International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 4 Issue 3, April 2020 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Privacy Preserving Public Aditing of Shared Data

Dr. H. Azath[1], Mohankmar M[2], Kishore Prasad M[2], Nagulraj A[2]

[1]Associate Professor, [2]Student,
[1,2]Department of Computer Science and Engineering,
[1,2]Sri Krishna College of Engineering and Technology,
[1,2]Coimbatore (An Autonomous Institution), Tamil Nadu, India

ABSTRACT

In this paper, we propose a novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud[2]. Particularly, we exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. Using our idea , of the signer on each block in shared data is kept private from all public verifiers, who are able to easily verify shared data integrity without getting the entire file. In addition, our mechanism is able to perform various auditing tasks at same time instead of verifying them one by one.

The propose system, a privacy-preserving public auditing mechanism for shared data in any source particularly cloud. We can use ring signatures to build homomorphism authenticators, so that a public verifier is able to perform auditing task shared data integrity without retrieving the entire data to improve efficiency further extend our mechanism to support batch auditing. There are two problems we will continue to study for our future work. One of them is traceability of data , which means the group manager can able to reveal the identity of the signer based on verification metadata in some special situations.

KEYWORDS: *Deep Learning, My Sql database, Netbeans IDE, Java, HTML, jsp, local host*

How to cite this paper: Dr. H. Azath | Mohankmar M | Kishore Prasad M | Nagulraj A "Privacy Preserving Public Aditing of Shared Data" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 | Issue-3, April 2020, pp.850-854, URL: www.ijtsrd.com/papers/ijtsrd30680.pdf

IJTSRD30680

AES Algorithm

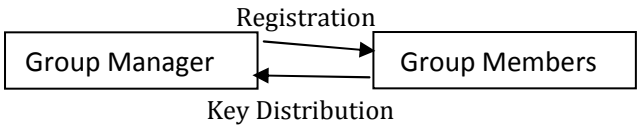
The Advanced Encryption Standard (AES) is an algorithm used for encrypting and securing the data ,unclassified material by U.S. like many of the public agencies and, as a likely consequence, may eventually become the defacto encryption standard for commercial transactions in the private sector. In year of 1997, a process was initiated by the National Institute of Standards and Technology (NIST), a unit of the U.S. Department, of commerce to find a more robust replacement for the Data Encryption Standard (DES) and to a lesser degree Triple DES. The process is 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. This is new algorithm strategy was used 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 defended against various attack techniques. The whole selection process was completely open to public scrutiny and comment, it being decided that full visibility would ensure the best possible analysis of the designs. In the year of 1998, the people from NIST selected 15 candidates for the AES, which were then subject to preliminary analysis by the world cryptographic community, including the National Security Agency. , in August 1999Also with this, NIST selected five algorithms for more extensive analysis. These were:

- The technique MARS, given by a large team from IBM Research
- RC6, submitted by RSA Security

Submissions of all idea were tested in ANSI C language and JAVA language, for speed and efficient reliability in such measures as encryption and decryption speeds, key and algorithm time setup and resistance to different attacks, both in hardware- as well as software-centric systems. The repeated detailed analysis was provided by the global cryptographic community. The result was that on Oct 2, 2000, NIST declared that Rijndael had been selected as the proposed standard. Where in Dec 6, 2001, the Secretary of Commerce officially declared Federal Information Processing Standard (FIPS) 197, which specifies that all sensitive, unclassified documents will use Rijndael as the AES. We can 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), fingerscanning (fingerprint scanning),munging, insider threat, authentication server, defense in depth, nonrepudiation.

1. User Registration:
The Registration of user ID the group manager randomly selects a number. Then the manager from group adds into the group user list which is used in the traceability phase. After the registration, user has private key which will be used for generation of signature groups and file decryption.



2. Public Auditing:
The users verification of metadata explained from individual data blocks and audit data in linear combination of data and file blocks is correctly computed verified only by the aggregated authenticator. The main task to achive privacy-preserving public auditing, we propose to unique integratation the Homomorphic authenticator with random mask technique. By using our rules, the linear combination of sampled blocks in the server's response is masked with randomness generated by a pseudo random function (PRF).

The new schema is as follows:
1. Setup Phase
2. Audit Phase

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

4. Integrity Checking:
In Dynamic data privacy-preserving public risk auditing is also of paramount importance. We show how to develop dynamic data including block level operations of modification, deletion and insertion. It can able to perform in this feature in our design to achieve privacy-preserving public risk auditing with support of data dynamics. User can able to download the particular file not download entire file.

Algorithm
Advanced Encryption Standard (AES):
The Advanced Encryption Standard (AES) is an algorithm used for encrypting and securing the data ,unclassified material by U.S. like many of the public agencies and, as a likely consequence, may eventually become the defacto encryption standard for commercial transactions in the private sector. In year of 1997, a process was initiated by the National Institute of Standards and Technology (NIST), a unit of the U.S. Department, of commerce to find a more robust replacement for the Data Encryption Standard (DES) and to a lesser degree Triple DES. The process is 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. This is new algorithm strategy was used 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 defended against various attack techniques. The whole selection process was completely open to public scrutiny and comment, it being decided that full visibility would ensure the best possible analysis of the designs. In the year of 1998, the people from NIST selected 15 candidates for the AES, which were then subject to preliminary analysis by the world cryptographic community, including the National Security Agency. , in August 1999Also with this, NIST selected five algorithms for more extensive analysis.

Explanations
AES is an algorithm based on the principle known as a Substitution permutation network. It is fast in both software and hardware On otherhand predecessor, DES, AES algorithm does not use a Feistel network. AES algorithm has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, on otherhand Rijndael can be specified has a single block and key sizes in any multiple of 32 bits, has a minimum of 128 bits. The blocksize 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. Maximum number of AES calculations are performed in a special field. The AES cipher has various repeations of transformation rounds that convert the input plaintext into the output of ciphertext. Every round has several processing steps, including one that depends on the encryption key. A group of reverse rounds are applied to convert ciphertext back into the original plaintext using the same encryption key.

High-level description of the algorithm
1. Key expansion: The round keys are getting from the cipher key using Rijndael's key scheme.
2. Initial Round
    1. Add Round Key: every byte of the state is combined with the round key using bitwise xor

3. Rounds
    1. Sub Bytes: It is non-linear substitution having each byte is replaced with another according to alookup table.
    2. Shift Rows: each row of the state is shifted cyclically a certain number of steps.

4. Final Round (there is no MixColumns)
1. SubBytes
2. ShiftRows
3. AddRoundKey

Diagrams

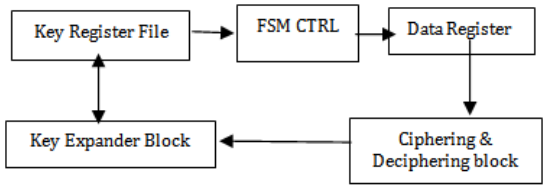


Fig 1 AES algorithm workflow

Examples
In this problem, twenty examples are provided for the MAC generation process. The block which is underlying is the cipher is in form of either the AES algorithm or TDEA. A cipher key is fixed for various key sizes, i.e., AES-128, AES-192, AES-256, two key TDEA, and three key TDEA. In every

key, the generation of the associated subkeys is given, followed by four examples of MAC generation with the key. The contents in every set of examples are derived by truncating a common fixed string of 64 bytes. Every strings are represented in hexadecimal notation, with a space inserted every 8 symbols, for readability. As in the body of the Recommendation, K1 and K2denote the subkeys, M denotes the message, and T denotes the MAC. For this algorithm examples, Tlen is 128, i.e., 32 hexadecimal symbols, and K denotes the key. In the TDEA example, Tlen was 64, i.e., 16 hexadecimal symbols, and the key, K, is the ordered triple of strings, (Key1, Key2, Key3). For two key TDEA, Key1 = Key3. D.1 AES-128

In examples below, the block cipher is the AES algorithm with the following 128 bit key:
K 2b7e1516 28aed2a6 abf71588 09cf4f3c.

Subkey Generation
CIPHK(0128)
7df76b0c 1ab899b3 3e42f047 b91b546f
K1 fbeed618 35713366 7c85e08f 7236a8de
K2 f7ddac30 6ae266cc f90bc11e e46d513b

Example Explanations
The Advanced Encryption Standard (AES) algorithm denotes a FIPS-approved cryptographic algorithm that can be used to protect electronic data[3]. This algorithm is a simple symmetric block cipher it can encrypt and decrypt information. Encryption is a technique that converts data to an unintelligible form called ciphertext. That is decrypting the ciphertext converts the data back into its original form, called plaintext. The AES is an algorithm is capable of using various cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits.

MD5 Algorithm Definition
MD5[1] 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 standard to that specific data as a fingerprint is to the specific individual one. MD5, which was designed by Professor called Ronald L. The MIT Reverts, is intended for using with digital signature applications, which includes that large files must be compressed by a secure method before it is encrypted with a key which is secretly maintained, under a public key cryptosystem. MD5 is an algorithm which is currently a standard, Internet Engineering Task Force (IETF) Request for Comments (RFC) 1321. On basis of 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 otherwise that a false message could be created through apprehension of the message digest. MD5 is the third Message Digest algorithm developed by an expert called as Rivets. All three algorithms like MD3 , MD4 and have same structures, but MD2 able to optimized for 8-bit machines, in comparison with the two later formulas, which are optimized for 32-bit machines. The MD5 is algorithm that is an extension of MD4, which the critical review found to be fast, but possibly not absolutely secure. Comparing, MD5 is not quite as fast as the MD4 algorithm, but offers much more assurance of data security.

Explanations
The MD5 expands Message-Digest-5 Algorithm that is a widely used cryptographic hashing function that produces a 128-bits (16-bytes) hash value. In RFC grade1321, where MD5 has been employed in a wide variety of security applications, and is also commonly used to check data integrity. May it has been conclude that MD5 is not collision resistance as, MD5 is cannot suitable for applications like SSL certificates or the various digital signatures that rely on this property. An MD5 is an algorithm for hashing, hash is typically expressed as a 32-digit of hexadecimal. .MD5 was designed by Ron Rivets in 1991 to replace an earlier hash function, MD4. In the year of 1996, a flaw or the fatal was found with the design of MD5 algorithm. While it is not like weakness in fatal, cryptographers has been recommending the usage of different algorithms, such as SHA-1 algorithm. In the year of 2004, more serious fatal were discovered, that are making further use of the algorithm for various security reasoning purposes questionable; specifically, a group of experts described how to create a pair of files or a set of data that share the same MD5 checksum. Various advance models of versions were made in breaking MD5 in 2005, 2006, and 2007. While attack on MD5 published in December 2008, a group of researchers used this technique to fake SSL certificate validity

EXISTING SYSTEM:
The existing system has a new significant privacy issue introduced in the case of shared data or the files with the use of the leakage of identity privacy to public verifiers. Where in Data checking using traditional approach correctness is to retrieve the entire data from the cloud or other storage devices, and then verify data integrity by checking the correctness of signatures.

Security purposes, they introduce a new effective third party auditor (TPA), the following two vital requirements have to be met: 1) Third Party Auditor should be able to efficiently auditing the cloud data storage or local data storage without requesting the local copy of data, and introduce no additional on-line burden to the cloud or local storage user; 2) The third party auditing process should provide in no new problems towards user data privacy.

LIMITATIONS
- As users has no time for physically possess the storage of their data and their files, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted.
- They do not perform the multiple auditing task in simultaneously.

PROPOSED SYSTEM:
The In Fig.2: propose system, a privacy-preserving public auditing mechanism for shared data in the cloud. We use ring type of signatures to construct homomorphism for various authenticators, so that a public verifier is able to auditing shared data on different storages and audit integrity but not retrieving the entire data, and it cannot distinguish who is the signer on each block.

To empower the efficiency of verifying multiple or various auditing tasks, we further extend our mechanism to support batch auditing. There has some problems we will continue to study for our future work. One of the future work is traceability of data and shared users, which means the ability for the group manager to reveal the identity of the signer based on verification metadata in some special situations

ADVANTAGES:

- The proposed system can perform multiple auditing tasks simultaneously
- They improve the efficiency of verification for multiple auditing tasks.
- High security provide for file sharing.

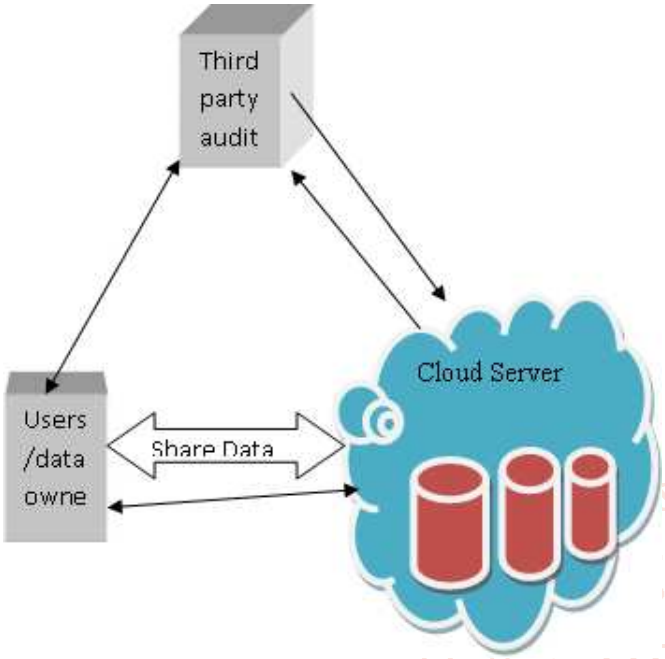


Fig.2: System Architectre

Results:

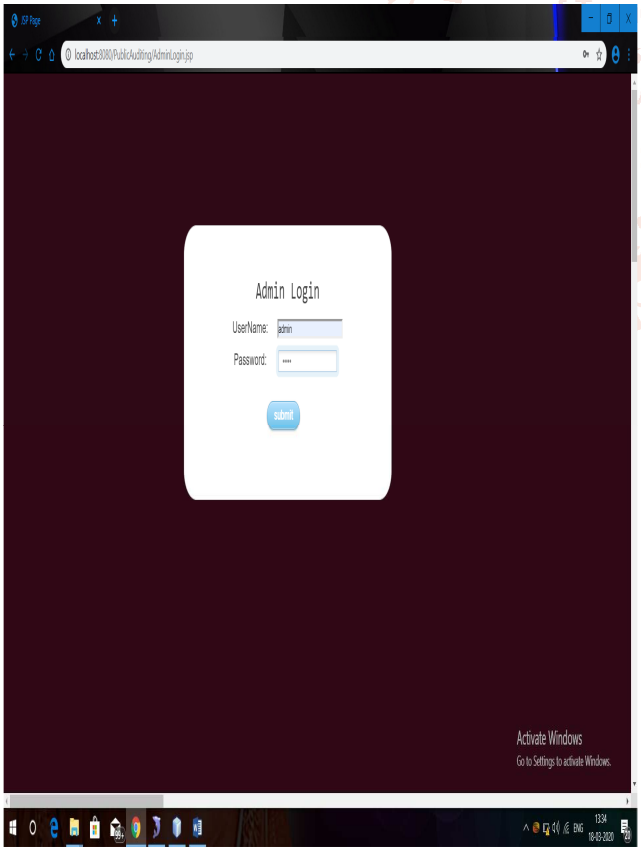


Fig.3: Admin page View

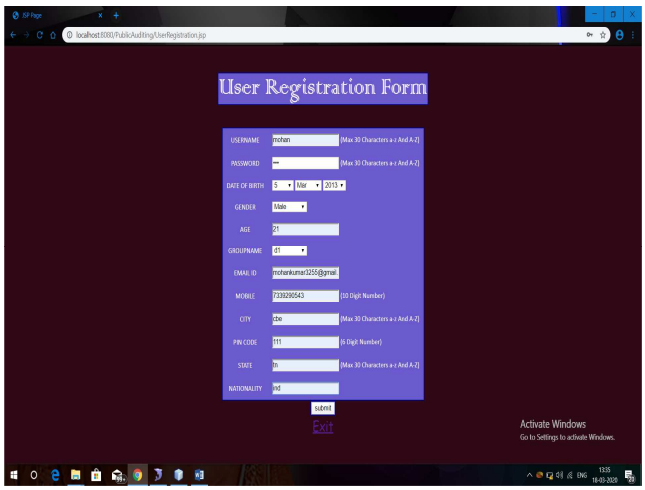


Fig.4: Registration page view

Fig.5 Audit page view

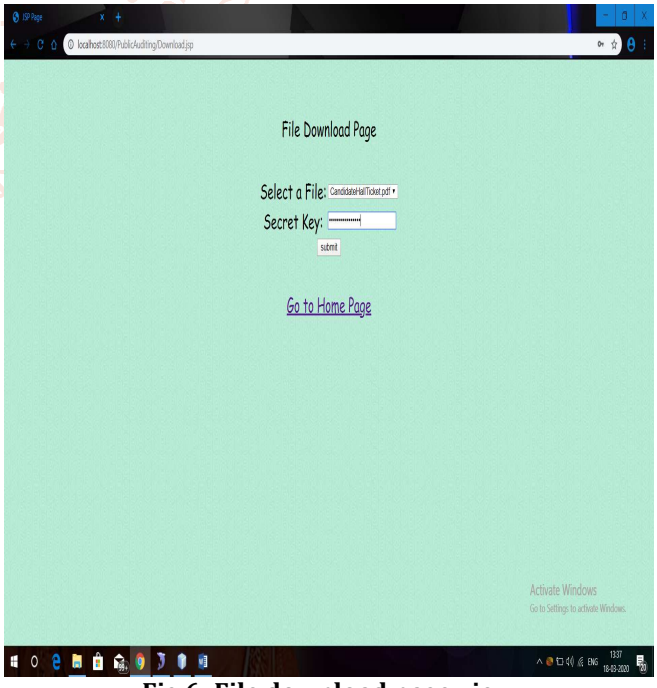


Fig.6: File download page view

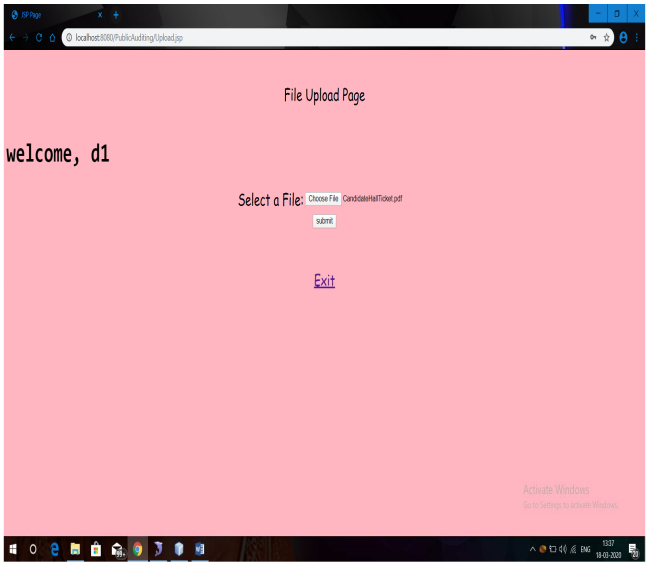


Fig.7: File upload page.

Conclusion and Future Enhancements:

In this paper(privacy preserving public auditing of shared data) we propose a solution for dynamic public auditing mechanism for a group of users they shared data in the cloud or other primary storages. In this mechanism ring signatures are used to construct authenticators, so that a public verifier is able to audit shared data and audit data without retrieving the entire data. To improve the efficiency[4] of verifying multiple auditing tasks, we further extend our mechanism to support batch auditing. There are some problems we will continue to study for our future work in this project. One of the process in our future work is traceability of user , which means the ability for the group manager (i.e., the original user) to retrive the upcoming activity of existing user based on verification metadata in some special situations. Since is based on ring signatures, where the identity of the signer is unconditionally protected, the current design of ours does not support traceability. With the best of our knowledge, designing an efficient public auditing mechanism with the capabilities of preserving identity privacy and supporting traceability is still open and anyone can able to make a future changes. And also one of our future work is how to prove data freshness (prove the cloud possesses the latest version of shared data) while still preserving identity privacy.

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.