Countermeasure to Resource-Inflated Denial-of Service Attacks”, IEEE Transactions on Information Forensics and Security, Vol. 10(1), Jan 2015, pp.168-177.