Security over Cloud Data through Encryption Standards

Santhi Baskaran\(^1\), Surya A\(^2\), Stephen Pius C\(^3\), Sudesh Goud G\(^4\)
\(^1\)Professor, \(^2,3,4\)Student, Department of Information Technology, Pondicherry Engineering College, Pondicherry, India

Abstract—In cloud computing, data owners upload their data to the public cloud server. The server may not be secured so it can be attacked easily and data can be taken from the server. For many security reasons this data has to be encrypted before uploading it to the cloud server. In the proposed system data owners encrypt their data before uploading by creating a key for security, and the users search their required data in the cloud server. If the data is found, it is received by the user in the encrypted form. To decrypt the data the user has to be authenticated by the data owner. After verifying the user, the data owner provides the security key to the user along with the Message Authentication Code (MAC). The message authentication code is generated by the data owner at the time of uploading itself. The user can decrypt the data using the key provided by the data owner and check the integrity of data using Message Authentication Code (MAC). Data integrity is also very important to check whether any data is missing or not. Security for data is provided by using various cryptographic methods. We establish a strict encryption algorithm i.e. Blowfish algorithm so that our system will have greater security. We encrypt the data index for efficient matching of data in the server so that the server will provide accurate search results to the user. Our proposed system also avoids Data piracy in the cloud server.

Keywords—Cloud computing, Data owner, Message authentication code, Security key.

I. INTRODUCTION

Cloud computing is the area where cloud users can remotely store their data into the cloud server so as to use the on-demand high-quality applications and services from a various configurable computing resources so that they can retrieve or use those data at any time for their purposes [1], [2]. Cloud computing ensures various benefits for businesses and users. The services of cloud computing can be private, public or hybrid. Because of its greater economic savings and flexibility it makes both users and many concerns to upload their data to the cloud server. To improve the security and privacy of the sensitive data for example, personal health records, tax documents, e-mail, photo albums, financial transactions, and so on, has to be encrypted before they are uploaded to the cloud server by the data owner, but the traditional data utilization service is based on plaintext keyword search. The trivial solution of downloading all the data and decrypting locally is clearly impractical, due to the huge amount of bandwidth cost in cloud scale systems.

Moreover, eliminating the local storage management, and storing data into the cloud servers serves no purpose unless they can be searched and utilized easily. Thus, exploring security service over encrypted cloud data is more vital. Considering the potentially large number of on-demand data users and huge amount of uploaded data documents in the cloud, this problem is particularly challenging as it is extremely difficult to meet the requirements of performance, usability, and scalability.

II. RELATED WORK

2.1. The Versa Key Framework

Security in multicasting or broadcasting is hard in dynamic group communication. In this method newly joining members are not able to understand past group traffic, and the leaving members are not allowed to know future communication [3]. It supports a close range communication for key management. Advantages of this are it allows participants to join and leave at any time. It has low complexity for join and leave the group.

2.2. Simple and Fault-Tolerant Key Agreement for Dynamic groups

In a group, if the person needs to transfer data they need to transfer it by providing key along with the data [4]. Group communication needs the secure key for data transfer in that group. Centralized methods are responsible for key distribution in large groups. Distributed key agreement techniques are mostly preferred by many groups.

2.3. Encrypting Keyword

A user sends the query or keyword of the data required by him as a request. Traditional keyword searchable encryption schemes usually build an encrypted searchable index such that its contents are hidden from the server unless it is given appropriate trapdoors generated through secret key in the symmetric key setting, and improvements and advanced security definition [3], [4], [5], [6], [7], [8]. Early works solve secure ranked keyword search which utilizes keyword frequency to rank results instead of returning same results. However, they only support single keyword search rather than multi keyword.
In the public key setting, present the first searchable encryption, whereas anyone with public key can write to the data stored on the cloud server, but only authorized users having their own private key can search through it. Public key solutions are usually very computationally expensive however, the keyword privacy could not be protected in the public key setting since server could encrypt any keyword with public key and then use the received trapdoor to evaluate this cipher text.

2.4. Boolean Encryption

Search functionalities have to be improved by, conjunctive keyword search over encrypted data have been introduced [9], [10], [11], [12]. These schemes incur large overhead caused by their fundamental primitives, such as computation cost by bilinear map or communication cost by secret sharing. As a more general search approach, predicate encryption schemes are recently proposed to support both conjunctive and disjunctive search. Conjunctive keyword search returns “all-or-nothing,” which means it only returns those documents in which all the keywords specified by the search query appear whereas disjunctive keyword search returns undifferentiated results, which means it returns every document that contains a subset of the specific keywords, even only one keyword of interest. The existing Boolean keyword searchable encryption schemes does not support multiple keywords ranked search over encrypted cloud data.

2.5. Key-insulated Public Key Cryptosystems

Insecure device are not a trustable to maintain secrecy of the private key [13]. This minimizes the damage caused by secret-key exposures to attackers. The secret key stored in the insecure device are refreshed at certain time periods, as the key is valid for only certain time period and becomes invalid.

III. MODEL OF THE SYSTEM

A cloud data hosting service consists of three different entities which is illustrated in Fig. 1: the data owner, the data user, and the cloud server. The data owner has a collection of data D to be uploaded to the public cloud server in the encrypted form E. To enable the searching capability over E for effective data utilization, the data owner, before uploading, will first build an encrypted searchable index I from D, and then upload both the index I and the encrypted data collection E to the cloud server. The data owner also generates a security key along with the message authentication code so that the data is in the control of the data owner. To search the data the user will give the given keywords, the user is then authorized.

The data owner uploads the data to the public cloud server along with the index and both are encrypted. Then the user searches for particular data they want and if the match is found the server will return the list of data in encrypted form. The users submit their information to the data owner. The data owner will check the user integrity and if the user is a valid user appropriate key along with the message authentication code is provided. At last the user can decrypt the data using that key and verify the data integrity using message authentication code.

Consider the cloud server as honest but it is really acting as curious in our model, which is consistent with related works on cloud security [14], [15]. Particularly, the cloud server acts in an “honest” way and correctly follows the designated specification. However, it is “curious” to infer and analyze data (including index) in its storage and message flows received during the protocol so as to learn more information. Depending upon what information the cloud server knows, two threat models are considered. They are Cipher text Known model and Background known model. In Cipher text Known model, the cloud server is supposed to know the encrypted data set E and index I, both of which are uploaded by the data owner at the beginning. Background Known model is the strongest model the cloud server is supposed to have more knowledge than what can be accessed in the known cipher text model. Such information may include the correlation relationship of given search requests, as well as the data set related statistical information. As an instance of possible attacks in this case, the cloud server could use the known trapdoor information combined with document/keyword frequency [16] to deduce/identify certain keywords in the query.

3.1. Used Notations

i. D—the plaintext document collection, denoted as a set of m data documents

ii. E—the encrypted document collection stored in the cloud server.

iii. W—the dictionary, i.e., the keyword set consisting of n keyword.

iv. I—the searchable index associated with C, denoted as (I_1; I_2; . . . ; I_m) where each sub index I_i is built for D_i.

v. f_w—the subset of W, representing the keywords in a search request.

3.2. Binary data generation

Data owner select the data and create bit vector for that data. Using that bit vector the binary data is generated. The binary data is the index for the data in the data owner side.
The bit vector is the bytes form of the data in the data owner. The bit vector is converted into the binary data. This bit vector and the binary data are ready for the data ciphering.

3.3. Data ciphering

The data owner encrypts the original data and sends it to the server. It also encrypts the binary data or the index and sends it to the server. Service provider does not know about the original content sent by the data owner. These indexes are used to refer the data in the server. It gives more security in the server side, so that the attackers can't use the data. We use blowfish algorithm for data ciphering.

3.4. Data user access control

The user needs data from the server. The user has different choices and the user send the query to the server or service provider. To access from the data owner the user sends the details about him or her to the data owner. Then only the data owner receives the information from client and ready to send the decryption key.

3.4. Data user query

The data user query is processed by the service provider. The service provider generates the bit vector for the query from the client. Then the service provider converts the bit vector into binary data. Server finds the similar data from the index, and sends the encrypted data to the user. Then the user decrypts the received data using the key received from the data owner.

3.5. Data Integrity

Data integrity of the data is also very important. For that the message authentication code is used as the future enhancement. Using this message authentication code the integrity of the data is verified. The data owner creates the message authentication code before the data upload in the service provider. Client receives the data from the service provider and decrpts the data by using the key from the data owner. After that the user uses the message authentication code generated by him with the code generated by the data owner.

IV. PROPOSED WORK

In our proposed work, in order to provide a security to the data we are using Blowfish algorithm for encryption. Blowfish algorithm is more secured than AES and processing time is less compared to AES. No attack is known to be successful against this. Better Key length will provide better symmetric algorithm implementation and security.

The coordinate matching is used to select the similar data from the cloud for the user query. We also check data integrity.

Fig1. Architecture of encrypted cloud data and request from user

4.1. Encryption

Blowfish is a Feistel network block cipher with a 64 bit block size and a variable key size up to 448 bits long [17]. The Blowfish algorithm is unencumbered by patents and it is free to use for any one in any situation.

Blowfish consists of two parts: key-expansion and data encryption. During the key expansion stage, the inputted key is converted into several sub key arrays a total of 4168 bytes. There is a P-array, which is eighteen 32-bit boxes, and the S-boxes, which are four 32-bit arrays with 256 entries each. All of these boxes are initialized with a fixed string, the hexadecimal digits of pi.

After the string initialization, the first 32 bits of the key are XORed with P1 (the first 32-bit box in the P-array). The second 32 bits of the key are XORed with P2, and so on, until all 448, or fewer, key bits have been XORed. Cycle through the key bits by returning to the beginning of the key, until the entire P-array has been XORed with the key.

After the string initialization, the first 32 bits of the key are XORed with P1 (the first 32-bit box in the P-array). The second 32 bits of the key are XORed with P2, and so on, until all 448, or fewer, key bits have been XORed. Cycle through the key bits by returning to the beginning of the key, until the entire P-array has been XORed with the key.

Data Encryption is as follows:

Blowfish has 16 rounds.
The input is a 64-bit data element, x.
Divide x into two 32-bit halves: xL, xR.
Then, for i = 1 to 16:
xL = xL XOR Pi
xR = F(xL) XOR xR
Swap xL and xR
After the sixteenth round, swap xL and xR again to undo the last swap.
Then, xR = xR XOR P17 and xL = xL XOR P18.
Finally, recombine xL and xR to get the cipher text.
Decryption is exactly the same as encryption, except that $P_1, P_2, \ldots, P_{18}$ are used in the reverse order.

4.2. **Simulation Results of Blowfish**

The simulation results showed that Blowfish has a better performance and has more efficiency than other common encryption algorithms used. Blowfish is not having any known security weak points so far, so it is an excellent candidate to be considered as a standard encryption algorithm.

The following graphs show the performance of blowfish algorithm comparing to other standard encryption algorithm.

The simulation is carried out in Encryption mode and Cipher mode for varying block sizes as shown in fig.2 and fig.3.

From the graph it is verified that the Blowfish algorithm has taken minimum encryption time, compared to other algorithm for all the data block sizes.

4.3. **Data Integrity using Message authentication code**

Message authentication code is created by the data owner at the beginning. When the user asks the security key for decryption, the data owner will transfer the security key along with the message authentication code to the user. Using message authentication code, the user checks the integrity of the data. This will ensure that the data is original.

V. **Conclusion And Future Work**

In this paper we define and solve the problem of security over cloud data through various encryption standards. Thorough analysis investigating privacy and efficiency guarantees of proposed schemes is given, and experiments on the real-world data set show our proposed schemes introduce low overhead on both computation and communication.

In future the security of the proposed work may be improved by analyzing user authentication. The upcoming encryption algorithms such as two fish, three fish encryption algorithms may be used for encryption by considering the cloud server as non-secured.
REFERENCES


[12] Ning Cao, Cong Wang, Ming Li, Kui Ren, Wenjing Lou, “Privacy-Preserving Multi-KeyWord Ranked Search over Encrypted Cloud Data,” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 1, JANUARY 2014.


