A Cloud Computing Security Solution

Deepak. D. M
(Mtech,CS)
SVIT
Bangalore, India

Nithin. K
(B.E,CS)
SVIT
Bangalore, India

Vidya. C. S
(B.E,CS)
SVIT
Bangalore, India

Uha. P. J
(B.E,CS)
SVIT
Bangalore, India

***Abstract*— Cloud computing is a general term for the delivery of hosted services over the internet .In recent years, outsourcing large amount of data in cloud and how to manage the data raises many challenges with respect to privacy with rapid development of cloud computing, more and more users deposit their data and application on the cloud. But the development of cloud computing is hindered by many cloud security problem. Cloud computing has many characteristics, ex: multiuser, virtualization, scalability and so on.**

Among the various cryptographic encryption schemes, Homomorphic scheme allow to perform meaningful computations on encrypted data. In this context, the research deals with homomorphic encryption scheme for maintaining privacy and security in cloud by detecting the error incurred while transferring the data using RSA cryptosystem.

Keywords —Distributed implementation, Cloud services, Cloud security, Fully homomorphic encryption.

I. INTRODUCTION

Cloud Computing is an innovative service mode. It is started off with grid computing, where large number of systems are used for solving scientific problem that require high levels of parallel computation. This technology expanded exceptionally, which eventually stimulated concerns over ensuring data security in public networks. It enables users to get almost unlimited computing power and abundant a variety of information services from internet. Although cloud computing has become a mature service model, and have large commercial, cloud computing is still facing many problems.

According to a recent survey conducted by Cisco global cloud networking academy, it has been revealed that 72% of IT professionals stated that security of data is a major hindrance to implement the services in cloud. Recent development in cloud storage and the services rendered by it allows users to outsource storage. As a result, it allows companies or organizations to offload the task of maintaining datacenters. In the past few years, the security requirements for data are very strong and many algorithms have evolved. Only few algorithms play a comprehensive role in creating and maintaining a secure session over vulnerable public networks. Public key cryptography is one of the commonly used algorithms for this type of operation. The communicating parties share their private keys amongst them before exchanging information. In the case of transmitting a message over a public channel, the work of Diffie helman and RSA provides way to encrypt a message into cipher text using private key. Consequently, the receiver on the other side has to read the cipher text by decryption with the help of their private key. The encryption scheme shows that the secret decryption key allows retrieving the actual text but if the secret key is lost, the cipher text is of no use .In 1978, RDA decided to propose a technique on performing arbitrary computations on encrypted data. Such techniques give rise to useful applications to privately perform manipulations on encrypted data.

Homomorphic encryption is evolved to solve such critical issues. The homomorphic properties of ciphers have been implemented in various real time applications. Some of them include privacy protection during electronic voting, computation in multiparty environment computation and analyzing traffic in distributed environment. Basically, homomorphic encryption enhances the security measures of cloud data. Data protection is achieved through the homomorphic encryption scheme, which allows additive and multiplicative operations over encrypted bits.

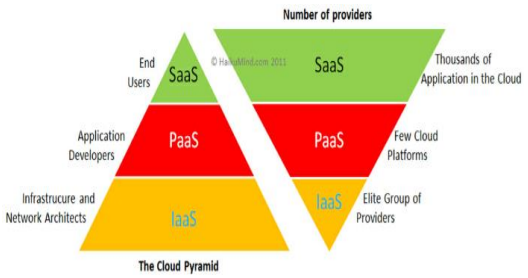
According to the survey in 2009, Google expose to the risk of mixed data by unauthorized sharing user's spreadsheets data and document data.

Three security requirements are often considered: confidentiality, integrity and availability for the most Internet service providers and cloud users. In the order of Saas, Paas and Iaas, the providers gradually release the responsibility of security control to the cloud users.

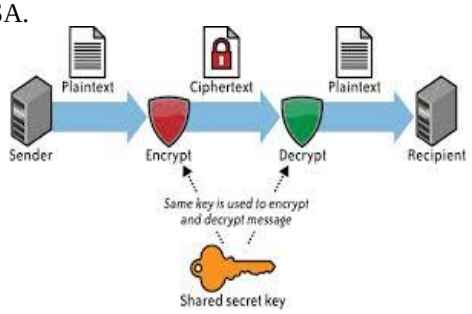
Saas[Software as a Service] is a software distributed model in which applications are hosted by a vender or service provider and made available to customers over a network, typically the Internet.

Paas[Platform as a Service] is a cloud model in which provider deliver apps over the Internet and host users' hardware and software on their infrastructure.

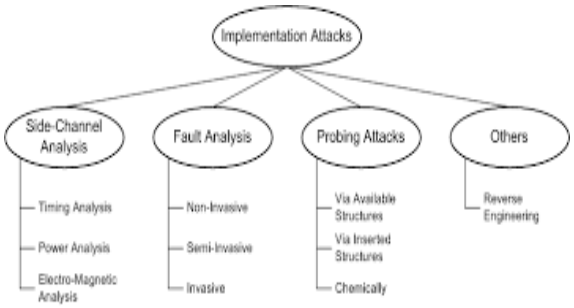
Iaas [Infrastructure as a Service] is a type of cloud computing in which a third-party provider hosts virtualized computing resources over the Internet.



The public key and private key cryptosystems are designed with various fault attacks. Error Detection(ED)-based countermeasures have been developed for both private-key cryptosystem such as AES, public-key cryptosystem such as RSA, ECC. In this research, the focus is on detecting the fault attack using public-key cryptography, RSA. It is identified that counter measures for RSA can be devised. It is achieved through the digital signature mode which is based on CRT-RSA.



2. FAULT ATTACK ON CRYPTOGRAPHIC IMPLEMENTATIONS

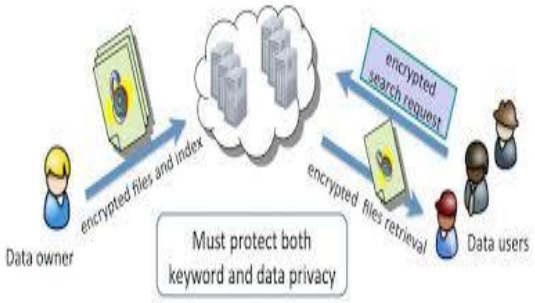


- Cryptographic algorithms like symmetric ciphers, asymmetric ciphers, and hash functions are designed with a set of primitives that meet specific objectives. The cryptographic implementations on evaluation show their resistivity against attacks. It is necessary to determine countermeasures against such attacks and evaluate the feasibility and applicability of such attacks. Side channel attacks assist in breaking the hardware or software implementations of many cryptosystems including block ciphers(DES, AES), stream ciphers(RC4,RC6), public key ciphers (RSA-type ciphers, Megamall-type ciphers, ECC, XTR, etc.), to break the implementations of signature schemes, chunk authentication code schemes, cryptographic protocols, cryptosystems, and networking systems. Side channel faults are of two kinds. The first kind of induced during cryptographic computation. These faults are either random or intentional, caused by a voltage manipulation. The second kinds of faults occur by intentionally injecting corrupted input data. This research focuses on such computation wherein the receiver while noticing a mismatch identifies that the chunk is fault.

3. FAULT ATTACK ON RSA

There are many formal definitions for public key cryptosystem such as RSA and paler cryptosystem. Public-key cryptography is asymmetric since one of the participants has a secret key, while the others have access to the public key that matches the secret key. But, the symmetric system has only one key which should be shared between the two participants. The complexity of the systems indicates that the computation of public key systems is time consuming. The objectives is to exchange data between two users without sharing a common secret. RSA labs embarked on an effort to differentiate the security level of symmetric key and the RSA key size. The security of RSA depends on the key size. In integer factorization problem based algorithms, security depends on the difficulty to factorize a large number to obtain large primes.

4.SECURITYARCHITECTURE



1) The Privacy protection: User transmits and save their data to the cloud by encrypted. Both ensure the security of data in the process of transmission, and ensure safe storage od data. Although the cloud computing service providers handle, they can't easily obtain the information of plaintext.

2) Data processing: Fully homomorphic encryption mechanism enables users or the trusted third party process cipher text data directly, instead of the original data. Users can obtain arithmetic results to decrypt to get good data. For example, in the medical information system, electronic medical records are in the cipher text is store in the cloud server. When the health department deals with potential safety problems, they must know some areas of certain disease location and age distribution. They can give encrypted electronic medical record data to the professional data processing services. Then they can get the correct data after decryption.

3) The cipher text retrieval: Fully homomorphic encryption technology based on cipher text retrieval method can search directly on the cipher text data. It is not only ensure query privacy and improve the efficiency of retrieval, the retrieval data can be added and multiply without changing the corresponding plaintext.

Three generations of network defense technologies have appeared in the past. In the first generation, tools were designed to prevent or avoid intrusions. These tools usually manifested themselves as access control policies or tokens, cryptographic systems, and so on. However, an intruder could always penetrate a secure system because there is always a weak link in the security provisioning process. The second generation detected intrusions in a timely manner to exercise remedial actions. These techniques included firewalls,

intrusion detection systems (Ides), PKI services, reputation systems, and so on. The third generation provides more intelligent responses to intrusions.

5. HOMOMORPHIC BASED ERROR DETECTION SCHEME

The error detection scheme includes input block that contains all the input chunks. Based on the size of the input, chunks are created. The number of chunks decides the type of error detection scheme. When the count of chunks is two, the basic error detection scheme for even chunks is evaluated as discussed above. When the count of the chunks is odd the error detection scheme for odd chunks is evaluated. Under this scheme a constant chunk is generated and it is used during encryption. The large dataset involves large number of chunks relatively of the order of n follow the enhancement error detection scheme. The enhanced error detection scheme is allowed to run in parallel framework. K denotes number of input chunks taken for all three schemes.

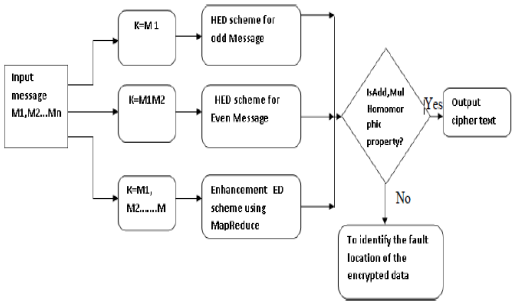


Figure. Homomorphic error detection scheme

5.1. MULTIPLICATIVE PROPERTY

Basic ED scheme select two successive chunks and perform encryption of two chunks. It calculates the multiplication of two chunks and performs the encryption the multiplication of two chunks. Then calculate the multiplication of two chunks then perform the encryption the multiplication. The chunk count to be processed is odd to generate constant chunk and add this chunk to input chunk list. The encryption process is initiated by encryption of two input chunks and buffers the corresponding results. The product of two chunks is calculated continued by performing encryption of the calculated chunks product. All the three schemes are decided to check the homomorphic property. When homomorphic property is satisfied it indicates no mismatch between the product of cipher texts and the cipher text of the product of chunks. When the homomorphic property not satisfied, it indicates fault on the encrypted data.

Homomorphic Encryption Scheme:

Step a: $C_1=E(M_1)$

Step b: $C_2=E(M_2)$

Step c: $C_3=E(M_1 \times M_2 \bmod n)$

Step d: If $C_3=(C_1 \times C_2) \bmod n$

No Fault -Output(C_1, C_2)

else

Fault Injected.

Figure. Homomorphic Property of RSA

5.2. ADDITIVE PROPERTY

Basic ED scheme select two successive chunks and perform encryption of two chunks. It calculates the addition of two chunks and performs the encryption the addition of two chunks. The chunk count to be processed is odd to generate constant chunk and add this chunk to input chunk list. The encryption process is initiated by encryption of two input chunks and buffers the corresponding results. The addition of two chunks is calculated continued by performing encryption of the calculated chunks addition. All the three schemes are decided to check the homomorphic property. When homomorphic property is satisfied it indicates no mismatch between the addition of cipher texts and the cipher text of the addition of chunks.

Homomorphic Encryption Scheme:

Step a: $C_1=E(M_1)$

Step b: $C_2=E(M_2)$

Step c: $C_3=E(M_1+M_2 \bmod n)$

Step d: If $C_3=(C_1+C_2) \bmod n$

No Fault -Output(C_1, C_2)

else

Fault Injected.

FIGURE. HOMOMORPHIC PROPERTY OF PAILLER CRYPTOSYSTEM

5.3. ED SCHEME FOR EVEN CHUNKS

The homomorphic error detection scheme operates on encrypted data in three steps namely one verification operation after three normal operations. The process starts with the encryption of two input messages and stores the result. The next step involves encryption of the product of the two messages which occupies buffer. The comparison of the stored results is achieved through a comparator. The mismatch between the product of the cipher text and the cipher text of the product of the chunks shows an injection has occurred.

On the other side, during such operation fault can even occur inside comparator. Such problem can be resolved using a self-checking comparator or a duplicate running in parallel. The results are stored in registers along with the secret information. This prevents the attacker to steal the cipher text before verification is done. The drawback in the scheme is that it requires always even number of chunks.

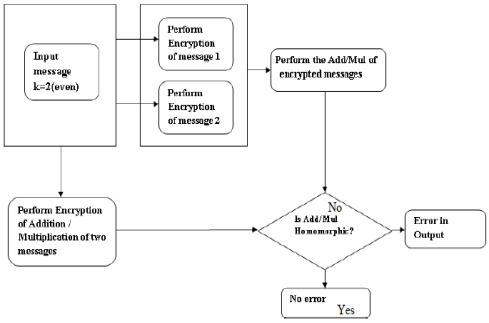


FIGURE. ERROR DETECTION SCHEME FOR EVEN CHUNKS

5.4. ED SCHEME FOR ODD CHUNKS

The basic ED scheme supports even chunks. Therefore to support the odd chunks the next scheme is proposed. Here introduction padding solves the problem. In such cases the

simple chunk M_1 has to follow the previous algorithm proposed with a constant chunk M_2. The chunk M_2 can either be a constant chunk or a random generated value. The scheme encrypts input chunk M_1. It is followed by the encryption of chunk M_1*M_{ons} mod n for RSA and encryption chunk M_1+M_{cons} mod n for Paler cryptosystem. The comparison of the result with the product and addition of encryption of M_1 and constant chunk M_{ons} helps to identify the fault.

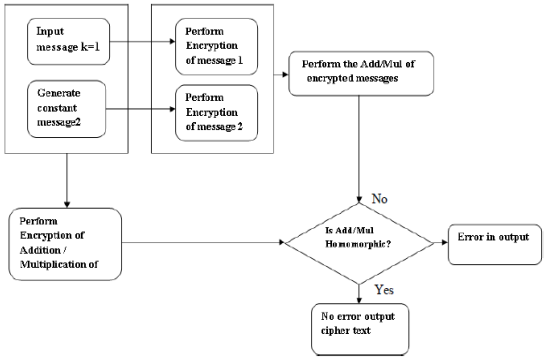


FIGURE. ERROR DETECTION SCHEME FOR ODD CHUNKS

5.5. *ENHANCED ED SCHEME:*

This scheme is designed for the chunks which are relatively large. The previous schemes time overhead ranges from 50% to 100%. To reduce the time overhead further, the third scheme is proposed by performing the same multiplicative and additive operations using Map Reduce concept. In this parallelized scheme, each mapper holds n chunks where the chunks split based in the available number of mappers to compute the encrypted form of the input chunks. Mapper i….. Mapper k are designed to perform encryption on the chunks resulting in $E(M_I)….E(M_y)$ under mapper i to $E(M_K)…E(M_P)$ under mapper k. the leftover odd chunks follow the padding scheme of constant chunk as explained previously.

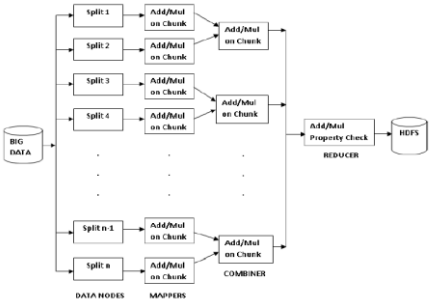


FIGURE. ENHANCED ED SCHEME USING MAP-REDUCE

CONCLUSION

In this research, an effective low cost and high performance error detection scheme that uses additive homomorphic property of RSA and papillae is introduced. The time overhead for the error detection scheme for odd and even chunks varied from 50% to 100%. The one time encryption of individual chunks using Map reducing algorithms has significantly improved the fault detection latency in both RSA and papillae cryptosystem. The memory overhead depends on the mapper output values that are stored in buffers. All the schemes support for output latency is relatively low as the verification to find the fault occurs only after comparing the output of the mapper.

Security problem is a big problem for the development of cloud computing, encryption is a central technology to ensure the cloud computing data security. We also need to deploy mechanism to prevent online piracy and copyright violations of digital content. We will study reputation systems for protecting cloud systems and data centers.

Security infrastructure is required to safeguard web and cloud services. At the user level, one needs to perform trust negotiation and reputation aggregation over all users. At the application end, we need to establish security precautions in worm containment and intrusion detection against virus, worm, and distributed Do's(Dodos) attacks.

Based on the cloud data security problem faced, this article introduced the homomorphic encryption mechanism, proposes a cloud computing data security scheme. The scheme ensures the transmission data between the cloud and the user safety. And in the cloud storage their data is still safe. It is convenient for users and third party agency to search data to dispose. At present fully homomorphic encryption scheme has high computation problem needs further study.

ACKNOWLEDGEMENT

The authors are grateful for Ability Sir for guiding us based on the thoughts collection of Fang Zhao. Fang Zhao was graduated from Beijing university of posts and telecommunications, master, majoring in automation.

REFERENCES

i. Rivest R, Adleman L, Dertouzos M. on data banks and privacy homomorphisms[M]. Newyork: Academic Press, 1978:169-180.
ii. Wikipedia. Cloud computing[EB/OL]. http://en.wikipedia.org/wiki/Cloud_*computing*,*2012-12-05*
iii. TALBOT D. Security in the ether[j]. technology review, 2010,113(1),36-42.