Volume 5, Issue 1, January 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Permit Access Control of Data Stored In Cloud to Authenticate Users

Velampudi Anusha, Anitha Jalumuri, Amarendra Kothalanka
Department of Computer Science & Engineering,
Dadi Institute of Engineering & Technology,
Anakapalli, A.P., India

Abstract: ***In cloud computing storing data into cloud and retrieving data from the cloud with more securely is not a simple task and providing security to stored data is very difficult. This paper discusses how to overcome these problems. Before storing data the cloud must verify the authenticity of the series, without knowing the user's identity it should not allow user to operate on the data. In this paper we are proposing concepts of authentication of users, secret key sharing, and encryption of data and decryption of data using cryptography technique. The scheme prevents replay attacks and supports creation, modification, and reading data stored in the cloud. In this approach we are also provide user revocation and prevent replay attack. So that our proposed authentication process and access control schemes are decentralized and robust. By using those techniques the computation, communication and storage are more effective than the centralized approach.***

Key words: Encryption, Decryption, Shamir secret sharing scheme

I. INTRODUCTION

Research in cloud computing is receiving a lot of attention from both academic and industrial worlds. In cloud computing, users can outsource their computation and storage to servers (also called clouds) using Internet. This frees users from the hassles of maintaining resources on-site. Clouds can provideseveral types of services like applications (e.g., Google Apps, Microsoft online), infrastructures (e.g., Amazon's EC2, Eucalyptus, Nimbus), and platforms to help developers write applications (e.g., Amazon's S3, Windows Azure). Authentication of users using public key cryptographic techniques has been studied in many homomorphic encryption techniques have been suggested to ensure that the cloud is not able to read the data while performing computations on them. Using homomorphic encryption, the cloud receives cipher text of the data and performs computations on the cipher text and returns the encoded value of the result. The user is able to decode the result, but the cloud does not know what data it has operated on. In such circumstances, it must be possible for the user to verify that the cloud returns correct results.

II. EXISTING SYSTEM

Much of the data stored in clouds is highly sensitive, for example, medical records and social networks. Security and privacy are, thus, very important issues in cloud computing. In one hand, the user should authenticate itself before initiating any transaction, and on the other hand, it must be ensured that the cloud does not tamper with the data that is outsourced. User privacy is also required so that the cloud or other users do not know the identity of the user. The cloud can hold the user accountable for the data it outsources, and likewise, the cloud is itself accountable for the services it provides. The validity of the user who stores the data is also verified. Apart from the technical solutions to ensure security and privacy, there is also a need for law enforcement. Recently, Wang et al. addressed secure and dependable cloud storage. Cloud servers prone to Byzantine failure, where a storage server can fail in arbitrary ways. The cloud is also prone to data modification and server colluding attacks. In server colluding attack, the adversary can compromise storage servers, so that it can modify data files as long as they are internally consistent. To provide secure data storage, the data needs to be encrypted.However, the data is often modified and this dynamicproperty needs to be taken into account while designingefficient secure storage techniques.

III. PROPOSED SYSTEM

As we know cloud computingis playing a vital role in storage of data into cloud. We see the rise in demand for data out sourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. So that storing data increasing day by day and also provide security of that data. In this paper we propose the technique for providing authentication users, security of data and support for creation, modification and reading data stored into cloud.

Signature Generation

This module is used for the generation of signature for the purpose authentication of users. The generation of signature we are using digital signature algorithm for users' authentication. The process of digital signature algorithm as follows.

Steps:

1. User chooses a large prime P with the range between 512 to 1024 and multiple of 64.
2. User chooses q with the range of 160 bit divisor by (P-1).
3. Calculate g=h (P-1)/q.
4. User randomly select x greater than q.
5. User calculate public key $y=g^x$ % q.
6. Calculate r = (g^k % p)% q
7. User generate signature s == [k^{-1}(H(M)+ xr)] % q.

After generating signature each and every user sent the signature to trusted party.

Users Authentication

This module is used for performing authentication of user can be done by trusted center. The trusted center retrieves the signature from the users and generate each user signature compare it. If both signatures are equal they are the authenticated user. If the signature are not equal they are not authenticate users.

The authentication process as follows.

1. Trusted party will calculate $w=s^{-1}$ % q.
2. Trusted party will calculate u1= [H(M)w]% q and u2= (rw)% q.
3. v = [(g^{u1} y^{u2})% p]%q

If both v and s are equal they are authenticated user.

IV. GENERATE SHARED KEY

After completion of authentication the trusted center will generate shared key and sent to each user. The generation of secret key trusted center will use Shamir secret key and Lagrange polynomial equation. The process of Shamir and Lagrange polynomial equation as follows.

1. The trusted center will choose a random secret key k and select two random numbers.
2. After completion of choosing number the trusted center will generate polynomial equation.
3. After generating polynomial equation the trusted center will divide secret key into 6 parts.
4. The trusted center will sent any subset of parts into users.
5. After retrieving three parts the users reconstruct secret key using those three parts. By reconstruction of secret key we are using Lagrange polynomial equation.

Shamir secret Sharing scheme:

The essential idea of Adi Shamir's threshold scheme is that 2 points are sufficient to define a line, 3 points are sufficient to define a parabola, 4points to define a cubic curve and so forth. That is, it takes points to define a polynomial of degree k-1.Suppose we want to use a(k ,n) threshold scheme to share our secret , without loss of generality assumed to be an element in a finite field F of size P where 0<k<=n<P;S<P and P is a prime number.Choose at random k-1 positive integers a1,.....,ak-1 with ai<P, and let a0=S. Build the polynomial f(x)=a_0+a_1x+a_2x^2+a_3x^3+.....+a_{k-1}x^{k-1}. Let us construct any n points out of it, for instance set i=1....n to retrieve (i ,f(i)). Every participant is given a point (an integer input to the polynomial, and the corresponding integer output). Given any subset of k of these pairs, we can find the coefficients of the polynomial using interpolation. The secret is the constant term a0.After generation of points the trusted party will send the all user and the user will generate secreat key using these points. So that the user will use these secreat key to encrypt the store data. As well the user also retrieve the data with cipher and decrypt using these secreat key.

V. DESCRIPTION OF IDEA

The block cipher IDEA operates with 64-bit plaintext and cipher text blocks and is controlled by a 128-bit key. The fundamental innovation in the design of this algorithm is the use of operations from three different algebraic groups. The substitution boxes and the associated table lookups used in the block ciphers available to-date have been completely avoided. The algorithm structure has been chosen such that, with the exception that different key sub-blocks are used, the encryption process is identical to the decryption process.

Key Generation

The 64-bit plaintext block is partitioned into four 16-bit sub-blocks, since all the algebraic operations used in the encryption process operate on 16-bit numbers. Another process produces for each of the encryption rounds, six 16-bit key sub-blocks from the 128-bit key. Since a further four 16-bit key-sub-blocks are required for the subsequent output transformation, a total of 52 (= 8 x 6 + 4) different 16-bit sub-blocks have to be generated from the 128-bit key.The key sub-blocks used for the encryption and the decryption in the individual rounds are shown in Table 1.

Table 1 Encryption of the key sub-blocks

Round 1	$Z_1^{(1)}\ Z_2^{(1)}\ Z_3^{(1)}\ Z_4^{(1)}\ Z_5^{(1)}\ Z_6^{(1)}$
Round 2	$Z_1^{(2)}\ Z_2^{(2)}\ Z_3^{(2)}\ Z_4^{(2)}\ Z_5^{(2)}\ Z_6^{(2)}$
Round 3	$Z_1^{(3)}\ Z_2^{(3)}\ Z_3^{(3)}\ Z_4^{(3)}\ Z_5^{(3)}\ Z_6^{(3)}$
Round 4	$Z_1^{(4)}\ Z_2^{(4)}\ Z_3^{(4)}\ Z_4^{(4)}\ Z_5^{(4)}\ Z_6^{(4)}$
Round 5	$Z_1^{(5)}\ Z_2^{(5)}\ Z_3^{(5)}\ Z_4^{(5)}\ Z_5^{(5)}\ Z_6^{(5)}$
Round 6	$Z_1^{(6)}\ Z_2^{(6)}\ Z_3^{(6)}\ Z_4^{(6)}\ Z_5^{(6)}\ Z_6^{(6)}$
Round 7	$Z_1^{(7)}\ Z_2^{(7)}\ Z_3^{(7)}\ Z_4^{(7)}\ Z_5^{(7)}\ Z_6^{(7)}$
Round 8	$Z_1^{(8)}\ Z_2^{(8)}\ Z_3^{(8)}\ Z_4^{(8)}\ Z_5^{(8)}\ Z_6^{(8)}$
Output Transform	$Z_1^{(9)}\ Z_2^{(9)}\ Z_3^{(9)}\ Z_4^{(9)}$

The 52 16-bit key sub-blocks which are generated from the 128-bit key are produced as follows:

1. First, the 128-bit key is partitioned into eight 16-bit sub-blocks which are then directly used as the first eight key sub-blocks.
2. The 128-bit key is then cyclically shifted to the left by 25 positions, after which the resulting 128-bit block is again partitioned into eight 16-bit sub-blocks to be directly used as the next eight key sub-blocks.
3. The cyclic shift procedure described above is repeated until all of the required 52 16-bit key sub-blocks have been generated.

Encryption

The functional representation of the encryption process is shown in Figure 1. The process consists of eight identical encryption steps (known as encryption rounds) followed by an output transformation. The structure of the first round is shown in detail.

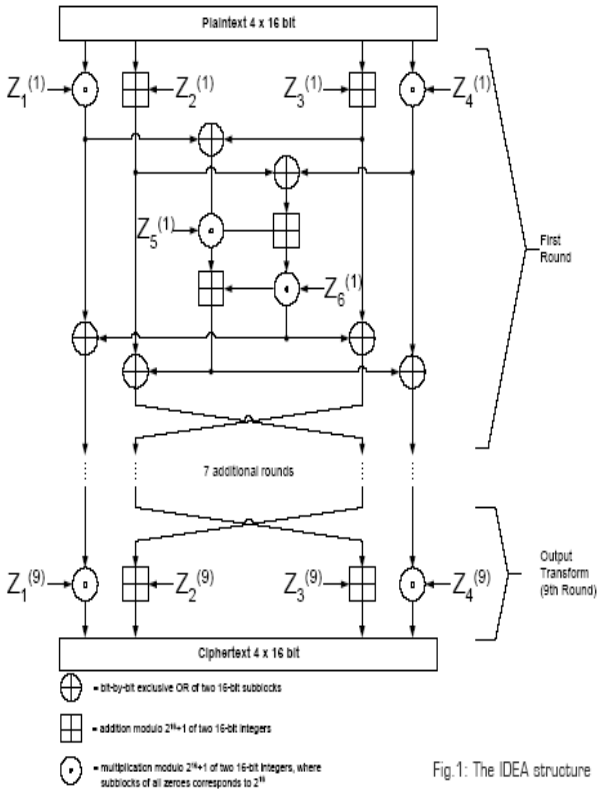


Fig.1: The IDEA structure

In the first encryption round, the first four 16-bit key sub-blocks are combined with two of the 16-bit plaintext blocks using addition modulo 2^{16}, and with the other two plaintext blocks using multiplication modulo $2^{16} + 1$. The results are then processed further as shown in Figure 1, whereby two more 16-bit key sub-blocks enter the calculation and the third algebraic group operator, the bit-by-bit exclusive OR, is used. At the end of the first encryption round four 16-bit values are produced which are used as input to the second encryption round in a partially changed order. The process described above for round one is repeated in each of the subsequent 7 encryption rounds using different 16-bit key sub-blocks for each combination. During the subsequent output transformation, the four 16-bit values produced at the end of the 8^{th} encryption round are combined with the last four of the 52 key sub-blocks using addition modulo 2^{16} and multiplication modulo $2^{16} + 1$ to form the resulting four 16-bit ciphertext blocks.

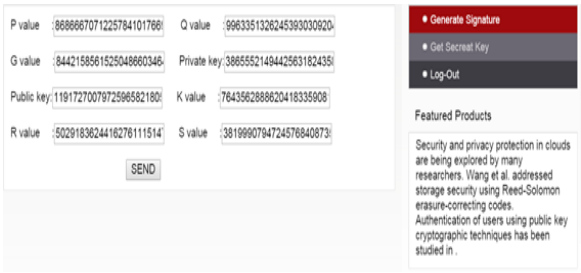
Decryption

Table 2 Decryption of the key sub-blocks

Round 1	$Z_1^{(9)-1}$ -$Z_2^{(9)}$ -$Z_3^{(9)}$ $Z_4^{(9)-1}$ $Z_5^{(8)}$ $Z_6^{(8)}$
Round 2	$Z_1^{(8)-1}$ -$Z_3^{(8)}$ -$Z_2^{(8)}$ $Z_4^{(8)-1}$ $Z_5^{(7)}$ $Z_6^{(7)}$
Round 3	$Z_1^{(7)-1}$ -$Z_3^{(7)}$ -$Z_2^{(7)}$ $Z_4^{(7)-1}$ $Z_5^{(6)}$ $Z_6^{(6)}$
Round 4	$Z_1^{(6)-1}$ -$Z_3^{(6)}$ -$Z_2^{(6)}$ $Z_4^{(6)-1}$ $Z_5^{(5)}$ $Z_6^{(5)}$
Round 5	$Z_1^{(5)-1}$ -$Z_3^{(5)}$ -$Z_2^{(5)}$ $Z_4^{(5)-1}$ $Z_5^{(4)}$ $Z_6^{(4)}$
Round 6	$Z_1^{(4)-1}$ -$Z_3^{(4)}$ -$Z_2^{(4)}$ $Z_4^{(4)-1}$ $Z_5^{(3)}$ $Z_6^{(3)}$
Round 7	$Z_1^{(3)-1}$ -$Z_3^{(3)}$ -$Z_2^{(3)}$ $Z_4^{(3)-1}$ $Z_5^{(2)}$ $Z_6^{(2)}$
Round 8	$Z_1^{(2)-1}$ -$Z_3^{(2)}$ -$Z_2^{(2)}$ $Z_4^{(2)-1}$ $Z_5^{(1)}$ $Z_6^{(1)}$
Output Transform	$Z_1^{(1)-1}$ -$Z_2^{(1)}$ -$Z_3^{(1)}$ $Z_4^{(1)-1}$

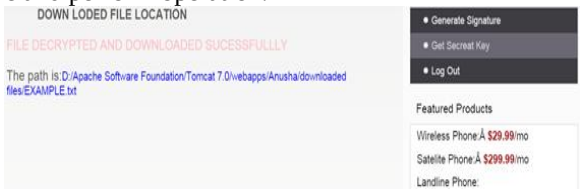
The computational process used for decryption of the ciphertext is essentially the same as that used for encryption of the plaintext. The only difference compared with encryption is that during decryption, different 16-bit key sub-blocks are generated.More precisely, each of the 52 16-bit key sub-blocks used for decryption is the inverse of the key sub-block used during encryption in respect of the applied algebraic group operation. Additionally, the key sub-blocks must be used in the reverse order during decryption in order to reverse the encryption process as shown in Table 2.

V. EXPERIMENTAL RESULS



This figure shows generated points of individual user which are sent to administrator. This points are used to generate secrete key.

Here administrator sends a secret key to user by which user can access data. Here encrypt as well as decrypt options are seen by which user can encrypt file and perform operation.



This figure shows the path of file which is decrypted by user. By clicking on path that particular file will be opened.

VII. CONCLUSION

The storing of data in the cloud is place a vital role for the purpose giving security of data. Another problem for facing in the cloud computing is the authentication of users. In this paper we are using generation signature for user authentication are used the technique for digital signature algorithm. In this we are using another technique for generation of secret key for the encryption and decryption of transmitted data. For the generation key we are using Shamir secret key and langrage's polynomial equation. The encryption and decryption of transmitted data we are using IDEA algorithm. Afterperforming encryption of data stored into cloud in the form encrypted format. By providing those technique we are provide more security and efficiency for transferring data.

REFERENCES

[1] S. Ruj, M. Stojmenovic, and A.Nayak, "Privacy Preserving AccessControl with Authentication for Securing Data in Clouds," Proc. IEEE/ACM Int'l Symp. Cluster, Cloud and Grid Computing, pp. 556- 563, 2012.An Insight to OAD (Book with CD)

[2] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-Based Authentication for Cloud Computing," Proc. First Int'l Conf. Cloud Computing (CloudCom), pp. 157-166, 2009.

[3] A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-Based Cloud Computing," Proc. Third Int'l Conf. Trust and Trustworthy Computing (TRUST), pp. 417-429, 2010

[4] R. Lu, X. Lin, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," Proc. Fifth ACM Symp. Information, Computer and Comm. Security (ASIACCS), pp. 282-292, 2010.

[5] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. ACM Symp.Information, Computer and Comm. Security (ASIACCS), pp. 261-270, 2010.

[6] F. Zhao, T. Nishide, and K. Sakurai, "Realizing Fine-Grained andFlexible Access Control to Outsourced Data with Attribute-Based Cryptosystems," Proc. Seventh Int'l Conf. Information Security Practice and Experience (ISPEC), pp. 83-97, 2011.

[7] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-Based Access Control in Social Networks with Efficient Revocation," Proc. ACMSymp. Information, Computer and Comm. Security (ASIACCS), 2011.

