ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Key Management Using Newton's Interpolation Method for MANET

[I]K.R. Ramkumar, [II]Dr. C.S.Ravichandran
[I]Associate Professor, Sri Venkateswara College of Engineering, Chennai
[II]Professor and Dean, Dept. of EEE, Sri Ramakrishna College of Engineering, Coimbatore

Abstract
The wireless medium is uncovered , vulnerable to different routing attacks[2] and intruders can hack[1][2] nodes effortlessly. The confidentiality[4] and authentication[4] are the main elements of security framework. The wireless networks are unstable and unreliable because of various factors like topology changes are frequent, limited bandwidth and absence of a centralized control. A reliable key management and chaining is required to implement a standard security framework. The proposed work leverages the advantages of Newton's interpolation method to implement the key chaining for MANET.

Keywords
Key Chaining, Secret Sharing, Certificate, Polynomial Functions

I. Introduction

The "active attacks" [1,2] execute harmful functions such as packet discarding, corrupting payload and routing messages. The "passive attacks" [1,2] mainly read network functions and collect information about network. Furthermore, a malicious node [4] [7,9] can take part in the network to disrupt the normal routing process. The malicious node is an unauthorized node that causes congestion, propagates incorrect routing messages, prevents services or shuts them down completely. These extortions exist because of intrinsically limited physical security[11] of mobile ad hoc networks. Undeniably, it is easier to interrupt communications and infuse corrupted messages in the wireless communication[13] medium than in an equivalent wired network. In general , a dedicated server[14] is constituted to manage certificates in normal scenario or asymmetric keys[12] are used. These traditional methods are not applicable to wireless networks.

II. Security Issues

The spoofing is the main problem that destructs the entire network .The immediate dominance of spoofing attack[3][5] is the "over all" corruption of network information trailed by network loops[6] and partitioning of network.

The proposed security frame work for MANET is made up of the following building blocks.

i) Distributed Key Management
ii) Security Association (SA)
iii) Key Chaining

A. Security Association

A certificate [2] contains (Px: public key, toc: time of creation and IPx: Ip address of a device). A dedicated server [2] takes the responsibility of issuing and revoking certificates. But establishing a separate server is against to the nature of MANET .Therefore this work requires a hybrid approach with Security Association(SA) "Trust Model" [11], the SA is made up of set of trusted nodes. The members of SA take the responsibility of overall key management.

B. Key distribution concepts

In symmetric key cryptography[6], the prerequisite is exchanging the symmetric keys between source and destination before the encryption and decryption. Distribution of secret keys[1] has been problematic until recently. In public key cryptography, the key distribution is done through key servers. The key-pair[3] [14] contains public and private keys[14], the source retains private key and gives public key to receivers. In secret sharing, a secret is used as a seed to generate a number of distinct secrets that are shared by security association members' subset of key shares can be used to regenerate original key and the recipients of can jointly authenticate themselves and use the secret information.

The basic idea behind key sharing is, 'N' secretes are distributed among M nodes so that any M<N of them can regenerate the original information, but no smaller group up to M-1 can do so. There are several mathematical approaches to solve this problem, such as the number of points needed to identify a polynomial of a certain degree (used in Shamir's scheme),[7] or the number of intersecting hyper planes needed to specify a point (used in Blakely's scheme). When applying this type of secret sharing trust model, an entity is trusted[10] if any k trusted entities approve so. A locally trusted entity is globally accepted and a locally suspected entity is looked upon unreliable all over the network.

C. Key Chaining

1. Newton Interpolation

Newton Interpolation is based on developing difference tables for a given set of data points. The degree interpolating polynomial obtained by interpolating data points will be identical to that obtained using Lagrange formulae . One of the most important features of Newton's formula is that one can gradually increase the support data without re computing what is already computed.

Two types of Newton Interpolation

1) Newton's forward difference
2) Newton's backward difference

2. Forward Difference Table

Newton forward difference table is used to calculate the difference in values until it reaches '0'**.** The forward divided differences can be written in the form of a table. For example, for a function $f(x)$ is to be interpolated on points

x_o $f(x_0)$

$$\frac{f(x_1)-f(x_0)}{x1-x0}$$

x1 f(x1)

$$\frac{\frac{f(x2)-f(x1)}{x2-x1}-\frac{f(x1)-f(x0)}{x1-x0}}{x2-x0}$$

$$\frac{f(x2)-f(x1)}{x2-x1}$$

x_2 $f(x_2)$

x_n $f(x_n)$

Polynomial Generation :

$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots + a_n(x - x_0) \ldots (x - x_{n-1})$ (2.1)

3. Worked out example

The list of key shares are given in Table 2.1 , the key shares are taken in linear order.

The Table 2.2 shows the forward difference table that follows Newton's forward difference method, the values are calculated till it reaches zero. The calculation method is given as an example.

Table 2.1: Key share Table

x	0	1	2	3	4
f(x)	1	7	23	55	109

Table 2.2: Forward Difference Table

x	f(x)				
0	1				
		6 ⟷	$\frac{f(x1)-f(x0)}{x1-x0}=\frac{7-6}{1-0}=6$		
1	7	5			
			16	1	
2	23	8			0
			32	1	
3	55	11			
			54		
4	109				

The Equation 2.1 is substituted with proper values to regenerate the polynomial Equation(2.2)

$\Rightarrow$ $1 + 6(x-0) + 5(x-0)(x-1) + 1(x-0)(x-1)(x-2) + 0(x-0)(x-1)(x-2)(x-3)$

$\Rightarrow$ $1 + 6x + 5(x^2-x) + (x^2-x)(x-2)$

$\Rightarrow$ $1 + 6x + 5x^2 - 5x + x^3 - 2x^2 - x^2 + 2x$

$\Rightarrow$ $x^3 + 2x^2 + 3x + 1$ (2.2)

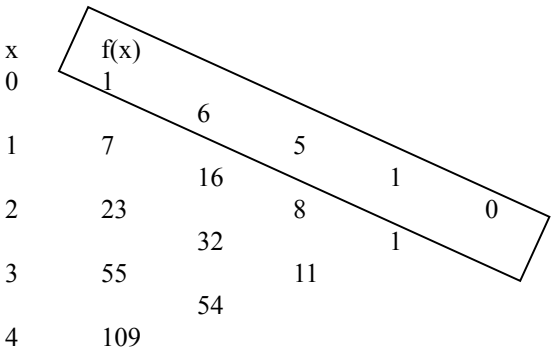
The symmetric key is generated by substituting any x value to generated f(x), the x value is decided by destination node. As an example the equation (2.2) is solved by substituting x = 4 and f(x) = 64 + 32 + 12 + 1 = 109 , this is the session symmetric key for a particular payload delivery.

III. Proposed Forward Ant Structure

The forward ant (16) is used to discover new routes in all possible ways and reaches destination node. The proposed forward ant is updated with key shares that are generated as intermediate values of Newton's interpolation method. The key shares are distributed among all intermediate nodes to implement key chaining . All key shares are encrypted by the public keys of respective intermediate nodes ,these key shares cannot be viewed by other nodes.

It looks like the process requires more processing and memory requirements but it is obligatory to implement a tight key management system in wireless networks, moreover recent developments show the application feasibility of implementing asymmetric encryption algorithms in mobile devices. These asymmetric keys are more suitable to encrypt small messages in a strong way. The process starts from collecting key shares from Newton's forward table. The f(x) values are taken in diagonal form ,which are used to generate polynomial functions.

Table 2.3. Newton's forward difference table.



The proposed forward ant is updated with a polynomial function that is encrypted by the private key of destination. This polynomial function is used to generated symmetric key for that session. The intermediate nodes store the partial function elements 6(x-0),

5(x-0)(x-1) + 1(x-0)(x-1)(x-2) , 0(x-0)(x-1)(x-2)(x-3) that are generated as part of Newton's forward difference table. The count of these values are depending upon the key shares taken , for example, if six ,x and f(x) values are given as input then five intermediate calculations are done as $c_1(x-0)$,$c_2(x-0)(x-1)$,$c_3(x-0)(x-1)(x-2)$, $c_4(x-0)(x-1)(x-2)(x-3)$ and c_5 $(x-0)(x-1)(x-2)(x-3)(x-4)$. The count of intermediate values are decided upon the hop count to reach destination. The complete structure of proposed forward ant is given in Table (3.1)

Table 3.1: Proposed forward ant structure

Source Address	1	
Destination Address	5	
Next Hop	5	
Maximum Hop Count	4	
Hop count	4	
Polynomial function	$[x^3 + 2x^2 + 3x + 1]Dp+$	
Key Shares	Node Number	[Key Share]
Dynamic Stack	[1]	1 ,
	[2]	6 , 6(x-0)
	[3]	5, 5(x-0)(x-1)
	[4]	1, (x-0)(x-1)(x-2)
	[5]	0

The Table 3.1 has the proposed structure of forward ant , the polynomial function is encrypted by the public key of

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

destination node. The key shares are encrypted by respective intermediate nodes. So this ant structure is completely secured and encrypted.

A. Key Share Issue

The key shares are distributed as given in Figure 3.1 the maximum hop count is already calculated by a forward ant when route discovery is completed.

Algorithm

Step 1: Start route discovery from source to destination
Step 2: Issue source node's public key to destination
Step 3: Receive backward ant and update pheromone values
Step 4: Prepare a unicast forward ant as given in Table 3.1
Step 5: Fetch all intermediate key shares in to forward ant
Step 6: Start unicast process towards destination
Step 7: Issue key shares to intermediate nodes that are encrypted by public keys of respective intermediate nodes.
Step 8: Issue the polynomial function to destination node, which is encrypted by the public key of destination node.

The Figure 3.1 shows the snapshot of key chaining the intermediate nodes have proper partial elements of Newton's forward difference.

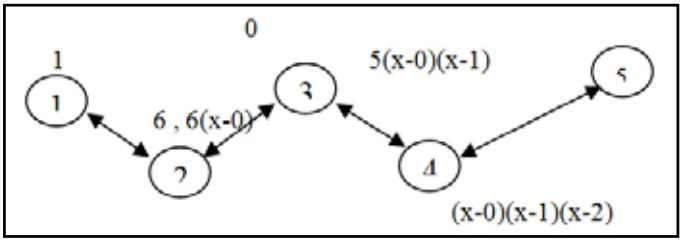


Fig. 3.1: Snapshot of key chaining

B. Key Chaining

The destination node starts key chaining management after receiving a forward ant, backward ant is updated with key shares table in reverse order. The key shares are encrypted by public keys of respective intermediate nodes . These values are compared with old values from node table and algorithm 3.2.1 is invoked to update path and to issue the symmetric key to source node.

3.2.1 Key Chaining algorithm

Step 1: Generate a backward ant with key shares as given in Table 3.2
Step 2: Generate a polynomial function with the help of collected key shares.1 + 6(x-0) + 5(x-0)(x-1) + 1(x-0)(x-1)(x-2) + 0(x-0)(x-1)(x-2)(x-3)
$x^3 + 2x^2 + 3x + 1$
Step 4: Calculate f(x) for a given random value of x
Step 5: For instance , if x=5 then f(x)=191
Step 6: Encrypt x by using the public key of source node
Step 7: Send backward ant to source node
Step 8: Compare key shares value at each and every intermediate node
Step 9: if values are equal then evaluate equations at intermediate nodes
Step 10: For instance , node 4 calculates value= (x-0)*(x-1-)*(x-2) = 5*4*3=60.
Step11: Calculate sum=sum+value
Step12: Encrypt sum by using the public keys of respective intermediate nodes.
Step13: Repeat steps from 8 to 11 to calculate total sum.
Step14: Deliver sum to source node

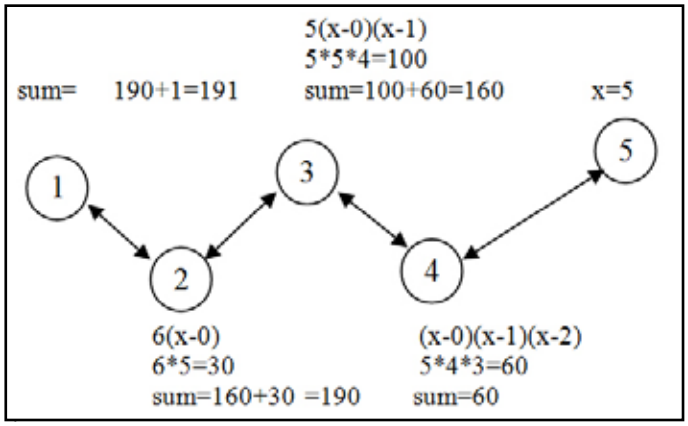


Fig. 3.2 : Path Management

The Figure 3.2 shows the working principle of proposed algorithm , the 'x' value is applied in each and every intermediate nodes, and small self descriptive program calculates sum and encrypts by itself . The sum is calculated at each and every stage and source node collects total . This sum is the value that acts as the symmetric key between source and destination.

Table 3.2 :Proposed Backward ant

Source Address	5	
Destination Address	1	
Next Hop	4	
Maximum Hop Count	4	
Hop count	4	
Key Shares Dynamic Stack	Node Number	Key Share
	[5]	0
	[4]	1
	[3]	5
	[2]	6
	[1]	1

IV. Simulation Results

The detailed analysis has been done to evaluate the Maximum percentage Error between Newton's Interpolating Polynomial and Lagrange's Interpolating polynomial. The analysis has been done for all basic functions and Maximum percentage error values have been taken for various basic functions and a linear function. The Figure 4.1 clearly shows that Newton's interpolating polynomial outperforms Lagrange's method.

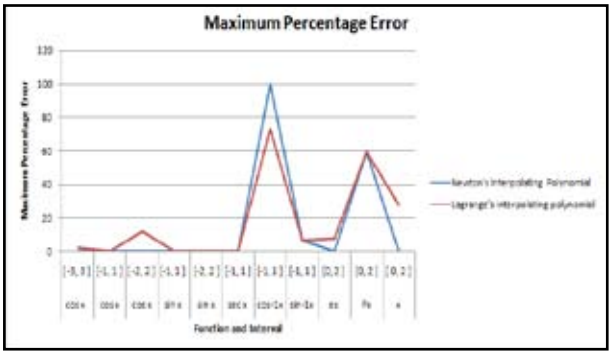


Fig. 4.1 : Maximum Percentage Error

V. Conclusion And Future Enhancements

This work mainly focuses on key chaining concept of exchanging symmetric keys between source and destination nodes. The key exchange should be an authenticated one and the proposed work uses intermediate nodes to implement key management. The backward ant is a best example of key chaining , the proposed backward ant solves the intermediate equations with the help of random generated 'x' value and sum is carried over to source node. The sum of all intermediate values should be equal to symmetric key values that is generated directly by source node. The entire path decides the correctness of symmetric key. The existing standards use a single handed approach of key management but the proposed work makes this as a collective effort . The intruders and hackers cannot access any part of symmetric key and they cannot make brute force attacks , Moreover , it is possible to generate 'n' symmetric for all available paths between source and destination. Mainly the proposed work focuses on path based key chaining instead of a single node. The other interpolation methods are Lagrange's Interpolation and Curve fitting but both are costlier in terms of memory and processing requirement. The number of multiplication and division in Lagrange's method is greater than Newton's methods. The number of additions and subtractions are lesser in Lagrange's method. The running times of the two methods are sensibly equal with a small advantage for Newton's interpolation. The future enhancement could be improved with different session keys for different paths and 'k' out of 'n' key concept could be used to implement the chaining.

References

[1]. *H. Miranda and L. Rodrigues, 'Preventing selfishness in open mobile ad hoc networks,' in Proc. Of the Seventh CaberNet Radicals Workshop, October 2002.*

[2]. *D. Djenouri, L. Khelladi, N. Badache, "A Survey of Security Issues in Mobile Ad Hoc Networks," IEEE Communications Surveys, Vol. 7, No. 4, Fourth Quarter 2005.*

[3]. *S. Sen and J. A. Clark, "Intrusion Detection in Mobile Ad Hoc Networks". In: Guide to Wireless Ad Hoc Networks, S. Misra, I. Woungang, S. C. Misra (Eds.), Springer, 2009.*

[4]. *T. Anantvalee, J. Wu, "A Survey on Intrusion Detection in Mobile Ad Hoc Networks," Wireless/Mobile Network Security, Springer, Chapter 7, pp. 170 - 196, 2006.*

[5]. *C. Xenakis, C. Panos, I. Stavrakaki,"A Comparative Evaluation of Intrusion Detection Architectures for Mobile Ad Hoc Networks" Computers and Security , Elsevier, pp. 63-80,2011.*

[6]. *Y. Jain, R. Ahirwar, "Secure Mobile Agent Based IDS for MANET", International Journal of Computer Science and Information Technologies, Vol. 3 (4), pp. 4798-4805, 2012.*

G. A. Jacoby, N. J. Davis, "Mobile Host-Based intrusion Detection and Attack Identification, "IEEE Wireless Communications, vol. 14, issue 4, pp. 53-60, August 2007.

[7]. *K. Nadkarni, A. Mishra, "A Novel Intrusion Detection Approach for Wireless Ad Hoc Networks," IEEE Wireless Communications and Networking Conference (WCNC. 2004), Vol. 2, pp. 831 – 836, March 2004.*

[8]. *A. Lauf, R. A. Peters, W. H. Robinson, "A Distributed Intrusion Detection System for Resource-Constrained Devices in Ad Hoc Networks". Elsevier Journal of Ad Hoc Networks, vol. 8, issue 3, pp. 253-266, May 2010.*

[9]. *P. Kabiri, M. Again, "Feature Analysis for Intrusion Detection in Mobile Ad Hoc Networks", International Journal of Network Security, 12 (2), 80-87, 2011.*

[10]. *J. Joseph, B. Lee, A. Das, & B. Seet, "Cross-Layer Detection of Sinking Behavior in Wireless Ad Hoc Networks Using SVM and FDA", Dependable and Secure Computing, IEEE Transactions, 8 (2), 233-245,2011.*

[11]. *H. Nakayama, S. Kurosawa, A. Jamalipour, Y. Nemoto, & N. Kato," A Dynamic Anomaly Detection Scheme for AODV-Based Mobile Ad Hoc Networks", Vehicular Technology, IEEE Transactions on, 58(5), 2471-2481, 2009.*

[12]. *S. Bansal and M. Baker, "Observation-Based Cooperation Enforcement in Ad hoc Networks," Research Report cs. NI/0307012, Stanford University, 2003.*

[13]. *S. Srivastava, N. Gupta, S. Chturvedi, S. Ghosh, "A Survey on Mobile Agent based Intrusion Detection System", ISDMISC, IJCA, 2011.*

[14]. *D Kheyri1 & M Karami1,"A Comprehensive Survey on Anomaly-Based Intrusion Detection in MANET", Computer and Information Science; Vol. 5, No. 4; 2012.*

[15]. *Y Huang, & L Wenke," A Cooperative Intrusion Detection System for Ad Hoc Networks", proceedings of the ACM Workshop on Security in Ad Hoc and Sensor Networks2003.*

[16]. *Di Caro, G., Ducatelle, F., Gambardella, L.M. "AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad Hoc Networks", Tech. Rep. No. IDSIA-27-04-2004,IDSIA/USI-SUPSI (September 2004).*

Authors' Information

K.R.Ramkumar received his M.E.Computer Science degree in 2007 from Anna University, Chennai ,India.Now he is pursuing PhD and He is a Associate Professor of Department of Computer Applications / Sri Venkateswara College of Engineering. Since 2007 he has authored 7 Papers .His research interest includes Mobile Adhoc networks, Swarm Intelligence , Routing algorithms and Mobile adhoc network security

Dr. C.S.Ravichandran has completed Doctorate from Anna University , India. Now he is working as Professor and Dean at Sri Ramakrishna Engineering College ,Coimbatore ,Tamilnadu, India. He has authored and co authored more than 20 papers and his area of interests are Power Systems and Swarm Intelligence.