International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

KEY MANAGEMENT WITH PAIRING AND WITH CERTIFICATELESS CRYPTOGRAPHY IN MANETS

Preeti Sheoran, HCTM Technical Campus; Virender Kumar, Assistant Professor, HCTM Technical Campus

Abstract

This paper discusses the basis of authenticated key management with pairing and network key management with certificateless cryptography in mobile ad hoc network (MANET). A mobile ad hoc network is an autonomous collection of mobile devices (laptops, smart phones, sensors, etc.) that communicate with each other over wireless links and cooperate in a distributed manner in order to provide the necessary network functionality in the absence of a fixed infrastructure. In this work, we adopt this system's advantage over MANET. To implement CL-PKE over MANET and to make it practical, we incorporate the idea of Shamir's secret sharing scheme. The master secret keys are shared among some or all the MANET nodes. This makes the system self-organized once the network has been initiated The study of tripartite key agreement has great theoretical and practical significance. Based on bilinear pairing and MA (message authentication) schemes, an improved secure tripartite authenticated key agreement protocol is proposed. In this paper we study proposed protocol and enhance the key strength according to simulation performed in MATLAB and we present an idea of adopting certificateless public key encryption (CL-PKE) schemes over mobile ad hoc network (MANET).

Keywords— Pairing, Key Agreement, Manets, Message Authentication

Introduction

A MANET is a most promising and rapidly growing technology which is based on a self-organized and rapidly deployed network [1]. Mobile Ad Hoc Networks (MANETS) are wireless mobile nodes that cooperatively form a network without infrastructure. In other words, ad hoc networking allows devices to create a network on demand without prior coordination or configuration. Thus, nodes within a MANET are involved in routing and forwarding information between neighbors, because there is no coordination or configuration prior to setup of a MANET. MANETs are self-configuring networks of mobile nodes without the presence of static infrastructure. They can also be heterogeneous, which means that all nodes don't have the same capacity in term of resources (power consumptions, storage, computation, etc.). Due to its great features, MANET attracts different real world application areas where the networks topology changes very quickly.

A good example is given by military battlefield networks. In that case, mobile devices have different communications capability such as radio range, battery life, data transmission rate, etc.

MANETs have many potential applications in both military and civilian domains. Their self-organized and adaptive form of node communications is particularly attractive in certain scenarios where communication infrastructures are either too expensive to build or too vulnerable to maintain. However, due to Manets' characteristics, they are susceptible to many types of attacks [5]. Wireless communication, for example, is open to interference and interception, and malicious nodes might create, alter, or replay routing information to interrupt network operation. These nodes may also launch a Sybil attack, in which a single node presents multiple identities to others, or an identity replication attack, in which clones of a compromised node are put into multiple network places. Moreover, malicious nodes may inject bogus data into the network to consume its scarce resources, and selfish nodes can drop data packets of other nodes.

Characteristics and complexities of mobile ad hoc networks [3]:

- Autonomous and infrastructure less
- Multi-hop routing
- Dynamic network topology
- Device heterogeneity
- Energy constrained operation
- Bandwidth constrained variable capacity links
- Limited physical security
- Network scalability
- Self-creation, self-organization and self-administration

Key management can be defined as a set of techniques and procedures to support the establishment and maintenance of keying relationships between authorized parties [4][5]. A keying relationship is the process by which network nodes share keying material to be used by cryptographic mechanisms. The keying material can include public/private key pairs, secret keys, initialization parameters, and non-secret parameters supporting key management in various instances.

Key management should also define methods to revoke keys from compromised nodes and update keys from non-compromised ones.

Key management for MANETs must deal with dynamic topology that is self-organized and decentralized [1] [2]. It must also satisfy some requirements, such as:

- Not having a single point of failure
- Being compromise-tolerant; that is, the compromise of a certain number of nodes does not affect the security between non-compromised nodes
- Being able to efficiently and securely revoke keys of compromised nodes and update keys of non-compromised ones
- Being efficient in terms of storage, computation, and communication

In ID-based schemes the node or user identity, such as an email or IP address, is used to derive its public key, while the private key is generally provided by an external entity. ID-based key management has been gaining interest recently, and has been used by routing protocols, cooperation mechanisms, cryptographic systems, and others.

The main advantages of IBC are the simple key management process and the reduced memory storage cost, compared with traditional public key methods. Nodes must maintain only the PKG parameters and not the public key of all other nodes.

The major problem with ID-based schemes is that the private key of all users must be known by the PKG. In conventional networks this is not an issue, but in MANETs in which the PKG must be distributed or emulated by an arbitrary entity, this might be a major issue.

Identity-based schemes are normally specified by four randomized algorithms [5]:

1 Setup: takes security parameters as input and returns a master public/private key pair for the system. The master private key is only known by the PKG.

2 Extract: takes the master private key and an identity of a node as input, and returns the personal private key of the node.

3 Encrypt: takes the master public key, the public key of the destination node (derived from its identity), and the message as input, and returns the corresponding cipher text.

4 Decrypt: takes the master public key, the private key of the node, and a cipher text as input and returns the decrypted message.

Preliminaries

A. PAIRING: Let G1 be a cyclic additive group of prime order q, and G2 be a cyclic multiplicative group of the same order q, and e: $G1 \times G1 \longrightarrow G2$ be a pairing which satisfies the following properties [22], [23]:

1) Bilinear:

$e(P1 + P2,Q) = e(P1,Q)e(P2,Q)$,

$e(P,Q1 + Q2) = e(P,Q1)e(P,Q2)$,

$e(aP, bQ) = e(P,Q)ab$,

where for all $P,P1, P2,Q,Q1,Q2 \in G1$ and $a, b \in Z^*q$

2) Non-degenerate: If P is generator of G1, then $e(P,P) \neq 1$

3) Computable: There is an efficient algorithm to compute $e(P,Q)$ for all $P,Q \in G1$.

The security of bilinear parings is based on difficulty of the computational Diffie-Hellman problem and bilinear Diffie-Hellman problem which are defined in the following subsection [22], [23].

B. ABELIAN GROUPS: An abelian group is a set, A, together with an operation "•" that combines any two elements a and b to form another element denoted a • b. The symbol "•" is a general placeholder for a concretely given operation. To qualify as an abelian group, the set and operation, (A, •), must satisfy five requirements known as the abelian group axioms:

1. Closure

For all a, b in A, the result of the operation a • b is also in A.

2. Associativity

For all a, b and c in A, the equation (a • b) • c = a • (b • c) holds.

3. Identity element

There exists an element e in A, such that for all elements a in A, the equation e • a = a • e = a holds.

4. Inverse element

For each a in A, there exists an element b in A such that a • b = b • a = e, where e is the identity element.

5. Commutativity

For all a, b in A, a • b = b • a.

More compactly, an abelian group is a commutative group. A group in which the group operation is not commutative is called a "non-abelian group" or "non-commutative group".

CYCLIC GROUP: A group G is called cyclic if there exists an element g in G such that G = <g> = { gn | n is an integer }. Since any group generated by an element in a group is a subgroup of that group, showing that the only subgroup of a group G that contains g is G itself suffices to show that G is cyclic. For example, if G = { g0, g1, g2, g3, g4, g5 } is a group, then g6 = g0, and G is cyclic. In fact, G is essentially the same as (that is, isomorphic to) the set { 0, 1, 2, 3, 4, 5 } with addition modulo 6. For example, 1 + 2 = 3 (mod 6) corresponds to g1•g2 = g3, and 2 + 5 = 1 (mod 6) corresponds to g2•g5 = g7 = g1, and so on. One can use the isomorphism φ defined by φ(gi) = i. For every positive integer n there is exactly one cyclic group (up to isomorphism) whose order is n, and there is exactly one infinite cyclic group (the integers under addition). Hence, the cyclic groups are the simplest groups and they are completely classified. The name "cyclic" may be misleading: it is possible to generate infinitely many elements and not form any literal cycles; that is, every gn is distinct. (It can be said that it has one infinitely long cycle.) A group generated in this way is called an infinite cyclic group, and is isomorphic to the additive group of integers Z.

C. COMPUTATIONAL PROBLEMS:

1) Discrete Logarithm Problem (DLP): Given P and Q ∈ G, to find an integer n ∈ Z , such that Q = nP.

2) Decision Diffie-Hellman Problem (DDHP): Given P, aP, bP , and cP , to decide whether c = ab mod q, where a, b, and c ∈ Zp*

3) Computational Diffie-Hellman Problem (CDHP): Given aP, and bP , to compute abP , where a, and b ∈ Zp*

Elliptic Curve Cryptography Based On Group Theory

ECC [21] [24] has become the cryptographic choice for ad hoc networks and communication devices due to its size and efficiency benefits. Elliptic curve cipher uses very small keys and is computationally very efficient, which makes it ideal for the smaller, less powerful devices being used today by majority of individuals to access network services. The elliptic curve crypto system (ECCS) is a crypto-algorithm method of utilizing a discrete logarithm problem (DLP) over the points on an elliptic curve. Groups which also obey commutative or symmetric property are known as Abelian groups. Abelian groups are extensively used in cryptography, as the order of the sender-receiver transmission should not confuse the common key. The abelian group of points of an elliptic curve, due to the smaller key size (and hence lower, number of members of the closed set), that is much smaller in size, at the same time maintains the same level of security. Closure, a fundamental property of groups, is used. The modulo (n) operation causes the domain to have finite number of members. This ensures the problem is solvable for the valid receiver, as well as for the problem to be hard eg: discrete log (for Diffie-Hellman, or Elliptic Curves, and prime factorization for RSA). We note that for a non-group say, y = xa, which is not limited (not closed), but over infinite real numbers, or integers. It is easy for an intruder over time to map, or guess, the exponential pattern, from the random samples eavesdropped. If we modify this to y = xa mod(n), where a, x, y, n are integers and x, and y values now becomes more random, and hence it becomes much harder for an intruder to guess any pattern. At the same time, given y, and n, publicly known values in public key cryptography, it becomes very difficult to guess x. This is due to the hardness of the discrete log problem which is due to the group closure requirements The typical representation of an elliptic curve is $y^2 = x^3+ax+b$ with a,b are integers. (x,y) are the points on x and y coordinates. We avoid curves where points (x,y), such that, x, and/or y is irrational, or transcendental. In cryptography, elliptic curves restricted over the domain of rational numbers (Q), is found to provide sufficient hardness in the discrete logarithm problem. For k to be an integer, we have to allow the coordinates of points (x,y) to be rational numbers. Thus points M, and P on the elliptic curves are allowed to take (x, y) values in rational numbers, such that M = kP where this operation is called scalar multiplication. The much smaller size keys, makes ECC very promising for the wireless, smaller size, smaller memory, bandwidth and power limited devices. 160 bit keys in elliptic curves provide same levels of security as 1024 bit RSA. Likewise 224 bit key in elliptic curve provide same levels of security as 2048 bit key in RSA.

Protocol

Diffie-Hellman key exchange: One application of CDH is the Diffie-Hellman key exchange protocol [14] [15]. Suppose two people, traditionally named Alice and Bob, want to share a secret key (which is a random element in some group). Sharing this secret needs to be done by communicating over an insecure channel and should not require any prior interaction between the two parties. Assuming the agreement between the two parties on a group G of large prime order with generator g, and also the hardness of CDH in G, the sharing of a secret key can be done in one round using the following steps:

1. Alice generates a random positive integer a, which should be less than the group order. The Information she sends to Bob is: g^a

The integer a is kept private.

2. Bob also generates a random positive integer b, which should be less than the group order. The information he sends to Alice is: g^b

The integer b is kept private.

After these two steps Alice computes $(g^b)^a = g^{ab}$ and Bob computes $(g^a)^b = g^{ab}$. This shared secret gab cannot be recovered without solving CDH in G, because any eavesdropper watching the insecure channel only has the following information: G; g; g^a and g^b

Practical Consideration

A. We consider the following scenario. Assume that Alice selects a random nonce a ∈ Zp* and computes aP . Alice wishes to send the message aP to Bob, Bob is able to ascertain that a P is not modified or fabricated and the original sender is indeed Alice. Let Alice has the public key certificate Cert A, containing her long-term public key Ya = Xa P and her long term private key Xa. Let H1 be a public cryptographic hash function H1: {0,1}* →G1 where . We describe the message authentication scheme [18] as follows (depicted in Fig. 1).

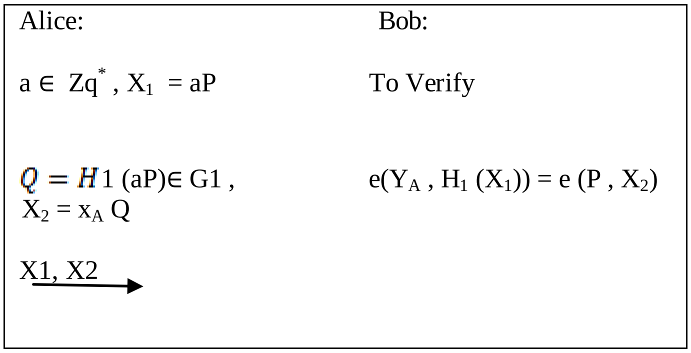


Fig. 1 Message Authentication Scheme

(1) Alice selects a random nonce a ∈ Zq* and computes X1=aP, Q = X1 (H1) , X2 = xA Q then sends (X1, X2) to Bob;
(2) Upon receipt of (X1, X2), Bob can compute e (P,X2) and e(YA , H1 (X1)) ,then verify whether they are equal. If they are equal, then the authentication is successful, otherwise, is failed.

We consider the following model for CL-PKE over MANET

We assume that at the beginning of the network there is a Key Generator Center (KGC) which generates partial secret keys for all the users. We also denote n to be the number of original nodes and t to be the pattern of security level of the threshold system. Those n nodes collectively form a Distributed Key Generator Center(DKGC). After the initiation, the KGC will go offline, and the network becomes self-organized. We define those nodes that get partial secret keys from the KGC to be the original nodes, those nodes that get partial secret.
keys from DKGC to be the new-joint nodes and those nodes that collectively form the DKGC to be DKGC nodes.

- **Setup:**
This algorithm takes as input a security parameter 1^k and returns the master private key *msk* and master public key *mpk*. This algorithm is run by the KGC, in order to setup a Certificateless ad hoc system.
- **Extract-partial-secret-key:**
This algorithm takes as input the master public key *mpk*, the master private key *msk* and an identity ID=i∈{ 0,1}*. It outputs a partial private d_i. This algorithm runs by KGC once at the initiation of the network.
- **Extract-master-secret-key-shares:**
This algorithm takes as input the master private key *msk* and an identity ID=i∈{ 0,1}*.It outputs a master secret key shares *msks*$_i$. This algorithm runs by KGC once at the initiation of the network.
- **Extract-partial-secret-key-share-and master-secret-key-share:**
This algorithm takes as input the master public key mpk, the master private key share *msks*$_i$ from a DKGC node and an identity *new* of a new-jointly node. It outputs a share of partial user private key $ds_{new,i}$ and a share of master secret key share *msks*$_{new,i}$, i ∈{ 0,1...n}. This algorithm runs by DKGC nodes.
- **Extract-master-secret-key-shares-DKGC:**
This algorithm takes as input the master public key *mpk*, an identity ID=*new* ∈{ 0,1}*, and *t* shares of master private key share *msks*$_{new,i}$, i ∈{ 0,1...n}. It outputs a master secret key share *msks*$_{new}$. This algorithm runs by the new-joint node.
- **Extract-partial-secret-key-DKGC:**
This algorithm takes as input the master public key *mpk*, a user identity ID=*new* and *t* shares of partial user private key $ds_{new,I}$, i ∈{ 0,1...}. It outputs a user partial secret key d_{new}. This algorithm runs by the new-joint node.
- **Set-user-keys:**
This algorithm takes as input the master public key *mpk*, a user identity ID=i, a partial private key d_i and a secret value x_i. It outputs a user public/private key pair (pk_i/sk_i) or an error symbol. This algorithm runs by all the nodes.
- **Encryption**:
This algorithm takes as input the master public key *msk*, a user's identity ID=*i*, a user's public key pk_i and a message *msg*. It outputs a cipher text *c*.
- **Decryption**:
This algorithm takes as input the master public key *msk*, a user's private key sk_i and a cipher text *c*. It outputs a message *msg*.

Fully Distributed System In the fully distributed system, all the nodes will have a share of *msk*. They together maintain the stability of the system. At the initiation stage, the KGC generates a master public/private key pair (*mpk*/*msk*) using **Setup** algorithm. It then generates user partial keys using **Extract-partial-secret-key** algorithm and divides *msk* with **Extract-master-secret-key-shares**. The user partial keys d_{ID} and master secret key shares *msks*$_{ID}$ are distributed to all the origin nodes. Once this is done, the KGC goes offline, and all the original nodes become DKGC nodes. We use the threshold cryptography to provide authentication for new jointly nodes. A new-joint nodes need to successfully contact at least *t* DKGC nodes. Those DKGC nodes will run **Extract-partial-secretkey-share-and-master-secret-key-share** algorithm for the new-joint node. Once this new-joint node obtains *t* shares of $msks_{new,i}$ and *t* shares of $dss_{new,I}$, it will be able to derive a master secret key share $msks_{new}$ and a partial secret key d_{new} by **Extractmaster- secret-key-shares-DKGC** and **Extractpartial-secret-key-DKGC** respectively, and it becomes a DKGC node. The number of DKGC nodes rises with the increase of node numbers.

DKGC nodes use **Set-user-keys** algorithm to calculate their own public/private keys. The public keys will be broadcasted all through the network so that nodes can communicate to each other with **Encryption** and **Decryption** algorithms.

Partially Distributed System In a partially distributed system, a certain number of nodes will become DKGC nodes. The *msk* is only shared between these nodes. They are responsible for issuing partial secret key for new coming nodes. This system differs from fully distribution system that:

1. For a new-joint node, the DKGC nodes only issue partial secret key shares $dss_{new,I}$, without any master secret key shares $msks_{new,i}$.
2. Once a DKGC node goes offline, a random non- DKGC node will be picked. Other DKGC nodes will give this node master secret key shares $msks_{new,i}$, so that this chosen one will become a new DKGC node. In this model, the number of DKGC nodes does not increase.

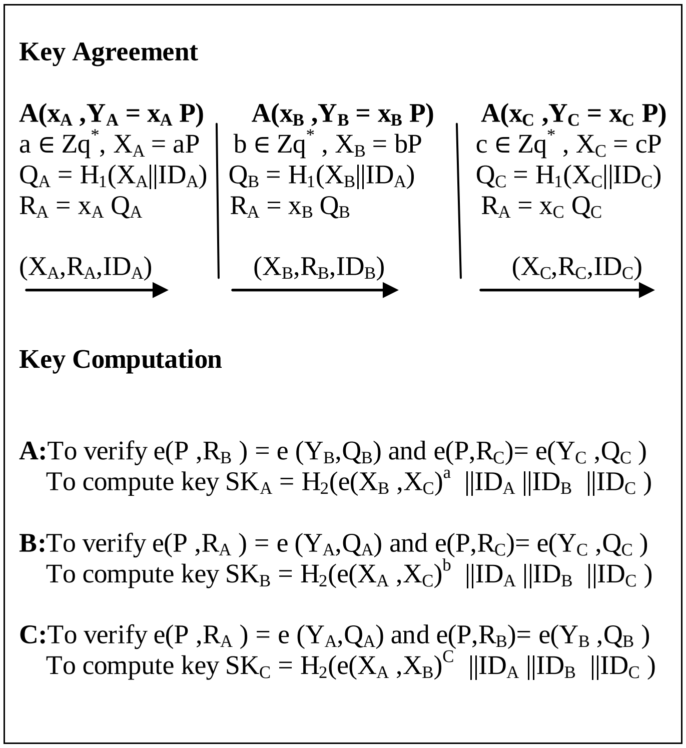
In our model, we pick all the initiation nodes to be the DKGC nodes.

B. Proposed Scheme

Setup: Let A, B and C be parties and *H* be cryptographic hash function. Choose group G_1 and G_2 of prime order q such that an admissible pairing[6] e: $G_1 \times G_1 \rightarrow G_2$ can be constructed and pick a generator *P* of G_1. Let H_1 be a cryptographic hash function where $H_1:\{0,1\}^* \rightarrow G_1$, and H_2 be a key derivation function where $H_2:\{0,1\}^* \rightarrow \{0,1\}^K$ and k is a security parameter. The public parameters are $<G_1, G_2, P, q, e, k, H_1, H_2>$ and k is a security parameter. Let A, B and C be parties who participate in this protocol. Let ID_A, ID_B and ID_C denote the identities of A, B and C respectively. Each party has his own private key *x* and the public key $Y = xP$. Assume that the broadcast channel is available and "Broadcasting " is denoted by "→".

Key Agreement Party A selects a random number $a \in Zq^*$ and computes: $X_1=aP$, $Q_A = H_1(X_A \| ID_A)$, and $R_A = x_A Q_A$. Then A broadcasts (X_A, R_A, ID_A) , Similarly, B broadcasts (X_B, R_B, ID_B) and C broadcasts (X_C, R_C, ID_C).

Key Computation Upon receipt of (X_B,R_B,ID_B) and (X_C, R_C, ID_C). , A can compute $Q_B = H_1(X_B \| ID_B)$ and $Q_C = H_1(X_C \| ID_C)$, then verify whether $e(P, R_B) = e(Y_B,Q_B)$ and $e(P,R_C)= e(Y_C, Q_C)$ If the equalities do not hold, A terminates the protocol. Otherwise, A can compute the session key $SK_A = H_2(e(X_B, X_C)^a \|ID_A \|ID_B \|ID_C)$ Similarly, B can compute the session key *SK* and C can compute the session key SK_C .



Key Agreement

A(x_A ,$Y_A = x_A P$)	**A(x_B ,$Y_B = x_B P$)**	**A(x_C ,$Y_C = x_C P$)**
$a \in Zq^*$, $X_A = aP$	$b \in Zq^*$, $X_B = bP$	$c \in Zq^*$, $X_C = cP$
$Q_A = H_1(X_A\|\|ID_A)$	$Q_B = H_1(X_B\|\|ID_A)$	$Q_C = H_1(X_C\|\|ID_C)$
$R_A = x_A Q_A$	$R_A = x_B Q_B$	$R_A = x_C Q_C$
$\xrightarrow{(X_A,R_A,ID_A)}$	$\xrightarrow{(X_B,R_B,ID_B)}$	$\xrightarrow{(X_C,R_C,ID_C)}$

Key Computation

A:To verify $e(P, R_B) = e(Y_B,Q_B)$ and $e(P,R_C)= e(Y_C, Q_C)$
To compute key $SK_A = H_2(e(X_B, X_C)^a \|ID_A \|ID_B \|ID_C)$

B:To verify $e(P, R_A) = e(Y_A,Q_A)$ and $e(P,R_C)= e(Y_C, Q_C)$
To compute key $SK_B = H_2(e(X_A, X_C)^b \|ID_A \|ID_B \|ID_C)$

C:To verify $e(P, R_A) = e(Y_A,Q_A)$ and $e(P,R_B)= e(Y_B, Q_B)$
To compute key $SK_C = H_2(e(X_A, X_B)^C \|ID_A \|ID_B \|ID_C)$

Fig.2 Key Computation

C. Experimental Results And Analysis

Simulation with MATLAB: This simulation runs over following scenarios:

1. Network establishment.
2. Network scenarios.
3. Variable initialization.
4. Parameter initialization.
5. Simulation of Network.
6. Encrypting the data packets, it is done on the basis of random generator.
7. Public key is randomly chosen and private key is changed according to change in node position.(Peer to peer connection)
8. Formulas used to generate key and key strength is improved on the basis of group changing.

9. Network data flow and nodes linking is analyzed. This is done to show the efficiency of data and key strength.
10. Time or simulation time is increased up to 0.5% and key strength improved.

These steps are performed in simulation using MATLAB. Results are shown in the following figures:

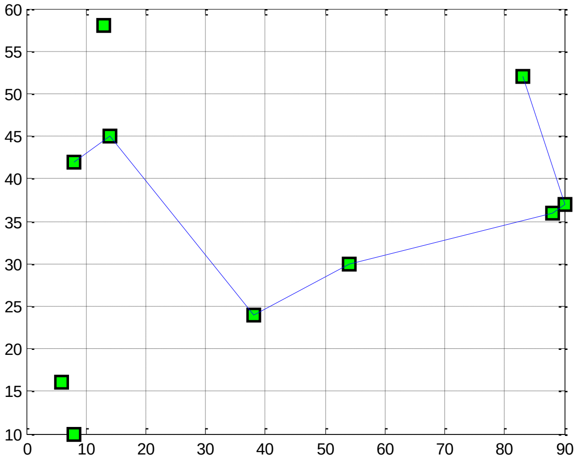


Fig. 3

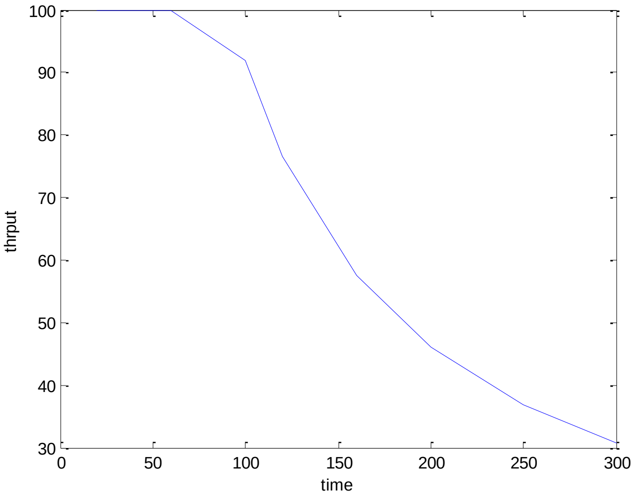


Fig. 4

Steps 1-7 are performed in Fig. 3 and Fig. 4

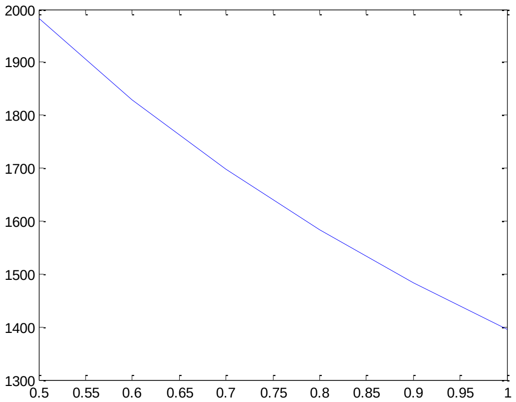


Fig.5

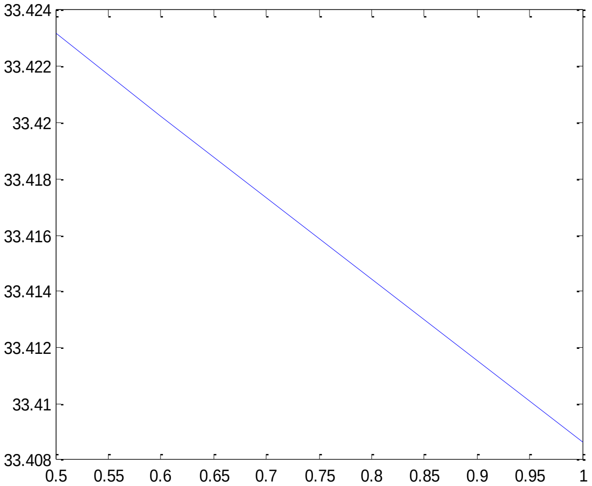


Fig.6

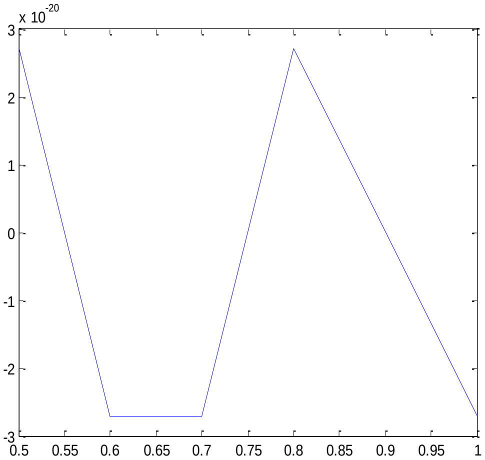


Fig.7

Step 8 is performed in Fig. 6, 7.

ISSN:2319-7900

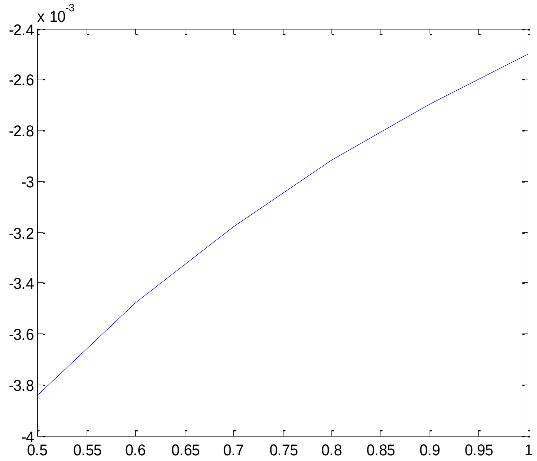


Fig.8

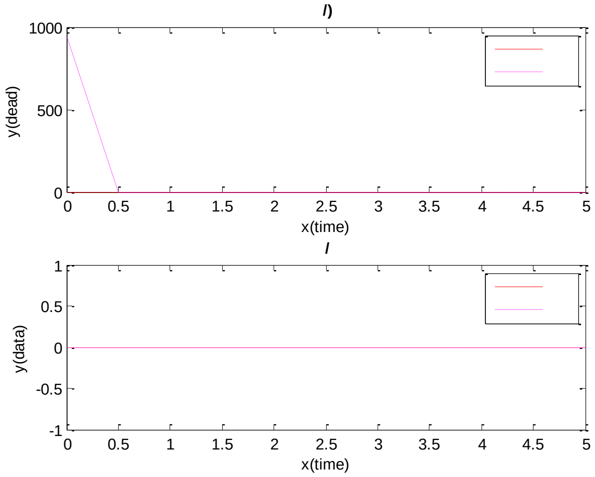


Fig.9

Steps 8 and 9 are performed in Fig. 5, 8, 9. Final results are shown in Fig. 9. In this figure key strength is improved

As we can see from the figures(Fig 10,11), in a network with 10 nodes, our scheme generates around 30 percent more traffic but the packet drop rate decreases to one quarter of pure network The average route discovery time (0.38s) is a little higher than pure network (0.32) at first but then decreases to 0.13s which is 60 percent of the pure network (0.20s).

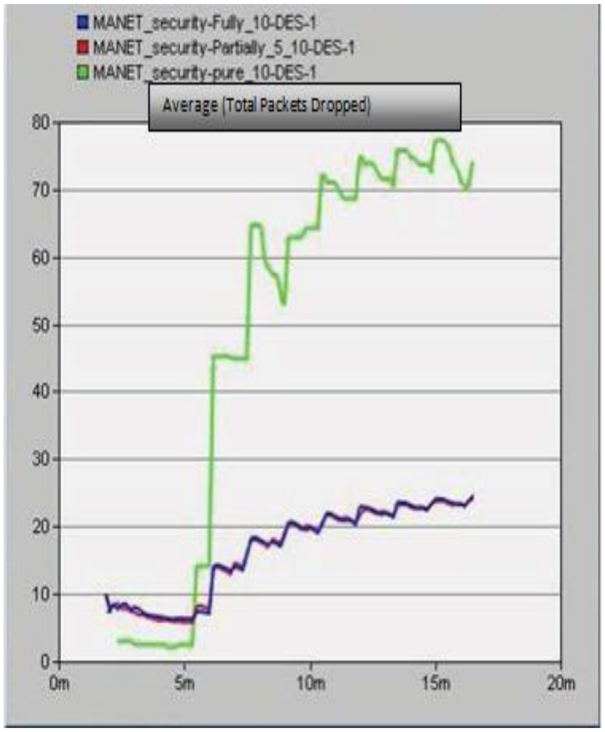


Fig.10a

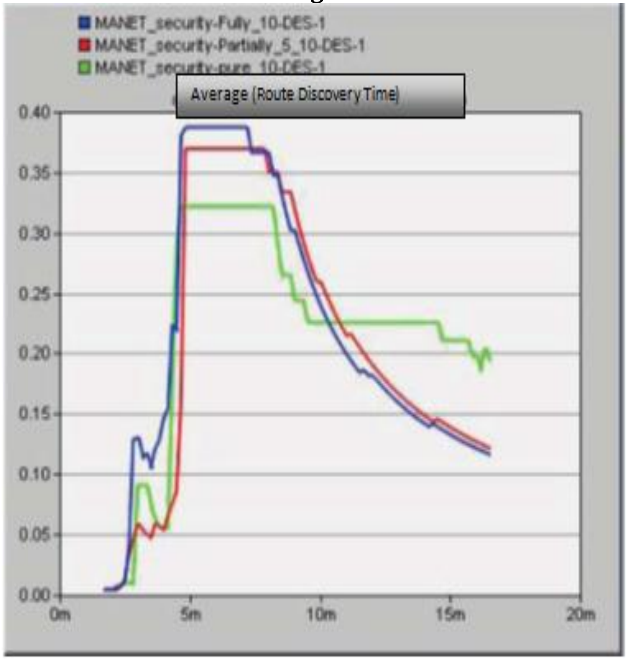


Fig.10b

In a network with 20 nodes, our scheme contributes to the average route discovery time as well, around 0.41s with CL-PKE while 0.71s without CL-PKE.

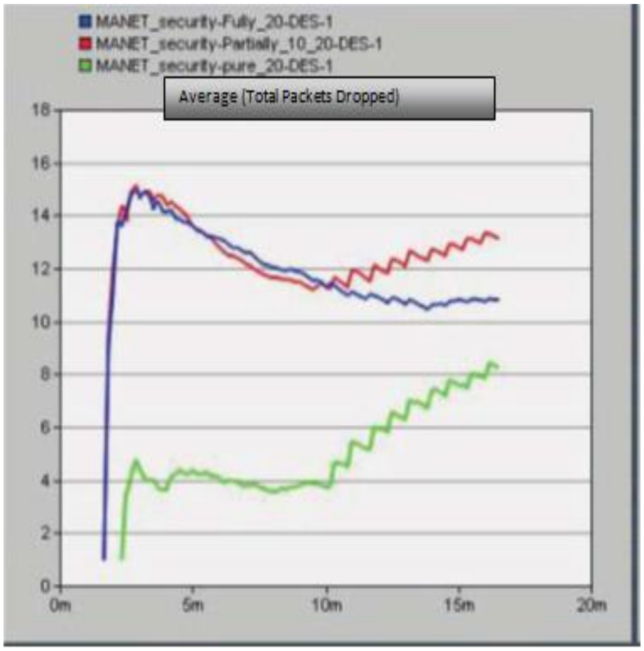


Fig.11a

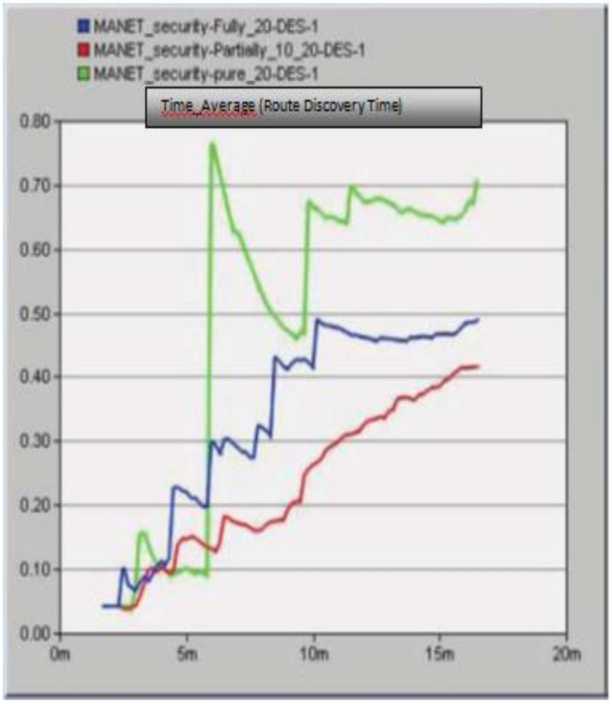


Fig.11b

Conclusions

This paper presents the simulation of a key distribution scheme over mobile ad hoc network, based on the message authentication scheme using bilinear pairing. From the simulation result, it is found out that scheme works extremely well in a small size of MANET. It improves the key strength efficiency and slightly increases the simulation time (0.5%). This scheme also ensures that system can work on self-organized networks after the initiation.

References

[1] Samba Sessay, Zongkai Yang and Jianhua He ,"A Survry on Mobile Ad Hoc Wireless Network, " Information Technology Journal 3(2):168-175, 2004.

[2] van der Merwe, J., Dawoud, D., and McDonald, " A survey on peer-to-peer key management for mobile ad hoc networks," ACM Comput. Surv. 39, 1, Article 1 , April 2007.

[3] Jeroen Hoebeke, Ingrid Moerman, Bart Dhoedt and Piet Demeester," An Overview of Mobile Ad Hoc Networks: Applications and Challenges ,"

[4] Anne Marie Hegland, Eliwinjum, Stig F. Mjølsnes, Chunming Rong, Øivind Kure, and Pål Spilling," A Survey of Key Management in Ad Hoc Networks, " IEEE Communications Surveys & Tutorials, VOLUME 8, NO. 3, 3RD QUARTER 2006.

[5] Eduardo Da Silva, Aldri L. Dos Santos, and Luiz Carlos P. Albini," Identity-based Key Management in Mobile Ad hoc Networks: Techniques and Applications ," IEEE Wireless Communications, 1536-1284, Aug. 2008.

[6] Cheng-Chi Lee and Chin-Ling Chen," Authenticated Multiple Keys Exchange Protocol based on Bilinear Pairings," International Conference on E-Business and E-Government, 978-0-7695-3997, Oct. 3, 2010.

[7] Wan An Xiong, Bin Tang," A Secure and Highly Efficient Key Management Scheme for MANET, " Advances on Information Sciences and Service Sciences. Volume 3, Number 2, March 2011.

[8] Yanchao Zhang, Wei Liu, Wenjing Lou, Yuguang Fang, and Younggoo Kwon," AC-PKI: Anonymous and Certificateless Public-Key Infrastructure for Mobile Ad Hoc Networks , " IEEE 0-7803-8939, May 5, 2005.

[9] Renu Dalal, Yudhvir Singh, and Manju Khari," A Review on Key Management Schemes in MANET, " International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.4, July 2012.

[10] Pin-Chang Su," New Authenticated Key Exchange Using Pre-shared Key Model for Ad-hoc Networks , " JOURNAL OF C.C.I.T., VOL.37, NO.1, NOV., 2008.

[11] Johann van der Merwe, Dawoud Dawoud and Stephen McDonald ," A Survey on Peer-to-Peer Key Management for Military Type Mobile Ad Hoc Networks , " Armaments Corporation of South Africa, 2005.

[12] YANG Ya-tao, ZENG Ping, FANG Yong, CHI Ya-Ping," A Feasible Key Management Scheme in Adhoc Network , " Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, IEEE, 0-7695-2909,July 7, 2007.

[13] Rakesh Chandra Gangwar and Anil K. Sarje," Secure and Efficient Dynamic Group Key Agreement Protocol for an Ad Hoc Network ," IEEE, 1-4244-0731,june 1, 2006.

[14] Hongji Wang, Gang Yao, Qingshan Jiang," An Identity-Based Group Key Agreement Protocol from Pairing , " Third International Conference on Availability, Reliability and Security, IEEE, 0-7695-3102, Aug. 4, 2008.

[15] Gang Yao, Hongji Wang, and Qingshan Jiang," An Authenticated 3-Round Identity-Based Group Key Agreement Protocol ," Third International Conference on Availability, Reliability and Security, IEEE, 0-7695-3102, Aug. 4, 2008.

[16] Zhenfei Zhang, Willy Susilo, and Raad Raad," Mobile Ad-hoc Network Key Management with Certificateless Cryptography, " IEEE, 978-1-4244-4242, Aug. 3, 2008.

[17] Daisuke MORIYAMA, and Hiroshi DOI," Efficient ID-based Key Agreement Protocol under the DLDH Assumption Without Random Oracles ," International Symposium on Information Theory and its Applications(ISITA) ,Auckland, New Zealand, 7-10, December, 2008.

[18] Peng HE, Qiuliang XU, and Ruiguang LIU," An Improved Tripartite Authenticated Key Agreement Protocol From Pairings , " Second International Workshop on Education Technology and Computer Science, IEEE, 978-0-7695-3987, oct. 4, 2010.

[19] Chen Yixiang," Certificateless Key Agreement Protocol , " IEEE, 978-1-4244-5895, Dec., 2010.

[20] S. Tapaswi1, Virendra Singh Kushwah," Securing Nodes in MANETs Using Node Based Key Management Scheme , " International Conference on Advances in Computer Engineering, IEEE, 978-0-7695-4058, Dec., 2010.

[21] K.Muthumayil, Dr.V.Rajamani, Dr.S.Manikandan, and M.Buvana," A Group Key Agreement Protocol based on stability and power using Elliptic curve cryptography ," ICETECT, IEEE, 978-1-4244-7926, Nov. 9, 2011.

[22] Amit K Awasthi and Sunder Lal," ID-based Ring Signature and Proxy Ring Signature Schemes from Bilinear Pairings , " arXiv:cs/0504097V1[cs.CR],Apr. 23, 2005.

[23] Dan Boneh and Matthew Franklin," Identity-Based Encryption from the Weil Pairing," SIAM J. of Computing, Vol. 32, No. 3, pp. 586-615, 2003.

[24] Mengbo Hou and Qiuliang Xu," An Efficient and Secure One-Round Authenticated Key Agreement Protocol without Pairings, " IEEE, 978-1-61284-774, Nov., 2011.

Biographies

Ms. PREETI SHEORAN received the B.Tech. degree in Electronics & Communication Engineering from HCTM Technical Campus, Kaithal, Haryana, in 2012,. Currently, she is pursuing M.Tech degree in Electronics & Communication Engineering from HCTM Technical Campus, Kaithal, Haryana. Her research areas include key management with pairing and with certificateless cryptography in MANETS.