RESEARCH ARTICLE OPEN ACCESS

Identity-Based Secure Data Storage Schemes

Unmesh Mhatre, Dhondiram Jadhav, Shrikant Bhirud, Sumit Desai
Prof. Sunil Jadhav
Department of Computer Science and Engineering
Y.T.I.E.T, Chandhai
Raigad, Mumbai
Maharashtra - India

ABSTRACT
Secure data storage can shift the burden of maintaining a large number of files from the owner to proxy servers. Proxy servers can convert encrypted files for the owner to encrypted files for the receiver without the necessity of knowing the content of the original files. In practice, the original files will be removed by the owner for the sake data must be addressed carefully. In this paper, we propose two identity-based secure data storage (IBSDS) schemes. Our schemes can capture the following properties: (1) The file owner can decide the access permission independently without the help of the private key generator (PKG); (2) For one query, a receiver can only access one file, instead of all files of the owner; (3) Our schemes are secure against the collusion attacks, namely even if the receiver can compromise the proxy servers, he cannot obtain the owner's secret key. Although the first scheme is only secure against the chosen plaintext attacks (CPA), the second scheme is secure against the chosen cipher text attacks (CCA).

Keywords:- PKG, CCS, IBSDS

I. INTRODUCTION

CLOUD computing provides users with a convenient mechanism to manage their personal files with the notion called database-as-a-service (DAS). In DAS schemes, a user can outsource his encrypted files to untrusted proxy servers. Proxy servers can perform some functions on the outsourced ciphertexts without knowing anything about the original files. Unfortunately, this technique has not been employed extensively. The main reason lies in that users are especially concerned on the confidentiality, integrity and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, as the cloud is managed by an untrusted third party. After outsourcing the files to proxy servers, the user will remove them from his local machine.

Therefore, how to guarantee the outsourced files are not accessed by the unauthorized users and not modified by proxy servers is an important problem that has been considered in the data storage research community. Confidentiality is proposed to prevent unauthorized users from accessing the sensitive data as it is subject to unauthorized disclose and access after being outsourced. Since the introduction of DAS, the confidentiality of outsourced data has been the primary focus among the research community. To provide confidentiality to the outsourced data, encryption schemes are deployed. Integrity can prevent outsourced data from being replaced and modified. Some schemes have been proposed to protect the integrity of the outsourced data, such as proof of irretrievability and provable data possession. In these schemes, digital signature schemes and message authentication codes (MACs) are deployed. Query in data storage is executed between a receiver and a proxy server. The proxy server can perform some functions on the outsourced ciphertexts and convert them to those for the receiver. As a result, the receiver can obtain the data outsourced by the owner without the proxy server knowing the content of the data.

II. EXISTING SYSTEM

Cloud computing provides users with a convenient mechanism to manage their personal files with the notion called database-as-a-service (DAS). In DAS schemes, a user can outsource his encrypted files to untrusted proxy servers. Proxy servers can perform some functions on the outsourced ciphertexts without knowing anything about the original files. Unfortunately, this technique has not been employed extensively. The main reason lies in that users are especially concerned on the confidentiality, integrity and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, as the cloud is managed by an untrusted third party.

After outsourcing the files to proxy servers, the user will remove them from his local machine. Therefore, how to guarantee the outsourced files are not accessed by the unauthorized users and not modified by proxy servers is an important problem that has been considered in the data storage research community. Furthermore, how to guarantee that an authorized user can query the outsourced files from proxy

servers is another concern as the proxy server only maintains the outsourced ciphertexts. Consequently, research around these topics grows significantly.

III. PROPOSED SYSTEM

In this paper, we propose two identity-based secure data storage (IBSDS) schemes in standard model where, for one query, the receiver can only access one of the owner's files, instead of all files. In other words, an access permission (re-encryption key) is bound not only to the identity of the receiver but also the file. The access permission can be decided by the owner, instead of the trusted party (PKG). Furthermore, our schemes are secure against the collusion attacks. There are four entities in an identity-based secure distributed data storage scheme: the private key generator, the data owner, the proxy server and the receiver. The PKG validates the users' identities and issues secret keys to them.

The data owner encrypts his data and outsources the ciphertexts to the proxy servers. Proxy servers store the encrypted data and transfer the ciphertext for the owner to the ciphertext for the receiver when they obtains an access permission (re-encryption key) from the owner. The receiver authenticates himself to the owner and decrypts the re-encrypted ciphertext to obtain the data. In this module, first the new data owner registers and then gets a valid login credentials. After logged in, the data owner has the permission to upload their file into the Cloud Server. The data owner encrypts his data and outsources the cipher texts to the proxy servers.

Fig: Data Owner

In this module, the private key generator (PKG) validates the users' identities and issues secret keys to them. The key is generated and sent to their respective mail id's with the file name and the corresponding key values.
Proxy servers store the encrypted data and transfer the cipher text from the owner to the receiver when they obtain access permission (re-encryption key) from the owner. In these systems, proxy servers are assumed to be trusted. They authenticate receivers and validate access permissions.

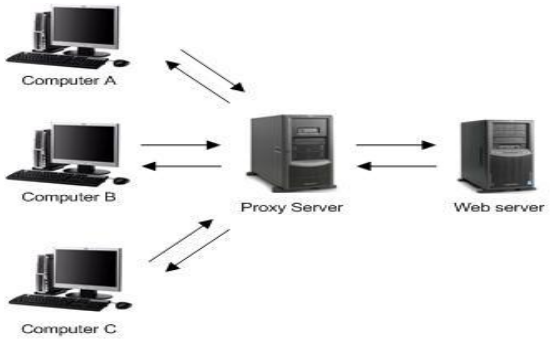

Fig: Proxy Server

IV. SYSTEM ARCHITECTURE

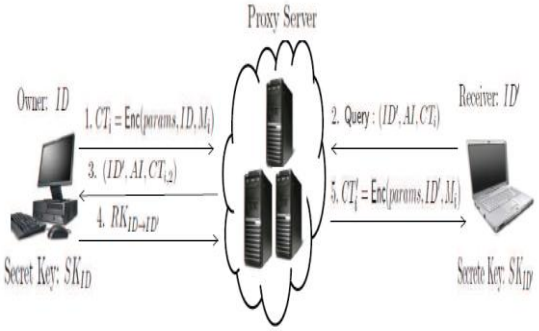

Fig:System Architecture

In this section, we propose an identity-based secure distributed data storage scheme which is secure against chosen plaintext attacks. At first, the file owner encrypts his files and outsources the ciphertexts to the proxy servers. The proxy servers validate the outsourced ciphertexts and store them for the owner. For one query, the receiver computes an authentication information (AI) using his secret key and sends it to the proxy server. The proxy server sends the identity of the receiver, AI and the partial intended ciphertext to the owner. Suppose that the owner can know which file the receiver wants to access from the partial ciphertext. To check whether the is a legal user in the system, the owner validates the received AI. If the AI is correct, the owner computes an access permission (re-encryption key) using his secret key, the partial ciphertext and the identity of the receiver, and sends it to the proxy server. Otherwise, the access is denied. The proxy server transfers the intended ciphertext to a ciphertext for the receiver using the received access permission.

V. DATA ENCRYPTION STANDARD

Data encryption is used pervasively in today's connected society. The two most basic facets of modern day data encryption are data privacy and authentication. As modern society becomes more connected, and more information becomes available there is a need for safeguards which bring data integrity and data secrecy. In addition, authenticating the source of information gives the recipient, with complete certainty that the information came from the original source and that it has not been altered from its original state. Both, the needs for information privacy and data authentication have motivated cryptography.
Cryptosystem or cipher system a method of disguising messages so that only certain people can see through the disguise.
Cryptography-The art of creating and using cryptosystems.

Cryptanalysis-The art of breaking cryptosystems, and seeing through the disguise even when you are not supposed to be able to.

Cryptology-The study of both cryptography and cryptanalysis.

Plaintext -The original message

Cipher text -The disguised message

Encryption -A fundamental security mechanism in which the ordinary data (plaintext) are transformed by the encryption process into cipher text.

Decryption - A procedure to convert ciphertext back into plaintext .The DES is based on the work of IBM Corporation, and was adopted as the American National Standard (ANSI) X3.92-1981/R1987.

The DES algorithm was adopted by the U.S. government in 1977, as the federal standard for the encryption of commercial and sensitive-yet-unclassified government computer data and is defined in FIPS 46 (1977). (FIPS are Federal Information Processing Standards published by NIST). A "block cipher" refers to a cipher that encrypts a block of data all at once, and then goes on to the next block. The DES, which is a block cipher, is the most widely known encryption algorithm. In block encryption algorithms, the plaintext is divided into blocks of fixed length which are then enciphered using the secret key. The DES is the algorithm in which a 64-bit block of plaintext is transformed (encrypted/enciphered) into a 64-bit cipher text under the control of a 56-bit internal key, by means of permutation and substitution. It has been mathematically proven that block ciphers are completely secure. The DES block cipher is highly random, nonlinear, and produces cipher text which functionally depends on every bit of the plaintext and the key. At least five rounds of DES are required to guarantee such dependence. In this sense, a product cipher should act as a "mixing" function which combines the plaintext, key, and cipher text in a complex nonlinear fashion. The DES has more than 72 quadrillion (72 x 1015) possible encryption keys that can be used. For each given message, the key is chosen at random, from among this enormous number of keys. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key. The process can run in several modes and involves 16 rounds or operations.

Triple-DES Algorithm

The DES algorithm is popular and in wide use today because it is still reasonably secure and fast. There is no feasible way to break DES, however because DES is only a 64-bit (eight characters) block cipher, an exhaustive search of 255 steps on average, can retrieve the key used in the encryption. For this reason, it is a common practice to protect critical data using something more powerful than DES.

A much more secure version of DES called Triple-DES (TDES), which is essentially equivalent to using DES three times on plaintext with three different keys. It is considered much safer than the plain DES and like DES,TDES is a block cipher operating on 64-bit data blocks. There are several forms, each of which use the DES cipher three times. Some forms of TDES use two 56-bit keys, while others use three. TDES can however work with one, two or three 56-bit keys. With one key TDES = DES. The TDES can be implemented using three DES blocks in serial with some combination logic or using three DES blocks in parallel. The parallel implementation improves performance and reduces gate count. Using standard DES encryption, TDES encrypts data three times and uses a different key for at least one of the three passes. The DES "modes of operation" may also be used with triple-DES. This 192-bit (24 characters) cipher uses three separate 64-bit keys and encrypts data using the DES algorithm three times. While anything less than that can be considered reasonably secure only the 192 bit (24 characters) encryption can provide true security. One variation that takes a single 192 bit (24 characters) key and then: encrypts data using first 64 bits (eight characters), decrypts same data using second 64 bits (eight characters), and encrypts same data using the last 64 bits (eight characters). For some time, it has been a common practice to protect and transport a key for DES encryption with triple-DES. This means that the plaintext is, in effect, encrypted three times. A number of modes of TDES have been proposed:

• DES-EEE3: Three DES encryptions with three different keys.

• DES-EDE3: Three DES operations in the sequence encrypt-decrypt-encrypt with three different keys.

• DES-EEE2 and DES-EDE2: Same as the previous formats except that the first and third operations use the same key.

Triple DES uses a "key bundle" that comprises three DES keys, K_1, K_2 and K_3, each of 56 bits (excluding parity bits).

The encryption algorithm is:
Cipher text = $E_{K3}(D_{K2}(E_{K1}(\text{plain text})))$i.e., DES encrypt with K_1, DES *decrypt* with K_2 then DES encrypt with K_3.
Decryption is the reverse: plaintext = $D_{K1}(E_{K2}(D_{K3}(\text{cipher text})))$ I.e., decrypt with K_3, encrypt with K_2, then decrypt with K_1.

Each triple encryption encrypts one block of 64 bits of data. In each case the middle operation is the reverse of the first and last. This improves the strength of the algorithm when using keying option 2, and provide backward compatibility with DES with keying option 3.
The standards define three keying options:
Keying option 1: All three keys are independent.
Keying option 2: K_1 and K_2 are independent, and $K_3 = K_1$.
Keying option 3: All three keys are identical, i.e. $K_1 = K_2 = K_3$.
Keying option 1 is the strongest, with $3 \times 56 = 168$ independent key bits.
Keying option 2 provides less security, with $2 \times 56 = 112$ key bits. This option is stronger
than simply DES encrypting twice, e.g. with K_1 and K_2, because it protects against meet-in-the-middle attacks. Keying option 3 is equivalent to DES, with only 56 key bits. This option provides backward compatibility with DES, because the first and second DES operations cancel out.

This core is a full implementation of the TDES encryption algorithm. Both encryption and decryption are supported. The TDES algorithm was proposed by IBM Corporation when it became clear that the security of the DES had been compromised by advances in computer technology. Compared to the DES algorithm, the TDES provides a significantly higher level of security. Each TDES encryption/decryption operation (as specified in ANSI X9.52) is a compound operation of the DES encryption and decryption operations. The X_3DES triple DES core supports the most popular ECB mode, while providing customization for other modes as well. The TDES algorithm coincides with the DES algorithm, providing backward compatibility.

VI. CONCLUSIONS

Data storage schemes provide the users with convenience to outsource their files to untruhsasted proxy servers. Identity-based secure distributed data storage (IBSDS) schemes are a special kind of distributed data storage schemes. In IBSDDS users are identified by their identities and can communicate without the need of verifying the public key certificates. The future of IBSDDS is very good, as it provides data integrity and confidentiality. The owners have the full access on sharing of file. It also secures the data from unauthorized access and collusion attacks.

ACKNOWLEDGMENT

It is matter of great satisfaction and pleasure to present seminar on **"IDENTITY-BASED SECURE DATA STORAGE SCHEMES".** We wish to express our sincere thanks to **H.O.D. Mrs. Prof. Nilima Nikam** who extended their valuable support during the course of seminar. We wish to express our sincere thanks and gratitude to our honorable guide **Mr. Prof. Sunil Jadhav** for his constant guidance and motivation. We also thank her for her valuable support and encouragement throughout the preparation of seminar without which the seminar would have not been completed. We also thank our colleagues who have helped in successful completion of the seminar. Last but not least we would like to thank all our friends, who helped us not directly or indirectly. Helpful hand rendered by all of them will remain for long time in our memory. Finally we admit the cooperation, coordination & hard work are our keywords for success.

REFERENCES:

[1] Jinguang Han, Student Member, IEEE, Willy Susilo, Senior Member, IEEE, and Yi Mu, Senior Member, IEEE-"Identity-Based Secure Distributed Data Storage Schemes" ieee transactions on computers, 2013.

[2] H. Hacig¨um¨us, B. R. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model," in Proceedings: SIGMOD Conference - SIGMOD'02 (M. J. Franklin, B. Moon, and A. Ailamaki, eds.), vol. 2002, (Madison, Wisconsin, USA), pp. 216–227, ACM, Jun. 2002.

[3] L. Bouganim and P. Pucheral, "Chip-secured data access: Confidential data on untrusted servers," in Proc. International Conference on Very Large Data Bases - VLDB'02, (Hong Kong, China), pp. 131– 142, Morgan Kaufmann, Aug. 2002.

[4] U. Maheshwari, R. Vingralek, and W. Shapiro, "How to build a trusted database system on untrusted storage," in Proc. Symposium on Operating System Design and Implementation - OSDI'00, (San Diego, California, USA), pp. 135–150, USENIX, Oct. 2000.

[5] A. Ivan and Y. Dodis, "Proxy cryptography revisited," i+n Proc. Network and Distributed System Security Symposium - NDSS'03, (San Diego, California, USA), pp. 1–20, The Internet Society, Feb. 2003.