Improving Performance Efficiency for Searching in the Partially Encrypted Indexed XML File Process

V. Sankar, [2]Dr. G. Zayaraz
[1]Research Scholar, Bharathiar University, Coimbatore, India.
[2]Professor, Department of CSE, Pondicherry Engineering College, Pudhucherry. India.

Abstract: XML format applied in system configuration, custom settings, web service and software metadata etc., Confidential XML data required to protect by encryption. RSA is a proven asymmetric encryption algorithm. Asymmetric encryption uses separate keys for encryption and decryption. Encrypting entire XML document leads performance degradation. Balancing between performance and security is a key parameter. Encrypting confidential data and other node value save as original format. This approach improves the searching performance.

Index Terms **- Encryption and decryption; XML security; XML searching; symmetric key; Asymmetric Key, XML File, XML Encryption, XML Signature.**

I. INTRODUCTION

Data to be secured when transferred over unreliable network [3], making data security a major concern in the digital world. Securing confidential data is from intruders is critical. Encryption address the security concerned of confidential data.

Encrypting entire XML file drastically degrades the performance. They needs to decrypt for every query processing entire data. Element-wise encryption used in XML data. Decrypt every encrypted elements leads performance degradation.

This paper proposes a balancing between performance and security. Storing confidential data element in encrypted format and non-confidential stored as plain text. The file has indexed and encrypted confidential data, rest will be stored as plain text. This system improves query performance in the encrypted XML.

The source data, index XML and web service hosted in the cloud environment. It improves the performance of huge XML files. The following definition and terms are used in this paper. In Encryption is the process of converting normal message (plaintext) into meaningless message (Ciphertext). It is a process of converting normal data into an unreadable form. It helps you to avoid any unauthorized access to data. In Decryption is the process of converting meaningless message (Ciphertext) into its original form (Plaintext). It is a method of converting the unreadable/coded data into its original form. In symmetric encryption, there is only one key, and all communicating parties use the same key for encryption and decryption. In asymmetric, or public key, encryption, there are two keys: one key is used for encryption, and a different key is used for decryption.

In Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. In XML Encryption specifies a process for encrypting data and representing the result in XML. In the XML Security standards define XML vocabularies and processing rules in order to meet security requirements.

The RSA public key cryptosystem was invented by R. Rivest, A. Shamir and L. Adleman. The RSA cryptosystem is based on the dramatic difference between the ease of finding large primes and the difficulty of factoring the product of two large prime numbers.

II. **Organization of this research paper**

This research paper is arranged as follows. Section II explains the related works. Section III provides the XML source and XML index file. Section IV shows architecture of the proposed searching technique and its algorithmic steps, Section V elaborates experimental study and result, and finally section VI Presents the conclusion and future work.

III. RELATED WORK

Encryption of an entire file, which is not desirable XML files. XML follows element based structure to store information [7]. Encrypt confidential data using RSA algorithm and leave rest of the element. Mathematical attack is possible because of factoring two prime numbers and brute force attack can be done by trying possible private keys [6]. Choosing large RSA keys are typically 1024 to 2048 bits long is impossible to break. Our solution makes use of index tables to allow for fast keyword and location queries [5].

Decrypting every instance of the encrypted element significantly reduces the performance of the XQuery [3]. Maintaining a separate index XML file along with unique-ID in plain text format improves the performance. The index file will be smaller than the original XML file.

Ravi Chandra Jammalamadaka, et al [9]this paper proposes techniques to query encrypted XMLdocuments..1) primitives using which a client can specify the sensitiveparts of the XML documents; 2) mechanisms to mapthe XML documents to encrypted representations that hides sensitive portions of the documents; and 3. techniques torun SPJ (Selection-projection-join) queries over encrypted XML documents.

RA. K. Saravanaguru, et al[10] aimed to evaluate the importance of XML Signature and XML Encryption for WS-Security. In today's e-business scenario, organizations are investing a huge amount of their resources in Web Services.

Hoi Ting Poon, et al[11] investigate the problem of processing a large amount of encrypted documents in XML-like formats where a user may wish to search or compute based on certain elements in the XML tree. Our solution makes use of index tables to allow for fast keyword and location queries.
Gu Yue-sheng, et al[12] formulates the XML signature and encryption as the core of web services security technology, and describes how to create and verify XML signature, how to encrypt and decrypt XML data. The application of XML signature and encryption in the Web services ecurity is illustrated.

Nithin N and Harshitha.K.S. , *et al[13]* this paper involve that The encryption machine takes the key Value and the input file (XML file) and generate the encrypted text using Symmetric algorithm (Caesar cipher and Vigenere cipher).The decryption Machine takes the encrypted file and key Value to generate the original XML file.

M. Preetha et al[14] this paper analysis Cryptography is a first abstraction to separate specific algorithms from generic cryptographic processes in order to eliminate compatibility and upgradeability problems. The core idea is enhance the security of RSA algorithm. In this dissertation public key algorithm RSA and enhanced RSA are compared analysis is made on time based on execution time.

IV. XML DATA

In Fig. 1. <PaymentDetails> node has online transaction details. The child nodes are classified two categories. <Payment>, <MerchantID>, <CustomerID>, <SalesID>, <SalesID>, <SalesValue> elements are not contains sensitive information. Encrypting entire document leads processing overhead. This data do not required to protect. <CardNo>, <Name>, <ExpiryMonth>, <ExpiryYear> and <CVV> elements contains payment details. Financial details are confidential. These category elements encrypted and index file generated for searching.

a. XML data as a plain text

The XML data contains financial payment data. It contains Merchant, Customer and payment details in the plain format.

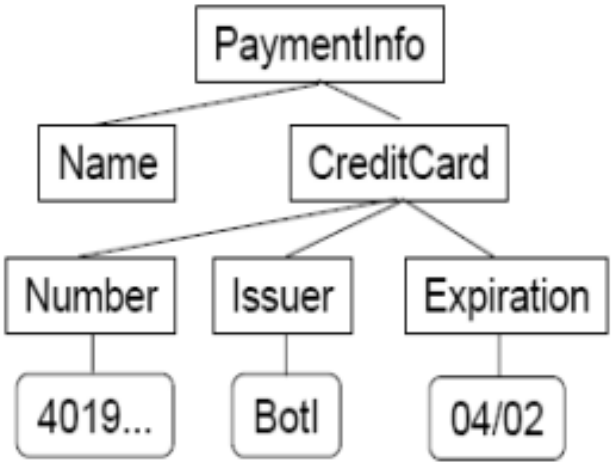
```
<PaymentDetails>
  <Payment>
    <PaymentID>648734434</PaymentID>
    <MerchantID>53485763</MerchantID>
    <CustomerID>64645434</CustomerID>
    <SalesID>201645834932</SalesID>
    <IssueBy>SBI Cards</IssueBy>
    <CardNo>5241826452643485</CardNo>

    <Name>Sankar V</Name>

    <ExpiryMonth>09</ExpiryMonth>
    <ExpiryYear>2022</ExpiryYear>
    <CVV>3254</CVV>
    <SalesValue>7850</SalesValue>
    <Currency>INR</Currency>
  </Payment>

    <Payment>
    <PaymentID>648734435</PaymentID>
    <MerchantID>53485763</MerchantID>
    <CustomerID>64535325</CustomerID>
    <SalesID>201645834933</SalesID>
    <IssueBy>ICICI Cards</IssueBy>
    <CardNo>5241114111786164</CardNo>
    <Name>Zayaraz G</Name>
    <ExpiryMonth>05</ExpiryMonth>
    <ExpiryYear>2025</ExpiryYear>
    <CVV>3211</CVV>
    <SalesValue>22500</SalesValue>
    <Currency>INR</Currency>
  </Payment>
  …….
</PaymentDetails>
```

Fig. 1. Payment data in plain XML format



b. XML data in the encrypted format

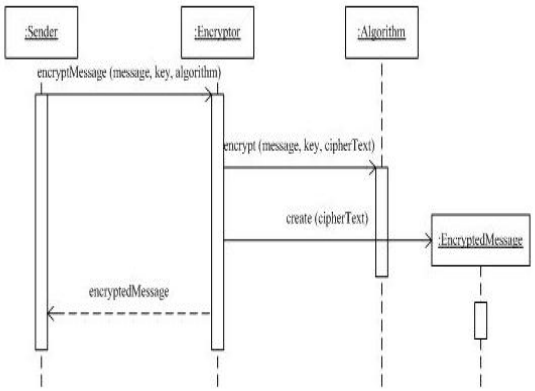


Fig. 1 is a source XML file without encryption. Encrypted XML data shown in the Fig. 3. By using RSA algorithm referred in the Fig. 2.

RSA algorithm

n = p*q where p and q are larger prime number, at least 1024 bits length.

φ = (p-1)*(q-1) where φ is the product of p-1 and q-1.

e = 1 < e < φ where e value is between 1 and φ

gcd(e, φ) = 1 where d is the greatest common divisor of e and φ.

d = 1 < d < φ where d value is between 1 and φ

public key = (n, e)

private key = (d, p, q)

Fig. 2. RSA asymmetric encryption

XML data encrypted by using public key n and e. Decryption parameters d, p, q and φ values are kept confidential. Encrypt all nodes in the XML file.

```
<PaymentDetails>
  <Payment>
    <Encrypt>
      <PaymentID>Encrypted Data</PaymentID>
      <MerchantID>Encrypted Data</MerchantID>
      <CustomerID>Encrypted Data</CustomerID>
      <SalesID> Encrypted Data</SalesID>
      <IssueBy>Encrypted Data</IssueBy>
      <CardNo>Encrypted Data</CardNo>
      <Name>Encrypted Data</Name>
      <ExpiryMonth>Encrypted Data</ExpiryMonth>
      <ExpiryYear>Encrypted Data</ExpiryYear>
      <CVV>Encrypted Data</CVV>
      <SalesValue>Encrypted Data</SalesValue>
      <Currency>Encrypted Data</Currency>
    </Encrypt>
  </Payment>
  <Payment>
    <Encrypt>
      <PaymentID>Encrypted Data</PaymentID>
      <MerchantID>Encrypted Data</MerchantID>
      <CustomerID>Encrypted Data</CustomerID>
      <SalesID> Encrypted Data</SalesID>
      <IssueBy>Encrypted Data</IssueBy>
      <CardNo>Encrypted Data</CardNo>
      <Name>Encrypted Data</Name>
      <ExpiryMonth>Encrypted Data</ExpiryMonth>
      <ExpiryYear>Encrypted Data</ExpiryYear>
      <CVV>Encrypted Data</CVV>
      <SalesValue>Encrypted Data</SalesValue>
      <Currency>Encrypted Data</Currency>
    </Encrypt>
  </Payment>
  …….
</PaymentDetails>
```

Fig. 3. Encrypted payment data in XML format

Encrypting entire XML data increases processing time of reading, writing and modification function. The <Payment>, <MerchantID>, <CustomerID>, <SalesID>, <SalesID>, <SalesValue> and <Currency> elements does not contain sensitive details. The Credit/Debit card details are should not access by unauthorized person.

c. Partially encrypted XML data

To improve the performance of encrypted confidential data, partially encrypt. The confidential data contains in <CardNo>, <Name>, <ExpiryMonth>, <ExpiryYear> and <CVV> elements. The elements are Partially encrypted payment data in XML format and stored into Fig.4.

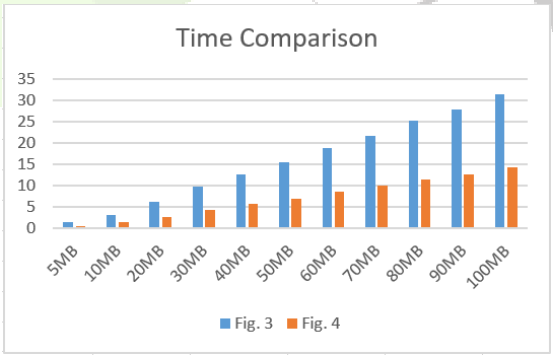
```
<PaymentDetails>
  <Payment>
    <PaymentID>648734434</PaymentID>
    <MerchantID>53485763</MerchantID>
    <CustomerID>64645434</CustomerID>
    <SalesID> 201645834932</SalesID>
    <IssueBy>SBI Cards</IssueBy>
    <Encrypt>
      <CardNo>Encrypted Data</CardNo>
      <Name>Encrypted Data</Name>
      <ExpiryMonth>Encrypted Data</ExpiryMonth>
      <ExpiryYear>Encrypted Data</ExpiryYear>
      <CVV>Encrypted Data</CVV>
    </Encrypt>
    <SalesValue>7850</SalesValue>
    <Currency>INR</Currency>
```

```
  </Payment>
  <Payment>
    <PaymentID>648734435</PaymentID>
    <MerchantID>53485763</MerchantID>
    <CustomerID>64535325</CustomerID>
    <SalesID> 201645834933</SalesID>
    <IssueBy>ICICI Banks</IssueBy>
    <Encrypt>
      <CardNo>Encrypted Data</CardNo>
      <Name>Encrypted Data</Name>
      <ExpiryMonth>Encrypted Data</ExpiryMonth>
      <ExpiryYear>Encrypted Data</ExpiryYear>
      <CVV>Encrypted Data</CVV>
    </Encrypt>
    <SalesValue>22500</SalesValue>
    <Currency>INR</Currency>
  </Payment>
  …….
</PaymentDetails>
```

Fig. 4. Partially encrypted payment data in XML format

XML File Size	Fig. 3	Fig. 4
5MB	1.47214	0.6237
10MB	3.18176	1.4458
20MB	6.15154	2.7507
30MB	9.69158	4.2689
40MB	12.68114	5.7187
50MB	15.52046	7.0093
60MB	18.95136	8.5688
70MB	21.81354	9.9607
80MB	25.26228	11.4374
90MB	27.85696	12.6168
100MB	31.51886	14.2813

d. Partially encrypted XML data



To reduce performance overhead, creating single index file Fig. 4. The index files has confidential N number of elements. Each elements corresponding <PaymentID>, stored as index key. <PaymentID> is unique to the all transaction node.

```
<PaymentDetails>
  <Payment>
    <CaPaymentID>648734434</CaPaymentID>
    <CardNo>Encrypted Data</CardNo>
    <NaPaymentID>648734437</NaPaymentID>
    <Name>Encrypted Data</Name>
    <EmPaymentID>648734454</EmPaymentID>
    <ExpiryMonth>Encrypted Data</ExpiryMonth>
    <EyPaymentID>648734476</EyPaymentID>
    <ExpiryYear>Encrypted Data</ExpiryYear>
     <CvPaymentID>648734454</CvPaymentID>
```

```
    <CVV>Encrypted Data</CVV>
  </Payment>
   <Payment>
    <CaPaymentID>648734435</CaPaymentID>
    <CardNo>Encrypted Data</CardNo>
    <NaPaymentID>648734443</NaPaymentID>
    <Name>Encrypted Data</Name>
    <EmPaymentID>648734422</EmPaymentID>
    <ExpiryMonth>Encrypted Data</ExpiryMonth>
    <EyPaymentID>648735433</EyPaymentID>
    <ExpiryYear>Encrypted Data</ExpiryYear>
     <CvPaymentID>648733543</CvPaymentID>
    <CVV>Encrypted Data</CVV>
  </Payment>
   ……
</PaymentDetails>
```

Fig. 5. Indexed encrypted payment data in XML format

The Fig. 4 is an indexed file, which contains confidential data in encrypted form and paymentID as plain text.

e. *Searching in XML*

Algorithm 1 Searching encrypted and plain text

```
1: Input:          {{element1,key1},op1,...{elementN,keyN}}
2:  Output: {List of nodes}
3:  foreach({element,searchkey} ← search) do
4:     if element data is plaintext then
5:       result ← Search(sourceFile)
6:     else
7:       result ← Search(indexFile)
8:     end if
9:     result.Append(result)
10: end for
11: if (result node satisfies operators) then
12:    Display Nodes
13: end if
14: end
```

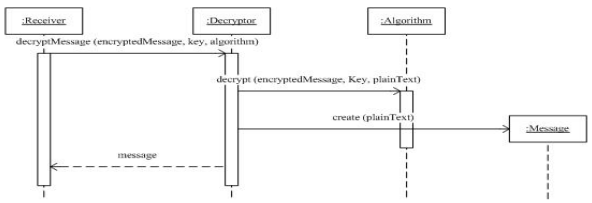


Fig. 5. Searching encrypted and plain text

The Fig. 4. XML parent node and element tags are not encrypted the data inside the elements. This approach reduce the amount of encryption and decryption. It improves the performance marginally.

Case 1: The search key element is non-confidential. Search type is exact keyword search or wildcard search. The search algorithm find the PaymentID in the source XML. Because the data stored as plain text. Encryption or decryption overhead will be is eliminated.

Case 2: The search key elements are confidential. Search type is exact keyword. The search key will encrypted and search algorithm find the PaymentID in the indexed XML. Comparing encrypted key against encrypted data in the index file. Decryption of each element in the index file is not required. This approach improves the efficiency.

Case 3: The search key elements are confidential. Search type is wildcard string. The data from indexed XML will decrypted and search algorithm find the PaymentID in the indexed XML. Comparing key against decrypted data in the index file. Decryption each element in the index file is unavoidable.

Algorithm 2 Searching text in the XML file

```
1: Input: xpath, Searchkey
```

```
2: Output: Node index
3: nodesList ← selectNodes(xpath)
4: midIndex ← 0
5: maxIndex ← Count(nodesList)
6: if Searchkey <> wildcard then
7:   Searchkey ← Encrypt(Searchkey)
8: end if
7:  while (midIndex <= maxIndex)
8:     midIndex ←  (midIndex + maxIndex) /2
9:     dataNode ← nodesList[midIndex]
10:    if Searchkey is wildcard then
11:      dataNode ← Decrypt(dataNode)
12:    end if
13:    if Searchkey = dataNode then
14:      return paymentID
15:    else if Searchkey < dataNode then
16:       maxIndex ← midIndex -1
17:     else if Searchkey > dataNode then
18:       midIndex ← midIndex +1
19:     end if
20:  end while
21: end
```

Fig. 6. Searching algorithm

Step 1 – Select nodes from Fig. for given xpath
nodesList ← *SelectNodes(xpath)*
Step 2 – Encrypt search key
encrptSearchKey ← encrypt(Searchkey)
Step 3 – Using binary search, elements in the Fig. 4.
Step 4 – If elements found in the list, returns unique paymentID
Step 5 – if search returns null, returns -1.

V. EXPERIMENTAL STUDY

The XML searching algorithm implemented in the Visual Sudio 2015 and C# program. The data indexed and encrypted by RSA. An XML file size is from 5 MB to 100MB. This experiment tested on i7 2.5GHz Quad core Intel processor and 16GB RAM, running on Windows 10 and 64-bit architecture.
The data stored in the Fig. 4 encrypted partially and ID encrypted in the Fig. 4. Table 1 time calculated for decryption and searching the proposed technique.

Table 1. Proposed algorithm

XML File Size	Proposed Algorithm	Existing Algorithm
5MB	0.2162	0.6237
10MB	0.5169	1.4458
20MB	0.8439	2.7507
30MB	1.3951	4.2689
40MB	1.8345	5.7187
50MB	2.3332	7.0093
60MB	2.7439	8.5688
70MB	3.2552	9.9607
80MB	3.6690	11.4374
90MB	4.1998	12.6168
100MB	4.5731	14.2813

We tested from 5MB XML and 100MB files Fig. 2 and Fig. 4. Test data contains simple and complex query and collected average test result.

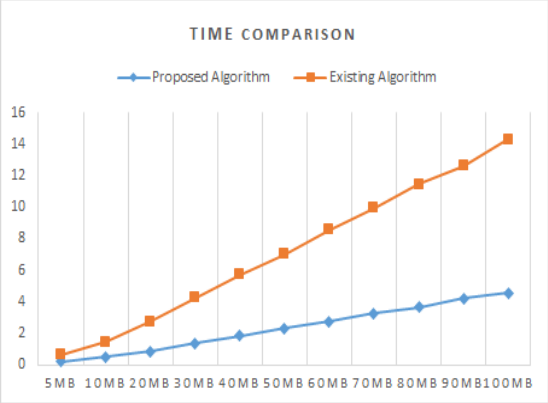


Fig. 7. Comparison between conventional and proposed system from 5 MB to 100 MB file.

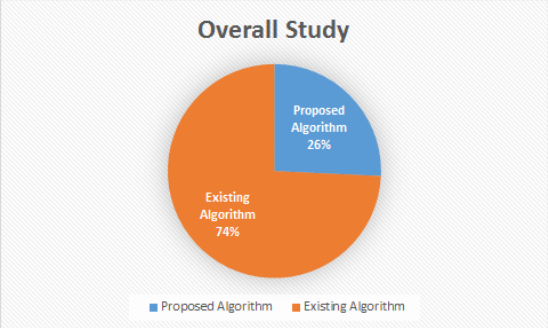


Fig. 8. Overall comparison between existing and proposed algorithm.

Algorithm1 ran 54% faster than fully encrypted XML file, shown in Fig. 7. The time were calculate by proposed partially encrypted index XML. Handling Fig. 2. Data by Fig. 4. Improves efficiency.

VI. Conclusion and Future Work

The partially encrypted document in XML indexed file with a low computational, communication and storage cost. Create a partially encrypted XML data with encrypted ID improves efficiency. Our solution minimize the number decrypted elements in the XML. The proposed algorithm reduces computational cost and network traffic. Choose larger prime pair to secure the encrypted data. Otherwise powerful computing hardware with efficient logic can break the encryption. We proposed to implement the technique in cloud platform to increase security and performance.

References

[1] X. Fang and W. Hongfu, " The research of asymmetrical encryption algorithm XRSA based on XML," IEEE Second International conference on Artificial Intelligence, Management Science and Electronic Commerce, pp. 3526-3529, Aug. 9-10, 201 l.

[2] S. C. Seak and N. K. Siong, "A file based implementation of XML encryption," IEEE fifth Malaysian conference on Software Engineering, pp. 418-422, Dec. 13-14, 2011.

[3] Azhar Rauf, Waqas Ali, Maher Ahmed, Shah Khusro and Shaukat Ali, "Efficient XQuery over Encrypted XML Documents, " in The 10th International Conference on Computer Science & Education (ICCSE 2015) July 22-24, 2015. Fitzwilliam College, Cambridge University, UK.

[4] Li Juan and Ming De-ting, "Research and application on the query processing for encrypted xml data,"in IEEE International Conference on Advanced Management Science,2010, pp. 707–711.

[5] Hoi Ting Poon and Ali Miri, "Computation and Search over Encrypted XML Documents", Department of Computer Science, Ryerson University, Toronto, Ontario, Canada.

[6] Anupkumar M Bongale, Dept. of Computer Engineering,"LRXE: Lite-RSA for XML Encryption Suitable for Computational Constraint Devices",Dr. D.Y. Patil College of Engineering, Ambi, Pune, India.

[7] Allan Delon Barbosa Araujo and Paulo Caetano da Silva "Middleware for Multiple Encryption in Web Services," in 2015 - 12th International Conference on Information Technology - New Generations.

[8] F. Kerschbaum, "Secure conjunctive keyword searches for unstructured text," in International Conference on Network and System Security, 2011, pp. 285–289.

[9] Ravi Chandra Jammalamadaka, Sharad Mehrotra, "Querying Encrypted XML Documents"Donald Bren School of Information and Computer Sciences University of California, Irvine, CA 92697, USA 10th International Database Engineering and Applications Symposium (IDEAS'06) 0-7695-2577-6/06 $20.00 © 2006

[10] RA. K. Saravanaguru1, George Abraham2, Krishnakumar Venkatasubramanian3, Kiransinh Borasia4 "Securing Web Services Using XML Signature and XML Encryption" School of Computer Science and Engineering,VIT University, Vellore, India.

[11] Hoi Ting Poon and Ali Miri "Computation and Search over Encrypted XML Documents" Department of Computer Science Ryerson University Toronto, Ontario, Canada, 978-1-4673-7278-7/15 $31.00 © 2015 IEEE.
[12] Gu Yue-sheng, Ye Meng-tao, Gan Yong "Web Services Security Based on XML Signature and XML Encryption" JOURNAL OF NETWORKS, VOL. 5, NO. 9, SEPTEMBER 2010. © 2010 ACADEMY PUBLISHER doi:10.4304/jnw.5.9.1092-1097.
[13] Nithin N and Harshitha.K.S. "Analysis of Symmetric algorithm for XML document security" International Journal of Innovations in Engineering and Technology (IJIET), Vol. 3 Issue 4 April 2014 , ISSN: 2319-1058.
[14] M. Preetha1, M. Nithya " A Study And Performance Analysis Of Rsa Algorithm" International Journal of Computer Science and Mobile Computing Section, IJCSMC, Vol. 2, Issue. 6, June 2013, pg.126 – 139.
[15] S. Zittrower and C. C. Zou, "Encrypted phrase searching in the cloud," in IEEE Global Communications Conference,2012, pp. 764–770.
[16] Yinqi Tang, Dawu Gu, Ning Ding, and Haining Lu, "Phrase search over encrypted data with symmetric encryption scheme," in International Conference on Distributed Computing Systems Workshops, 2012, pp. 471–480.
[17] He Tuo and Ma Wenping, "An effective fuzzy keyword search scheme in cloud computing," in International Conference on Intelligent Networking and Collaborative Systems, 2013, pp.786–789.
[18] Wenliang Du and Mikhail J. Atallah," Protocols For Secure Remote Database Access With Approximate Matching", 87–111,2001
[19] Bart Goethals, Sven Laur, Helger Lipmaa, and Taneli Mielikinen, "On private scalar product computation for privacy preserving data mining," in International Conference in Information Security and Cryptology. 2004, pp. 104–120, Springer-Verlag.
[20] A. Abdelsalam and A.Uounis, "Developing a Cryptosystem for XML Documents," IEEE second International conference on Computer Technology and Development, pp. 240-244, Nov. 2-4, 2010.
[21] N. Nithin and A. M. Bongale, "XBMRSA: A new XML encryption algorithm," IEEE International conference on World Congress on Information and Communication Technologies, pp. 567-571, Oct. 30 2012-Nov. 2 2012.
[22] Pascal Paillier, "Public-key cryptosystems based on composite degree residuosity classes," Lecture Notes in Computer Science, vol. 1592, pp. 223–238, 19992014 IEEE Global Conference on Wireless Computing and Networking (GCWCN)
[23] Brent Waters, Dirk Balfanz, Glenn Durfee, and D. K. Smetters, "Building an encrypted and searchable audit log," in Network and Distributed System Security Symposium, 2004
[24] Dale Gundersen Encryption http://www.w3 Requirements XML.Org/TR/XML - encryption req. 2005-05
[25] Abdel-Karim Al Tamimi , "Performance Analysis of Data Encryption Algorithms" http://www.cse.wustl.edu/~jain/cse567-06/ftp/ encryption_perf
[26] D. Luciano and Gordon Prichett, "Cryptology: From Caesar Ciphers to Public-Key Cryptosystems," The College Mathematics Journal, vol. 18
[27] S. T. F. Al-Janabi and M. A. Rasheed, "Public-Key Cryptography Enabled Kerberos Authentication," In IEEE conference on Developments in E-systems Engineering, pp. 209-214, Dec. 6-8, 201 l.
[28] H.M. Sun, ME. Wu, W.C, Ting, and M.J. Hinek, "Dual RSA and Its Security Analysis," IEEE TRANSACTIONS ON INFORMATION THEORY, vol. 53, no.8, Aug. 2007.
[29] Hongfu Wang, "The Research of Asymmetrical Encryption Algorithm XRSA Based on XML" School of Computer Science and Information, Engineer, Anyang Normal University, Anyang, China
[30] R. L. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of ACM,NY, USA, vol. 21, Issue 2, Feb. 1978.
[31] Nithin N and Anupkumar M Bongale, "XBMRSA:A New XML Encryption Algorithm.", Proceedings of Information and Communication Technologies (WICT), 2012 World Congress,pp 567-571,2012.
[32] T. Bray, J. Paoli, C. Sperberg-McQueen, Extensible markup language (XML) 1.0., Technical report, W3C Recommendation, 1998.
[33] D. Chamberlin, D. Florescu, J. Robie, J. Simeon, M. Stefanescu, XQuery: a query language for XML, Technical report, W3C Working Draft, February 2001.
[34] Nadeem2005]Aamer Nadeem , "A Performance Comparison of Data Encryption Algorithms", IEEE 2005.
[35] Ravi Varma1, Dr. G. Venkat Rami Reddy "Schema Based Parallel XML Parser: A Fast XML Parser Designed forLarge XML Files" ,International Journal of Computer Science and Mobile Computing, Vol.3 Issue.8, August-2014, pg. 379-389.
[36] Israt Jahan, Mohammad Asif, Liton Jude Rozario "Improved RSA cryptosystem based on the study of number theory and public key cryptosystems" American Journal of Engineering Research (AJER)e-ISSN : 2320-0847 p-ISSN : 2320-0936Volume-4, Issue-1, pp-143-149