A SURVEY ON MCELICE PUBLIC KEY CRYPTOSYSTEM USING GOPPA CODES TO IMPLEMENT IN CLOUD

Chanda Raju[1], **Arif Mohammad Abdul**[2]

[1]*M. tech scholar, Department of Computer Science and Technology, GITAM University, Hyderabad, India*
[2]*Asst Professor, Department of Computer Science and Engineering, GITAM University, Hyderabad, India*

Abstract

As the technology is emerging its level into virtualisation in the current computing. Among these virtualisation, the most used and highlighted concept is cloud computing. Cloud computing is the technology where a user can share his resources for storing and maintaining the user data in the shared environment confidentially by having different levels of agreements. It is a pay per use system where the resources are serviced on the user request and requirement. In this system the main concern is how much secured the user data is, in the shared environment. Many theories proposed different encryption techniques (for example AES, DES, RSA etc.,) till the date, among those techniques one of the oldest encryption technique is McElice cryptography, based on the linear algebraic function for the data encryption which was rarely used so far in any environment. The proposed system is to implement this McElice encryption technique on the data to store in the cloud, that uses Goppa codes [6] for error correction. The proposed system is to prove that the McElice PKC [9] technique is good enough to use in different systems by comparing the results with other encryption techniques. The work clearly specifies regarding different encryption techniques that are used so far.

Keywords: *Cryptography, McElice PKC, Encryption, Decryption, Goppa Codes.*

--***--

1. INTRODUCTION

Cloud computing is a technology that is introduced based on the very old concept of computing devices using mainframes that uses huge equipments to store and compute large programs in the early days of computers. By modernizing the concept of mainframes the cloud computing is built for providing the space to store the data and many other services for developing their own applications and infrastructure according to the user requirement. It works on the pay per use concept. It has brought a huge change in the way the computing services were traditionally being delivered. This cloud offers platform as a service, software as a service and infrastructures as service, the users are not aware of the location where these services are offered and performed. This model enables the cloud users to increase their usage of data storage capacity and capability through dynamic access without investing in other infrastructure, licensing new software etc. In cloud computing there are two major issues to be consider for which the security is ensured one is authentication and the other constrain is data storage. The proposed system mainly concentrates on the data storage using a new type of encryption called as *McElice cryptography* [2][7] as far as cloud data storage is concerned.

1.1 Data Encryption

Encryption is the process of converting the plain text into cipher text. In past the message were sent in the form of cipher text in any field to protect sensitive data by performing different cryptographic techniques. Slowly the same has been implemented in the field of computer technology, by performing different encryption algorithms like RSA, ECC, DES, AES etc.

There are two types of encryption techniques.

- Symmetric key encryption.
- Public key encryption.

1.1.1 Symmetric Key Encryption

In this encryption both the sender and the receiver has the same key for exchanging the message. By comparing the two similar keys either sides the communication ends successfully.

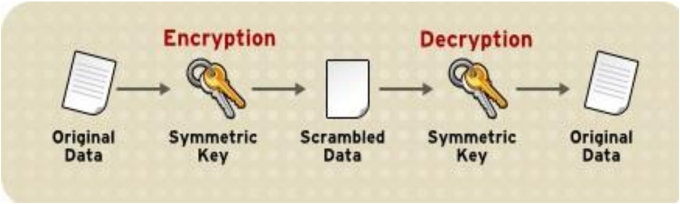

Fig .1 Symmetric Encryption Standard

1.1.2 Public Key Encryption

In this encryption there will be two different keys one is for encryption public key that is open to all and the other is for decryption private key that is kept with the receiver himself and use at the time of accessing the data.

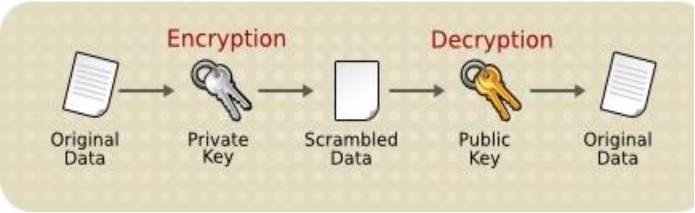

Fig. 2 Public key Encryption Standard

1.2 Mcelice Public Key Cryptography

In the year 1978 McElice proposed a new crypto system using algebraic coding technique. This cryptography uses the error correcting code for which the decoding is fast with the help of *Goppa codes* [5]. In this technique the generator matrix transforms the code and hides the structure. Then the transformed matrix becomes the public key.

1.3 Goppa Codes

The *McElice Cryptosystem* [4] uses the *linear codes* [9] with a set of same length and dimensions, in that case these goppa codes are used for the error correction and minimizing the length of the text after encryption. This goppa codes are constructed mainly considering a function as mentioned below.

1. Select a finite field $\mathbf{F}_{2m}$ of n=2^m elements
2. Select an irreducible polynomial g(z) € $\mathbf{F}_{2m}$ [*z*] of a vector **c =** (c_0, $c_{0, \ldots,}$ $c_2{}^m$ **-** $_1$) is the *irreducible Goppa code* [1] Γ(*L*, *g*) if and only if

$$\sum_{i=0}^{2m-1} ci(z - \zeta i)^{-1} \equiv 0 \text{ mod g}(z)$$

2. LITERATURE REVIEW

Encryption is the process of converting the plain text as cipher text. Many different encryption techniques are been introduced from early days of computing. As the encryption algorithms are classified into two categories.
The following are the common algorithms that are introduced and widely used for symmetric key encryption.

- DES (Data encryption standard-1970): It is a 64-bit block cipher using 56-bit key, having 16 rounds for complete encryption.
- AES (Advanced encryption standard-2001): It has three different key sizes 128, 192, 256 and the rounds for completion are 10, 12, 14.
- Triple DES (1998): It has two different key sizes 112, 168 and the rounds for completion are 48.

Following are the asymmetric key encryption algorithms.

- RSA (Rivest, Shamir, Adleman-1977): It considers 1024-bit for minimum security encryption and 2048 in the current usage. But RSA has not satisfied the requirements of IND requirements. NIST says that RSA 15360-bits is equivalent to strength of 256-bit.
- DSA (Digital Signature Algorithm-1991): Using minimum of 1024-bit to maximum 15360-bit as to generate the private key it hold maximum time for encryption and decryption.
- Diffie-Hellman (1976): It is also similar to RSA and DSA in terms of maintaining the key size and security levels.
- ECC (Eliptic crypto curve- 2004): It carries minimum 160-bits to 512-bits key size. Comparing with RSA, DSA, DH it is better in key size where as it is more expensive in usage.

After all the algorithms having their own level of performance advantages and disadvantages, the proposed system is to introduce the an old McElice crypto system for better advancement of encryption standards in cloud computing.

In the year 1978, McElice proposed public key cryptosystem based on the error correcting codes for encryption. This encryption technique mainly uses Goppa codes as the *error correction code* [10], as it functions on the linear algebraic polynomial equations. A polynomial g(x) over GF(2^m) of degree *t* is taken, for each g(x) there exists a binary *Goppa code* [3] of length n=2^m and dimension *k* >= *n – mt*. The linear code is described as *k* x *n* generator matrix *G*. By the use of *k* x *k* matrix *S* and *n* x *n* permutation matrix *P*, a new generator matrix is constructed that hides the structure of *G*.

$$G^1 = S * G * P$$

3. PROBLEM STATEMENT

1. The current system of data encryption in cloud using different encryption standards are been implemented according to the type of data to be stored and the type cloud services provided.

2. As the existing cloud applications are mostly implementing the symmetric encryption standards like AES, DES, etc.,which has to maintain the same key for encryption and decryption.

3. In this case the key generation and maintenance is done by the service provider or the third party who is maintaining the keys, which is not trust worthy.

Disadvantages:

- The keys are maintained in blocks.
- Sharing the encrypted key blocks with the third party and the service provider is risk.
- An attacker can easily decrypt the text if he knows the key, as the key is public for encryption and decryption.
- Decryption is done using the same key.

4. PROPOSED SYSTEM

The proposed system uses an old public key encryption technique known as *McElice cryptography* [8] that is classified into three processes as follows.

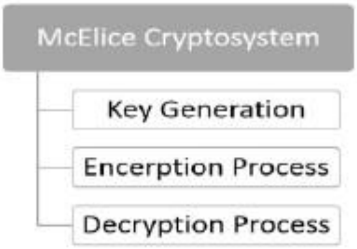

Fig .3 McElice Cryptosystem

4.1 Key Generation:

McEliece Cryptosystem Public key cryptosystem based on two types of keys (public and private).A public key is published and used encrypt the data, while a private is kept secret and used it to decrypt the message. To prepare keys, the following approaches should be used

1. The secret key of *McEliece PKC* [1] depends on three Parameters:

* The first is select a random polynomial g(z) of degree t over GF(2m). The Goppa code Γ(L; g(z)) has parameters [n; k ≥ n -mt; d ≥ 2t + 1]. Calculate the k × n private generator matrix G of the Goppa code.

* Pick a random k × k matrix S, S × B= I. The matrix B is derived from Gauss elimination method. This approach is faster than to determining the determinant of a matrix.

* Pick an arbitrary n × n permutation matrix P, Where P is a matrix that contains one ones in each row and each column.

2. The Public generator matrix $G*$ is calculated by $G* = S \times G \times P$ and should be published with degree of random generator polynomial t.

4.2 Encryption Process in McEliece Cryptosystem

The following steps for encryption.

1. In the encryption process there is public generator matrix $G^*_{K\,x\,n}$ and degree of an arbitrary generator polynomial t.
2. Convert each character to decimal number using ASCII code of 7 bits length.
3. Collect all binary string together.
4. If the length of the message mod k ≠ 0, the message with zero's in the last of the message, should be padding.
5. Each process for k bits from the message should perform steps 6-10.
6. Calculate fetched message × G*.
7. Create an error vector e with size n and include (≤ t) errors.

4.3 Decryption Process in McEliece Cryptosystem

The following steps should be done for the decryption.

1. The receiver has the following information:
Goppa code secret generator matrix$G_{k\times n}$, Fig 4. Decryption process
Matrix S, and Permutation matrix P.
2. Compute the matrix S, and the inverse of Permutation matrix.
3. Divide the cipher message into k bits parts.
4. For each entity should perform steps 5-9
5. Compute
$mSG' = c \times P{-}1 = mSGP + P' = mSG + e'$.
6. Use efficient decoding algorithm either the separable Goppa code or irreducible to find error location e'.
7. Calculate $mSG = mSG' + e'$.
8. Remove secret generator matrix G using Gaussian elimination method to get (mS).
9. Compute $m = mS \times S{-}1$.
10. Collect all computing message together.
11. Check the length of the message mod 7 =0, if not we can remove the length of the message (mod 7) zero's from the last of the message.
12. Fetch every time 7 bits from the message and convert it to decimal number.
13. Convert each decimal number to character using ASCII table.

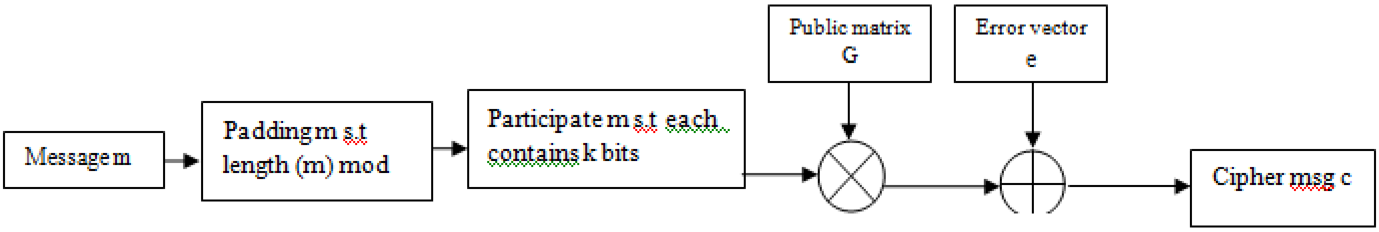
Message m
Padding m s.t length (m) mod
Participate m s.t each contains k bits
Public matrix G
Error vector e
Cipher msg c

Fig .4 Encryption Process

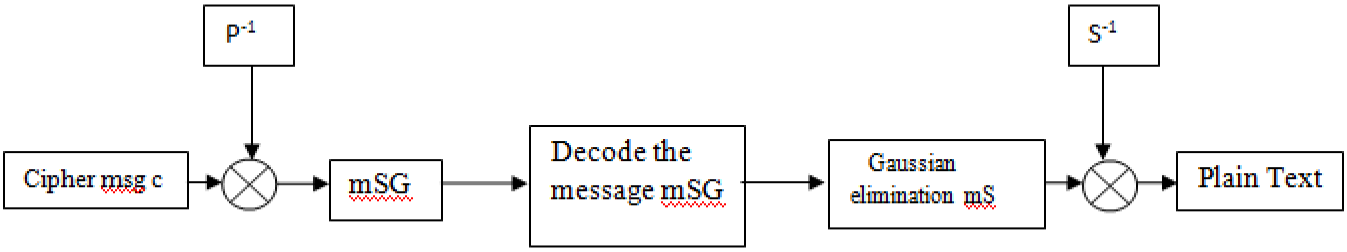

Fig .5 Decryption Process

Table .1 Comparison between different encryption standards

Algorithm with KEY size	Megabytes processed	Time taken (seconds)	Mega bits/sec
AES			
128-bits	256	4.196	64.336
192-bits	256	4.817	53.145
256-bits	256	5.308	48.229
McEliece cryptosystem	**On microcontroller**		

1024-bits	16-bit CPU @13MHz	970ms(encryption)
		690ms(decryption)
			62

2048-bits	16-bit CPU @33MHz	1390ms(encryption)
		1060ms(decryption)
			102

	On FPGA		

2048-bits	1400FPGA Enc @ 150MHz	1.07ms(encryption)
		1.082ms(decryption)
			80

2048-bits	Xilinx virtex-6 163MHz	0.5ms(encryption)
		1.4ms(decryption)
			102

DES			
56-bits	128	5.998	21.34
3DES			
112-bits	128	6.159	20.783

5. BLOCK DIAGRAM

Phase- I: Uploading a File/Data

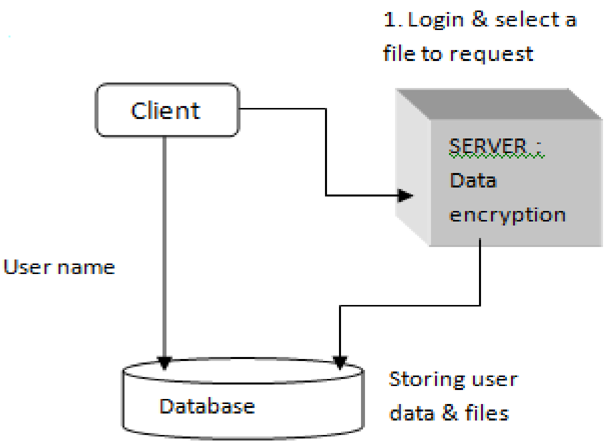
1. Login & select a file to request
Client
SERVER : Data encryption
User name
Database
Storing user data & files

Fig .6 Block diagram for Uploading a file

Phase-II: Retrieving a File

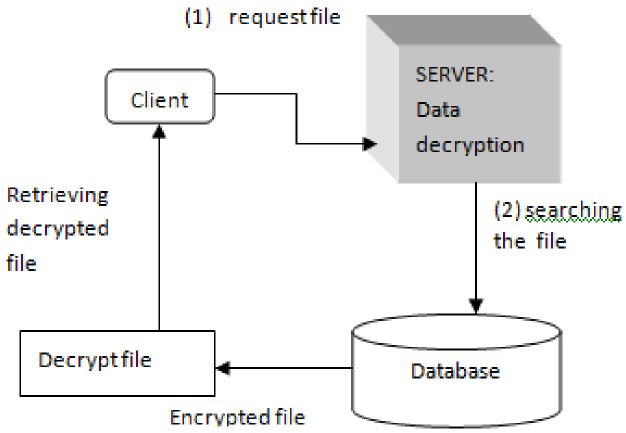

Fig.7 Block diagram for retrieving file

6. CONCLUSION

As the cloud computing is the emerging technology and vastly used in the current trends. Simultaneously the user number is increasing gradually, so the main concern here is how the user data is being stored and secured. To ensure the data security there must be a trusted service provider with the strongest encryption standards. In order to this other than the traditional encryption standards in cloud computing the proposed system is to introduce an old and strong data encryption standard McElice cryptosystem, which uses the error correction coding for making the key more stronger and shorter. This technique follows the mathematical and algebraic linear codes, to balance and secure the encrypted data. So far many applications are followed mostly using the symmetric encryption algorithms, but the proposed system using public key encryption standard is better with its unbreakable linear coding and safe to store and transfer the data in the cloud.

REFERENCES

[1] Newroz Aldabagh, Comparison between Separable and Irreducible Goppa Code in McEliece Cryptosystem , Salahaddin University – Erbil,2015.

[2] Marek Repka — Pavol Zajac, Overview of the McElice Cryptosystem and its Security, Slovak University of Technology, Bratislava, Ilkoviˇcova-2013.

[3] Sergey Bezzateev and Natalia Shekhunova, Optimal Quasi-Cyclic Goppa Codes, Saint Petersburg State University of Aerospace Instrumentation, Russia, 2013.

[4] Roberto M. Avanzi, Simon Hoerder1, Dan Page, Michael Tunstall, Side-Channel Attacks on the McEliece and Niederreiter Public-Key Cryptosystems, HGI and Faculty of Mathematics, Ruhr-University Bochum-2011.

[5] Thomas Risse, DSI GmbH Bremen, How SAGE helps to implement Goppa Codes and McEliece PKCSs, IIA Faculty EEE & CS, Hochschule Bremen, University of Applied Sciences-2009.

[6] Key One Chung ,Goppa Codes, Department of Mathematics, Iowa State University, Ames-2004.

[7] Suanne Au Christina Eubanks-Turner Jennifer Everson, The McEliece Cryptosystem-2003.

[8] Bart Preneel, Antoon Bosselaers, Rene Govaerts and Joos Vandewalle, A Software Ementation of the McELIECE Public-Key Cryptosystem. Laboratorium ESAT, Belgium-1992.

[9] R. McElice. A Public-Key Cryptosystem Based on Algebraic Coding Theory Technical report, NASA, 1978.

[10] Edoardo Persichetti, under the supervision of Ass. Prof. Steven Galbraith, Improving the Efficiency of Code-Based Cryptography, Auckland, 2012