

Data Security for Virtual Data Centers by Commutative Key

Maram Balajee¹, Challa Narasimham², and Y. Ramesh Kumar³

¹ Dept. of IT, G M R Institute of Technology,
Rajam, Andhra Pradesh, 532127, India
balajee.m@gmrit.org

² Dept of IT, V R Siddhartha Engg. College, Vijayawada
narasimham_c@yahoo.com

³ Dept. of IT, Avanthi Inst. of Technology,
Vizianagaram, AP, India
javaramesh143@gmail.com

Abstract. The maintenance cost of a Data Center (In Small Organizations) is very high and difficult also. So an economically better choice is to use cloud computing and cloud storage instead of manage data centers by itself. But in Cloud Storage, Cloud user's sensitive data is in the control of a third party. Here the customer can't trust the Cloud Storage. But Cloud storage providers' claims that they can protect the data, but no one believe them. So this paper presents a framework to ensure data security in cloud storage system. In this framework, we use Commutative property, Bitwise XOR for managing keys between cloud storage and customer. And UNICODE and Colors for encrypt and decrypt the data both in cloud storage and customer's system.

Keywords: UNICODE, Colors, commutative key, cloud storage, data center, bitwise XOR.

1 Introduction

As of now, many technologies have introduced to store the data in cloud storage. But in some industries, the data are being dynamically created. And the data sources are geographically distributed all over the world. But cloud storage has the potential of providing geographically distributed storage services since cloud can integrate servers and clusters that are distributed all over the world and offered by different service providers into one virtualized environment. This can potentially resist disastrous failures and achieve low access latency and greatly reduced network traffic by bringing data close to where they are needed.

Cloud computing can be defined as a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned, and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers [1].

cloud storage

Cloud storage is a new distribution model, however, with the potential for economies of scale. Aside from cost, its benefits are outsourced operation, simple, unlimited growth and 'enterprise' features for smaller users - like high availability, security, data protection, etc.

There are nearly as many definitions of cloud storage as there are providers of cloud services. In simplest terms, cloud storage is data storage or services hosted remotely on servers and storage devices on the Internet or a similar private network, usually hosted by a third party.

Cloud storage is a subset of cloud computing, in which the term cloud refers to the wide area network infrastructure, including switches and routers, for a packet-switched network. When capitalized, cloud usually refers to the public data network, including the Internet.

Cloud computing has probably been best defined by the National Institute of Standards and Technology (NIST) as:

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Data is deployed to cloud storage either through Web-based applications or through Web services application programming interfaces (APIs). Web-based applications are often used for manual access to data or management functions, while APIs are used for more automated or transparent approaches. Since standard APIs and communications protocols are used, the physical location of the data becomes irrelevant, since it can be made available virtually anywhere via the Internet or private network. This also means that cloud data can be easily replicated to multiple locations for fault tolerance, high availability and other purposes, often without involvement of the customer.

2 Existing Systems

In cloud storage, the data is stored in Remote system which is owned by third party vendor. Whenever the customer wants to get data, the data will be transferred from third party vendor to customer. In this scenario, how the owner of the data trust the third party vendor. Because the plain data is available in the server which is being maintained by third party vendor. So the owner of the data can't trust the third party vendor. So there is a need of alternative solution. The following section explains that alternative solution.

2.1 UNICODE

ASCII which stands for American Standard Code for Information Interchange became the first widespread encoding scheme. However, it is limited to only 128 character definitions. Which is fine for the most common English characters, numbers and punctuation but is a bit limiting for the rest of the world? The people in the world naturally wanted to be able to encode their characters too.

So there is a need of a new character encoding scheme was needed and the UNICODE [2] standard was created. The objective of UNICODE [2] is to unify all the different encoding schemes so that the confusion between computers can be limited as much as possible. These days the UNICODE [2] standard defines values for over 105,000 characters and can be seen at the UNICODE [2] Consortium. It has several character encoding forms, UTF standing for UNICODE [2] Transformation Unit:

- UTF-8: only uses one byte (8 bits) to encode English characters.
- UTF-16: uses two bytes (16 bits) to encode the most commonly used characters.
- UTF-32: uses four bytes (32 bits) to encode the characters. UTF-32 is capable of representing every UNICODE [2] character as one number.

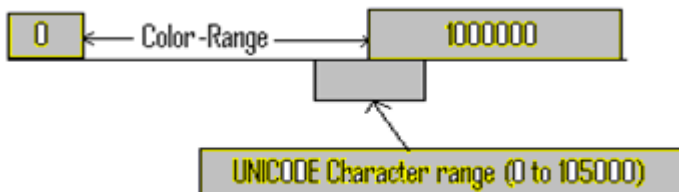
2.2 Colors Supported by Computer

Careful measurements of our visual system's best performance have been made by psychophysicists (people who study human responses, like seeing Color, to things in the world, like light). They have shown that we can see about 1000 levels of light-dark, 100 levels of red-green, and 100 levels of yellow-blue for a single viewing condition in a laboratory. This means that the total number of Colors we can see is about $1000 \times 100 \times 100 = 10,000,000$ (10 million). A computer displays about 16.8 million Colors to create full Color pictures, really more than necessary for most situations.

2.3 Cryptography with UNICODE and Colors

This method is fully based on Private-key cryptography. In secret-key cryptography schemes, a single key is used to encrypt & decrypt data. A secret key may be held by one person or exchanged between the sender and the receiver of a message. If secret-key cryptography is used to send secret messages between two parties, both the sender and receiver must have a copy of the secret key. The method is to send it via another secure channel or even via overnight express, but this may be risky in some cases.

In this method, the sender will decide the range of Colors which will be assigned to 1, 05,000 UNICODE [2] characters. The selection of Colors is in the following way:



In this proposed new policy, the sample binding between character / symbol/digit, UNICODE [2] and Color is in the following way:

Ch	UNICODE	COLOR
a	U+0061	Grey
b	U+0062	Red
c	U+0063	Green

Fig. 1. Sample mapping of Char/symbol/digit, UNICODE and Color

2.4 Use of Receiver-Key

A computer displays about 16.8 million Colors to create full Color pictures, really more than necessary for most situations. Now we are considering 10 million Colors only. And the UNICODE [2] standard defines values for over 105,000 characters and can be seen at the UNICODE [2] Consortium. Now 10 million Colors and 105,000 characters are available in a computer system.

Now we can create a dynamic mapping table like Fig 1. If we select starting position is 825,001. Then 825,001st Color is assigned to the first UNICODE [2] character. The 825,002nd Color is assigned to the second UNICODE [2] character. And so on. Finally the 930,000th Color is assigned to 105,000 UNICODE [2] character.

2.5 Encryption

This is kind of cryptography is fully based on UNICODE [2] and Colors. First of all, it checks each and every character in the given file. Then it finds concerned UNICODE [2] of each character is. Then it gives the corresponding Color for each UNICODE [2] character according to the predefined mapping.

According to the starting-point, the dynamic mapping table is created before encryption. Next, it takes the first character from text and finds the UNICODE [2]. According to the UNICODE [2], it will take concerned Color. Then it will take the 2nd character and so on. Now all characters are translated into corresponding Colors. It means, all characters are encrypted into Color-charts.

Now it takes the first character from shared receiver-key and finds the concerned Color. And overlap all Color-charts with this new Color. Then it takes the 2nd character, finding the concerned Color and overlaps recent Color-charts with Color and so on.

After encryption, the data is transferred from owner of the data to Cloud Storage. So the Cloud Storage is having encrypted data, which is not decrypted by the owner of the Cloud Storage also.

2.6 Decryption

Now the receiver receives encrypted data and temporary-key. After receiving a temporary-key from sender, the receiver calculates the sender-key. According to the

sender-key, the receiver prepares a mapping between alphabet/special character/digit, UNICODE [2] and Color. By using this chart, the receiver easily identify the COLOR->UNICODE->character.

3 Proposed System

The following Steps explain how Proposed Systems works.

Step 1: In this proposed method, there is a need to apply Commutative Property. According to Commutative Property, there is a need of one secret-key i.e Receiver-key and one public key i.e Intermediate-key. Here the sender creates Session-keys dynamically. Here no need to send any key with the message to the receiver.

Step 2: When the owner of the data wants to use Cloud Storage, then he/she simply encrypt the data by using the existing method (which is explained in existing systems) and shared secret-key i.e receiver-key. Now the Cloud Storage is having encrypted data only.

Step 3: When customer (Receiver) wants to get data from Cloud Storage, he/she has to send request. After receiving a request, a Session-key will be created and a Intermediate-key will be calculated based on session-key, shared secret-key i.e Receiver-key and bitwise XOR. And the encrypted data will be encrypted once again with Cloud's session-key.

Step 4: Now the data is transferred from Cloud Storage to Customer. Here no need to any key, because the Intermediate-key is public key. Now the Customer is having encrypted data, Intermediate-key and Receiver key.

Step 5: Based on available keys (Intermediate- key & Receiver-key), the receiver can calculate the required key i.e Cloud's Session-key. Here Intermediate-key is public-key.

Step 6: Now the receiver can decrypt the encrypted data by using Cloud's Sender-key and Receiver's receiver-key.

So no need to send any key while transmission of data from Cloud's Storage Centre and Receiver. Because the Receiver is having receiver-key and Intermediate-key is public key, so the receiver can calculate Cloud's Session-key by using Intermediate-key, receiver-key and bitwise XOR.

After calculating Cloud's Session-key, the receiver can decrypt the data by using Cloud's session-key and Receiver's receiver-key. So commutative property plays vital role in this proposed system.

3.1 The Importance of Commutative Property

According to Mathematics, the commutative property is $a.b=b.a$. According to Proposed system, a and b are sender-key, receiver-key respectively. By default, the data is encrypted with receiver-key (which is shared between End-user and owner of the data) i.e a. As and when required, the encrypted data is again encrypted with sender-key i.e. b.

In this way, sender is having sender-key and receiver is having receiver-key only. But the sender sends encrypted data & temporary key (t) to the receiver. Here t is bitwise XOR of a and b. So Temporary-key (t): $a \wedge b$.

Then the receiver receives encrypted data and temporary-key (t) only. Based on receiver-key and temporary-key (t), the receiver calculates sender-key (a) using bitwise XOR (\wedge).

While data transmission, the hacker can't get neither original data nor original keys (sender-key & receiver-key). In this way, commutative key provides more security to the data which is stored in Cloud Storage/ Data Center. So third-party vendor also not able to get the data, which is stored in Cloud Storage.

In the proposed method, we can take data from different languages in the world like English, French, German, Latin, Russian, Hindi, Telugu, Tamil, Kannada, Bengali, Malayalam, Urdu etc. And we can take sender key, receiver key from those languages also. So those keys would not be guessed by hackers.

4 Explanations with an Example

Suppose we want to encrypt the message "rajam" with shared secret key "abc". Now the above message is translated into the following UNICODE [2] characters:

rajam->U+0072 U+0061 U+006A U+0061 U+006D

4.1 Encryption

Then these UNICODE [2] characters are translated into the following Color-chart.



Fig. 2. Basic Color Chart of given Message

Here shared receiver-key is "abc". Assigned Colors are as follows:

Ch	UNICODE	COLOR
a	U+0061	grey
b	U+0062	red
c	U+0063	olive

In first iteration, the Basic Color Chart (BCC) is overlapped with the corresponding Color of the first character in shared receiver-key 'a'. It is called First Color Chart (First CC). In 2nd iteration, the recent Color chart overlapped with the corresponding Color of 'b' and so on. After completion of overlapping, the Final Color Chart (FCC) is looking like the following:



Then a temporary key is calculated based on sender-key, receiver-key and bitwise XOR in the following way. Assume,

Receiver-key: “abc”; Sender-key: “xyz”

Temporary key will be calculated by using Receiver-key, Sender-key and bitwise XOR in the following way:

```

01100001(a)  01100010(b)   01100011(c)
01111000(x)  01111001(y)   01111010(z)
.....
00011001(↓)  00011011(←)   00011001(↓)
.....
    
```

Here the result is bitwise XOR of “abc” and “xyz”. Now the temporary-key is “↓←↓”. This temporary-key also converted into Colors [3]. Now encrypted data and temporary-key are ready.

When receiver wants to get data, then the receiver will receive encrypted data and temporary-key (“↓←↓”). Now the receiver will calculate sender-key by using receiver-key, temporary-key and bitwise XOR. After performing bitwise XOR, the receiver will get sender-key.

By using sender-key and receiver-key, the receiver simply decrypts the received encrypted data. Initially the receiver should decrypt using receiver-key then sender-key in the following way:

After receiving the Color-chart, the receiver prepares Mapping between alphabet / special character / digit, UNICODE [2] and Color. According to receiver-key it is very simple to find equaling UNICODE [2] then character.

4.2 Decryption





After receiving encrypted data and temporary-key, the receiver calculates the sender-key by using receiver-key, temporary-key and bitwise XOR in the following way:

Receiver-key (“abc”) & temporary-key (“↓←↓”):

```

01100001(a)  01100010(b)   01100011(c)
00011001(↓)  00011011(←)   00011001(↓)
.....
01111000(x)  01111001(y)   01111010(z)
.....
    
```

Here the result is bitwise XOR (^) of receiver-key and temporary-key i.e sender-key (“xyz”). By using the mapping table like Fig 1, we can decrypt the encrypted message like the following:

In this the first cell  is converted into  by applying receiver-key. After applying sender-key, the cell  is converted into , which indicates ‘b’. If we apply same procedure to remaining cells then we can get the actual message “rajam”

5 Pictorial Representations

Here the owner of the data is encrypt the data by using shared secret-key i.e receiver-key. And it will be uploaded to Cloud Storage by using above said method i.e. UNICODE [2] AND colors [3] combination. And the owner of the data will calculate temporary-key based on sender-key, receiver-key and bitwise-XOR. Here sender-key and receiver-key are 2 components in Commutative property. The procedure is in the Fig3.

After receiving encrypted data and temporary-key, the receiver calculates sender-key using receiver-key, temporary-key and bitwise-XOR (^). Then receiver decrypt the data using both sender-key and receiver-key like the Fig4.

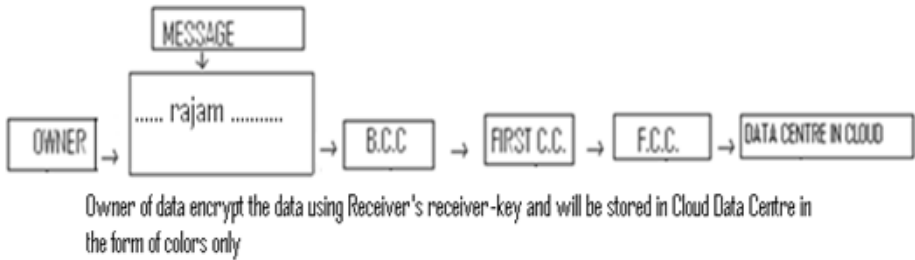


Fig. 3. Data Encryption and storage of data in Cloud's Virtual Data Center

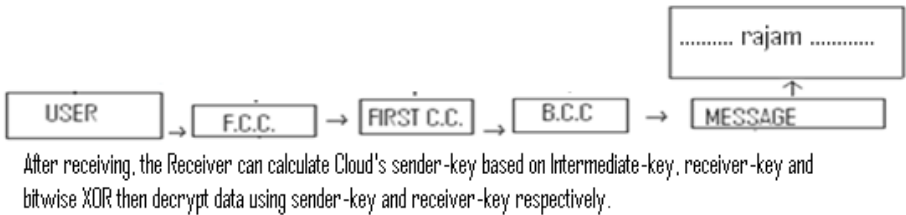


Fig. 4. Data Decryption using sender-key, receiver-key, bitwise XOR, UNICODE & Colors

References

- [1] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. *Future Generation Computer Systems* 25(6), 599–616 (2009)
- [2] Balajee, M.: UNICODE [2] and Colors Integration for Encryption and Decryption. *IJCSE* 3(3) (March 2011)
- [3] Balajee, M., Narasimham, C.: IPVDD: Intrusion prevention for virtual Data Centers (A Framework for Encryption and Decryption). *IJCST* 2(4) (October- December 2011); ISSN: 0976-8491 (online), 2229-4333 (print)