International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 11 (2015) pp. 28875-28887

http://www.ripublication.com

Data Security on Cloud: A Survey, Research Issues and Challenges

Deeksha Vimmadisetti
vdeeksha23@gmail.com
Tushar Sharma
tusharsharma3@outlook.com
Ashwini Bhaskar
ashwinibhaskar.ashu@gmail.com
Kavitha C R
Assistant Professor, cr_kavitha@blr.amrita.edu
Dept. of Computer Science and Engineering,
Amrita VishwaVidyapeetham, Kasavanahalli,
Carmelaram P.O., Bangalore - 560035, Karnataka, India

Abstract

Cloud computing is a recent emergent technology that is almost used in all fields. It can be used to resolvethe hardware, software and storage availability problems that we face day to day. Every application in the current world is linked to the cloud.Though cloud computing and virtualization of the systems, help organizations to break the barriers between the end users and architecture, there is always a danger of data being tampered and misused by unauthorized entities. Hence to make cloud computing more efficient and secure, it is necessary to provide strong security to data by using some encryption techniques.

In this paper, we discuss the various symmetric and asymmetric algorithms such as RSA, ECC, Diffie-Hellman, RC5, Homomorphic Encryption and AES that can be implemented in order to provide security for data stored on the cloud. We also discuss the KNN classifier that is used to ensure the data confidentiality.

Keywords: 'Data Security', 'Cloud Computing', 'Encryption and Decryption', 'KNN Classifier', 'RSA', 'Diffie-Hellman', 'AES', 'RC5', 'ECC', 'Homomorphic Encryption'

Introduction

Every application in the current world is linked to the cloud. Cloud computing has become an inevitable part of every organization. The hardware, software and storage

availability problems that we face can be resolved by using Cloud computing. It involves the sharing of computing resources rather than having them on every local PC. Cloud services also deliver infrastructure, software and storage based on the user demands, over the internet.

All these services face the problem of data integrity, authentication and confidentiality. Since the cloud security concern has been recognized by the cloud service providers and the data users, many encryptionalgorithms and techniques are being implemented to keep the data secured [3][8].

Many users save their private data on the cloud. There is a need to encrypt this sensitive data before outsourcing it to the cloud server. The outsourced encrypted files stored on the cloud can be accessed using data structures called as Block Status Table (BST). Linked lists are used to implement BST. The process of securely storing the data on cloud is as follows:

- The data owner divides the data into smaller blocks of 128 characters and encrypts them using an efficient encryption algorithm and prepares the Block Status Table for this encrypted block and sends it to the Trusted Third Party (TTP) auditor [1].
- Combined hash value is calculated for the BST and the encrypted file by the TTP and only the encrypted file and BST are sent to the cloud for storage.
- When an authorized user wants to access the data, a request has to be sent to both TTP and the cloud service provider.
- The authenticity of the user is verified by TTP and once the user identity is validated, the TTP sends authorization signal to the cloud service provider.
- The hashed value of the BST and encrypted file are sent to the user by TTP.
- The BST and the encrypted file are sent to the user by the cloud provider.
- The hash value of the BST and encrypted file received from the cloud server is calculated by the user and it is verified with the hash values received from the TTP.
- Finally, ifvalues are matched, the decryption key is obtained by the user and the data blocks are decrypted. [1]

KNN Classifier For Data Confidentiality

Providing security for data stored on the cloud has always been a challenging threat for being able to provide quality of service and can lead to people not wanting to use the cloud services.

Usually, there are two ways by which data is stored in the cloud:

- Store data with encryption
- Store data without encryption

Both methods face data confidentiality issues as the customer is not sure where his data is being stored on the vast cloud.

The method that has been proposed reduces computing time by a large amount because it follows the simple concept of Classification, or supervised learning. The data is classified into two distinct classes:

- Sensitive data
- Non-sensitive data

This classification is done using the K Nearest Neighbor Algorithm wherein the distances between neighborhood data is taken into consideration when classifying.

The sensitive data is then encrypted using an RSA encryption algorithm and then stored on a virtual machine. The non-sensitive data is directly stored onto a virtual machine after being classified so.

A. The KNN Classifier

The K nearest neighbor (K-NN) is a supervised machine learning technique that is used in various fields such as pattern recognition, prediction and estimation.

The classifier itself works on fundamental recognition problems. Using a large K could lead to pixels being chosen that aren't relevant; on the other hand, picking a K that is too small may lead to pixels being left out.

The K-NN algorithm has a set of n labelled samples; n is the number of data items in the set. This is shown as:

T = {t_1 , t_2 , t_3 ,..., t_n } where D is the set of total samples and t_1 , t_2 , t_3 ,..., $t_n$are different samples. The T must be assigned n labels. The set of n labelled samples can be represented as:

T = {t_1, t_2, t_3... t_n | c} Where c is a class for the targeted values. In algorithm we specified only one class for confidential attributes. [2]

B. Working of the Algorithm

Step 1: Determine the set of n labelled samples: D
Step 2: Determine value of K
Step 3: Calculate the distance between the new input and all ofthe training data
Step 4: Sort the distance and determine the K-nearest neighbors based on the K^{th} minimum distance
Step 5: Find the classes of those neighbors
Step 6: Determine the class of the new input based on a majority vote.

Digital Signature With RSA Encryption

One of the techniques used to ensure security of data on the cloud is to use digital signatures with RSA encryption.

The mathematical technique used to validate the authenticity and integrity of a message document is called as a digital signature. A valid digital signature tells the receiver that the message was sent by the known sender. The software at the sender hashes the message to be sent and compresses it. This compressed message is called as the message digest. The sender then encrypts this message digest using his/her private key. This is the digital signature. The software at the receiver decrypts the message and validates the signature using the public key of the sender. Here digital signatures are used so that it is possible to distribute the software and other transactions over the network.

RSA is an asymmetric algorithm that is used for public and private key encryption and generation.

Here both of the above are used to increase the security of data stored in the cloud.

A. Working of the Algorithm

Step 1: Algorithm to generate the key

- Two distinct large prime numbers k and l are chosen.
- The modulus for the public and private key , n, is calculated by z = k * l
- Quotient is computed Phi(z) = (k-1) * (l-1)
- An integer e_1 which is the encryption exponent is chosen such that it falls in the range of 1 and phi(z) and e and phi(z) share no common factors other than 1
- The decryption exponent d_1 , which is kept as the private exponent, is calculated such that D * e_1 = 1 * modulus * phi(z)
- All the values d, k, l and phi are kept secret. The public key generated is (z,e_1) and the private key is (z,d_1) [3]

Step 2: Digital Signing by the sender

- Message digest is created by the sender, using a hash function
- This digest m is represented as an integer between 0 and z-1
- The private key (z,d_1) is used to compute the signature s,
- $S = m^{d}_{1} \bmod(z)$ and this is sent to the receiver

Step 3: Encryption

- The public key (z,e_1) of the recipient is obtained
- The plain text is represented as a positive integer m
- Cipher text is computed by encrypting m with the public key of the recipient $c = m^{e}_{1} \bmod (z)$
- c is then sent to the recipient B

Step 4: Decryption

- Private key (z,d_1) is used by recipient B to compute

  $m = c^{d}_{1} \bmod (z)$ (1)

- Hashed message m is used to extract the plain text

Step 5: Verification of the Signature

- The public key (z,e_1) of the sender is used to compute integer

  $v = s^{e}_{1} \bmod (z)$ (2)

- This integer v, is used to extract the message digest
- A message digest is computed for the signed information by B
- The signature is considered to be valid if both the message digests are identical[3]

ECC Algorithm

The data stored in the cloud is being accessed from mobile devices. But these small computing devices need an algorithm that can provide equivalent amount of security using less battery resources, run time memory and processor speed.

Elliptic Curve Cryptography (ECC) is used to implement public-key cryptography technique. Public Key algorithms use a pair of keys (public key and private key) for encryption and decryption purpose. These keys are distributed among many entities in a large system in order to exchange information. The public key is known to all users whereas the private key is known only to one a particular user. Basis of ECC is discrete logarithms.

A. Working of the Algorithm

An elliptic curve is a cubic curve consisting of a set of points that satisfy a mathematical equation and is represented by a curve intersecting two axes. ECC algorithm is based on the concept of elliptic curves and is used to generate efficient cryptographic keys that are smaller and in a shorter period of time. In this algorithm, the properties of the elliptical curve equation are used to generate the key instead of using product of prime numbers as in most of the traditional encryption techniques. ECC is based on the properties of an equation which is derived from a set of points where the line intersects with the axes.

Elliptical curves have properties that are easy to perform but difficult to reverse. Hence they can be of great use in cryptography.

The ECC algorithm provides security by computing a point that is new on the curve using the product points and encrypts this new point as the information to be exchanged between the two entities. Since this algorithm involves small computations, it can be evaluated in polynomial time.

The equation of an elliptical curve is given as:

$$Y^2 = X^3 + aX + b \tag{3}$$

Step 1: Key Generation

The sender uses the public key of the receiver to encrypt the message while storing it on the cloud and the receiver uses his private key to decrypt it. The public key K can be generated using the equation:

$$K = e^* R \tag{4}$$

Where, e is the private key which is a number chosen between ranges of 1 and n-1, R is any point on the elliptical curve.

Step 2: Encryption

If M is the message that has to be sent and it represents a point on the elliptical curve E, two cipher texts T1 and T2 can be generated. Consider k to be in the range of 1 and n-1, then

$$T1 = k * R \tag{5}$$

$$T2 = M + (k * R) \tag{6}$$

Step 3: Decryption

The receiver can get back the initially sent message M by doing the following action

$$M = T1 – (e * T2) \quad (7)$$

Thus the original message is obtained. [4]

V. Diffie - Hellman key exchange and Two Factor authentication

Many users upload private and confidential data to the cloud. Efficient security must be provided so that the users will not be worried about the data confidentiality. There is always a threat of unauthorized users accessing this data. Diffie-Hellman key exchange protocol can be used to securely exchange the symmetric keys over an insecure public channel. The key generated by this algorithm will be used to do the two-factor authentication.

When data has to be transferred in an insecure medium, this algorithm can be used to provide security to the data that is stored on the cloud. The Diffile-Hellman algorithm is based on the concept of modulus (say 35 mod 10 is 5). It is easy to generate the powers modulo a prime number but reversing the process is a difficult task. If we run 2^2mod 11, 2^3mod 11 and so on, 11 is the prime number and 2 is called as the generator.

A. Working of the Algorithm

Assuming that users A and B have to exchange data over an insecure medium:

Step1:

Users A and B decide on a prime number p_1 and a generator g_1

Step 2:

A then chooses a number XA < p_1 and similarly B chooses an integer XB < p_1. XA and XB are the private keys of A and B respectively.

Step 3:

A's public key is computed by $p_A \equiv g_1{}^{x}{}_A \pmod{p_1}$ and it is sent it to B using insecure communication. B also computes public key $p_B \equiv g_1{}^{x}{}_B$ and sends it to A (0 <p_A < p_1, 0 <p_B < p_1). These public keys can be sent over an insecure medium as it is very hard to compute the private keys from these public keys.

Step 4:

A computes $z_A \equiv p_B{}^{x}{}_A \pmod{p_1}$ and B computes $z_B \equiv p_A{}^{x}{}_B \pmod{p_1}$. Here z_A <p_1, z_B <p_1. But $z_A = z_B$, since $z_A \equiv p_B{}^{x}{}_A \equiv (g_1{}^{x}{}_B)^{x}{}_A = g_1{}^{(x}{}_A{}^{x}{}_B{}^{)} \pmod{p_1}$ and similarly $z_B \equiv (g_1{}^{x}{}_A)^{x}{}_B = g_1{}^{(x}{}_A{}^{x}{}_B{}^{)} \pmod{p_1}$. So this is the shared key that is private, which can be used to encrypt and decrypt the data when it is stored and retrieved from the cloud. [5]

B. Client-Server System

This concept can also be extended to a client-server based storage of data on the cloud.There are two major entities in this system - the data owner, who is the user,

and the cloud service providers who have a very large storage space and computational speed. Cryptographic algorithm like Diffie-Hellman key exchange and Symmetric key encryption is used to make communication secure between the data owner and the service provider. Diffie-Hellman is used to exchange key between the data owner and cloud service provider and this eliminates the issue of over load for key distribution.

Step 1: Creating a user account

To securely upload the data onto the cloud, the data owner must first create an account on the cloud by entering important information (username, password, mobile number). To authenticate the user, a message is sent to the user's phone containing the key generated by the Diffie-Hellman algorithm. This key is valid only for a specific amount of time.This has to be entered by the user while creating an account. Once this password and the one generate by the server match, then an account is created and user can utilize the benefits offered by the service provider.

Step 2: Using the cloud service

When the user has to login to his/her account, the username and password have to be provided. After these credentials are verified with the information stored in the database, a message is sent to the user's phone containing a key generated by the Diffie-Hellman algorithm. The user then has to enter the key he/she has received and after the key is match with the one that was generated, the user is granted access.

Step 3: Encryption and Decryption

The data that is stored in the cloud by the user is kept encrypted using the same key generated by the Diffie-Hellman algorithm. And similarly, decryption can be done with the same key after verifying the authenticity of the user. This data is secure on the cloud. [5]

RC5 Algorithm

RC5 is a fast symmetric block cipher and its interesting feature is that it makes use of data-dependent rotations. The complete name of the algorithm is RC5-w/r/b. It consists of a variable word size, variable number of rounds and a variable length secret key. The word size in bits is denoted by 'w', the value is either 16 or 32 or 64 bits, the number of rounds is denoted by 'r' and its value ranges from 0 – 255, and the length of the user's secret key is denoted by 'b', which is in bytes and the value ranges from 0 – 255.

A. Working of the Algorithm

The RC5 algorithm is composed of three parts:-

1. Key Expansion Algorithm
2. Encryption Algorithm
3. Decryption Algorithm

Step 1: Key Expansion Algorithm

A key of b bytes is supplied by the user,

Considering, a secret key = K [0-(b-1)]

An array = A [0 – (c-1)],

Where,

c = ceil (b/o) ando = w/8 (o is in little-endian order). The secret key is copied into the array. The empty byte positions, i.e., the byte positions that aren't filled are filled with zeros.

In case,

b = c

c = 0

set c = 1 and L[0] = 0.

2(r+1) w-bit words will be generated for additive round and these are stored in the array S [0 – (2r+1)].

We define two magic constants S_w and T_w for arbitrary w. [6]

S_w = Odd ((e-1) 2w) (8)

T_w= Odd ((v-1) 2w) (9)

e is the base of natural algorithms (e = 2.718281828459)

v is the golden ratio (v = 1.618033988749)

Odd (x) is the odd integer nearest to x [2].

Step 2: Encryption

It's assumed that,

1. Two w-bit registers, C and D contain the input block.
2. Key expansion is performed, the array s[0 – (t-1)] is computed .

Now,

- C = C + S[0] (10)
- D = D + s[1] (11)

Fori=1 to r **do**

- C = ((C XOR D) <<< D) + S[2*i]
- D = ((D XOR C) <<< C) + S[2*i+1]

The registers C and D contain the output. [6]

Step 3: Decryption

The encryption routine paves way to the decryption routine.

Fori = r to 1, **do**

- D = ((D-S[2*i+1] >>> C) XOR C)
- C = ((C-S[2*i] >>> D) XOR D)
- D = D-S[1]
- C = C-S[0]

Experimentally, it was determined that some rotation amount was affected after eight rounds in RC5 – 32. [6]

Fully Homomorphic Encryption

A. Principle of fully homomorphic encryption

This algorithm includes four methods:-

1. Key generation algorithm
2. Encryption algorithm
3. Decryption algorithm
4. Additional Evaluation algorithm

Involves two basic homomorphism types – the multiply homomorphic encryption algorithm and the additively homomorphic encryption algorithm. The method shown here, uses a symmetrical fully encryption homomorphic algorithm which is proposed by Craig Gentry. Here, we show the encryption and decryption methods.

Step 1: Encryption Algorithm
Three numbers p, q and r are selected as parameters for encryption, where,
p, is a positive odd number
q, is a large positive integer
r, is a random number, encrypted when selected
In which p and q are determined during the key generation phase. [7]
m is the text,

$$c = m + 2r + pq \tag{12}$$

Step 2: Decipherment Algorithm
$t = (c \bmod p) \bmod 2$
$p \times q$ is much more lesser than $2r + t$,
$(c \bmod p) \bmod 2 = (2r + t) \bmod 2 = t$

AES Algorithm

AES (Advanced Encryption Standard) was picked as a strong encryption algorithm after DES (Data Encryption Standard) was found to fall weak. It is a block cipher – this means that the output bit size will be the same as the input bit size. It is not a Feistel network, but an iterative cipher. For the mathematical operation substitution, it makes use of the S-box. It has a block size of 128 bits or 192 bits or 256 bits (128 bits block described here), for 128 bits, the algorithm takes 10 rounds, the array dimension it's stored in is 4x4. The key is of variable length. The AES is designed to be efficient in the following major considerations-

1. Security
2. Cost
3. Design Simplicity

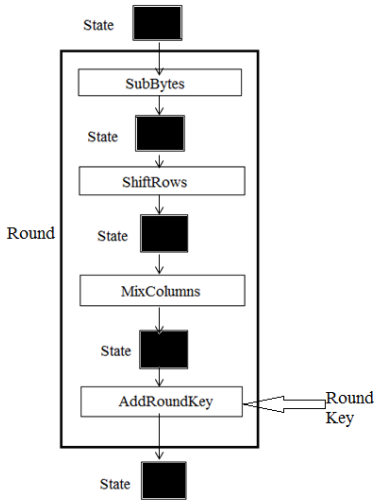

Figure 1: Structure of Each Round

A. Working of the Algorithm

It has four iterating operations and a key expansion operation (Figure 1). The different operations result in a 'state' which can be used for further transformations.

Step 1: Sub Bytes

This step is the byte substitution step .The plaintext is first converted to its hexadecimal equivalent. This results in hexadecimal content of 16 bytes. This is used as index for substitution from the S-box, where the left digit or alphabet is the row index and the right, the column index. The S-box is applied on each of the 16 bytes, this means that each byte in the state is substituted with a byte from the S-box as a result of indexing. Each of the bytes is therefore represented as hexadecimal values.

Step 2: Shift Rows

This step is a transposition step, each row of the state is cyclically shifted to the left. Starting from the 1^{st} row, which is shifted 0 positions to the left ,i.e., this row is kept the same without any shift, the 2^{nd} row is shifted 1 position to the left, similarly 3^{rd} row by 2 positions and 4^{th} row by 3 positions.

Step 3: Mix Columns

In this step, each column of the resultant state of the previous step is multiplied with the matrix

2 3 1 1
1 2 3 1
1 1 2 3
3 1 1 2

Here,

when multiplied by1, there is no change,

when multiplied by 2, shift to left(fill with zeros) and then XOR with 0X1B(if MSB before the shift is 1),

when multiplied by 3, the byte is multiplied with the result of 10 XOR 01, where 10 and 01 are hexadecimal values.

Step 4: Add Round Key

In this step, each byte of the round key is combined with each byte of the state. A byte at a position in the round key is XORed with a byte at the same position in the state.

All the rounds are similar, except for the last round. The last round doesn't perform the mix columns operations, apart from this, all the other operations are the same.

Key Expansion

The first four words are directly taken from the cipher key, which are w_0, w_1, w_2 and w_3. The rest of the words are generated as follows(i is the position of the byte)

- If (i mod 4) is not equal to 0, then $w_{i=} w_{i-4}$ XORed with w_{i-1}
- If (i mod 4) is equal to 0, then $w_i = t_i$XORed with w_{i-4}

To obtain the temporary word t_i, the following operations are performed,

- RotWord – each byte is shifted to the left with wrapping(i.e., circular)
- SubWord – each byte is substituted using the S-box
- XOR the result with Rcon – Rcon is round constant, in which the 3 bytes to the extreme right are always 0. Its value is different for each round.

The words w_4 to w_{43} are generated in this manner.

Conclusion

The paper addresses the security of data present on the cloud, which is becoming a major concerns for users all around, as more and more data is being stored on it.

Here, we presented a brief survey on the different methodologies present in providing the aforementioned services. We began by mentioning what 'storing on the cloud' means and gave a brief introduction on why confidentiality and encryption are essential.

Next, we moved our discussion to BSTs (Block status table) and how encrypted files are stored on it. Further, we touched on confidentiality being important and the K-NN classifier. Classifying data into non-sensitive and sensitive, just by studying the K neighbors of the data in question resulted in dramatic reduction of computer power with respect to encryption, and thus it was chosen. Then, we saw various methods of encryption by first looking at the RSA algorithm with key generation. This algorithm provides users with a unique digital signature which needs to be verified each time the data needs to be accessed thus encrypting it completely. Next we moved towards the ECC algorithm which was mainly implemented for mobile devices and smaller equipment. This relied on the public key generation method for providing encryption. It used two public keys, one each for the client and server, which needed to be verified in order to access the data. Further, we studied more complex algorithms, such as the Diffie Hellman Algorithm which used the symmetric key concept and a

two factor authentication method based on the modulus. This is an effective method by which the question of confidentiality can be removed by providing an effective encryption algorithm. Next, we looked at the RC5 algorithm and how its 3 stages provide full proof encryption. The 3 stages being –

1. Key Expansion
2. Encryption

Next, we saw the Homomorphic Algorithm for encryption which used two homomorphic methodologies – additive and multiplicative algorithms. Finally we pored over the effectiveness of the AES algorithm which ensures remote integrity by using spot checking and error detection. It is mostly used to ensure encryption of files on the system.

In summary, research on data security of data stored in the cloud is quite broad and a number of research issues and challenges lay ahead. Nevertheless, it is in favor of both the users and providers to find a competent way of securing data on the cloud, as it is being used more and more every passing day. This article attempts to briefly explore the current technology with respect to some aspects related to cloud security and we discuss future research that may prove beneficial in pursuing this vision.

References

[1] Prakash, Prateek, M., Singh, I., 2014, "Data Encryption and Decryption Algorithms using Key Rotations for Data Security in Cloud System", Signal Propagation and Computer Technology (ICSPCT), International Conference, Ajmer, India.

[2] Ali, M., Zardari, Jung, L. T., Zakaria, N., 2014, "K-NN Classifier for Data Confidentiality in Cloud Computing", Computer and Information Sciences (ICCOINS), International Conference, Kuala Lumpur, Malaysia.

[3] Somani, U., Lakhani, K., Mundra, M., 2010, "Implementing Digital Signature with RSA Encryption Algorithm to Enhance the Data Security of Cloud in Cloud Computing**",** Parallel Distributed and Grid Computing (PDGC), 1st International Conference, Solan.

[4] Gharshi, R., Suresha, "Enhancing Security in Cloud Storage using ECC Algorithm", International Journal of Science and Research (IJSR), India.

[5] Patil, D.H., Rakesh, Bhavsar, Thorve, A.S, 2014, "Data Security over Cloud", Emerging Trends in Computer Science and Information Technology, Proceedings published in International Journal of Computer Applications.

[6] Singh, J., Kumar, B., Khatri, A., 2012, "Improving Stored Data Security In Cloud UsingRc5 Algorithm", Engineering (NUiCONE), Nirma University International Conference, Ahmedabad.

[7] Zhao , F., Li, C., Liu, C.F., 2014, " A cloud computing security solution based on fully homomorphic encryption",Advanced Communication Technology (ICACT), 16th International Conference, Pyeongchang.

[8] Sachdev, A., Bhansali, M., 2013, "Enhancing Cloud Computing Security using AES Algorithm", International Journal of Computer Applications (0975 – 8887) Volume 67– No.9.

[9] Thiyagarajan, B., Kamalakannan, 2014, "Data Integrity and Security in Cloud Environment Using AES Algorithm",Information Communication and Embedded Systems (ICICES), International Conference, Chennai.