IJCTA, 9(10), 2016, pp. 4819-4831

End to End Encryption by Algebraic OR/ XOR

***Geetanjali Rathee *Hemraj Saini**

Abstract : Wireless Mesh Network (WMN) is deliberated as a key technology due to its self-healing and self-configuring characteristics with the provisions of large scale exposure in industrial and academic fields. Security is considered as a vital constraint in WMN owing to its broadcasting and dynamic nature. Due to nature of WMN where information is being passed over multiple hops, data encryption is taken to be an important parameter. Researchers have proposed various encryption techniques to provide the message security, but foremost shortcoming in most of the approaches is their processing time. An encryption technique having large encryption/decryption timing increases overhead which may cause copious perilous attacks (*i.e.* passive eavesdropping etc.). Further an encryption technique with large file size may increase the load on the server during file transmission. In order to overcome these hitches, the manuscript proposes an end to end encryption based on algebraic operations i.e. Advanced Encryption through Homomorphic operation (AEHO) with reduced processing time where a cipher text is generated using OR/XOR operations. Further, a Trusted Party Authority (TPA) server is anticipated to provide the authenticity. To establish the legitimacy of the proposed solution, the experimental results are explained in terms of reduced encryption/decryption timing and increased throughput.

Keyword : Wireless Mesh Network; AEHO; NTRU; Network Security; Authentication

1. INTRODUCTION

WMN [1] is the most admired proxy technology for a last mile anchor for home, community and proximate networks. It comprises of mesh clients and mesh routers where clients are divided into different zones depending to their signal strength (as depicted in fig.1). In WMN, security [2-3] can be easily compromised due to its distributed, broadcasting and dynamic nature. If a node either inter-domain or intra-domain wants to send some message to destination node, information is passed among multiple hops. So, to prevent the data exposure at each intermediate node, message must be encrypted by some technique or an ornate encryption technique is requisite to guarantee that even if the message is forged by an attacker then it may not be able to decrypt it anyway. Encryption time is defined as the time required by a client to encrypt a message (to change the plaintext into cipher text).

Researchers have proposed numerous message encryption [4-5] techniques but the foremost shortcoming in most of the approaches is encryption/decryption timing. With the ease of data privacy, encryption/decryption delay is taken to be an important parameter. A large deferment encryption technique transpires a long delay, resulting passive eavesdrop, security threats [6]. A complex encryption technique may prevent the data from an attacker but may increase the size of file in the network. A technique having large file size may chock the server during transmission. So, there is a need to propose a technique which takes less encryption/decryption time and does not increase the load on server during file transmission from source to destination.

* Department of Computer Science and Engineering Jaypee University of Information Technology Waknaghat, India - 173234
geetanjali.rathee123@gmail.com, hemraj1977@yahoo.co.in

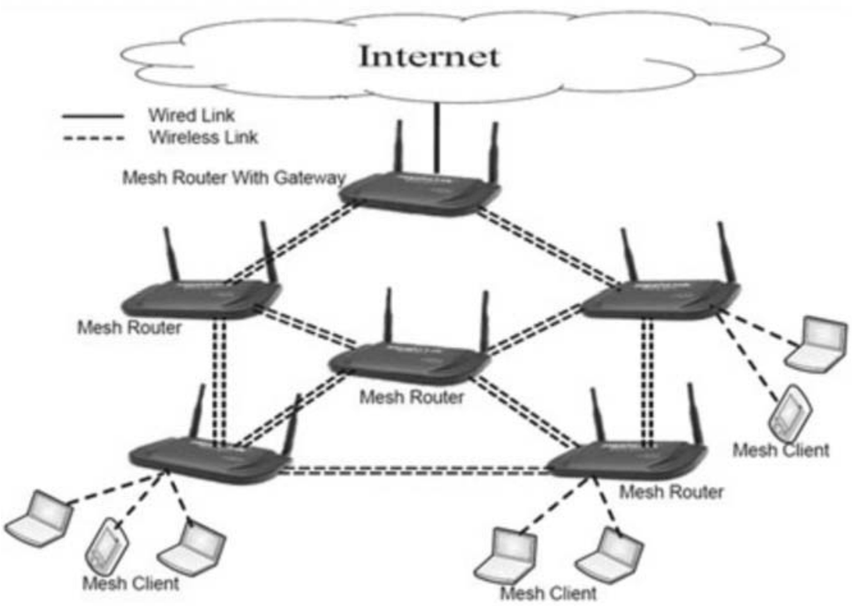

Fig. 1. Wireless Mesh Network

2. RELATED WORK

Multilevel [7] and end to end [8] encryptions are the 2 ways to encrypt the data, Soft encryption [9] and hard encryption [10] are the further categories to encrypt the data in WMN. Although numerous folks manipulated that end to end encryption WOULD work in WMN but nobody spoke with a firm that it would work due to the dynamic nature of WMN. Technique discussed by Li Xi [11] and NTRU [12] are multilevel encryptions where data is verified at each node and forwarded to next node after judging its integrity. Li Xi encrypts the client data using soft encryption in which author is encrypting the parameter of client MAC address using Pair-wise Master Key (PMK) which is generated by Authentication Server (AS) based on MAC address of access point from where the user authenticates itself for first time in WMN. The technique proposed by the author is infeasible because every time when a new client joins WMN then the client simply connects to the access point based on which message can be easily decomposed by black nodes. Further it is easy to decrypt as there is no encryption algorithm applied other than the MAC address of access point only.

On the other hand NTRU is a strong encryption technique which prevents the user data from an attacker and may not be easily decrypted. In this the message is represented in the form of polynomial ring based cryptosystem. But the major drawback of this approach is that it may increase the server load during transmission. The encrypted file size is very large using NTRU and may slow down the server during file transfer to destination node. Further Yahui Li [13] proposed an ID based broadcast encryption scheme where all the mesh routers which are selected to forward the data packets take part in trust domain and broadcast their transmission key to adopt the cryptographic protection on data packets.

The drawback with this approach is that all the intermediate mesh routers between source and destination will forward the data packet to next node after verifying with sender transmission key which means packet encryption or decryption at each node may increase the risk of different security threats *i.e*. DoS, passive eavesdrop and increase the encryption/decryption time. Although the researchers are able to resolve some limitations but major drawbacks in these approaches are that the data is exposed unblemished at each node and increases the processing time of encryption/decryption. Now, to remove the above limitations, several researchers felt that end to end encryption SHOULD work well in WMN because of its dynamic nature. Edward L. Witzke [14] gave some experiments to encrypt the data through IP Sec but the approach was not able to satisfy the challenges of RF shadow and shifting paths.

So, if existing techniques are considered then to optimize one parameter other parameters are affected adversely. Even though Li Xi encryption approach does not increase the load on server but it can be easily decrypted by applying some permutation and combination on the other hand NTRU approach resolves the Li Xi limitation but suffers from large size, further Yahui increases the encryption/ decryption time. Therefore, there is a need to propose an end to end technique which is resilient against these parameters. A brief summary of previous approaches is shown in table 1. In addition to that, in order to overcome the listed drawbacks, a new desired encryption technique is to be deliberated by using some of the existing methodologies i.e. homomorphic encryption [15], identity based cryptosystem [16] and quantum cryptosystem [17]. Homomorphic encryption is a direct arithmetic operation performed on a plaintext. It encrypts the plaintext by applying some algebraic operations, for example addition, subtraction and multiplication and outputs the results by decrypting the operations in reverse order.

A. Manuscript Contribution

This manuscript proposes an end to end encryption that has the advantage of not decrypting the data at each node. End to end encryption not only lessens processing overhead but also eradicates exposing the data unblemished at intermediate nodes. The proposed technique takes less encryption/decryption time with reduced file size and increased throughput. The approach is based upon two different techniques, i.e. polynomial ring cryptosystem (NTRU) and homomorphic encryption. NTRU algorithm is used to generate the private keys in order to send the data and to produce the cipher text in order to strengthen the security and lessen the possible attacks. While Homomorphic Encryption is used to further remove the limitation of NTRU i.e. reduce the file size by eliminating the white spaces from the file.

The structure of the paper is organized as follows. In section two, the taxonomy and background knowledge of the entire manuscript is discussed including polynomial based ring cryptosystem, homomorphic encryption, binary operations and network architecture. Section three deliberates the proposed technique i.e. Advanced Encryption through Homomorphic Operation (AEHO). Further, the performance evaluation of proposed technique in terms of encryption/decryption time and throughput is debated in section four and an empirical study is given in this section only. Finally section five concludes the paper.

Table 1. Previous Approaches Comparison

Author	*Intermediate/ end-to-end encryption*	*Type*	*Technique used*	*Limitation*
Li Xi [9]		Soft encryption	Encrypt AP MAC address through PMK	Decryption is easy
Jeffrey [10]	Multilevel	Hard encryption	Polynomial based ring cryptosystem	Increase server load, decrease throughput
Yahui Li [11]	Encryption	Hard encryption	ID based cryptosystem	Increased processing time, authentication delay
Edward L. [12]	End to End encryption	Hard encryption	IP Sec function	Security threats

3. TAXONOMY

This section discusses the background knowledge, terms, concepts and additional assumptions of this manuscript.

A. Polynomial Ring Cryptosystem (NRTU)

In the designing of proposed scheme, a polynomial based ring concept *i.e*. NTRU algorithm [18] is being utilized. The network is divided into a number of domains. Each domain has fixed parametric values in order to encrypt the plaintext message. The message M is coded in binary and represented by polynomial p as:

$P = (Z[M]/(M^N - 1)$, Where, z is the integer and N is the number of degree.

Since NTRU eliminates the degree of data exposure at each node and security threats but the major drawback of this approach is that it may increase the file size after encryption because text may include multiple white spaces which increases the file size. The proposed approach removes this drawback by eliminating the white spaces through homomorphic operation and is able to reduce the file size during transmission.

B. Homomorphic Encryption

Homomorphic encryption [19-20] is defined as into and onto mapping of a specific algebraic operation performed on plaintext equivalent to another algebraic operation performed on cipher text. It can be applied on a numeric value or non-numeric value. The example of homomorphic encryption is shown in fig. 2. Homomorphic encryption is the one where algebraic operations are involved consistently between plaintext and cipher text. Operations applied on cipher text will change the plaintext accordingly. Let E is the encryption operation and D is the decryption operation performed over two integers x and y. The operations of the corresponding plaintext and cipher text will be shown as below:

$$E\text{ (algebraic operation }(E(X) + E(Y)) = E\,(E(X+Y))$$

$$E\text{ (algebraic operation }(E(X)*E(Y)) = E\,(E(X*Y))$$

There exist three types of homomorphic encryptions. Partial homomorphic encryption which accomplishes one operation, *i.e.* multiplication or addition but not both at the same time on encrypted data. Somewhat homomorphic encryption is the one which executes more than one operation, but suppers only a limited number of addition and multiplication operations. Fully homomorphic encryption sustains both addition and multiplication by computing any function.

C. Binary Operations used in Homomorphic Encryption

Addition operation is used to add two integer functions *x* and *y* of length l bit. Each integer's function is firstly converted in binary form and then addition operation is performed bit by bit.

$$X = X_1, X_2, X_3\ldots \ldots X_1$$

$$Y = Y_1, Y_2, Y_3\ldots \ldots Y_1$$

$$X + Y = (X_1 + Y_1 + C_{(l-1)}) \ldots \ldots \ldots (X_1 + Y_1 + C_0)$$

Multiplication operation is used to multiply two integer functions x and y of length l bit. Each integer's function is added after being converted into binary form and then multiplication operation is performed using addition.

$$X = X_1, X_2, X_3\ldots \ldots X_1$$

$$Y = Y_1, Y_2, Y_3\ldots \ldots Y_1$$

$$X * Y = (X_1 * Y_1 * C_{(l-2)}) \ldots \ldots \ldots (X_1 * Y_1 * C_0)$$

In this manuscript OR and XOR operations are used in order to brace the encryption process. OR operation performs addition during the process. Let integers *x*, *y* and *z* of length l bit, where $X = xl, \ldots, x_2, x_1$, $Y = yl, \ldots \ldots y_2, y_1$ and $z = zl, \ldots z_2, z_1$ and perform OR operation. Every bit of *x*, *y* and *z* is added along with carrying on the previous stage. XOR is a universal operation. The output will be false in case of similar bits else true.

D. Network Architecture

In proposed model, a hierarchical architecture of WMN is considered which consists of three layers. The top most layer is of mesh servers which supply the internet service connectivity to stratum layer and TPA server which verifies the authenticity of the client nodes. The second layer consists of domain servers and mesh routers which forward the traffic to the main server and the third layer comprises of mesh clients which utilize the internet services.

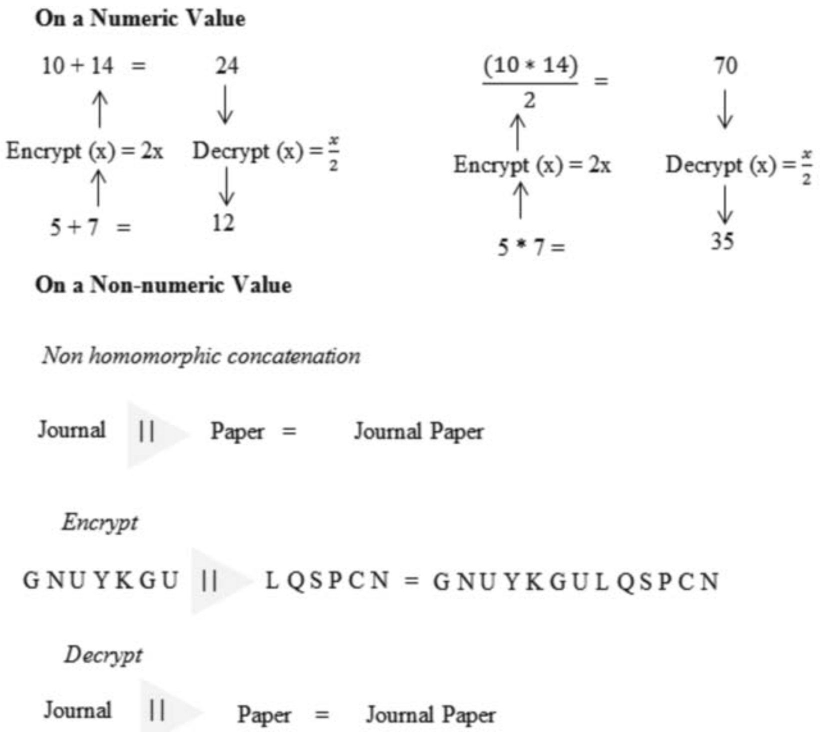

Fig. 2. Homomorphic Encryption

As shown in fig. 3, the network N is divided into different domains. Each domain has its own mesh server and mesh client. The purpose of dividing the network N into number of domains is to provide the services to the clients in continuous form or to reduce the load and waiting time of the clients in the network.

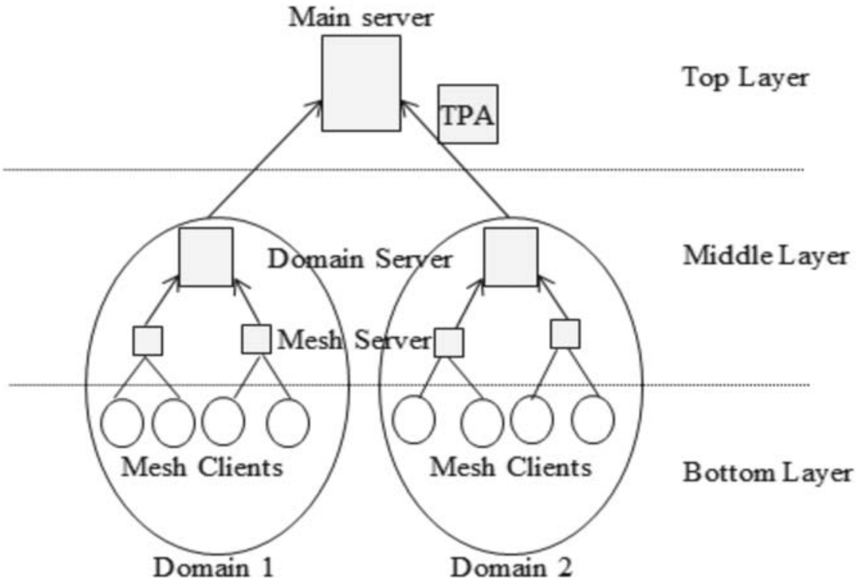

Fig. 3. Network Architecture Model

4. PROPOSED APPROACH

In this section, the proposed scheme is explained by dividing it into different number of phases. The two phases of AEHO are Authentication verification and encryption phase. Each phase clarifies the stepwise communication of proposed work. At last the entire working of communication is elucidated in phase three. The abbreviations used throughout the manuscript are depicted in table 2.

Table 2. Abbreviations Meaning

Abbreviations	***Meaning***
TPA	Trusted Party Authority
Sr	Source
DS/d_i	Domain Server
C_i, C_j	Clients
Auth$_{rrqt}$	Authentication Request
Pu	Public keys

A. Authentication Verification Phase

The purpose of this phase is to identify the legitimacy of the client nodes. TPA server is responsible to check the authenticity of each client for the first time. Whenever a client makes a request of data encryption to the domain server, the primary task of the server is to check the legitimacy of the node. The following steps describe the details of authentication verification.

Step 1 : Whenever a source client ‘Sr’ makes a communication request to its Domain Server (DS), the primary task of the domain server is to check the authenticity of the client node. For this purpose, DS will receive the source request and pass it to TPA server.

Step 2 : After getting the client’s request, TPA will send its address to DS which will forward the TPA’s address to the client. Now, the direct communication starts between TPA and the client.

Step 3 : TPA server sends acknowledgement at least 10 times for checking the authenticity of the client (if TPA gets 5 or 6 replies as an average communication out of 10 that means it is authentic client otherwise not).

Step 4 : Client will respond to each request of TPA by sending its IP address, user name, password, MD5 (MAC address) and keyword (yes).

Step 5 : As the client’s response matches with the TPA’s format, it will send SHA key as a response to both client and DS individually so that whenever the client presses encrypt button, then message with the SHA key will be sent to the domain server to cross verify the authenticity of the client. The steps discussed above are shown in fig. 4. An algorithm for the authentication verification process is described in table 3. Now, after verifying the authenticity of the client, encryption process starts. The next phase describes the encryption steps using homomorphic operation.

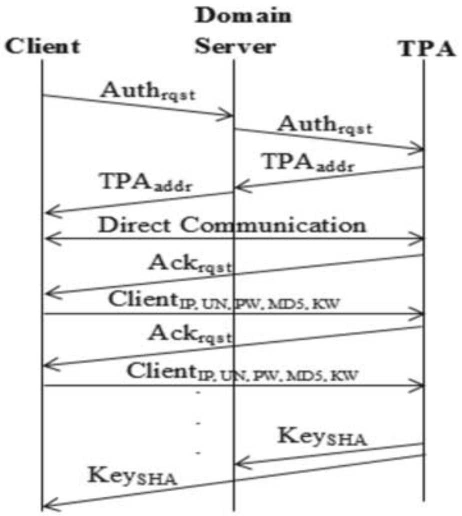

Fig. 4. Authentication Verification Steps

B. Encryption Phase

The plaintext message M will be converted into cipher text using the steps depicted in fig. 5. In this, initially the message is separated into an array and typecast into decimal form in order to apply the NTRU algorithm which generates the private key ‘pr’ through which message will be passed from source ‘Sr’ to destination and OR/XOR operations are applied to cipher text the message M.

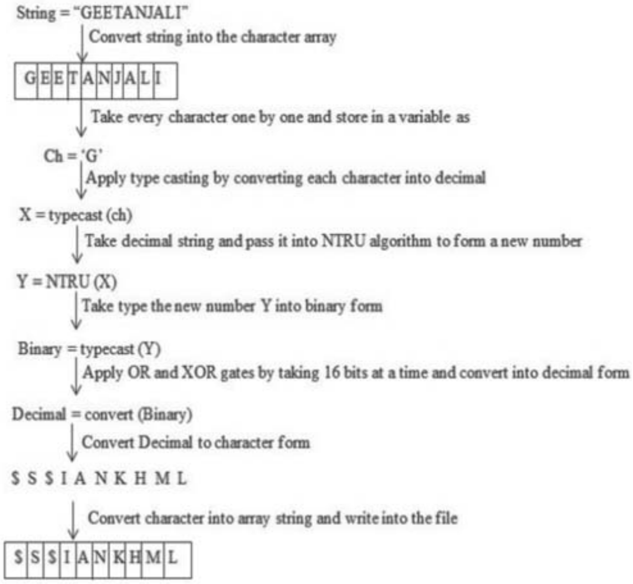
String = "GEETANJALI"
Convert string into the character array
G E E T A N J A L I
Take every character one by one and store in a variable as
Ch = 'G'
Apply type casting by converting each character into decimal
X = typecast (ch)
Take decimal string and pass it into NTRU algorithm to form a new number
Y = NTRU (X)
Take type the new number Y into binary form
Binary = typecast (Y)
Apply OR and XOR gates by taking 16 bits at a time and convert into decimal form
Decimal = convert (Binary)
Convert Decimal to character form
$ S $ I A N K H M L
Convert character into array string and write into the file
$ S $ I A N K H M L

Fig. 5. Secure Communication Steps

The complete process for encryption/decryption is depicted in table 4 which describes how a plaintext message M is converted into cipher text and the message is transmitted to the destination node by encryption with ‘pr’ key and whereas destination node decrypts the message further to get the plaintext.

C. Working of above Discussed Approach

The complete working of the proposed solution is described in a number of steps.

Step 1 : Whenever a client ci wants to communicate with a client cj, ci will contact to its domain server di which will further contact with the main and the TPA servers. The purpose of main server is to randomly generate two public keys and distribute them to individual clients while the purpose of TPA server is to prove the authenticity of clients the through authentication verification phase.

Step 2 : Both clients will exchange public keys in order to generate their private keys using the NTRU algorithm. The purpose of NTRU algorithm is to generate the private keys and one of the Pr key is used for cipher text communication and acts as a second bit during OR/XOR operation.

Table 3. Algorithm of Authentication Verification

Algorithm 1: Authentication Verification

```
1. Client $c_i$ will send a request to its domain server .
2. After getting the TAP address, $d_i$ forward the TAP address to client .
//Direct communication starts between $c_i$ and TAP
For ( $i$ = 1 to 10)
1. TAP send acknowledge ack to $c_i$.
2. $c_i$ will respond by sending its username, IP address, password, MD5(MAC), keyword(yes) to TAP
3. If (Response( ) > 4) then
```

```
        $c_i$ is authentic and TAP send SHA key to  as well as to Domain Server
    End If
    Else
    $c_i$ is not authentic
    End Else
    //whenever $c_i$ contact to $d_i$ for communication
    4. $d_i$ will verify the SHA key of its own with client SHA
    If ( SHA$d_1$= = SHA$c_i$ ) then
    Authentication successful and allow encryption
    End If
    Else
    Not Authentic
    End Else
    End For
```

Step 3 : Client ci will cipher text the message by following the steps as depicted in fig. 5. It will send the cipher text message by encrypting it with its pr key. Destination client *cj* will decrypt the message M using its pr key by following the process of fig. 5 in reverse order.

Table 4. Homomorphic Encryption Algorithm

```
Algorithm 2 : Homomorphic Encryption Algorithm
--------------------------------------------------------------------------
1. The main server will randomly generate two Pu keys and distribute to the corresponding clients.
2. On source client following homomorphic encryption algorithm will perform.
--------------------------------------------------------------------------
Input // Both ak and bk are the values of the selected domains
Input of A; ak and Input of B ;bk
Output: Lk + 1
--------------------------------------------------------------------------
Begin
 1.  Ck1 + 1 ← Generate Candidate key (Lk) // Both A and B calculate their private keys through public keys
     generated by the main server.
     For (A+1) do // loop continue till the content of the file length
 2.  A (plaintext) // for encryption process
 3.  W1 ← count(L) // extract the string via tokenization from line
 4.  T ← ak and bk // t is the cipher text
 5.  $\alpha$ ← Epk (cipher text) // conversion of cipher text via OR and XOR operations
 6.  Send to B ($\alpha$) // cipher text file send to the B
 7.  B (Cipher text) // for decryption process
 8.  Wpk ← count(Dpr)
 9.  T ← ck3(A) // compute pr key via A pu key
10.  B ← Epk(pass cipher text file)
11.  T ← homo comparison // apply tokenization on file that deduct private key from content of the file
12.  Send To A(T)
13.  (A)
```

```
R ← Dsk(T) // apply operations on content
File decrypt successfully
End if
Else
file cannot be decrypted
End for
End
```

D. OR/XOR Operation

This section describes the OR/XOR process in detail. The private key ‘pr’ acts as a second bit operator in order to perform the operation which will be selected by the server. The cipher text file generated in phase 1 is converted into binary form to perform the algebraic operations.

Initially, the whole file is divided into 16 bits binary form and again 16 bits are divided into 8-8 bits. The purpose of dividing 16 bits into 8-8 is to make the decryption operation complex. First 8 bits will perform OR operation using fixed 8 bit pr key generated by NTRU algorithm. The answer is then XOR with 8 bits pr key and finally the result is converted into decimal form. The operation is explained in table 5.

Table 5. OR/XOR Operation

1. Let plaintext P is	A
2. The ASCII value of P	65
3. Let pr key generated by NTRU is	10
4. Decimal number generated after addition is	75 (65+10)
5. Let Binary form of 75 is	1101(75 in binary form)
6. A constant no. selected by the server is	1011 (constant number)
7. Number generated after OR operation is	1111 (after OR operation)
8. Then XOR with constant number is	1011 (constant number)
9. XOR operation result is	1010(after XOR operation)
10. Covert cipher text generated into decimal	(89)8 (cipher text)

5. PERFORMANCE ANALYSIS

To prove the authenticity of proposed work, the technique is analyzed using java and is compared with existing approaches through performance metrics including authentication delay, encryption/ decryption time and throughput. Authentication Delay is defined as how much time an algorithm takes to check the authenticity of the client. Encryption Time is the time taken to convert a plaintext into corresponding cipher text. Decryption Time is the time taken to convert the cipher text message into plain text and finally the Throughput is the total time taken to transfer the number of packets to destination node. The corresponding formulas of all the mentioned parameters are defined as Encryption Time = $\sum_{i=1}^{n}\frac{\text{Encryption Time}}{\text{file size-number of bytes per KB}}$, Decryption Time = $\sum_{i=1}^{n}\frac{\text{Decryption Time}}{\text{file size-number of bytes per KB}}$ and throughput is defined in terms of file transmitted (in terms of bytes) in seconds. Let us discuss each parameter in detail with their graphical representation.

The defined parameters are evaluated against Yahui for encryption/decryption timing and throughput against Jaffrey technique.

A. Encryption/Decryption Time

In order to strengthen the proposed technique, individual encryption/decryption time of different file sizes is calculated. Encryption time is evaluated as the time required to convert a plain text into corresponding cipher text. Now, it can be scrutinized from fig. 6 and 7 that encryption timing ratio of proposed approach is less as compared to Yahui. The below fig. 6 and 7 show the corresponding encryption time of both the approaches on small and large file sizes.

- **Analysis of Encryption/Decryption**

This parameter is analysed over Yahui technique where all the intermediate mesh routers between source and destination forward the data packets after verifying them with sender transmission key. The encryption/decryption process is repeated at each step while in proposed approach there exist an end to end encryption which takes less processing time and improves security. Further, fig. 8 shows the decryption timing ratio of different file sizes over both the approaches. In proposed approach the time required for decryption is always less than encryption because cipher text needs to just follow the reverse process of encryption in order to get the plain text and there is no need to convert the strings into character and write each character in an array which is the main cause of delay.

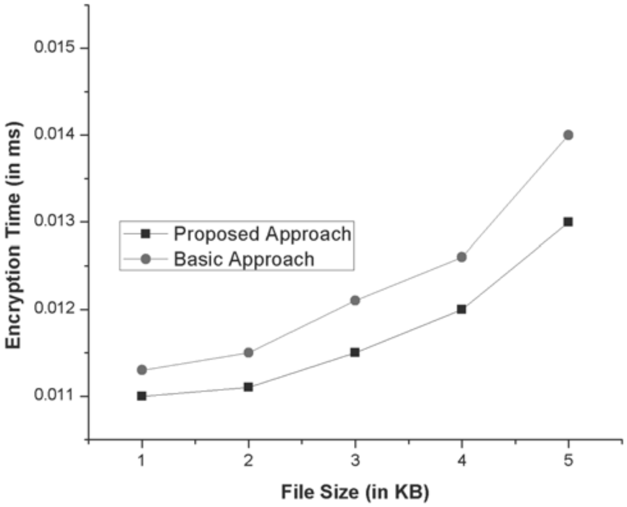

Fig. 6. Encryption Time (Over Small File Sizes)

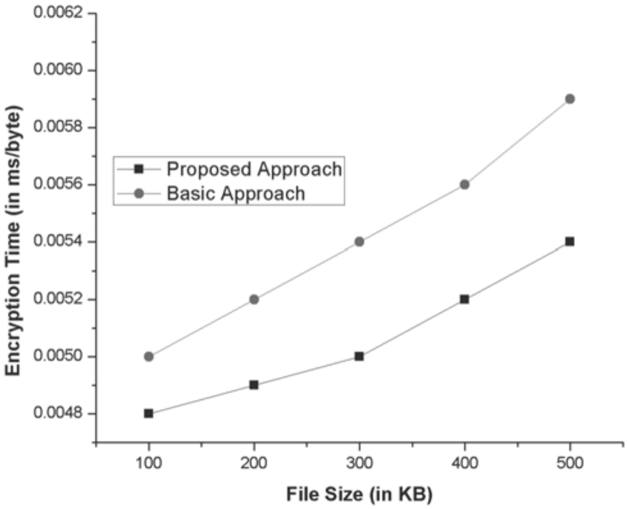

Fig. 7. Encryption Time (Over Large File Sizes)

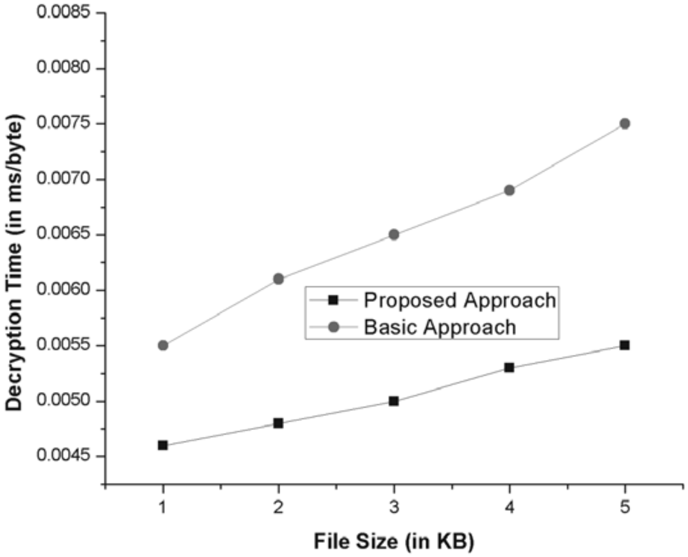

Fig. 8. Decryption Time (Over Small File Sizes)

B. Throughput

Fig. 9 shows the throughput graph in terms of file transmission in seconds. The proposed approach outperforms basic technique because of transmitting bytes in seconds.

- **Analysis of Throughput**

The file size of proposed approach is reduced to some extent by eliminating the white spaces which are the main cause of throughput (in terms of bytes transfer in seconds) decrement. In our case as compared to existing approach, the file transmitting time is reduced in seconds.

C. Empirical Proofs

- **Authentication**

All the Mesh Clients authenticate them with their domain server through TPA before communicating with other nodes. To reduce the authentication latency, there exists a direct communication between the TPA and the clients. Client authentication is done by getting a key SHA from the TPA.

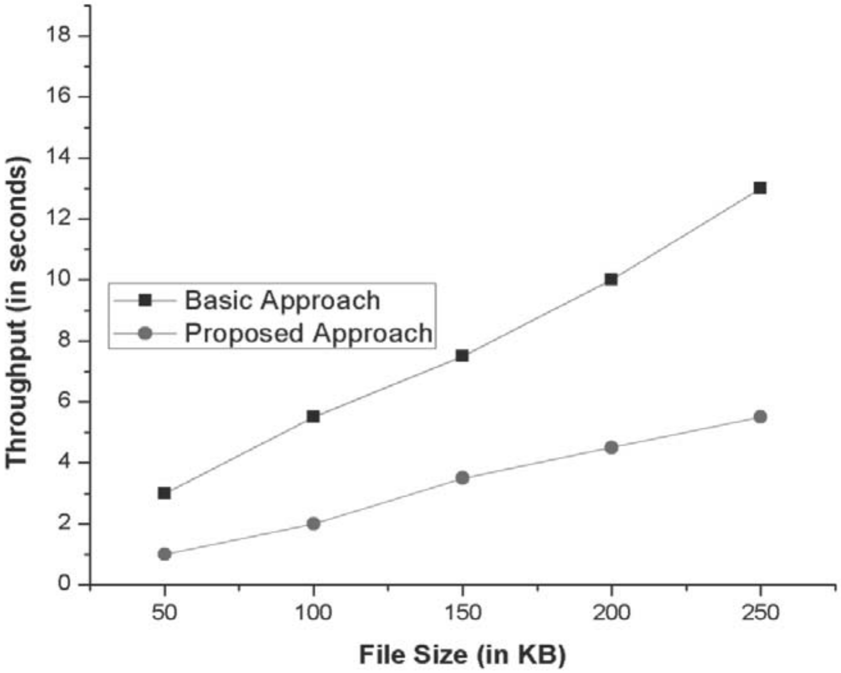

Fig. 9. Throughput

The Authenticity of client nodes before communication may reduce the issues of certain security threats, i.e. passive eavesdrop, traffic analysis.

Start

M_c, D_s, TPA are the mesh clients, domain servers
and Trusted Party Authorityserver respectively.
Send M_c (X request $\rightarrow$ D_s, where X request includes
Authentication request)
Pass D_s (M_c (X request)) $\rightarrow$ TPA
Send TPA $\rightarrow$ M_c (D_s (TPA_{addr}))
Send M_c (IP, UN, PW, MD5, KW) $\rightarrow$ TPA
TPA $\rightarrow$ Mc (Ds(Key_{SMA}))

End

- **Processing Delay**

End to End encryption reduces the overall encryption/decryption process between source and destination. The proposed technique uses end to end encryption process where client node encrypts the message through algebraic operations by passing it through multiple nodes. The data will be decrypted only after reaching at destination node. There will be no multiple encryption decryption process at each node which is the main cause of processing delay as well as security threats.

- **Attack Resistant**

The proposed technique is encrypted through OR/XOR operations. The second evaluating key for both OR and XOR operations will be selected by server only and is difficult to guess. Further even if the message is forged by an attacker then it will not be possible to decrypt it because it will be very difficult to guess the pairs of OR/XOR operations (*i.e*. in our case, firstly the file is broken into 16*16 then into 8*8 bits) (as depicted in fig. 10). Finally as there is no spacing in the encrypted file so it will be difficult to get the original text.

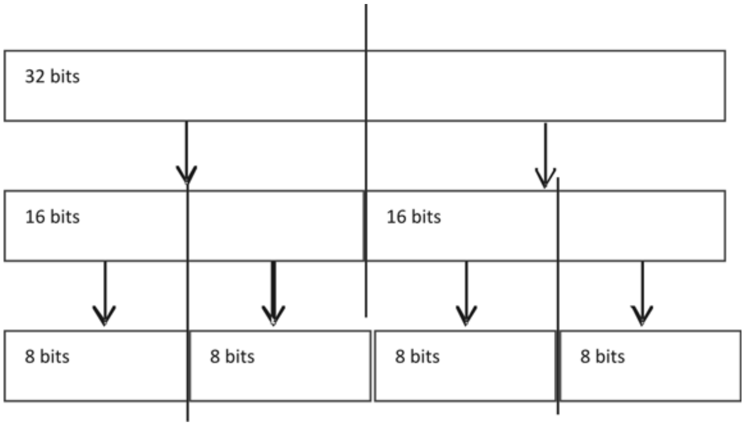

Fig. 10. Binary Files Separation Process

6. CONCLUSION

In this paper, an algebraic based encryption technique is proposed which ensures the security with reduced encryption/decryption time, processing delay and increased throughput. The key idea of the proposed technique is to generate the cipher text message using OR/XOR operations and transmit it with the private key generated by NTRU algorithm. Compared with existing approaches, the proposed technique reduces the processing delay and encryption/decryption time for different file sizes. Further, the proposed technique is compared in terms of throughput. According to the experimental results, our technique outperforms better in terms of throughput and processing delay with an efficient level of security.

7. REFERENCES

1. Akyildiz, Ian F., and Xudong Wang. "A survey on wireless mesh networks." Communications Magazine, IEEE. Vol. 43, no.9, pp. S23-S30, 2005.
2. A. A. Franklin and C. S. R. Murthy. "An introduction to wireless mesh networks". Security in Wireless Mesh Networks (book chapter), CRC Press, USA, 2007.
3. Lee, C. wei. "Security in Wireless Mesh Networks." Wireless Network Security. Springer Berlin Heidelberg, pp. 229-246, 2013.
4. B. Eli and A. Shamir, "Differential cryptanalysis of the data encryption standard." Springer Science & Business Media, 2012.
5. Mukherjee S, Sanyal G, Koner C. A Novel Approach for Authentication Technique in Wireless Sensor Network. International journal of communication and networking system. vol. 2 no. 1, 2013.

6. Ravichandran S. Secured identity based approach with privacy preservation for wireless mesh networks. International journal of communication and networking system. vol. 1, no. 2, 2012

7. L. Daniel . "Multi-level encryption access point for wireless network." U.S. Patent No. 6,526,506. 25 Feb. 2003.

8. .Radia. S. R. Hanna, and Y. K. Elley. "Content screening with end-to-end encryption." U.S. Patent No. 6,636,838. 21 Oct. 2003.

9. Narula, Prayag, et al. "Security in mobile ad-hoc networks using soft encryption and trust-based multi-path routing." Computer Communications vol. 31, no.4, pp. 760-769, 2008.

10. B. Dan, and M. Franklin. "Identity-based encryption from the Weil pairing." Advances in Cryptology-CRYPTO 2001. Springer Berlin Heidelberg, 2001.

11. X. Zhang, L. Guangsong and H. Wenbao. "Ticket-Based Authentication for Fast Handover in Wireless Mesh Networks." Wireless Personal Communications, vol. 85, no.3, pp. 1509-1523, 2015.

12. J. Hoffstein, P. Jill, and H. Joseph. "NTRU: A ring-based public key cryptosystem." Algorithmic number theory. Springer Berlin Heidelberg, pp. 267-288, 1988.

13. Y. Li, , X. Cui, L. Hu, Y. Shen. "Efficient security transmission protocol with identity-based encryption in wireless mesh networks." IEEE International Conference on High Performance Computing and Simulation (HPCS), 2010.

14. L. Edward , P. Joseph, K.L. green, L.E. Riblett, J.M. Wisemanl. "Encryption in mobile wireless mesh networks." IEEE International Carnahan Conference on Security Technology (ICCST), 2012.

15. V. D. Marten et al. "Fully homomorphic encryption ove r the integers." Advances in cryptology EUROCRYPT, Springer Berlin Heidelb erg, pp.24-43, 2010.

16. D. Boneh and M. Franklin. "Identity-based encryption from the Weil pairing." SIAM Journal on Computing, vol. 32 no.3, pp.586-615, 2003

17. I.S. Amiri et al. "Cryptography s cheme of an optic al switching system using pico/femto second soliton pulse." International Journal of Advances in Engineering Technology (IJAET), vol 5. No. 1,pp.176-184, 2012

18. Y. Mote, N. Paritosh and G. Shekhar. "Sup erior Security Data Encryption Algorithm (NTRU)." International Journal of Engineering Sciences vol. 6, 2012

19. D. Van, Marten et al. "Fully homomorphic encryption over the integers." Advances in cryptology EUROCRYPT Springer Berlin Heidel-berg, pp- 24-43, 2010

20. C. Gentry, H. Shai and P. S. Nigel. "Fully homomorphic encryption with polylog overhead." Advances in Cryptology EUROCRYPT, Springer Berlin Heidelberg, pp. 465- 482, 2012.