Wildcard Encryption System Using RSA Algorithm with K2C

S.Thanga Revathi[1], A.Kowsalya[2], R.Ramya[3], R.Revathi[4]
[1]*Assistant Professor, Dept of IT, MNMJEC, Chennai, India*
[2, 3, 4]*Student, Dept of IT, MNMJEC, Chennai, India*

Abstract: **Searching is one of the major task in today's internet. Many Search Engines are designed to enable this searching process. The search results are based on the word or phrase given for searching. The results have the content matching the pattern or phrases of the search input. But, today the world demands for a better search results, which is given by the new searching mechanism called the Wildcard Searching Technique. The Wildcard search is a technique which matches the entire character rather than only a normal string. The Wildcard Searchable Encryption Technique provides confidentiality and privacy of data. At present Paillier Homomorphic encryption has been used for wildcard search. Some disadvantages have been analyzed in its performance level like key size and its encrypted file size. In this paper, we proposed to implement RSA Algorithm, so as to increase the efficiency and performance of wildcard search.**

Keywords: **K2C, RSA, Cipher Text, Wildcard search, Encryption, Decryption**

I. INTRODUCTION

Encryption is the process of taking information represented in one fashion (usually human-readable), and representing it in another fashion (not usually human-readable). It is mathematically based, often using complex calculations. It uses an external piece of information, known as a key, to perform this re-representation. There are several different types of encryption, and they are used for various things. Common examples include connections to Internet merchants like Amazon, protected military communications, and hiding the password to your cell phone.

In regular encryption, the process follows a specific workflow. In particular, it looks as follows:

- The information is encrypted.
- The encrypted information is stored for future use.
- Retrieve the encrypted information for use.
- Decrypt the information to perform any operation on it.
- Once finished, delete, or re-encrypt the information.

So, from the time of decryption, to the time of deletion or re-encrypt, the information is exposed.

Homomorphic encryption works differently. Instead of decrypting before use, this process uses the information in its encrypted form. In other words, once encrypted, the information is never decrypted and exposed until the very end.

Homomorphic encryption is a form of encryption that allows computation on cipher texts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext. The purpose of Homomorphic encryption is to allow computation on encrypted data. Homomorphic Encryption is a method of performing calculations on encrypted information without decrypting it first. Because it could make cloud computing a lot more secure.

Cloud Computing platforms can perform difficult computations on homomorphically encrypted data without ever having access to the unencrypted data. Homomorphic encryption can also be used to securely chain together different services without exposing sensitive data.

For example, services from different companies can calculate 1) the tax 2) the currency exchange rate 3) shipping, on a transaction without exposing the unencrypted data to each of those services. Homomorphic encryption can also be used to create other secure systems such as secure voting systems, collision-resistant hash functions, and private information retrieval schemes.

II. HOMOMORPHIC CRYPTOSYSTEMS

Homomorphic encryption scheme supports both addition and multiplication operations on cipher texts, from which it is possible to construct circuits for performing arbitrary computation. It is possible to compute arbitrary number of additions and multiplications without increasing the noise too much. By "refreshing" the cipher text periodically whenever the noise grows too large, it is possible to compute arbitrary number of additions and multiplications without increasing the noise too much. Homomorphic Encryption cryptosystems scheme on the assumed hardness of two problems:

1. Worst-case problems over ideal lattices
2. Sparse (or low-weight) subset sum problem.

The 2nd generation of Homomorphic cryptosystems still follows first to construct somewhat a Homomorphic cryptosystem that handles noisy cipher text, and then convert it to a fully Homomorphic cryptosystem using bootstrapping

that is Boots trappable is capable of evaluating its own decryption circuit and then at least one more operation. Homomorphic encryption scheme can be converted into a fully Homomorphic encryption through a recursive self-embedding.

Functions of Homomorphic Encryption:

Homomorphic Encryption H is a set of four functions

H = {Key Generation, Encryption, Decryption, Evaluation}

1. Key generation: client will generate pair of keys public key pk and secret key sk for encryption of plaintext.
2. Encryption: Using secret key sk client encrypt the plain text PT and generate Esk(PT) and along with public key pk this cipher text CT will be sent to the server.
3. Evaluation: Server has a function f for doing evaluation of cipher text CT and performed this as per the required function using pk.
4. Decryption: Generated Eval(f(PT)) will be decrypted by client using its sk and it gets the original result.

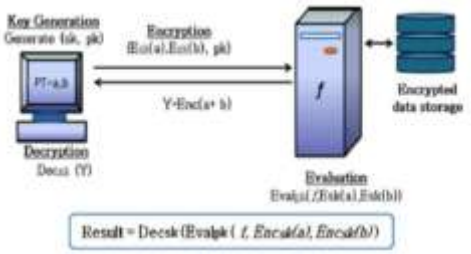


Fig 2: Functions of Homomorphic Encryption

The following are the two categories of cryptosystem.

A. Fully Homomorphic encryption

A cryptosystem that supports *arbitrary computation* on cipher texts is known as fully Homomorphic encryption (FHE) and is far more powerful. Such a scheme enables the construction of programs for any desirable functionality, which can be run on encrypted inputs to produce an encryption of the result. Since such a program never needs to decrypt its inputs, it can be run by an untrusted party without revealing its inputs and internal state. Fully Homomorphic cryptosystems have great practical implications in the outsourcing of private computations, for instance, in the context of cloud computing.

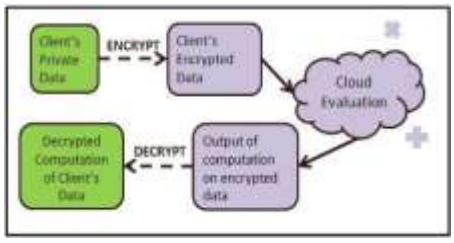


Fig 1: Homomorphic Encryption

B. Partially Homomorphic encryption

Partially Homomorphic encryption are classified into three types

- ElGamal encryption system is an asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie–Hellman key exchange.
- Paillier encryption is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing *n*-th residue classes is believed to be computationally difficult. The decisional composite residuosity assumption is the intractability hypothesis upon which this cryptosystem is based.
- RSA (Rivest–Shamir–Adleman) is one of the first public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and it is different from the decryption key which is kept secret (private).

III. WILD CARD SEARCH

Wildcard Search is a technique which matches the entire character pattern rather than only a normal string.

Searchable encryption scheme is a cryptographic technique that allows search of specific information in an encrypted content. Searchable encryption (SE) enables the users to generate a search token from the searched keyword in such way that given a token, the cloud server can retrieve the encrypted contents containing the searched keyword. Basically, the search token represents an encrypted query over the encrypted data and can be generated only by users with the appropriate secret key. The original goal of searchable encryption is to provide privacy-preserving keyword searches of encrypted data against an intermediate gateway such as a mail server or a network router, which involves a message exchange process between the sender and the receiver. The first searchable encryption scheme was the Public-key Encryption with Keyword Search (PEKS) scheme based on Identity-Based Encryption (IBE).

Example:

Electronic health record (EHR) storage system as an application example of SE. Consider, Alice's doctor Bob may use the keyword "05/**/2016" to search for all Alice's EHRs created during the month of May of 2016 and use the keyword "*ache" to search for Alice's EHRs containing "headache", "stomach-ache" or "heartache".

Wildcard Pattern Matching

Given a text and a wildcard pattern, implement wildcard pattern matching algorithm that finds if wildcard pattern is matched with text. The matching should cover the entire text (not partial text).

The wildcard pattern can include the characters '?' and '*'
'?' – matches any single character
'*' – Matches any sequence of characters (including the empty sequence)

For example,

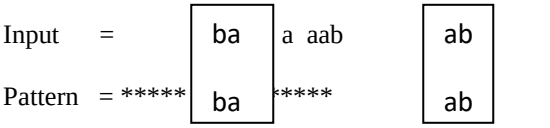
Text = "baaabab",

Pattern = "*****ba*****ab", output : true

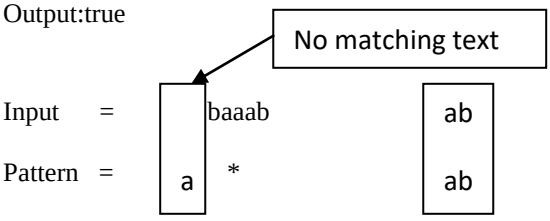
Pattern = "baaa?ab", output : true

Pattern = "ba*a?", output : true

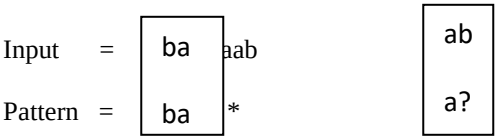
Pattern = "a*ab", output : false



Output:true



Output:false



Output:true

Each occurrence of '?' character in wildcard pattern can be replaced with any other character and each occurrence of '*' with a sequence of characters such that the wildcard pattern becomes identical to the input string after replacement.

Let's consider any character in the pattern.

Case 1: The character is ''*

Here two cases arise

1. We can ignore '*' character and move to next character in the Pattern.
2. '*' character matches with one or more characters in Text. Here we will move to next character in the string.

Case 2: The character is '?'

We can ignore current character in Text and move to next character in the Pattern and Text.

Case 3: The character is not a wildcard character

If current character in Text matches with current character in Pattern, we move to next character in the Pattern and Text. If they do not match, wildcard pattern and Text do not match.

IV. LITERATURE SURVEY

[1] The ElGamal Public Key Encryption Algorithm The ElGamal Algorithm provides an alternative to the RSA for public key encryption. 1) Security of the RSA depends on the (presumed) difficulty of factoring large integers. 2) Security of the ElGamal algorithm depends on the (presumed) difficulty of computing discrete logs in a large prime modulus. ElGamal has the disadvantage that the ciphertext is twice as long as the plaintext. It has the advantage the same plaintext gives a different ciphertext (with near certainty) each time it is encrypted.

The encryption time of Paillier shows huge time difference compared to RSA and ElGamal. RSA showed better performance over ElGamal and Paillier in term of encryption time . [2] ElGamal showed better performance over RSA and Paillier in term of decryption time . RSA showed better throughput over ElGamal and Paillier in encryption process.

Paillier showed worst result in encrypted file size and its encrypted file size increased exponentially with increase of input file size. The best result was shown by RSA.

[3] has discussed about the basic concepts and properties of homomorphic encryption. The paper stated that Paillier can be used for preserving the additive property of Homomorphic Encryption and RSA can be used for Multiplicative Property.

V. EXISTING SYSTEM

In the existing system, a part of plaintext is taken as a keyword. From the keyword, a key is generated using k2c algorithm .Using that key the entire plaintext is encrypted by paillier algorithm and stored in database. The wildcard search mechanism is used for searching an encrypted file in the database.

A. Paillier Algorithm:

It is a non-deterministic encryption system and based on the decisional composite residuosity assumption. The Paillier system could provide the confidentiality of the outsourced data in the cloud platform, and also realizes the Homomorphic properties.

The encryption performance of the Paillier cryptosystem, a partially Homomorphic cryptosystem that allows to perform sums on encrypted data without having to decrypt first. With a combination of both new and known methods, we increase the encryption performance by orders of magnitude compared to a naïve implementation. The new methods reduce the bottleneck of noise calculation by using pre-computed noise

to generate new noise in a much faster way than by using standard methods.

- Select two large primes, p and q.
- Calculate the product n=p x q, such that gcd(n,Φ(n)) = 1, where Φ(n) is Euler Function.
- Choose a random number g, where g has order multiple of n or gcd(L(gλ mod n2) n) = 1 where L(t)= (t 1) / n and λ(n)=lcm(p 1 q 1) yp p g gcd(L(g mod n), n) = 1,

where L(t)= (t-1) / n and

λ(n)=lcm(p-1,q-1).

- The public key is composed of (g, n), while the private key is composed of (p,q,λ).
- The Encryption of a message m < n is given by:
- c=gmr n mod n2
- The Decryption of cipher text c is given by
- m=(L(gλ mod n2)/L(gλ mod n2))modn

Encryption :

1) Let m Є Zn be the message

2) Choose r Є Z*n

3) Required Cipher text is c = g^m *r^n mod n^2

Decryption :

1) Compute m = L(c^λ mod n^2)*μ modn

K2C algorithm (Keyword 2 Ciphertext)

In order to encode a keyword to an element in ZN,we firstly transform each character in the keyword to its ASCII code. Then, the hexadecimal ASCII code is converted to a decimal. According to the position of the character in the keyword, each decimal is multiplied with a weight. Then, these weighted decimal numbers are added together and encrypted using PCTD algorithm. This algorithm is named as keyword to ciphertext algorithm (K2C)

Working procedure:

- Take a given keyword as plaintext.
- Convert it into ASCII code .
- After that convert it into decimal integer.
- Do multiply and add calculations and obtained results as big integer.
- Then,Encryption takes place.
- Finally,CiperText obtained as resuits.

To setup K2C, the root user needs to follow the steps listed below:

1. Sign up for cloud services required for hosting Meta-data Directory and Data Store.
2. Run Init procedure according to AB-HKU scheme to generate public parameters and the master key.
3. Save the master key and public parameters in the root's Key-store.
4. Share the public parameters with the cloud service providers that support K2C request authorization.
5. Create an entry in Meta-date Directory that corresponds to the root directory. The WAR and RAR numbers of the root directory entry are initialized to zero.

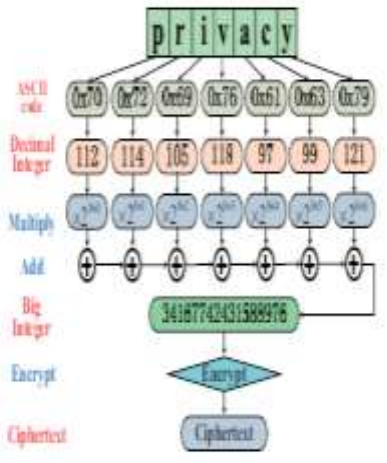


Fig 3 : K2C Algorithm

VI. PROPOSED SYSTEM

In the proposed system, we used same operations as that of the existing system. Here,k2c generates larger big integer and this is converted into two prime numbers .Using that the entire plaintext is encrypted by RSA algorithm and stored in database. This in turn generates key for downloading the file to whom we shared. Also the wildcard search mechanism is used for searching an encrypted file in the database without decrypting it with the help of Homomorphic encryption technique. By using RSA algorithm we can reduce the key space, encrypted file size to increase the throughput.

RSA Algorithm

RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024 bit keys could be broken in the near future. But till now it seems to be an infeasible task.

- Choose two very large random prime integers: p and q
- Computenandφ(n):
  n = pq and φ(n) = (p-1)(q-1)
- Choose an integer e, 1 < e < φ(n) such that: gcd(e, φ(n)) = 1(where gcd means greatest common denominator)
- Compute d, 1 < d < φ(n) such that: ed ≡ 1 (mod φ(n))

the public key is (n, e) and the private key is (n, d)

the values of p, q and φ(n) are private

e is the public or encryption exponent

d is the private or decryption exponent

Encryption

The cipher text C is found by the equation 'C = M^e mod n' where M is the original message.

Decryption

The message M can be found form the cipher text C ….. $M=C^d$ mod n.

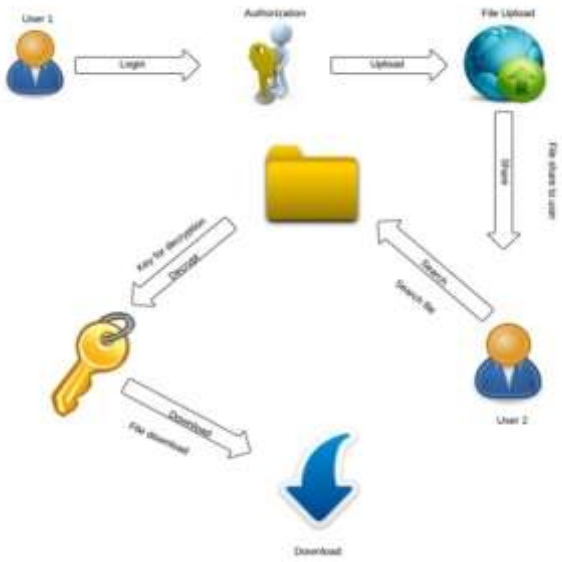


Fig 4 : Proposed System

Advantages of RSA Algorithm:

RSA algorithm is safe and secure for its users through the use of complex mathematics. RSA algorithm is hard to crack since it involves factorization of prime numbers which are difficult to factorize. Moreover, RSA algorithm uses the public key to encrypt data and the key is known to everyone, therefore, it is easy to share the public key

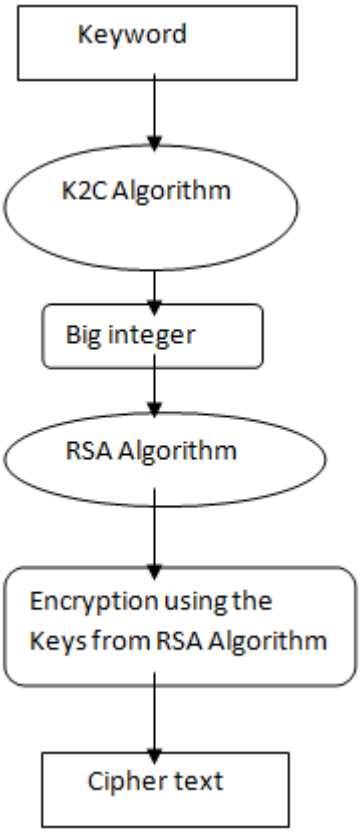


Fig 5: RSA Encryption using K2C

VII. PERFORMANCE COMPARISON

The table shows the comparison between the Paillier and the RSA Homomorphic Encryption algorithms[4]. It shows that RSA algorithm is comparitively better than the Paillier algorithm in terms of Key Size, File Encryption Size and Encryption Time. In Paillier the keys are generated randomly; wheareas in RSA, the keys are generated based on the Big Integer which is the output of K2C algorithm. And the time taken for encryption is more in Paillier when Compared to RSA.

	Paillier algorithm	**RSA algorithm**
ThroughPut	Paillier showed poor results in its throughput	RSA showed better throughput over Paillier.
Key Generation and size	In paillier,Randomly generates its key.	In RSA,key pair is derived from the product of two prime numbers chosen from the big integer.
File size	Paillier showed worst result in encrypted file size and its encrypted file size increased exponentially with increase of input file size.	Whereas RSA shown best results in its encrypted file size.

Encryption time	Paillier takes huge time in terms of encryption.	RSA showed better performance in term of encryption time.

Table 1: Comparison between Paillier and RSA algorithm:

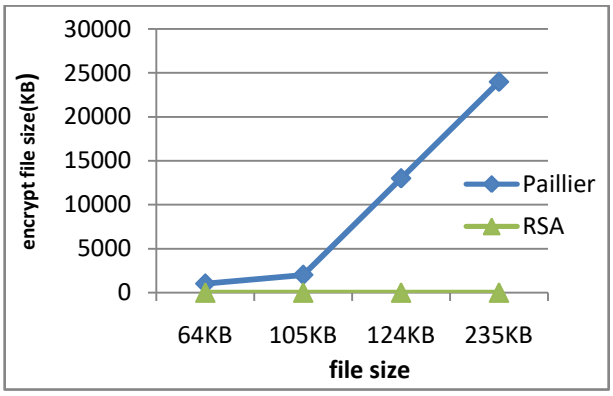


Fig 6: Comparsion of Decryption Time among Paillier and RSA algorithm

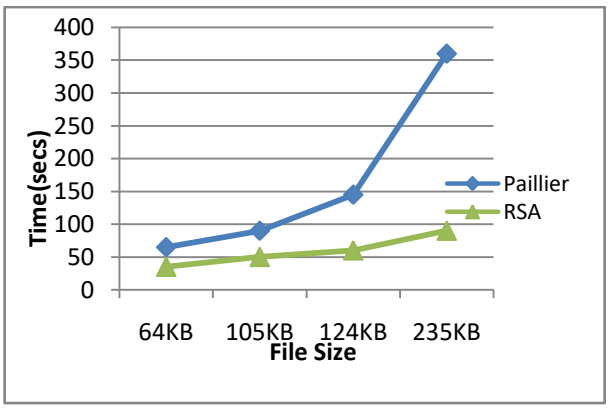


Fig 7: Comparison of Encrypted File Storage Space by Paillier and RSA

Fig 5 and Fig 6 shows the comparison in Decryption time and File Storage size between RSA and Paillier Homomorphic Encryption techniques. As the File size increases , the time of decryption for Paillier is increasing at a higher rate. But in RSA the increase in decryption time with the file size is comparatively less. Fig ---, displays the encrypted file size with that of the original file size, The encrypted file size is comparatively when using RSA algorithm

VII. CONCLUSION

In this paper, we used a wildcard searchable encryption for secure cloud storage and provide content search, user authentication and multiple file sharing. We proposed RSA algorithm along with K2C algorithm for secure data transmission. It increases the performance and efficiency of the system by reducing the key size and its encrypted file size compared with other Homomorphic encryption algorithm. Finally the proposed system eliminate the false probability and achieves high level of privacy in its data transmission.

REFERENCES

[1]. Kamara S, Lauter K. Cryptographic cloud storage[C]//International Conference on Financial Cryptographyand Data Security. Springer Berlin Heidelberg, 2010: 136-149.
[2]. Singh A, Chatterjee K. Cloud security issues and challenges: Asurvey[J]. Journal of Network and Computer Applications, 2017,79: 88-115.
[3]. SHAHZADI FARAH, M. YOUNAS JAVED, AZRA SHAMIM, TABASSAM NAWAZ "An experimental study on Performance Evaluation of Asymmetric Encryption Algorithms" Recent Advances in Information Science
[4]. Jarecki S, Jutla C, Krawczyk H, Rosu M, Steiner M. Outsourced symmetric private information retrieval. In Proc. ACM CCS 2013,pp. 875-888.
[5]. Sepehri M, Cimato S, Damiani E. Privacy-preserving query processingby multi-party computation. The Computer Journal, vol.58, no. 10, 2015, pp. 2195-2212.
[6]. Peter A, Tews E, Katzenbeisser S. Efficiently outsourcing multipartycomputation under multiple keys[J]. IEEE transactions on information forensics and security, 2013, 8(12): 2046-2058.
[7]. Shashi Mehrotra Seth, Rajan Mishra,Comparative Analysis Of EncryptionAlgorithms For Data Communication, IJCST Vol. 2, Issue 2, June 2011
[8]. Evaluation Of Performance Characteristics Of Cryptosystem Using Text Files.
[9]. Li J,Wang Q,Wang C, et al. Fuzzy keyword search over encrypted data in cloud computing[C]//INFOCOM, 2010 Proceedings IEEE.IEEE, 2010: 1-5.
[10]. Li J, Chen X. Efficient multi-user keyword search over encrypted data in cloud computing[J]. Computing and Informatics, 2013,32(4): 723-738.
[11]. Wang B, Yu S, Lou W, et al. Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud[C]//IEEE INFOCOM2014-IEEE Conference on Computer Communications. IEEE,2014: 2112-2120.
[12]. Bloom B H. Space/time trade-offs in hash coding with allowable errors[J]. Communications of the ACM, 1970, 13(7): 422-426.
[13]. Fu Z, Wu X, Guan C, et al. Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement[J]. IEEE Transactions on Information Forensics and Security, 2016, 11(12): 2706-2716.
[14]. Sedghi S, Van Liesdonk P, Nikova S, et al. Searching keywordswith wildcards on encrypted data[C]//International Conferenceon Security and Cryptography for Networks. Springer Berlin Heidelberg, 2010: 138-153.