HKDS: A Hierarchical Key Distribution Scheme for Wireless Ad Hoc Network

Kakali Chatterjee and Lalita Priya
(Corresponding author: Kakali Chatterjee)

Computer Science and Engineering Department, National Institute of Technology
Ashok Rajpath, Mahendru, Patna, Bihar 800005, India
(Email: kakali2008@gmail.com)
(Received June 4, 2016; revised and accepted Sept. 3, 2016 & Feb. 1, 2017)

Abstract

Wireless ad hoc networks are very popular in many areas such as border area protection, rescue operations etc. These networks suffer security problems due to their infrastructure less architecture. Attacks like passive eavesdropping, impersonation, replay etc. can be easily performed in such networks. Also the devices used in such networks are mostly resource constrained devices. Hence highly secured complex cryptographic algorithm cannot implement in such devices. This paper proposes a two level hierarchical key distribution scheme (HKDS) for wireless ad hoc networks. In the first level the secret key is distributed among the cluster heads using knapsack algorithm. In the second level, the secret shares generated by cluster head using Chinese Remainder Theorem (CRT) scheme is distributed among the nodes of the cluster. After that a mutual authentication scheme is introduced through which the node and the cluster head will mutually authenticate themselves and generate the secret session key for communication.

Keywords: Asmuth-bloom Secret Sharing; Ban Logic; Chinese Remainder Theorem; Knapsack Algorithm

1 Introduction

Wireless ad hoc networks are infrastructure less network which provides a basic framework for the ubiquitous computing. It supports anytime and anywhere deployment facility for easy communication. These networks are very disposed to security attacks such as routing attack, node stealing attack, impersonation attack, etc. [14, 20, 27]. Key management plays an important role in such networks as the data need to be encrypted before transmitting to neighboring nodes to resist those attacks. Implementation of suitable key distribution technique in this environment is challenging as the network having limited bandwidth and limited power [19, 21]. Public key management approaches mainly increase computational complexity and communicational overload. Symmetric key distribution techniques are suitable here due to the small key size and simple operations, but the security of the network totally depend upon one key. The network will compromise if the key is hacked. Hence it is better to distribute the secret in such a manner so that the adversary cannot able to get the key even if a node is captured. Different Key management schemes such as centralized group key management scheme [4, 28, 33, 36, 38], contributory group key management scheme [1, 15, 16], hybrid group key management protocol [22, 34], have been proposed in ad hoc networks.

In centralized group key management, key distribution center is involved in distribution of different types of key like group key, temporal key among the group members [8, **?**, 11, 31]. Hence these techniques are vulnerable of server spoofing attack and single point failure can occur in the network. To avoid this problem contributory group key management schemes came where contribution of every group member is considered in group key generation process [6, 12, 24]. However these schemes also suffer with scalability problem with high computational costs. Hence a hybrid of these two schemes has been proposed which is fault tolerant and also computationally efficient. But these key management schemes also suffers in node compromise attack and overall communicational overhead increased. To avoid the drawbacks, efficient secret sharing technique is preferable. Threshold Cryptography is suitable in such network to distribute the secret shares in the number of nodes. This reduces the chances of vulnerability and redundancy of secret key. Brickell [5] proposed a linear algebra based method which constructs the ideal secret sharing. Along with this he showed how to apply it to find ideal schemes for the multilevel and compartmented access structures. This schema has some scalability problem. Hence Luo *et al.* [25] described a scalable and distributed authentication technique for ad hoc network. They introduce the virtual certificate authority. After that some authentication techniques are found in literature. Zhu *et al.* [39] first introduced a threshold

cryptography based key management system for ad hoc network.

In their work, they define a group of N servers together with a pair of master public-private key which would be deployed by Certificate Authority. Each server was sharing the master private key and stores key pair of all nodes. But due to the lack of a prior knowledge of post-deployment configuration, when, N number of servers come together, they were not able to form a whole signature [26]. Condition for any node who wants to join the network was that they must collect all the N partial signatures from other nodes and compute the whole signature. Ma *et al.* [26] discussed the use of threshold cryptography in opportunistic network. They proposed identity based cryptography (IBC) security scheme where the nodes have to encounter t out of n public key generators to reconstruct their private key. Zou *et al.* [41] proposed an approach for weighted multi secret sharing scheme. Hui et.al [40] proposed a novel group key management scheme for mobile ad-hoc network where registration center handles complete registration of members and panel of key generator center handles key management. After the registration process, the user gets the shared key which is given by panel of key generation center. Farras *et al.* [9] proposed a work for constructing hierarchical secret sharing and the characterization of ideal access structures. Wang *et al.* [36] proposed a secret sharing scheme that distributes its share to currently available member of networks and threshold member will combines it to issue signature. Gharib *et al.* [10] proposed KERBEROS in mobile ad-hoc network. They assumed a predefined trusted third party. In their system, a mobile node sends resource ticket and authenticator to the service provider encrypted with the key. Wang *et al.* [37] proposed an identity based group key communication scheme based on bilinear pairing. In general all these approaches either need higher configuration effort before deployment or higher energy consumption for large traffic generation. Also most of the schemes cannot resist the major vulnerable attacks in ad hoc networks such as node-compromise attack, flooding attack, replay attack etc.

In this paper, we have proposed a two level hierarchical key distribution scheme (HKDS) for wireless ad hoc networks. The root node is the base station (BS). In the first level the secret key is distributed among the cluster heads (CH) using Knapsack algorithm. In the second level, the secret shares generated by cluster head using Chinese Remainder Theorem (CRT) scheme is distributed among the nodes of the cluster. After that in mutual authentication phase the node and the Cluster Head will mutually authenticate themselves and generate the secret session key for communication. We have compared proposed authentication scheme with some popular authentication protocol [17, 23, 29].

Our proposed scheme also includes the following aspects:

- This scheme efficiently covers two major issues like key management and node authentication in ad hoc network. We use Knapsack algorithm because one major practical advantage over RSA is speed. It can operates at throughput rates of 20 mbits/sec whereas in RSA through put rate is about 50 kbits/sec. Knapsack is suitable for resource constrained environment because it avoids complex operations like modular exponentiations.
- In first level, Knapsack key is used for communication between BS and CH which provide the strength of public key cryptography in this level. In second level, Secret shares are generated from the knapsack key and stored in cluster nodes. From the shared secret key the session key is generated which is used for message encryption (using AES) between cluster nodes and cluster head. Symmetric key cryptography is used in this level for reducing computational complexity.
- This scheme checks node validity and mutual authentication between the node and the CH. After that they will generate the secret session key.
- Authentication approach enforces very light computational load and detail security analysis shows that it resists all possible attacks.
- To resist node compromise attack, secret shares are generated and distributed in n number of nodes (participants). For recombination of secret need minimum t number of nodes (participants). Hence if a node is captured, then also an advisory cannot compute the secret key.
- This scheme uses AES symmetric encryption process to save energy and storage which is critical for constrained devices.

The rest of the paper is organized as follows: Section 2 presents Backgrounds, Section 3 provides Proposed Authentication Protocol; Section 4 discusses Implementation results; Section 5 presents Security Analysis; Finally, we conclude the paper in Section 6.

2 Backgrounds

In this section, we will discuss challenges of wireless ad hoc networks and threshold based cryptosystems.

2.1 Security Challenges in Ad Hoc Network

Ad hoc networks are decentralized and dynamic in nature [35]. The basic challenges for ad hoc network are:

- In ad hoc networks the packet of data is very insecure due to hostile environment.
- Each device contains low end processor having less speed and small programmable memory.

- Public key cryptosystems degrade the performance for complex mathematical operations.
- Physical capture of node by the adversary is a common problem.
- Due to dynamic nature, there is lack of post-deployment configuration knowledge.
- The nodes communicate each other and the base stations using low bandwidth and less transmission power.
- Group key management leads more communicational overhead.
- Ad hoc network devices are generally operated by batteries. Battery technology is lagging behind microprocessor technology. The life time of a battery have lower time range which implies the need of power conservation.
- Routing attacks (sinkhole, black hole), selective forwarding, node tampering, jamming and flooding attacks are possible attacks in these networks.

2.2 Threshold Based Cryptosystem

Secret sharing is a method in which we distribute the shares of the secret to the share-holders. The secret will be recovered only by certain predetermined groups as per access structure definition. The secret sharing schemes, where only a limited number (threshold) of participants in the reconstruction phase is important for recovering the secret is called threshold secret sharing scheme. When it calculates for total weight as threshold, it is named as weighted threshold secret sharing. Secret sharing scheme usually can be divided into following steps:

Dealer phase: This scheme starts from this phase, as dealer coordinates the whole share distribution scheme. Dealers generate a secret and its shares and distribute them in participants.

Combiner Phase: Combiner can be a participant or a special party that collect the shares from authorized participants and regenerate the secret.

For example, if a secret S_0 has to be distributed in n number of participants and threshold defined by access structure is t then dealer will generate n number of share counting from $S_1, S_2, \cdots, S_n$. Combiner will collect any t number of shares and recalculate the value S_0. According to the availability of secret to the dealer, the secret sharing scheme is defined as:

Explicit: Dealer receive the secret from outside and generate shares on it.

Implicit: Dealer create or have the secret and generate the shares on it. Generally dealer generates the secret from predetermined domain. Two types of secret sharing scheme is found in literature given below:

1) Shamir secret sharing scheme [32]: Shamir secret sharing scheme was based on polynomial interpolation. It's equation is in the form of any K-pairs $(X_1, Y_1), (X_2, Y_2), \cdots, (X_k, Y_k)$ with $x_i \neq x_j$. Dealer give a polynomial equation in the form of $p(x)$ degree $(t-1)$ such that $p(x_i) = y_i$ for all $1 \leq i \leq k$. Some features of Shamir secret sharing are:
   - Secret is chosen as free coefficient of a random polynomial.
   - Share is chosen as $l_i = P(x_i)$ for all $1 \leq i \leq n$ with x_i as a different public value.
   - Secret is recombined by using Lagrange's interpolation.
2) Blakey's Secret Sharing Scheme [3]: Blakey used n-dimensional vector space. It presents the secret as an element of GF_q^k vector space. Share were taken as any n-different $(t-1)$ dimensional vector space. Share were taken as any n different $(t-1)$ dimensional hyper plane subset of dimensional vector space as,

$$\begin{aligned} a_{11}x_{11} + a_{12}x_{12} + \cdots + a_{1t}x_{1t} &= a_1 \\ a_{21}x_{21} + a_{22}x_{22} + \cdots + a_{2t}x_{2t} &= a_2 \\ \vdots \quad \vdots \quad \vdots & \\ a_{n1}x_{n1} + a_{n2}x_{n2} + \cdots + a_{nt}x_{nt} &= a_n. \end{aligned}$$

Secret get recovered by intersection of K shares. The secret sharing scheme based on Chinese Remainder theorem has been proposed in [13]. It is a method to uniquely determine a number S modulo k many relatively prime integers $m_1, m_2, \cdots, m_k$, given that $S < \prod_{i=1}^{k} m_i$. The shares are generated by reduction modulo the integer m_i, and the secret is recovered by essentially using the Chinese remainder theorem.

3 The Proposed Authentication Protocol

In this section we propose a hierarchical authentication protocol for ad hoc network. Network model is given below.

3.1 System Network Model

Consider a system network model for border area protection. The ad hoc network is deployed for collecting the data regarding any motion or disturbances created in border area. The whole area is divided into small clusters and each cluster has a cluster head. The message from the nodes will transmit to the root node in an encrypted form [7]. During this process, each node authenticates itself to the cluster head before transferring data. Hence we design the two level hierarchical models

for key distribution and authentication. In this hierarchical model, let there are number of clusters and each cluster head will separately calculate and distribute its key among their cluster node. In deployment model, Level 1 connection (from root node to cluster head) is infrastructure based and Level 2 connection (from cluster head to cluster nodes) is infrastructure less. The network structure is described in Figure 1.

The proposed hierarchical key distribution scheme is based on Knapsack Cryptosystem and CRT based secret sharing scheme. First level is for key distribution of root node to cluster head using knapsack method and second level is for key distribution of cluster head to cluster node using CRT based secret sharing. In this process the generated symmetric key is used for node to node communication. While the node and cluster head will establish a session key for further data communication.

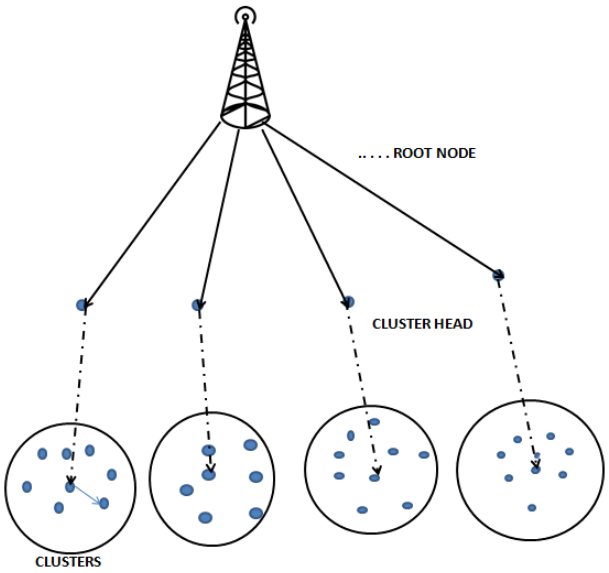

Figure 1: System model of HKDS

Assumptions used in this protocol:

- In this model, the root node is a trusted node. Mainly the Base Station is assumed here the root node.
- Cluster nodes are the mobile devices which are under the control of Cluster Heads.
- Cluster head, cluster nodes and root node are considered static and synchronized.
- In this system model, there is number overlapping between clusters.
- The communication between cluster head to cluster node is insecure.
- Threshold is calculated by static method.
- Signal intensity of all nodes is considered under threshold value.

Table 1 shows the notations used in this protocol.

Table 1: Notations used in this protocol

Symbols	Meaning
UN	User node
Bs	Base station/root node
CH	Cluster head
UID	Identification number of user node
CID	Identification number of cluster head
P_u	Random number chosen by user
R	Registration number
$h()$	Hash function (SHA-1 hash algorithm is used)
g	A generator on Z_p* where $(2 \leq g \leq p-2)$
P_b	One time key between root node and cluster node
P_{ch}	One time key between root node and cluster head
P_m	One time master key between cluster node and cluster head
P_k	Knapsack tuple
A	Diffie Hellman key of user
B	Diffie Hellman key of cluster
M_i	i^{th} message
T_i	i^{th} time stamp
N_i	i^{th} share of nodes
Key	Symmetric key in ad hoc network calculated by user node
K	Temporary session key
K_{SN}	Final session key

3.2 Description of Proposed Authentication Protocol

Proposed authentication protocol is divided in two phase. First phase is key distribution and registration which will be performed after deployment. Second phase is for Login and Authentication of nodes for data exchange.

Phase I (Key Distribution and Registration).

1) Key distribution Phase: This key distribution phase works in two level as discussed in the previous work [30]. In first level key distribution is performed from root node to cluster head using knapsack algorithm and in second level cluster head calculate secret key for each node using CRT based secret sharing and calculates its corresponding triplet,then send it to the cluster node. The key distribution is shown in Figure 2.

Level 1 (Root node to cluster head).

Root node generates Knapsack keys using N tuples super increasing key series where N is equals to the number of cluster head to which the key is to be distributed. Each tuple of the series will be given to a single cluster head. This tuple is the key for cluster head node and its share will be distributed among the cluster nodes. The process is given below: For N clus-

ter head, we choose super increasing series of N natural number.

$$w = (w_1, w_2, \cdots, w_N).$$

Randomly select a integer q such that

$$q > \sum_{i=1}^{N} w_i \tag{1}$$

and selects r such that $1 \leq r \leq q-1$ and $\gcd(q, r) = 1$ and now calculate

$$\beta_i = rw_i \bmod q. \tag{2}$$

So the calculated series

$$\beta = (\beta_1, \beta_2, \cdots, \beta_n).$$

Permute the β series and find new series

$$\gamma = (\gamma_1, \gamma_2, \cdots, \gamma_n).$$

This γ series will be the public key and the series

$$w = (w_1, w_2, \cdots, w_n), q, r,$$

will be the private key of root node. The public key tuples will be distributed to the cluster heads. For reconstructing the public key, the entire cluster node will exchange their share with their identity. Thus here the threshold $t = n$. Similar way CH will generate public-private key and send to root node. Finally root node will create a public key list of all CH.

Level 2 (Cluster head to cluster node connection).

In this level, CH will distribute the key to the node using CRT based secret sharing scheme. A special sequence of integer used here is known as Asmuth-bloom sequence [2];

$$p_0, p_1 < p_2 < \cdots < p_n.$$

Here n is the number of nodes in a cluster and threshold is decided at t. This sequence must satisfy the equation

$$A_0 \prod_{i=0}^{t-2} A_{n-i} < \prod_{i=1}^{t} A_i. \tag{3}$$

The dealer phase and combiner phase will separately run in each cluster by its cluster head.

Dealer phase.

p_0 is selected as the secret S belongs to element of z_{p_0}. The cluster node select a random number α so that

$$p_{n-t+2} \times p_{n-t+3} \times \cdots p_n < S + \alpha p_0 < p_1 \times p_2 \times \cdots \times p_t.$$

This value α will determine that without participation of nodes the secret key of cluster head cannot be retrieved. Secondly if value of $S + \alpha p_0$ is lower than the lower range decided for threshold can be reconstructed by combining less than threshold number of shares. Shares can be calculated by:

$$s_i = (S + \alpha p_0) \bmod p_i. \tag{4}$$

Combiner phase.

Cluster head will collect the threshold no of shares and calculate according to Chinese remainder theorem.

$$\begin{aligned} x &= s_1 \bmod p_1 \\ x &= s_2 \bmod p_2 \\ &\vdots \\ x &= s_t \bmod p_t. \end{aligned}$$

Here $Z = p_{i_1} p_{i_2} \cdots p_{i_t}$ and value of x can be calculated by $x = \sum_{i=1}^{t} \frac{z}{A_{i-1}} y_i s_i \bmod Z$. After calculating X, we can calculate the secret that is a key for each cluster node as $S = x \bmod p_t$.

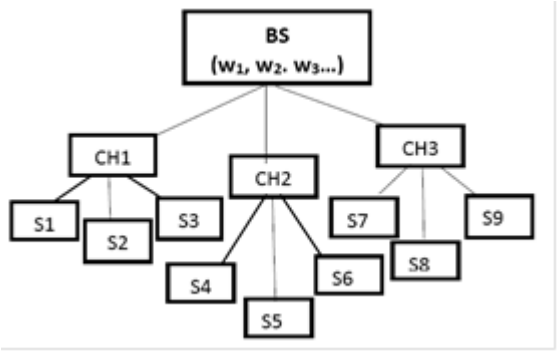

Figure 2: The Key distribution structure in HKDS

Now the cluster-head will calculate share for each node but it will send other factors along with the shared key to calculate the secret key. The steps involved in this process are given below:

Step 1: After calculating shared key it will find total multiple.

$$M = s_1 \times s_2 \times \cdots \times s_n \times p_{n+1}.$$

p_{n+1} is prime number which is greater than p_n.

$$M_1 = s_1 \times s_2 \times \cdots \times s_n.$$

Step 2: Calculate total sum

$$\begin{aligned} Sum &= s_1 + s_2 + \cdots + s_n + p_{n+1} \\ Sum_1 &= s_1 + s_2 + \cdots + s_n. \end{aligned}$$

Table 2: Message flow in registration phase

Message Flow	Message Name	Message Description
$UN \rightarrow BS$	Sub_{id}	$UID\|\|PIN\|\|P_u$
$BS \rightarrow UN$	Reg_{no}	$E_{P_b}(h(PIN)\|\|R)$
$BS \rightarrow CH$	$Node_{list}$	$E_{P_{ch}}(UID\|\|h(PIN)\|\|Q\|\|P_k)$
$CH \rightarrow UN$	$Node_{info}$	$E_{P_m}(secretshares)$

Step 3: Calculate symmetric key for cluster node as

$$Secretkey = M \bmod Sum.$$

Symmetric key will be stored as (M_1, Sum_1, p_{n+1}) Instead of sharing the symmetric key directly, cluster head will distribute this key in the form of triplet. Cluster-head will calculate triplet for each share applying conditions as follows:

Condition 1: If shared key = 1 or 0.

Step 1: Calculate array of prime number greater than total multiple . Size of array will be the total number of 0 and 1. Then calculate, multiplicative inverse of total multiplication.

$$M_1 = (M, Z_{[p_x]}*).$$

Step 2: Calculate Multiplicative inverse of Sum

$$ISum = (Sum, Z_{[p_x]}*).$$

Step 3: Calculate Triplet (M_1, $ISum$, $Sharedkey$) to be send to node.

Condition 2: If $sharedkey > 1$

Step 1: Calculate $M_1 = \frac{M}{SharedKey}$ and $ISum = Sum - SharedKey$.

Step 2: Calculate Triplet (M_1, $ISum$, $SharedKey$) to be send to node. After receiving the triplets, node will calculate the symmetric key by applying conditions as follows.

Condition 1: When shared secret is 1 or 0, calculate:

$$\begin{aligned} M &= inverse(M_1, Z_{[p_x]}*) \\ Sum &= inverse(ISum, Z_{[p_x]}*) \\ SecretKey &= M \bmod Sum. \end{aligned}$$

Condition 2: When $SecretKey > 1$, calculate:

$$\begin{aligned} M &= M_1 \times SharedSecret \\ Sum &= Sum_1 - SharedSecret \\ SecretKey &= M \bmod Sum. \end{aligned}$$

2) Registration Phase.
After deployment the nodes will registered to the base station by using following steps:

Step 1: The node will submit its UID and PIN with a random number p_u in registration form and submit it to base station. UID is a 6 digit hexadecimal number and PIN is a 4 digit number. UID is fixed (never be changed), but user can change his PIN when it is compromised. Base station will calculate $R = H(UID||p_u)$ and $Q = R \oplus H(PIN)$.

Step 2: BS will send hash of PIN concatenated with R, g to user node using the direct link. Both BS and User node will store hashed form of PIN.

Step 3: BS will send node-list UID, PIN, Q, and knapsack tuple key to cluster head. Encrypted by one time predefined key between BS and cluster head. When the registered cluster-node deployed to CH than CH will verify its UID and send it the shared triplet. Table 2. shows the message flow in this phase. This table shows sequence of messages in registration phase and what message parameter is passing from one node to another node.

Phase II (Login and Mutual Authentication Phase).
This phase discuss login process, session key generation and mutual authentication process. After authentication, session key is established. The mutual authentication phase is shown in Figure 3.

Step 1: During login process, user node submits it's UID and R to cluster head. The CH than calculate $Q' = R \oplus H(PIN)$ as CH already have $H(PIN)$ of the user node in user list table. If $Q' = Q$, where Q is already stored in the table, than login permitted and send $Grant_{login}$ message to user node. This message contains a nonce $h(N_1)$.

Step 2: Now user node calculates $M_1 = (R||h(UID)||h(N_1))$, $M_2 = (M_1||T_1)$, $A = g^a \bmod p$, $M_3 = h(A||M_1)$ and send $[CID, UID, M_2, A]_{E_{Key}}$ to cluster head.

Step 3: Cluster head verify the freshness of message by $T_2 - T_1 = \delta T$ and generate $M_1' = (R||h(UID)||h(N_1))$

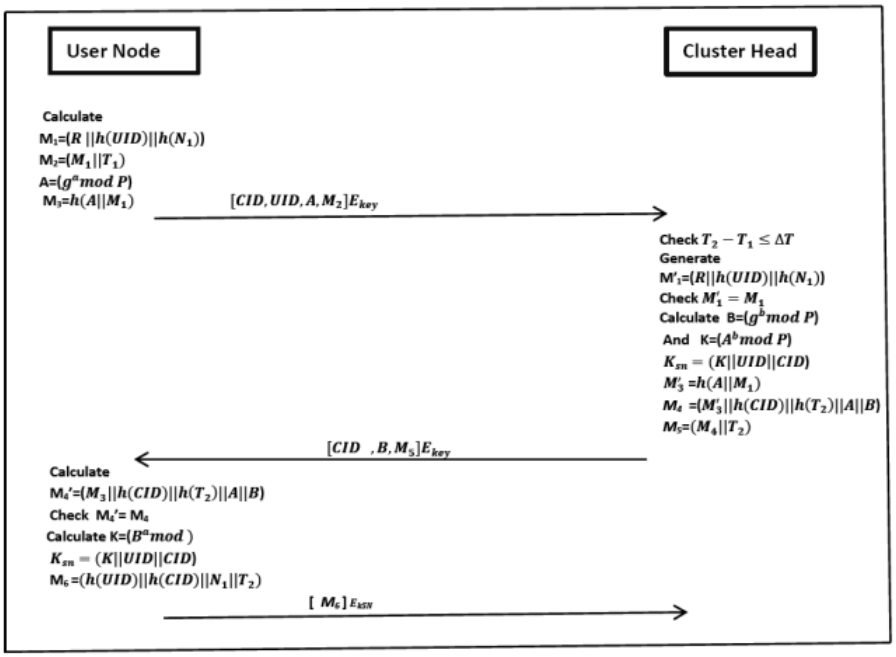

Figure 3: Mutual authentication phase

and verify it with M_1. Cluster head consider a random number b and calculate B by $g^b \bmod p$. Cluster head extract A from the message and generate $K = A^b \bmod p$. It calculates $M_2' = h(A||M_1)$ and the session key by $(K||UID||CID)$. Cluster head append M_3' in M_4. Calculate $M_4 = (M_3'||h(CID)||h(T_2)||A||B)$ and $M_5 = (M_4||T_2)$. Cluster head send message M_5 to user node as $[CID, B, M_5]_{E_{Key}}$.

Step 4: User node receive M_4 and N_2 User node calculate $M_4' = (M_2||h(CID)||h(T_2)||A||B)$. It checks $M_4' = M_4$ and calculates $K = B^a \bmod p$ and session key $K_{SN} = (K||UID||CID)$. User node send response message $M_6 = [(h(UID)||h(CID)||N_1||T_2)]_{E_{Key}}$.

Table 3 shows the message flow in this phase. In this way mutual authentication is performed.

4 Implementation and Performance Analysis

Implementation of the first phase is performed in Java Platform using Java Cryptographic Extension. We use a Laptop with Intel Core 2 DUO CPU T6400@2.00GHz with 4GB RAM and Windows 7 operating system having jdk1.8 as a Cluster head node. Implementation is distributed in three phases:- Knapsack Key generation phase, Share generation phase, and lastly the triplet generation phase. As an example, consider a cluster with 9 nodes and the share generated for 9 nodes is shown in Figure 4. Here a 15 digit random number is chosen and it is equal to 782490775074582. With this 9 generated shares, the cluster head will calculate triplet to send the nodes as shown in Figure 4 (screenshot) below using Netbeans IDE 6.8. Now the nodes can calculate the secret session key for encryption of message.

Next we calculate how much time is consumed for encrypting messages with this shared key in a mobile device (a tablet with 2GB RAM, ARM Cortex 1.7 GHz processor). For a Text File (File size: 870 KB) the encryption time is 30 ms and decryption time is 28 ms for same size data using the shared key.

To calculate how much energy it consumes over time, we find that the half of battery energy of the mobile device (a tablet with 2GB RAM, ARM Cortex 1.7 GHz processor with 3500 mAH polymer battery) consumes after 2 hours while using AES with fixed input block of length 128 byte and different key sizes (128 bytes,192 bytes, 256 bytes) for data transfer.

While calculating computation costs, we use the notation TH as the time complexity for the hashing function and TE as the time complexity for exponentiation function and TM as the time complexity for modular multiplication function. In cryptography, Exclusion-OR operations are usually neglected (due to very low computations) while considering its computational cost. Communicational cost depends upon the total no. of message exchanged for authentication. Here only three messages exchanged in mutual authentication phase. The comparison of our proposed protocol with other protocols is given in Table 4. Also functionality comparison is given in Table 5.

Table 3: Message flow in login and mutual authentication phase

Message Flow	Message Name	Message Description
$UN \rightarrow CH$	Req_{Login}	$UID\|\|R$
$CH \rightarrow UN$	$Grant_{login}$	$h(N_1)$
$UN \rightarrow CH$	$User_{Authn}$	$E_{key}(CID\|\|UID\|\|M_2\|\|A)$
$CH \rightarrow UN$	CH_{Authn}	$E_{key}(CID\|\|B\|\|M_5)$
$UN \rightarrow CH$	$Sesn_{Gen}$	$E_{key}(h(UID)\|\|h(CID)\|\|N_1\|\|T_2)$

Table 4: Performance comparison of our scheme

Schemes	Login phase	Authentication phase	Total
Lee *et al.* [17]	$7T_H$	$9T_H$	$16T_H$
Pippal *et al.* [29]	$2T_E + T_M + T_H$	$5T_E + T_M + 6T_H$	$7T_E + 2T_M + 6T_H$
Li *et al.* [23]	$T_E + 5T_H$	$3T_E + 8T_H$	$4T_E + 13T_H$
Our Scheme	T_H	$4T_E + 10T_H$	$4T_E + 11T_H$

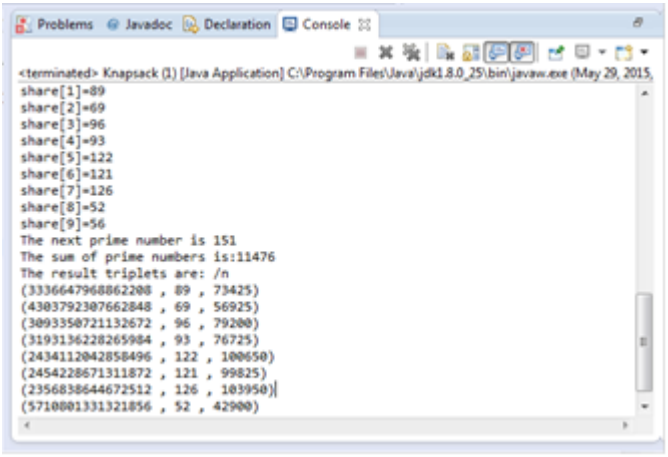

Figure 4: Triplet generation

5 Detail Security Analysis

In this section we discuss the security of proposed scheme. BAN logic is used here to proof the mutual authentication between the node and the cluster head and shared a session key. Then, we describe how the proposed protocol resists other network attacks.

5.1 Security Proof Using BAN Logic

We have proved that the authentication protocol provide high security using BAN Logic. BAN Logic is the defined set of logical rules for verifying the correctness of any protocol [23]. It also defines the beliefs of participants in the communication. Correctness of a protocol defines that both communicational parties confirms that they are sharing a fresh session key with each other after execution of the protocol.

For security verification, this work first starts with its normal definition found in [18]:

- R and S are principals i.e. the participants which communicate.
- I and J are statements.
- Key is the cryptographic keys.

Relationships and its uses for all principal, key and statement:

- $\#(I)$: The formula I is fresh.
- $R \models I$: R believes that I is true.
- $R \rightarrow I$: R is an authority and believes I.
- $R\delta I$: R receives some message including I from someone.
- $R \Vdash I$: R sent a message containing I sometime.
- (I, J): The formula I or J is one part of the formula $I, J)$.
- $< I > I$: The formula I combines with a secret parameter J.
- $\{I\}_{Key}$: The formula I is encrypted with the key Key.
- $(I)_h$: The formula I is hashed.
- R_Key_S: R and S use the shared key Key to communicate and Key will never be discovered by any principal except R and S.
- Message meaning rule: $\frac{R\models R\_Key\_S, R\delta\{I\}_{Key}}{R\models S\Vdash I}$.
- Freshness conjugation rule: $\frac{R\models\#(I)}{R\models\#(I,J)}$.

- Freshness-introduction rule: $\frac{R \text{ creat a random } I}{R \models \#(I)}$.
- The belief rule: If the principal R believes I and J, then the principal B believes (I, J): $\frac{R \models I \models J}{R \models (I,J)}$.
- The nonce-verification rule: If the principal R believes that I is fresh and the principal S sent I once then the principal R believes that S believes I: $\frac{R \models \#(I), R \models S \mid\sim I}{R \models S \models I}$.
- The jurisdiction rule: If the principal R believes that S has jurisdiction over I and S believes I, then R believes that I is true: $\frac{R \models S \rightarrow I \models S \models I}{R \models I}$.
- Introduction of the session keys: If the principal R believes that the session key Key is fresh and the principal S believes I. This is essential for a key, then R believes that he/she shares the session key Key with S: $\frac{R \models \#(S), R \models S \models I}{R \models R\_Key\_S}$.

For correctness measurement, the key agreement protocol must achieve the following goals:

$$Goal1 : CH \models CH\_K\_UN$$
$$Goal2 : UN \models UN\_K\_CH$$
$$Goal3 : UN \models UN\_K_{SN}\_CH$$
$$Goal4 : CH \models UN\_K_{SN}\_CH$$
$$Goal5 : UN \models CH \models UN\_K_{SN}\_CH$$
$$Goal6 : CH \models UN \models UN\_K_{SN}\_CH.$$

- Verification of this protocol is as following: This protocol will have three participants: $Root_{node}()$, $Cluster_{head}(CH)$, $User_{node}(UN)$. Verifying this protocol using BAN logic requires some assumption. They are as follows: From the registration phase before deployment they have

$$A1 : BS\_p_B\_UN$$
$$A2 : BS\_p_N\_CH$$
$$A3 : CH\_p_M\_UN$$
$$A4 : BS\_K_{NC}\_UN, UN\_K_{NC}\_UN$$
$$A5 : CH \models UID$$
$$A6 : UN \models UID$$
$$A7 : CH \models CID$$
$$A8 : UN \models CID.$$

Verification of this protocol using BAN logic follows: Starting from the First message $C_1 UN \rightarrow CH[CID, UID, A, M_2]_{K_{NC}}$. A is random variable calculated by Diffie Hellman process. Thus, we can assume that $A9 : UN \models \#(A)$. M_2 is calclated at UN as $M_2 = [M_1||T_1]$. Whereas, $M_1 = [R||h(UID)||h(N_1)]$. So again we can conclude that $A10 : UN \models \#(M_2)]$ and $S1 : CH\delta[CID, UID, A, M_2]_{K_{NC}}$. CH has seen the message. By message meaning rule:

$$S2 : CH \models UN \mid\sim [CID, UID, A, M_2]_{K_{NC}}. \quad (5)$$

By verifying $T_2 - T_1 = \delta T$, $S3 : CH \models \#(T_1)$. So, By Freshness conjugation rule:

$$S4 : CH \models \#(A). \quad (6)$$

By Freshness conjugation rule: $S5 : CH \models \#(M_1)$. From Equations (5) and (6),

$$S6 : CH \models UN \models (A). \quad (7)$$

Now, CH will calculate K with the help of A. So, we can conclude.

$$A11 : CH \models \#(K). \quad (8)$$

From Equations (7) and (8) and by introduction of session rule, we get $S4 : CH \models CH\_K\_UN$. Goal 1 is achieved.

Again, for Message 2, $C2 : CH \rightarrow UN[CID, B, M_5]_{K_{NC}}$. B is random variable calculated by Diffie Hellman process. Thus, we can assume that $A12 : CH \models \#(B)$. M_5 is calculated at UN as $M_5 = [M_4||T_2]$, whereas $M_4 = [M_2'||h(CID)||h(T_2)||A||B]$ and $S7 : UN||\delta[M_2'||B||CID||T_2]_{K_{NC}}$. UN has seen the message. By message meaning rule:

$$S8 : UN \models CH \mid\sim [M_3'||B||CID||T_2]_{K_{NC}}. \quad (9)$$

By verifying freshness of T_2 and M_5. We get

$$S9 : UN \models \#(B). \quad (10)$$

From Equations (9) and (10), we get

$$S9 : UN \models CH \models \#(B). \quad (11)$$

Since UN also calculate K thus, it can be assumed that

$$A12 : UN \models \#(K). \quad (12)$$

From Equations (11) and (12) and by introduction of session key rule, we get $S9 : UN \models UN\_K\_CH$. Goal 2 is thus achieved.

Now since, K is the temporary session key so both share this K. Thus, we can make assumption that:

$$A13 : UN \models K$$
$$A14 : CH \models K$$
$$A15 : UN \models CH \models K \quad (13)$$
$$A16 : UN \models CH \models K. \quad (14)$$

Both end CH and UN will calculate Session key K_{SN}. Our aim is to establish session key between user node and cluster head, thus

$$A17 : UN \models \#(K_{SN}) \quad (15)$$
$$A18 : CH \models \#(K_{SN}). \quad (16)$$

From Equations (13), (refe10), (15), (16) and using session key rule: $S10 : UN \models UN\_K_{SN}\_CH$. Goal 3 is thus achieved.

$S11 : CH \models UN\_K_{SN}\_CH$. Goal 4 is achieved.

Similarly from freshness rule and Equations (13), (refe10), (15), (16): $S12 : UN \models CH \models UN\_K_{SN}\_CH$. Goal 5 is thus achieved.

$S13 : CH \models UN \models UN\_K_{SN}\_CH$. Goal 6 is achieved.

Hence according to $S4$, $S9$, $S10$, $S11$, $S12$, $S13$, the proposed protocol achieves all the Goals and both user node and cluster head believe they share a session key K_{SN}.

5.2 Security Proof Using Attack Analysis

In this section we discuss the security of proposed scheme. The proposed protocol will be considered to be a secure authentication protocol, if it satisfies the following properties:

Man-in-the-middle attack: In this attack, the attacker establishes a common key between two parties and intercepts all message transmitted between them [7]. He modifies these intercepted messages within a valid time period. This protocol establishes a secret session key K_{SN} without revealing any information about the session key. The share key K of each node is transmitted in triplet form to the nodes. When this share is given to cluster nodes, each node calculates its secret key using an in built algorithm and communicates with other nodes. If adversary comes to know any random triplet of share, then also attacker will be unable to get the share of other node. Without knowing the algorithm he cannot deduce the symmetric key used for communication in cluster. Hence this attack cannot be successful.

Impersonation attack: This attack happens when an attacker impersonates as legitimate user by supplying valid credentials in login process. During login process, user node submits it's UID and R to cluster head. The CH than calculates $Q' = R \oplus H(PIN)$ as CH already have $H(PIN)$ of the user node in user list. If $Q' = Q$, where Q is already stored in the table, than login permitted. Now suppose, the attacker get the information about UID and R. He impersonates as a valid user and establishes a connection. But the shared key is unknown to him. So he will use different key while sending the message $[CID, UID, M_2, A]_{E_{Key}}$ to cluster head. The CH will immediately reject the message for using wrong key. Hence the proposed scheme can resist this type of attack.

Stolen-verifier attack: In this attack, the attacker may be able to steal the verification list from server. In this proposed scheme $H(PIN)$is stored in the verification table. If an attacker steal the verifier $H(PIN)$from the table, then also he will unable to calculate $R = H(UID||p_u)$ as p_u is unknown. So $Q = R \oplus H(PIN)$ is also impossible to compute. Hence this attack will be unsuccessful for determining valid login message.

Replay attack: Our protocol protects replay attack as it depends upon timestamp values (T_1, T_2). Also it depends upon random numbers a, b to confirm the freshness of the request message $[CID, UID, M_2, A]$ and response message $[CID, B, M_5]$. Even if an attacker intercept the request message, then also he will unable to compute M_6 and the correct key k for encrypt the message. It is impossible to compute a from $A = g^a \bmod p$ as it lies on discrete logarithm problem. Thus this protocol resists replay attack.

Perfect forward secrecy: In our protocol perfect forward secrecy is maintained even if the previous shared key is compromised. The attacker knows the previous shared secret N, but also unable to derive the previous session key $K_{SN} = (K||UID||CID)$ between the user node and cluster head because it had number relation with shared secret. Again suppose the attacker capture the request - response message and try to calculate the temporary session key $K = A^b \bmod p$ from the value of A. But it is impossible as it is based upon the assumption that the discrete logarithm problem is intractable and on the value of b. Thus the property of perfect forward secrecy is satisfied.

Insider attack: The user submits his UID and PIN concatenated with a random number to BS to generate $R = H(UID||p_u)$ and $Q = R \oplus H(PIN)$ which is stored in the memory of user node. During login process, user node submits it's UID and R to cluster head. The CH than calculates $Q' = R \oplus H(PIN)$ as CH already have $H(PIN)$ of the user node in user list table. If $Q' = Q$, where Q is already stored in the table, than login permitted.

Now from the registration message, an insider cannot able to calculate PIN because it is concatenated with a random number p_u called salt. Even the $H(PIN)$ value he can reveal, but that will not help him to generate Q because R cannot be calculated without p_u. Therefore insider attack cannot be possible in this protocol.

Server spoofing attack: This attack is very common in networks where the attacker manipulates the valuable data of legal user by setting up fake server. In order to set up a legal CH, the attacker needs to send the response message $[CID, B, M_5]$. As the request message is an encrypted message, hence to decrypt it the key must be known to the attacker. Suppose the shared symmetric key is known, then also the false CH does not know the share

Table 5: Functionality comparison of our scheme

Attacks	Our Scheme	Lee *et al.* [17]	Pippal *et al.* [29]	Li *et al.* [23]
Man-in-the-middle attack	Yes	Yes	Yes	No
Dictionary attack	Yes	No	Yes	Yes
Node-compromise attack	Yes	Yes	Yes	Yes
Mutual authentication	Yes	No	Yes	Yes
Impersonation attack	Yes	No	No	Yes
Stolen-verifier attack	Yes	Yes	No	No
Insider attack	Yes	Yes	No	Yes
Server spoofing attack	Yes	No	No	Yes
Replay attack	Yes	No	Yes	Yes
Perfect forward secrecy	Yes	No	No	Yes

N_1 for the user node. Hence the attacker cannot compute $M_1 = (R||h(UID)||h(N_1))$, Without M_1, it is impossible to calculate $M_5 = (M_4||T_2)$ because $M_4 = (M_2||h(CID)||h(T_2)||A||B)$ and $M_2 = h(A||M_1)$. Therefore the proposed scheme is secure against server spoofing attack.

Node compromise attack: In this attack, an adversary gets hold of a node physically and gain access of all data, intercept and modify message. In this protocol, the node authentication parameter R is embedded in the software which cannot be extracted. Hence if a node is compromised, the attacker can get the UID but not R. Hence valid login cannot possible. Now suppose the attacker capture the node and use it for valid login. After that he wants to send the request message to CH. But this time also he will unsuccessful for generating $M_1 = (R||h(UID)||h(N_1))$. Also further messages such as M_2, M_3 cannot generated. Hence he will unable to establish a session key between cluster head and user node. Thus proposed protocol resists node compromise attack.

6 Conclusion

A two stage hierarchical key distribution scheme and authentication protocol is proposed in this paper. Key distribution is a combination of knapsack public key cryptography and CRT based secret sharing scheme.

The unique feature of this architecture is that instead of symmetric key, secret shares generated by the cluster head is stored on the nodes. Any outsider cannot get the secret key without knowing n number shares. The threshold value minimum number of cluster node will need to generate share. It works with the symmetric key encryption among the cluster node. In this network, symmetric key is not directly distributed but send in the broken form. So that, if any node capture the message containing key then also capturing node will not be able to calculate the key. Performance analysis of the protocol also shows that our scheme resists the major vulnerable attacks in ad-hoc networks with low computational load.

References

[1] Y. Amir, Y. Kim, C. Nita-Rotaru, *et al.*, "Secure group communication using robust contributory key agreement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 5, pp. 468–480, May 2004.

[2] C. A. Asmuth, J. Bloom, "A modular approach to key safeguarding," *IEEE Transaction on Information Theory*, vol. IT-29, no. 2, pp. 208–210, 1983.

[3] G. Blakley, "Safeguarding cryptographic keys," in *National Conference*, vol. 8, pp. 313–317, AFIPS Press, 1979.

[4] M. S. Bouassida, I. Chrisment, O. Festor, "Group key management in MANETs," *International Journal of Network Security*, vol. 6, no. 1, pp.67–79, Jan. 2008.

[5] E. F. Brickell, "Some ideal secret sharing scheme," in *Advances in cryptology (Eurocrypt'89)*, LNCS, vol. 434, pp. 468–475, Springer, 2001.

[6] T. Y. Chang, W. P. Yang, M. S. Hwang, "Simple authenticated key agreement and protected password change protocol", *Computers & Mathematics with Applications*, vol. 49, pp. 703–714, 2005.

[7] K. Chatterjee, A. De, D. Gupta, "A secure and efficient authentication protocol in wireless sensor network," *Wireless Personal Communication*, vol. 81, no. 1, pp. 17–37, Mar. 2015.

[8] S. M. Chen, C. R. Yang, and M. S. Hwang, "Using a new structure in group key management for Pay-TV," *International Journal of Network Security*, vol. 19, no. 1, pp. 112-117, 2017.

[9] Z. Eslami, M. Noroozi, S. K. Rad, "Provably secure group key exchange protocol in the presence of dishonest insiders," *International Journal of Network Security*, vol. 18, no. 1, pp. 33-42, 2016.

[10] O. Farras, C. Padro, "Ideal hierarchical secret sharing scheme," *IEEE Transaction on Information Theory*, vol. 58, no. 5, pp. 3273–3286, May 2012.

[11] H. Gharib, K. Belloulat, "Authentication architecture using threshold cryptography in Kerberos for mobile ad hoc network," *Advances in Science and Technology Research Journal*, vol. 8, pp. 12–18, June 2014.

[12] P. Hiranvanichakorn, "Provably authenticated group key agreement based on braid groups - The dynamic case," *International Journal of Network Security*, vol. 19, no. 4, pp. 517-527, 2017.

[13] M. S. Hwang, W. P. Yang, "Controlling access in large partially-ordered hierarchies using cryptographic keys", *The Journal of Systems and Software*, vol. 67, no. 2, pp. 99-107, Aug. 2003.

[14] S. Iftene, "General secret sharing based on Chinese remainder theorem," *Electronic Notes in Theoretical Computer Science*, vol. 186, pp. 67–84, 2007.

[15] T. Jeyaprakash, R. Mukesh, "A new trusted routing protocol for vehicular ad hoc networks using trusted metrics," *International Journal of Network Security*, vol. 19, no. 4, pp. 537-545, 2017.

[16] Y. Kim, A. Perrig, G. Tsudik, "Group key agreement efficient in communication," *IEEE Transactions on Computers*, vol. 53, no. 7, pp. 905–921, 2004.

[17] Y. Kim. A. Perrigm, G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in *7th ACM Conference on Computer and Communications Security*, pp. 235–24, Nov. 2004.

[18] C. C. Lee, T. H. Lin, "A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13863–13870, 2011.

[19] J. S. Leu, W. B. Hsieh, "Efficient and secure dynamic ID-based remote user authentication scheme for distributed systems using smart cards," *IET Information Security*, vol. 8, no. 2, pp. 104–113, Mar. 2014.

[20] C. T. Li, M. S. Hwang, "A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks",*Information Sciences*, vol. 181, no. 23, pp. 5333–5347, Dec. 2011.

[21] C. T. Li, M. S. Hwang and Y. P. Chu, "An efficient sensor-to-sensor authenticated path-key establishment scheme for secure communications in wireless sensor networks", *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2107-2124, Aug. 2009.

[22] C. T. Li, M. S. Hwang, C. Y. Liu, "An electronic voting protocol with deniable authentication for mobile Ad hoc networks," *Computer Communications*, vol. 31, no. 10, pp. 2534–2540, June 2008.

[23] D. Li and S. Sampalli, "A hybrid group key management protocol for reliable and authenticated rekeying," *International Journal of Network Security*, vol. 6, no. 3, pp. 270–281, May 2008.

[24] X. Li, J. W. Niu, S. kumara, *et al.*, "An enhancement of a smart card authentication scheme for multi-server architecture," *Wireless personal Communications*, vol. 80, no. 1, pp. 175–192, 2015.

[25] I. C. Lin, M. S. Hwang, C. C. Chang, "A new key assignment scheme for enforcing complicated access control policies in hierarchy", *Future Generation Computer Systems*, vol. 19, no. 4, pp. 457–462, May 2003.

[26] H. Luo, P. Zerfos, J. Kong, *et al.*, "Self-securing ad hoc wireless networks," in *Proceedings of the Seventh International Symposium on Computers and Communications*, vol. 2, pp. 1346–1530, 2002.

[27] Y. Ma, A. Jamalipou, "Opportunistic node authentication in intermittently connected mobile ad hoc networks," in *16th Asia-Pacific Conference on Communications (APCC'10)*, pp. 543–548, 2010.

[28] L. T. Ngoc and V. T. Tu, "Whirlwind: A new method to attack routing protocol in mobile ad hoc network," *International Journal of Network Security*, vol. 19, no. 5, pp. 832-838, 2017.

[29] A. Perrig, D. Song, J. D. Tygar, "ELK: A new protocol for efficient large-groupkey distribution," *IEEE Symposium on Security and Privacy*, pp. 247–262, 2001.

[30] R. S. Pippal, C. D. Jaidhar, S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 72, no. 1, pp. 729–745, 2013.

[31] L. Priya, K. Chatterjee, "A secure authentication scheme in adhoc network using threshold cryptography," in *International Conference on Computing and Communication Technologies (ICCCT'15)*, pp. 152–155, 2015.

[32] R. V. Sampangi, S. Sampalli, "Metamorphic framework for key management and authentication in resource-constrained wireless networks," *International Journal of Network Security*, vol. 19, no. 3, pp. 430-442, 2017.

[33] A. Shamir, "How to share a secret," *Communication of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[34] A. T. Sherman and D. A. Mcgrew, "Key establishment in large dynamic groups using one-way function trees," *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444–458, May 2003.

[35] R. Srinivasan, V. Vaidehi, Rajavelu Rajaraman, *et al.*, "Secure group key management scheme for multicast networks," *International Journal Network Security*, vol. 11, no. 1, pp. 33–38, 2010.

[36] S. Tanwar, K. V. Prema, "Threats and security issues in ad hoc network: A Survey Report," *International Journal of Soft Computing and Engineering*, vol. 2, pp. 138–143, Jan. 2013.

[37] D. Wang, J. Teng, "Efficient and distributed authentication scheme for secure communication in MANET," *Journal of Computational Information Systems*, vol. 9, pp. 57–58, 2013.

[38] F. Wang, C. C. Chang, Y. C. Chou, "Group authentication and group key distribution for ad hoc networks," *International Journal of Network Security*, vol. 17, no. 2, pp. 199–207, Mar. 2015.

[39] W. H. Yang and S. P. Shieh, "Secure key agreement for group communications," *International Journal of Network Management*, vol. 11, no. 6, pp. 365–374, 2001.

[40] L. Zhu, Y. Zhang, L. Feng, "Distributed key management in ad hoc network based on mobile agent," in *Second International Symposium on Intelligent Information Technology Application*, pp. 600–605, 2008.

[41] H. Zong, L. Q. Chen, Q. Y. Zhu, "The application of threshold secret sharing in key agreement scheme for MANETs," in *International Conference on Computer Science and Service System*, pp. 837–840, 2012.

[42] X. Zou, F. Maino, E. Bertino, *et al.*, "A new approach to weighted multi-secret sharing," in *International Conference Computer Communication and Network (ICCC'11)*, pp. 1–6, 2011.

Biography

Kakali Chatterjee is an Assistant Professor in Computer Science and Engineering Department of National Institute of Technology Patna, India. She has many published research papers in LNCS (Springer) and reputed International Journals of springer. She is working in the field of Information Security and Cryptography.

Lalita Priya received B.Tech in Computer Science & Engineering from Uttarakhand Technical University, Uttarakhand. She has done her M.Tech from NIT Patna. Her area of research is Cryptography and Network Security. Presently she is working as Associate Software Engineer at CGI.