User Level Runtime Security Auditing For The Cloud

A.Haneef[1], K.M.Mohammed Mufeeth[2], Allan Jerome[3]
[1, 2, 3]Dept of Computer engineering
[1, 2, 3]DR.MGR Educational And Research Institute, Chennai ,India

Abstract- *The advent of the cloud computing makes storage outsourcing becomes a rising trend that promotes the secure remote data auditing a hot topic that appeared within the analysis literature. Recently some analysis contemplates the problem of secure and economical public information integrity auditing for shared dynamic information. However, these schemes area unit still not secure against the collusion of cloud storage server and revoked cluster users throughout user revocation in practical cloud storage system. During this paper, we have a tendency to discover the collusion attack within the exiting theme and supply associate economical public integrity auditing theme with secure cluster user revocation primarily based on vector commitment and verifier-local revocation cluster signature. We style a concrete theme supported our theme definition. Our theme supports the general public checking and economical user revocation and additionally some nice properties, like with confidence, efficiency, count ability and traceability of secure cluster user revocation. Finally, the security and experimental analysis show that compared with its relevant themes our scheme is additionally secure and economical.*

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).In this paper, to solve the above privacy issue on shared data, we propose Oruta,1 a novel privacy-preserving public auditing mechanism. Public verifier is able to verify the integrity of shared data without retrieving the entire data while the identity of the signer on each block in shared data is kept private from the public verifier.

PROPOSE, SCOPE AND APPLICABILITY

1. user registration.
2. public auditing.
3. sharing data.
4. integrity checking.

II. LITERATURE SURVEY

1) **Cong Wang, *Student Member, IEEE,* Sherman S.-M. Chow, Qian Wang, *Student Member, IEEE,* Kui Ren, *Member, IEEE,* and Wenjing Lou, *Member, IEEE]***
2) **Efficient and Secure Multi-Keyword Search on Encrypted Cloud Data** (1**Y. Prasanna,** 2**Ramesh)**
3) **Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud(**Boyang Wang †,††, Baochun Li †† and Hui Li † † State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China †† Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada Email:{bywang,bli}@eecg.toronto.edu, lihui@mail.xidian.edu.cn)
4) **Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud** (Boyang Wang, Baochun Li, Member, IEEE, and Hui Li, Member, IEEE)
5) **Remote Data Checking for Network Coding-based Distributed Storage Systems(**Bo Chen, Reza Curtmola Department of Computer Science New Jersey Institute of Technology {bc47,crix}@njit.edu, Giuseppe Ateniese, Randal Burns Department of Computer Science Johns Hopkins University {ateniese, randal}@cs.jhu.edu)

III. PROPOSED SYSTEM

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.

Advanced Encryption Standard (AES):

The Advanced Encryption Standard (AES) is an encryption algorithm for securing sensitive but unclassified material by U.S. Government agencies and, as a likely consequence, may eventually become the de facto encryption standard for commercial transactions in the private sector. (Encryption for the US military and other classified communications is handled by separate, secret algorithms.)In January of 1997, a process was initiated by the National Institute of Standards and Technology (NIST), a unit of the U.S. Commerce Department, to find a more robust replacement for the Data Encryption Standard (DES) and to a lesser degree Triple DES. 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
- Twofish, 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), fingerscanning (fingerprint scanning),munging, insider threat, authentication server, defense in depth, nonrepudiation

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 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* (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 ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

HIGH-LEVEL DESCRIPTION OF THE ALGORITHM

1. KeyExpansion—round keys are derived from the cipher key using Rijndael's key schedule
2. Initial Round
    1. AddRoundKey—each byte of the state is combined with the round key using bitwise xor
3. Rounds

    1. SubBytes—a non-linear substitution step where each byte is replaced with another according to alookup table.
    2. ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
    3. MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
    4. AddRoundKey
4. Final Round (no MixColumns)
    1. SubBytes
    2. ShiftRows
    3. AddRoundKey

III. CONCLUSION

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

IV. 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.