Kavin Kumar K,2021, 9:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Enabling Public Verifiability for Storage Security in Cloud Computing

Asst. Prof. V. S. Suresh Kumar, Praveen. S, Kavin Kumar K, Arun Prasanth. B, Dinesh Kumar
Department of Computer Science and Engineering,
Nandha College of Technology, Perundurai,
Tamilnadu, India

Abstract-In this paper, we propose a novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In particular, we exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. With our mechanism, the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. In addition, our mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. The propose system, a privacy-preserving public auditing mechanism for shared data in the cloud. We utilize ring signatures to construct homomorphism authenticators, so that a public verifier is able to audit shared data integrity without retrieving the entire data, yet it cannot distinguish who is the signer on each block. To improve the efficiency of verifying multiple auditing tasks, we further extend our mechanism to support batch auditing. There are two interesting problems we will continue to study for our future work. One of them is traceability, which means the ability for the group manager to reveal the identity of the signer based on verification metadata in some special situations. AES is associate degree unvarying instead of Feistel cipher. It's supported 'substitution–permutation network'. It contains of a series of joined operations, a number of that involve exchange inputs by specific outputs and other involve shuffling bits around. Interestingly, AES performs all its computations on bytes instead of bits. Hence, AES treats the 128 bits of a plaintext block as sixteen bytes. These sixteen bytes square measure organized in four columns and 4 rows for process as a matrix.

Keywords: -Storage Security, Cloud computing, Techniques, TPA.

I. INTRODUCTION

CLOUD service providers offer users efficient and scalable data storage services with a much lower marginal cost than traditional approaches. The shared file is divided into a number of small blocks, where each block is independently signed by one of the two users with existing public auditing solutions. Once a block in this shared file is modified by a user, this user needs to sign the new block using his/her private key.

Eventually, different blocks are signed by different users due to the modification introduced by these two different users. Then, in order to correctly audit the integrity of the entire data, a public verifier needs to choose the appropriate public key for each block (e.g., a block signed by Alice can only be correctly verified by Alice's public key). As a result, this public verifier will inevitably learn the identity of the signer on each block due to the unique binding between an identity and a public key via digital certificates under public key infrastructure (PKI).

International Journal of Science, Engineering and Technology

An Open Access Journal

II. PROBLEM STATEMENT

1. System Model:

There is a need to develop an effective public auditing protocol which overcomes the limitation of the existing auditing scheme. The proposed system is developed to verify the correctness of cloud data by TPA, periodically or on demand without retrieving the entire data or without introducing additional online burden to the cloud users and cloud servers. It assure that no data content is leaked to TPA during the auditing process. It maintains storage correctness of data, integrity and confidentiality of stored data. The proposed scheme consists of three basic entities; they are data owner, cloud server storage and TPA.

The data owner or the user is responsible for splitting the file into blocks, encrypting those using AES algorithm, generating a MD-5 hash value for each, concatenating the hashes and generates a AES signature on it. The cloud server is used to store the encrypted blocks of files. When the client or data owner request for data auditing to the TPA, it immediately request for the encrypted data from the cloud server. After receiving the data, it generated the hash value for each block of encrypted files. It uses the same MD-5 algorithm which was used by client. It later concatenate those hash values and generates a AES signature for that file. In the Verification process, the signature generated by TPA and the one stored in the TPA which is provided by the data user are compared by the TPA. If they matches with each other it means that the data is intact and data is not been tampered by any outsider or attacker. If it does not matches then it indicates that the data integrity has been affected or tampered. The result for the data integrity check is provided to the data owner.

2. System Architecture:

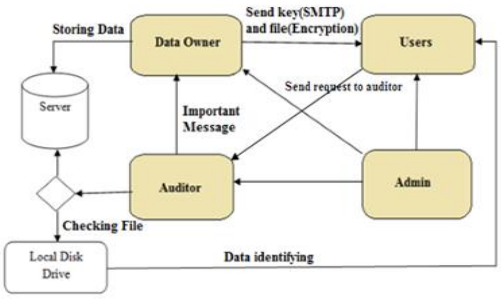

Fig 1.System Architecture

III. LITERATURE SURVEY

A privacy-preserving public auditing system for data storage security in Cloud Computing. We utilize the homomorphism linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage and It is provably secure and highly efficient [1]. The efficient and secure ranked multi-keyword search on remotely stored encrypted database model where the database users are protected against privacy violation and We appropriately increase the efficiency of the scheme by using symmetric-key encryption method rather than public- key encryption for document encryption and The ranking method proves to be efficient to return highly relevant documents corresponding to submitted search terms [2]. A new public auditing mechanism for shared data with efficient user revocation in the cloud and when a user in the group is revoked, we allow the semi-trusted cloud to re-sign blocks that were signed by the revoked user with proxy re-signatures and the group can save a significant amount of computation and communication resources during user revocation [3].

A secure and efficient RDC scheme for network coding-based distributed storage systems that rely on untrusted server and RDC-NC scheme can be used to ensure data remains intact when faced with data corruption, replay, and pollution attacks and The RDC-NC is inexpensive for both clients and servers [4]. A construction of dynamic audit services for unfrosted and outsourced storages. We also presented an efficient method for periodic sampling audit to enhance the performance of TPAs and storage service providers and which minimizes computation and communication costs [5].We believe is the right approach to achieve anonymity in storing data to the cloud with publicly-verifiable data-integrity in mind. The decouples the anonymous protection mechanism from the provable data possession mechanism via the use of security mediator and they minimize the computation and bandwidth requirement of this mediator, but also minimize the trust placed on it in terms of data privacy and identity privacy [6].

IV.ALGORITHM

1. Advanced Encryption Standard (AES):

The specification called for a symmetric algorithm (same key for encryption and decryption) using block encryption (see block cipher) of 128 bits in size, supporting key sizes of 128, 192 and 256 bits, as a minimum. The algorithm was required to be royalty-free for use worldwide and offer security of a sufficient level to protect data for the next 20 to 30 years. It was to be easy to implement in hardware and software, as well as in restricted environments (for example, in a smart card) and offer good defenses against various attack techniques. The entire selection process was fully open to public scrutiny and comment, it being decided that full visibility would ensure the best possible analysis of the designs. In 1998, the NIST selected 15 candidates for the AES, which were then subject to preliminary analysis by the world cryptographic community, including the National Security Agency.

On the basis of this, in August 1999, NIST selected five algorithms for more extensive analysis.

These were:

- MARS, submitted by a large team from IBM Research.
- RC6, submitted by RSA Security.
- Rijndael, submitted by two Belgian cryptographers, Joan Daemen and Vincent Rijmen.
- Serpent, submitted by Ross Andersen, Eli Biham and Lars Knudsen.
- Two fish, submitted by a large team of researchers including Counterpane's respected cryptographer, Bruce Schneier.

Implementations of all of the above were tested extensively in ANSI C and Java languages for speed and reliability in such measures as encryption and decryption speeds, key and algorithm set-up time and resistance to various attacks, both in hardware- and software-centric systems. Once again, detailed analysis was provided by the global cryptographic community (including some teams trying to break their own submissions). The end result was that on October 2, 2000, NIST announced that Rijndael had been selected as the proposed standard. On December 6, 2001, the Secretary of Commerce officially approved Federal Information Processing Standard (FIPS) 197, which specifies that all sensitive, unclassified documents will use Rijndael as the Advanced Encryption Standard.

Also see cryptography, data recovery agent (DRA)Related Glossary Terms: RSA algorithm (Rivest-Shamir-Adleman), data key, greynet (or graynet), spam cocktail (or anti-spam cocktail), finger scanning (fingerprint scanning), munging, insider threat, authentication server, defense in depth, non repudiation

V. HOW THEY WORK

AES is based on a design principle known as a Substitution permutation network. It is fast in both software and hardware. Unlike its predecessor, DES, AES does not use a Feistel network. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The block size has a maximum of 256 bits, but the key size has no theoretical maximum.AES operates on a 4×4 column-major order matrix of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special field. The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of cipher text. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key.

VI. HIGH-LEVEL DESCRIPTION OF THE ALGORITHM

Key Expansion—round keys are derived from the cipher key using Rijndael's key schedule.

1.Initial Round:

- **Add Round Key—**each byte of the state is combined with the round key using bitwise xor.

2.Rounds:

- **Sub Bytes:** A non-linear substitution step where each byte is replaced with another according to a lookup table.
- **Shift Rows:** A transposition step where each row of the state is shifted cyclically a certain number of steps.

- **Mix Columns:** A mixing operation which operates on the columns of the state, combining the four bytes in each column.
- Add Round Key

3.Final Round (no Mix Columns):

- Sub Bytes
- Shift Rows
- Add Round Key

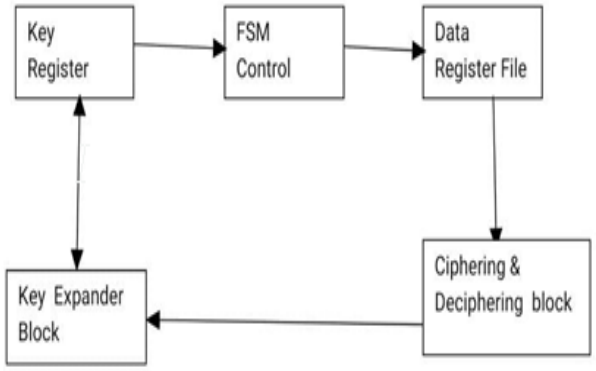

Fig 2.Flow Diagram

VII. MD5 ALGORITHM

MD5 is an algorithm that is used to verify data integrity through the creation of a 128-bit message digest from data input (which may be a message of any length) that is claimed to be as unique to that specific data as a fingerprint is to the specific individual. MD5, which was developed by Professor Ronald L. Rivest of MIT, is intended for use with digital signature applications, which require that large files must be compressed by a secure method before being encrypted with a secret key, under a public key cryptosystem. MD5 is currently a standard, Internet Engineering Task Force (IETF) Request for Comments (RFC) 1321. According to the standard, it is "computationally infeasible" that any two messages that have been input to the MD5 algorithm could have as the output the same message digest, or that a false message could be created through apprehension of the message digest.

MD5 is the third message digest algorithm created by Rivest. All three (the others are MD2 and MD4) have similar structures, but MD2 was optimized for 8-bit machines, in comparison with the two later formulas, which are optimized for 32-bit machines. The MD5 algorithm is an extension of MD4, which the critical review found to be fast, but possibly not absolutely secure. In comparison, MD5 is not quite as fast as the MD4 algorithm, but offers much more assurance of data security. The MD5 Message-Digest Algorithm is a widely used cryptographic hash function that produces a 128-bit (16-byte) hash value. Specified in RFC 1321, MD5 has been employed in a wide variety of security applications, and is also commonly used to check data integrity. However, it has been shown that MD5 is not collision resistant; as such, MD5 is not suitable for applications like SSL certificates or digital signatures that rely on this property. An MD5 hash is typically expressed as a 32-digit hexadecimal number.MD5 was designed by Ron Rivest in 1991 to replace an earlier hash function, MD4. In 1996, a flaw was found with the design of MD5.

While it was not a clearly fatal weakness, cryptographers began recommending the use of other algorithms, such as SHA-1 (which has since been found also to be vulnerable). In 2004, more serious flaws were discovered, making further use of the algorithm for security purposes questionable; specifically, a group of researchers described how to create a pair of files that share the same MD5 checksum. Further advances were made in breaking MD5 in 2005, 2006, and 2007. In an attack on MD5 published in December 2008, a group of researchers used this technique to fake SSL certificate validity.

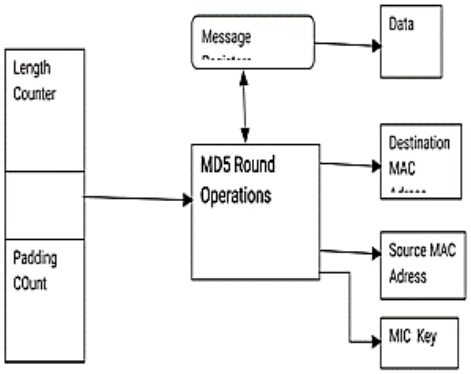

Fig 3.MD5 ALGORITHM

VIII. MODULE

- User registration.
- Public auditing.
- Sharing data.
- Integrity checking.

1. User Registration:

For the registration of user with identity ID the group manager randomly selects a number. Then the group manager adds into the group user list which will be used in the traceability phase.
After the registration, user obtains a private key which will be used for group signature generation and file decryption.

Fig 4.User Registration

2. Public Auditing:
Homomorphic authenticators are unforgeable verification metadata generated from individual data blocks, which can be securely aggregated in such a way to assure an auditor that a linear combination of data blocks is correctly computed by verifying only the aggregated authenticator. Overview to achieve privacy-preserving public auditing, we propose to uniquely integrate the Homomorphic authenticator with random mask technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by a pseudo random function (PRF).

The proposed scheme is as follows:
- Setup Phase
- Audit Phase

3. Sharing Data:
The canonical application is data sharing. The public auditing property is especially useful when we expect the delegation to be efficient and flexible. The schemes enable a content provider to share her data in a confidential and selective way, with a fixed and small cipher text expansion, by distributing to each authorized user a single and small aggregate key.

4. Integrity Checking:
Hence, supporting data dynamics for privacy-preserving public risk auditing is also of paramount importance. Now we show how our main scheme can be adapted to build upon the existing work to support data dynamics, including block level operations of modification, deletion and insertion. We can adopt this technique in our design to achieve privacy-preserving public risk auditing with support of data dynamics. The user download the particular file not download entire file.

IX. CONCLUSION

We propose a novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud.

X. FUTURE WORK

In Our future work will be how to avoid this type of re-computation introduced by dynamic groups while still preserving identity privacy from the public verifier during the process of public auditing on shared data.

REFERENCES

[1] The MD5 Message-Digest Algorithm (RFC1321). https://tools.ietf.org/html/rfc1321, 2014.
[2] B. Wang, B. Li, and H. Li, "Certificate less Public Auditing for Data Integrity in the Cloud," Proc. IEEE Conf. Comm. and Network Security (CNS'13), pp. 276-284, 2013.
[3] C. Wang, S.S. Chow, Q. Wang, K. Ren , and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.
[4] B. Wang, B. Li, and H. Li, "Public Auditing for Shared Data with Efficient User Revocation in the Cloud," Proc. IEEE INFOCOM, pp. 2904-2912, 2013.
[5] Nandagopal S, Arunachalam VP, KarthikS, "A Novel Approach for Mining Inter-Transaction Itemsets", European Scientific Journal, Vol.8, pp.14-22, 2012.
[6] Gokulraj P and Kiruthikadevi K, "Revocation and security based ownership deduplication of convergent key creating in cloud", International Journal of Innovative Research in Science, Engineering and technology. Vol. 3, Issue 10, ISSN: 2319-8753, October 2014.
[7] E.Prabhakar, V.S.Sureshkumar, Dr.S.Nandagopal, C.R.Dhivyaa, Mining Better Advertisement Tool for Government Schemes Using Machine learning ", International Journal of Psychosocial Rehabilitation, Vol.23,Issue.4, pp. 1122-1135, 2019.
[8] Prabhakar E, "Enhanced AdaBoost algorithm with modified weighting scheme for imbalanced problems, The SIJ transaction on Computer science & its application,Vol.6,Issue.4, pp.22-26, 2018.

International Journal of Science, Engineering and Technology

An Open Access Journal

[9] Nandagopal S, Malathi T, "Enhanced Slicing Technique for Improving Accuracy in Crowd Sourcing Database", International Journal of Innovative Research in Science, Engineering and Technology, Vol.3,Issue.1, pp.278-284, 2014

[10] Prabhakar E, Santhosh M, Hari Krishnan A, Kumar T, Sudhakar R," Sentiment Analysis of US Airline Twitter Data using New Adaboost Approach" ,International Journal of Engineering Research & Technology (IJERT),Vol.7,Issue.1, pp.1-6, 2019

[11] S Nandagopal, S Karthik, VP Arunachalam," Mining of meteorological data using modified Apriori algorithm", European Journal of Scientific Research , Vol. 47, no.2, pp. 295-308, 2010.

[12] P Gokulraj, K Kiruthika-Devi," Revocation and security based ownership deduplication of convergent key creating in cloud", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 3, no.10, pp16527-16533, October 2014.

[13] E Prabhakar, R Parkavi, N Sandhiya, M Ambika," Public Opinion Mining for Government Scheme Advertisement", International Journal of Information Research and Review, Vol. 3, no.4, pp2112-2114, February 2016.

[14] E Prabhakar, G Pavithra, R Sangeetha, G Revathy, "Mining Better Advertisement Tool for Government Schemes", International Journal For Technological Research In Engineering, Vol. 3, no.5, pp1023-1026, January 2016.

[15] Karthik.S. Nandagopal.S, Arunachalam.V.P.," Mining of Datasets with Enhanced Apriori Algorithm", Journal of Computer Science, Vol. 8, no.4, pp599-605, 2012.

[16] E. Prabhakar," Enhanced Adaboost Algorithm with Modified Weighting Scheme for Imbalanced Problems", The SIJ Tnsactions on Computer Science Engineering & its Applications (CSEA) , Vol. 6, no.4, pp22-26,July 2017.

[17] Nandagopal.S. Malathi.T.," Enhanced Slicing Technique for Improving Accuracy in Crowd Sourcing Database", International Journal of Innovative Research in Science, Engineering and Technology), Vol. 3, no.1, pp278-284, 2014.

[18] V Dharani S Thiruvenkatasamy, P Akhila, V Arjitha, K Bhavadharani," A MD5 Algorithm Approach to Monitor Village Using Mobile Application", South Asian Journal of Engineering and Technology, Vol. 8, no.s1, 2019.