Innovative Systems Design and Engineering www.iiste.org
ISSN 2222-1727 (Paper) ISSN 2222-2871 (Online)
Vol 2, No 3

Extension of Tinny Application for Cloud Based Custom Software with Encryption Queue: A Strategy to Protect Data on Cloud

Muhammad Ahmad Jan
Institute of Computing and Information Technology Gomal University, D.I.Khan, Pakistan
E-Mail: mr_ahmadjan@yahoo.com

Sheikh Muhammad Saqib
Institute of Computing and Information Technology Gomal University, D.I.Khan, Pakistan
E-Mail: saqibsheikh4@yahoo.com

Bashir Ahmad
Institute of Computing and Information Technology Gomal University, D.I.Khan, Pakistan
E-Mail: bashahmad@gmail.com

Shakeel Ahmad
Institute of Computing and Information Technology Gomal University, D.I.Khan, Pakistan
E-Mail: shakeel_1965@yahoo.com

Abstract

Data protection is in your hands, even your data lands in the lap of service providers. Cloud computing is a new and emerging concept in the field of IT. It means do everything over the internet and put burden on the shoulders of cloud's service providers. Individuals or business organizations are hesitant to do their tasks over public networks like internet. They can't leave their sensitive data in the hands of service providers. It is due to lack of confidence and trust on service providers. Protecting customer's sensitive data on cloud is the major issue. Here authors purposed a strategy to protect customer's sensitive data before sending it towards service providers. The strategy consists of a barrier (Encryption Queue) to protect customer's data. With the use of this strategy, encryption procedure will be in the hands of customers themselves. This concept will help to built up customer's trust and will prepare them to shift desktop based applications over the cloud.

Keywords: Cloud Computing, Business Organizations, Encryption Queue, Customer Trust, Desktop Based Application.

1. Introduction

Data should be protected from frauds, waste or any unauthorized person at any cost. Major runaway success of cloud computing is the security of data. Cloud is making now high security density zone with well procedures of encryptions [3].

In cloud environment there are many security management policies such as cloud's machine should installed firewalls and VPNs, setting of user permissions, passwords etc. Due to password, unauthorized person can not access the data but cloud provider is the owner of cloud environment. They can modify or visit consumer data although contract has been done between them. This is the major issue through which distrust is growing up in cloud environment. Data encryption is major idea for data security i.e. before

hosting sensitive data externally, one should encrypt it for fear that the data will be stolen or modified by the cloud provider [2].

Program like PGP or open-source TrueCrypt encrypts the data before sending it to the third party and can encrypt the file so that authorized members with their own passwords can access such data [6]. But in case of CBCS [4] (Cloud Base Custom Software), such application will be itself on third party, then how such encryption decryption can be lie on client side.

It is still under discussion who will keep the encryption/decryption keys, whether it should be kept on customer side or on cloud side. Most of the customers want their data should be encrypted both ways across the internet via SSL (Secure Socket Layer) and also when it is stored at cloud vender's storage pool [7].

Large companies commonly use custom software for critical functions such as content management, inventory management, customer relationship management and human resource management [5]. Mostly these applications are desktop based. Successfully installation of custom software requires different components such as: Switches, Database Server, Database Software, Operating System, Anti Viruses, and Backup Devices.

Each organization has its own software such as custom software known as Desktop Based Custom Software (DBCS). CBCS is the dedicated service for an organization and stored on cloud environment. Conversion of DBCS (Desktop Based Custom Software) to CBCS (Cloud Based Custom Software) has already been getting the all advantages of cloud computing but trust of consumer on cloud is still in discussion. TA (Tinny Application) makes work of CBCS just like DBCS i.e. TA restricts the access of CBCS outside the organization [4]. Though cloud computing has many advantages, however data security and privacy over the internet are some of the serious issues faced by organizations. Since CBCS and its database will be stored on third party spaces. It raises three types of issues for customer trust.

-Third party can delete our data

-Third party can change our data

-Third party can see our data

Although there are service level agreements (SLA) between cloud service providers and customers about the integrity and privacy of the data, however if people at cloud side delete or change customer, this will hurt customer's trust. Major failure of trust is the third option i.e. cloud side people can see customer data. This problem can be removed by the idea of encryption/decryption. Still issue is on study of road, either encryption can be done consumer side or on cloud side. It is observed that in case of CBCS, encryption/ decryption should be on cloud side because CBCS will be placed over here. As encryption/ decryption code will be added in CBCS, so customer can vacillate with such option because encryption code will be placed on cloud side. EQ (Encryption Queue) is solution of such problem, through which any customer can place the data (in CBCS) on cloud side without fear of being stolen of data. Setting of EQ will be done on client side and its running will be executed on cloud side with CBCS. In this way, CBCS will completely remove the DBCS from organization and can make its position more accurately. Implementation of CBCS will eliminate all drawbacks related to DBCS and will incorporate all the advantages of cloud computing with an organization.

2. EQ (Encryption Queue):

EQ (Encryption Queue) which will present the setting of encryption policy on client side can provide a relief to customer for security issues. Since with CBCS there is a TA application [4], which can be extended, with encryption procedures i.e. EQ. Consumer high-up can set encryption policy according to their desire order from encryption list. High-up can change the order of such options and remove or add some options in EQ with different intervals of time. Since cloud side people does not know whether what order they have selected and what choices they have added in encryption queue.

2.1. Values for EQ

Following are the some choices for encryption

1. Reverse string

2. Add any special symbol after 2 characters (Special Symbol and no of characters will be user defined)
3. Move character up 2 next position (2 is according to user choice)
4. Etc

2.2. Algorithm for EQ

Algoritm for encryption will contain the following items

Fields[]: This list contain the whole input fields which are going to be stored.

encryptionQueue[]: This queue will contain different integer values for encryption choices. And also such value will be stored on cloud side database.

encryptionQueue.count(): This methode will return value for total selected option.

SaveOnCloudSide(Fields[]): This event will save the encrypted data field to database on cloud side.

Encryption(Fields[],encryptionQueue[I]): this methode will apply encryption on Fields[] with respect to encryptionQueue[I].

Table-1 Algorithm for Encryption on Consumer Side

1. [Put all input fields in a list]
   Fields[]
2. [Select encryption options from list through TA in you desired order]
   encryptionQueue[]
   Perform step 4 for I=1 to encryptionQueue.count()
3. [Set encryption on all fields]
   Encryption(Fields[],encryptionQueue[I])
4. [After applying encryption on customer side store encrypted data to cloud side]
   SaveOnCloudSide(Fields[])
5. [Exit]

Algorithm shows that each next encryption will be applied on previous encrypted data which will be much secure and any one can not guess, judge or decrypt such data.

3. Extension of TA (Implementation of EQ)

This is implementation of encryption procedures in TA application, which will be installed on all client sides [2] but EQ will be permitted only to administrator or high-up. This interface will generate a queue for encryption (encryptionQueue []), this queue will be stored on database of cloud side, since in CBCS there are no database on consumer side. This queue will contain only integer values for different options. Cloud side people can not imagine which integer value relates to which type of encryption options.

Fig-1 is showing the different choices for encryption sequence. Here special characters, number of characters and move character are the user defined choice. User can set the values for such options according to their desires.

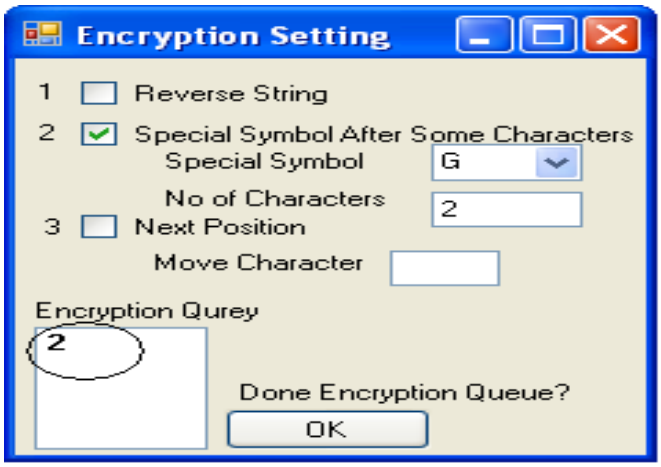

Fig-1: Selecting 1st Option

In Fig-1, 2nd option will be selected as 1st element of encryption queue. Here G will be added after 2 characters of each input data field.

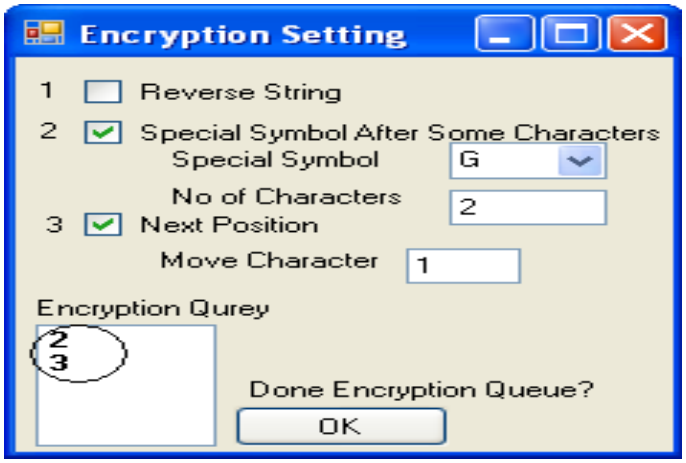

Fig-2: Selecting 2nd Option

In Fig-2, 3rd option will be selected as 2nd element of encryption queue. Here each character of previous encrypted data will be moved up to 1 character.

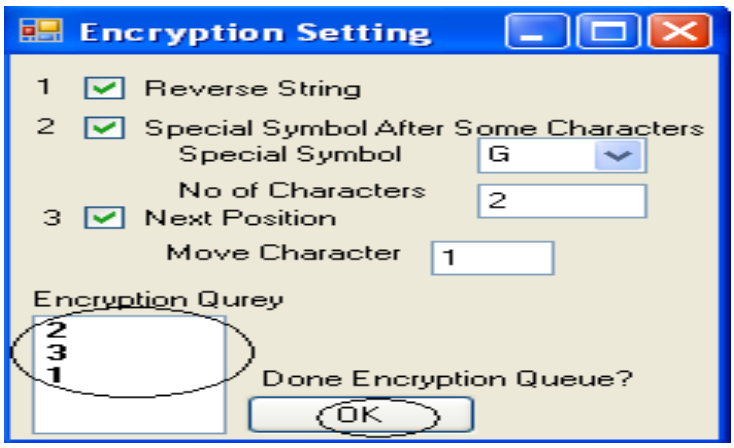

Fig-3: Selecting 3rd Option

In Fig-3, 1st option will be selected as 2nd element of encryption queue. Here previous encrypted data will be reversed.

Hence by selecting above order of encryption choices encryptionQueue(EQ) will be as:

EQ[1]	EQ[2]	EQ[3]
2	3	1

Value of encryptionQueue.count() will be 3.

3.1. Example (Encryption/ Decryption of word 'Cloud')

Suppose input Data is "Cloud".

Field="Cloud"

Now encryption code shown in Fig-4 will be added to CBCS, so any data on CBCS will be saved on database in encrypted form.

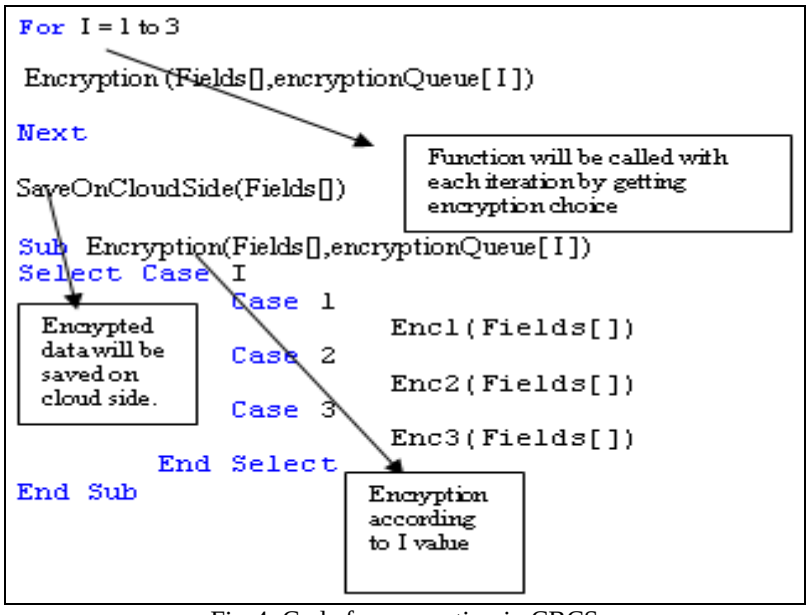

Fig-4: Code for encryption in CBCS

Now applying code (shown in Fig-4) on above data Field.

EQ[1]	EQ[2]	EQ[3]
2	3	1

Table-2: Encryption through Encryption Queue

Encrypting Data Filed 'Cloud'
When I=1 then option is 2. G will be added after 2 characters Now Cloud become as ClGouGd.
When I=2 then option is 3. Move each character upto next 1. Previous encrypted data is ClGouGd. C → D l →m G →H o →p u →v G →H d →e DmHpvHe
When I=3 then option is 1 Reverse the previous encrypted data. Final Encrypted Data: eHvpHmD.

Hence final encrypted word eHvpHmD will be stored in database. People of cloud side can not understand what its actual value is. Although they ma be familiar with different encryption options for CBCS, but they can not judge order of selection different options.
Now when user access data from database, it will be first decrypted then appeared on user interface. Decryption is the inverse of encryption. Now apply encryptionQueue[] in reverse order.

Table-3: Decryption through Encryption Queue

Decrypting Data Filed 'eHvpHmD'
When I=3 then option is 1 Reverse the previous encrypted data. Decrypted Data: DmHpvHe
When I=2 then option is 3. Move each character upto previous 1. Previous encrypted data is ClGouGd. D → C m →l H →G p →o v →u H →G e →d Decrypted Data: ClGouGd
When I=1 then option is 2.

G will be removed after 2 characters Now ClGouGd become as Cloud.

Hence eHvpHmD is decrypted to Cloud

4. Conclusion

Every new technology replaces the older one with more advantages. Since the trend of cloud computing is reducing the overheads of previous approaches for utilizing the different services/resources. This concept is very beneficial for business organizations. But when they utilize the storage resources for their personal business data, they can't make trust over such resources. Encryption queue provides a completely user defined options for encrypting their data. Only organizational high ups will have the knowledge to construct the Encryption Queue. By using encryption queue with tinny application they can easily replace their custom application over cloud side and can take different advantages of cloud computing such as [1]:

1. It should be able to quickly allot and relieve resources whenever required by clients
2. It should have real-time backup to offer maximum up time to clients
3. It should be able to cater to the needs of clients without having to involve clients into management of the service.
4. There is no need of Backup devices.
5. No efforts require on backup and recovery
6. After system crash, only installation of operating system will required for connecting with internet.

Besides above advantage, it will get the all advantages of CBCS[4] such as:

- No need of Database software.
- After system crash, only installation of operating system will required for connecting with internet.
- A new system can be easily replaced at once and CBCS will work properly on spot.
- CBCS does not require any supportable frame work, just internet connections.
- No need of database server.
- No need of technical securities on local area network even without local area network, CBCS can work properly with the connection of internet.

References

An Introduction to Cloud Computing, http://www.brighthub.com/environment/green-computing/articles/10026.aspx?image=149096

Arnon Rosenthal, "Cloud computing: A new business paradigm for biomedical information sharing", Journal of Biomedical Informatics 43 (2010) 342–353, ELSEVIER.

Subhas Chandra Misra, "Identification of a company's suitability for the adoption of cloud computing and modelling its corresponding Return on Investment", Mathematical and Computer Modelling 53 (2011) 504-521, ELSEVIER

Sheikh Muhammad Saqib, "Custom Software under the Shade of Cloud Computing", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 5, May 2011, PP 219-223

Custom software, http://en.wikipedia.org/wiki/Custom_software

Anthony T. Velte, Toby J. Velte, Ph.D, Robert Elsenpeter, "Cloud Computing: A Practical Approach", by The McGraw-Hill Companies, ISBN: 978-0-07-162695-8, 2010

John W. Rittinghouse, James F. Ransome, "Cloud Computing Implementation, Management, and Security" , CRC Press , ISBN: 978-1-4398-0680-7, 2010.

ISSN 2222-1727 (Paper) ISSN 2222-2871 (Online)
Vol 2, No 3