CHAPTER 4

AN EFFICIENT AUTONOMOUS KEY MANAGEMENT
WITH BLAKLEY’S SECRET SHARING SCHEME IN
MANET

4.1 INTRODUCTION

In MANET, devices may have different configurations, and must co-operate to ensure the existence of such networks (Hajami and Mohammed, 2010). MANET devices are free to move in the network, re-enter and leave, which shows the spontaneous nature of this type of networks. In addition, these networks do not support the existence of any supervisory or management authority, which provides equipments the same roles in the functioning of the network. To ensure communication between network devices, MANETs use the radio link. This allows a malicious node to infiltrate easily to disrupt the network. To prevent such behavior, a cryptographic authentication system should be established. However, the authentication system should include a trusted entity that will manage the cryptographic keys.

A secret sharing scheme is a method of distributing a secret, usually a key, among a group of users, requiring a cooperative effort to determine the key, so the plain text can subsequently be decrypted. Secret sharing schemes are designed with specific parameters that determine the number of shares needed to uncover the key, and the overall number of shares in the scheme
(Kang et al. 2010). The ultimate goal of the scheme is to divide the secret being hidden into \( w \) shares, but any subset of \( t \) shares can be used together to solve for the value of the secret. The general security of the secret sharing scheme is measured by how much information about the secret is given by each of the shares, as well as how much is given to a group shares. When a secret sharing scheme meets these criteria, it is known as a perfect secret sharing scheme (PSS).

The secure storage of the private keys of a cryptosystem is an important problem. The possession of a highly sensitive key by an individual may not be desirable as the key can easily be lost or as the individual may not be fully trusted. Giving copies of the key to more than one individual increases the risk of compromise (Bozkurt & Guloglu 2008). A solution to this problem is to give shares of the key to several individuals, forcing them to cooperate to find the secret key. This not only reduces the risk of losing the key but also makes compromising the key more difficult. In threshold cryptography, secret sharing deals with this problem, namely, sharing a highly sensitive secret among a group of \( n \) users so that only when a sufficient number \( t \) (size) of them come together can the secret be reconstructed. Well-known Secret Sharing Schemes (SSS) include Shamir based on polynomial interpolation, Blakley based on hyper plane geometry, and Asmuth-Bloom based on the Chinese Remainder Theorem.

A shortcoming of secret sharing schemes is the need to reveal the secret shares during the reconstruction phase. The system would be more secure if the subject function can be computed without revealing the secret shares or reconstructing the secret. This is known as the function sharing problem (Bozkurt & Guloglu 2008). A function sharing scheme requires distributing the function’s computation according to the underlying SSS such that each part of the computation can be carried out by a different user and
then the partial results can be combined to yield the function’s value without disclosing the individual secrets. Nearly, all the existing solutions for function sharing uses Shamir secret sharing as the underlying SSS.

A MANET is a collection of wireless mobile nodes dynamically forming a temporary network, without the use of fixed infrastructure or centralized entities, and this is exactly the environment envisioned for military operations by the Objective Force (Baras and Maria, 2004). Military command and control rely on secure (multicast) communications, and thus Key Management (KM) schemes that ensure secure communications under MANET constraints are required. However, without fixed infrastructure, e.g. Trusted Third Parties (TTPs), Certification Authorities (CAs), the design of KM becomes particularly difficult, since its most fundamental service – entity authentication, privileges update/revocation - rely on these entities to establish trust among nodes, terminate or renew participation to secure operations in a pre-agreed, global manner. Without this guarantee, all subsequent KM operations make no sense. So, it is of paramount importance to provide a secure authentication service that detects misbehavior and defends against dishonest users in the network. Thus, the challenge lies in dynamically generating mechanisms that provide individual nodes and KM groups with functionalities similar to those of the original CAs of fixed infrastructure, under MANET constraints. In this work, develop distributed, scalable, robust and efficient mechanisms for dynamically generating CAs in MANETs, by distributing the tasks of a CA among legitimate members of existing (preferably hierarchical) KM groups.

We create a (k, n) threshold scheme that allows a CA signing key to be split into n shares such that for a certain threshold k<n, any k entities could combine and recover the signing key whereas k-1 or fewer shares cannot do so. In this model, n is the number of subgroup members that have been
selected initially to participate to the CA generation and $k$ is a security threshold, selected on the fly, depending on certain KM and network parameters (Baras and Maria, 2004). The dynamic CA construction reduces to the generation of a pair of CA Public Key (PK) and Secret Key (SK): the SK is shared among the subset of Designated Members (DM), and the PK is propagated to nodes in the network.

4.2 PROPOSED AKM

Key management within a MANET is a security issue that cannot be ignored. Many researchers have dedicated themselves to this field. Some schemes are suitable for a limited number of nodes and are inefficient, insecure, or unreliable when the nodes increase (Lin and Chen-Yu, 2010). Nodes may join the MANET and leave later normally. Thus, the key management scheme in MANET must be dynamic. The main challenge of MANET is that each node handles the joining or leaving of nodes with the limited resources, such as CPU computation, storage, and the power consumption. The mobility of a MANET increases its unreliability and limits the bandwidth of wireless environment due to frequent topology changes.

The existing AKM uses a RSA algorithm as a public key cryptosystem. In recent years finding the possible alternative to RSA as ECC (Elliptic Curve Cryptography) is encouraged highly (Marimuthu & Gunavathi, 2014). All the aspects of cryptographic operations such as encryption, decryption, key exchange and digital signature, ECC is found a good alternative one. ECC uses very smaller key than RSA and DSA algorithm. For example, the key size of 128 in ECC is more powerful than 1024 bits of RSA in various aspects. Since the MANET is operating mostly on limited resources, need efficient security mechanism but as simple as possible. It is found ECC is a best suitable algorithm for MANET.
This section modifies the secret sharing of AKM, which runs dynamically in seven node-based/region-based operations. The seven operations are update, join, leave, merge, partition, expansion, and contraction (Lin and Chen-Yu, 2010). These operations are designed based on the following rules:

1. All leaves in the hierarchy of AKM are real nodes. Each real node \( i \) has its own secret key \( SK_i \), and \( PK_i = gSK_i \mod p \), where \( g \) is a random generator.

2. The non-leaf nodes are virtual nodes, and their secret keys are generated directly/indirectly from real nodes through some region-based operations.

3. A tree with node A as root is called Region A. For example, region A has virtual nodes \( B_1, B_2 \), and real nodes \( C_{1,1}, C_{1,2}, C_{1,3}, C_{2,1}, C_{2,2}, C_{2,3}, \) and \( C_{2,4} \). The number of the nodes that know the secret of region is Overall Region Size (ORS).

4. The Regional Trust Coefficient (RTC) is the ratio of the threshold to ORS, and indicates how secure the region is. The AKM sets a Global Trust Coefficient (GTC) as a lower bound of all the RTC.

4.3 **BLAKLEY'S SECRET SHARING SCHEME**

Two nonparallel lines in the same plane intersect at exactly one point. Three "nonparallel" planes in space intersect at exactly one point. More generally, any \( n \) nonparallel \((n-1)\)-dimensional hyperplanes intersect at a specific point (Naskar 2012). The secret may encoded as any single coordinate of the point of intersection. If the secret is encoded using all the coordinates, even if they are random, then an insider (someone in possession of one or more of the \((n-1)\)-dimensional hyperplanes) gains information
about the secret since he knows it must lie on his plane. If an insider can gain any more knowledge about the secret than an outsider can, then the system no longer has information theoretic security. The Blakley scheme is shown in Figure 4.1.

If only one of the $n$ coordinates is used, then the insider knows no more than an outsider (i.e., that the secret must lie on the $x$-axis for a 2-dimensional system). Each player is given enough information to define a hyper plane; the secret is recovered by calculating the planes point of intersection and then taking a specified coordinate of that intersection (Chaudhuri 2014).

![Blakley's scheme](https://en.wikipedia.org/wiki/Secret_sharing)

**Figure 4.1 Blakley's scheme**

Blakley's scheme in three dimensions: each share is a plane, and the secret is the point at which three shares intersect. Two shares are insufficient to determine the secret, although they do provide enough information to narrow it down to the line where both planes intersect. Blakley's scheme is less space-efficient than Shamir's; while Shamir's shares are each only as large as the original secret, Blakley's shares are $t$ times larger, where $t$ is the threshold number of players. Blakley's scheme can be tightened by adding restrictions on which planes are usable as shares. The resulting scheme is equivalent to Shamir's polynomial system (Chaudhuri 2014).
The following are some of the properties of Blakley’s \((t, n)\)-threshold scheme:

- Not Perfectly Secure: Because any unauthorized group knows that the secret lies on the intersection of their hyper planes. However, one can achieve perfect secrecy by choosing the secret as a single coordinate of a point, but this will affect the information rates that results in less efficient scheme. For example, in three shares system, having two shares will narrow the secret space to the line where both hyper planes intersect.

- Not Ideal: Because the size of each share exceeds the size of the original data (secret).

- Extensible: Share pieces can be dynamically added or deleted without affecting the other shares.

Blakley's secret sharing scheme is geometric in nature. The secret is a point in an m-dimensional space. N shares are constructed with each share defining a hyper plane in this space (Rishiwal & Ashutosh 2011). By finding the intersection of any m of these planes, the secret (or point of intersection) can be obtained. This scheme is not perfect, as the person with a share of the secret knows that the secret is a point on his hyper plane. Nevertheless, this scheme can be modified to achieve perfect security. This is based on the scenario where two shares are required to recover the secret. A two-dimensional plane is used as only two shares are required to recover the secret. The secret is a point in the plane. Each share is a line that passes through the point. If any two of the shares are put together, the point of intersection, which is the secret, can be easily derived.

Around the same time as the publishing of Shamir’s scheme, George Blakley published his own secret sharing scheme. Similar to Shamir’s scheme, Blakley’s scheme defined a threshold scheme based on hyper plane
intersections, instead of polynomial interpolation (Bozkurt & Guloglu 2008). The hyper planes used for this scheme are all in $t$ dimensions, which allows $t$ of the hyper planes to intersect at a single point, inside of a finite field.

Blakley secret sharing scheme has a different approach based on hyper plane geometry: To implement a $(t, n)$ threshold scheme, each of the $n$ users is given a hyper plane equation in a $t$ dimensional space over a finite field $GF(q)$ such that each hyper plane passes through a certain point. The intersection point of the hyper planes is the secret. When $t$ users come together, they can solve the system of equations to find the secret.

Figure 4.2 shows an example realization of Blakley SSS. Here $t = 2$, so each hyper plane equation is actually a line equation in the 2-dimensional space. Blakley proposed choosing the hyper planes that pass through the secret point randomly. If $q$ is sufficiently large and $t$ is not large, then the probability that any $t$ of the hyper planes intersect in some point other than the secret point is close to zero (Bozkurt, 2009). Thus generally it is possible to find the secret from any $t$ of the $n$ shares. However, it may not be possible to find the intersection point in some cases. In this case the resulting matrix is singular, i.e. the determinant is zero. The probability that a randomly chosen $t \times t$ matrix with elements chosen from the finite field $GF(q)$ is nonsingular can be computed by $\left(1 - \frac{1}{q}\right) \left(1 - \frac{1}{q^2}\right) \ldots \left(1 - \frac{1}{q^t}\right)$. This follows from the fact that the first column can be anything but the zero vector, the second column can be anything but the multiples of the first column and in general the $k$-th column can be any vector not in the linear span of the first $k - 1$ columns. When the prime $q$ is large, this probability is high enough to assure that the matrix will be invertible.
Blakley’s SSS uses hyper plane geometry to solve the secret sharing problem given in Figure 4.2. The secret is a point in a dimensional space and n shares are affine hyper planes that pass through this point (Bozkurt and Guloglu, 2008). An affine hyper plane in at-dimensional space with coordinates in a field $F$ can be described by a linear equation of the following form:

$$a_1x_1 + a_2x_2 + \cdots + a_t x_t = b$$  \hspace{1cm} (4.1)

The intersection point is obtained by finding the intersection of any $t$ of these hyper planes. The secret can be any of the coordinates of the intersection point or any function of the coordinates. Take the secret to be the first coordinate of the point of intersection.

### 4.3.1 Dealing Phase

Let $m$ be the prime and let $F = \mathbb{Z}_p$ be the field are working on. The dealer generates a secret point $x$ in $F^t$, where the first coordinate $x[1]$ is set to the secret value and sets the values of the other coordinates randomly from the field $F$. The $t$th user will get a hyper plane equation over $F$. 

![Figure 4.2 Blakley Secret Sharing Scheme for $t = 2$](image-url)
For a threshold scheme there will be \( n \) such hyperplane equations, and hence will have an \( n \times t \) linear system,

\[
a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{it}x_t = y_i
\]  \hspace{1cm} (4.2)

The dealer then sends the secret value of \( y_i \) along with \( a_{i1}, \ldots, a_{it} \) to user \( i \). The coefficients \( a_{ij} \) are not sensitive and can be made if needed.

#### 4.3.2 Share Combining Phase

Share combining step is simply finding the solution of a linear system of equations. Suppose that a coalition \( S = \{i_1, \ldots, i_t\} \) of users come together. They form a matrix \( A_S \) using their hyper-plane equations and solve

\[
A_Sx = y_S
\]  \hspace{1cm} (4.4)

where \( y_S \) is the vector of the secret shares of the users. The secret is found as the first coordinate of the solution.

#### 4.4 BLAKLEY’S THRESHOLD SCHEME

In the introduction, the threshold secret sharing scheme introduced by Shamir is presented (Stinson, 1992). Let us look at another example of secret sharing scheme: Blakley’s threshold scheme based on geometry over finite fields. Let us define \( t, n \) two integers such that \( 2 \leq t < n \) and denoted by

\[
P = \{P_i : 1 \leq i \leq n\},
\]

the set of the \( n \) participants, and by \( D \) the dealer (which is not a participant of \( P \)).
First, D fixes a line \( \ell \) in \( F_q^t \) where \( q \) is a (large) prime power. This line is public: the participants know that the secret will be one of the \( q \) points of this line.

Then, D chooses a secret point \( K \in \ell \) (the secret can be any of the coordinates of \( K \) or any function of its coordinates), and constructs a random hyper plane \( H \) (of dimension \( t-1 \)) intersecting \( \ell \) in the point \( K \). Afterwards, the dealer picks \( n \) points \( s_i \in H \) \( (1 \leq i \leq n) \) such that these points together with \( K \) are in general position, that is any \( t \) of the points generate a \( (t-1) \)-dimensional subspace. Participant \( P_i \) will receive the points \( s_i \).

This is a perfect secret sharing scheme since any \( t \) of the participants can use their points to determine the hyper plane \( H \), but for any \( t-1 \) of the participants, there is an hyper plane passing through their points and any given point on \( \ell \). This scheme can be modified slightly so that it is ideal. Let \( \ell \) be the first coordinate axis. When the dealer gives the point \( s_i \) to participant \( P_i \), he can make public all the coordinates except the first coordinate, and give only the first coordinate to \( P_i \) in secret (\( P_i \)'s share is only the first coordinate).

Blakley’s scheme can be modified into secret sharing schemes, which have interesting properties:

**Compartment schemes** – the participants are divided into subgroups called compartments. To obtain the secret, a quorum of compartments is required. However, for a compartment to participate in the quorum, another quorum of shares is required.

**Multi-level schemes** – the participants are divided into two ordered levels. To reconstruct the secret, a smaller quorum is required in the higher level. In
addition, each member of a ‘higher’ level can replace a member of the ‘lower’ level.

**Schemes with veto-capabilities** – to allow a qualified minority of participants to say ‘no’ and hence disallowing the quorum to obtain the secret

**Schemes capable of identifying cheaters** – to allow individual participants to verify that the other participants are honest and are giving their true shares

### 4.5 ANALYSIS OF BLAKLEY’S SECRET SHARING SCHEME WHEN THE SECRET IS AN IMAGE

Secret transmission of data is an important task to preserve the data from the probable threats, during the transmission (Rishiwal & Ashutosh, 2011). Various techniques has been proposed in literature for secure transmission of data but not much work has been done on the secret transmission of images, which is a very difficult task to accomplish. One of the secret sharing schemes, which are used in literature to share the image at transmission side can be applied either by using Blakley's secret sharing scheme or Shamir's scheme for sharing a secret. This section first analyzes the Blakley's secret sharing scheme and then discusses flaws in this approach. It also presents efficient schemes, which overcome all the flaws. The scheme is based on image segmentation techniques. The proposed scheme combines a subset images to recover the image at the receiver end.

#### 4.5.1 Review of Blakley's Secret Sharing Scheme

Blakley's secret sharing scheme is geometric in nature. The secret is a point in an m-dimensional space and n shares are constructed in such a way that each share defines a hyper-plane in this space. The secret can be revealed by finding the intersection of any m of n planes. This scheme can be
modified to achieve perfect security where two shares are required to recover
the secret. A 2 - dimensional plane is used as only two shares are required to recover the secret. Each share is a line that passes through the point of intersection. If any two of the shares are put together then the secret which is the point of intersection, can be easily derived. The secret message is a point in a k - dimensional space and n shares are affine-hyper planes that intersect at this point. The solution set $x'=(x_1, x_2, \cdots, x_k)$ to an equation $a_1x_1+a_2x_2+\cdots+a_kx_k=b$ forms an affine hyper-plane.

Suppose there is a secret image to be shared among n participants. The picture is divided into n transparencies (shares) such that if any m transparencies are placed together, the image becomes visible but if fewer than m transparencies are placed together nothing can be seen. Such a scheme is constructed by viewing the secret image as a set of black and white pixels and handling each pixel separately. The schemes in perfectly secure and can be implemented easily.

4.5.2 Flaws in Existing Secret Sharing Algorithms

Blakley's secret sharing scheme has been applied. The scheme given by Turk & Alex (1991) has some shortcomings (flaws) which can plays an important role in image reconstruction and may lead to faulty reconstructed images at received end.

A. Blakley's Sharing Algorithm

At the sender end, sharing process undergoes the following steps from chosen number of shares k to retrieve the secret with total number of shares n (Rishiwal & Ashutosh 2011).
Step 1: Divide the original image in set of k pixels to represent the $x' = (x_1, x_2, \ldots, x_k)$ in the given plane.

Step 2: Choose n sets of $(a_1, a_2, \ldots, a_k, b)$ such that equation $a_1x_1 + a_2x_2 + \ldots + a_kx_k = b$ is satisfied, where $(a_1, a_2, \ldots, a_k, b)$ are the parameters.

Step 3: Repeat steps 1 and 2 for every set of k pixels and get n equations.

Step 4: From the k shares by putting the values got from the solution of above equations.

The secret image-sharing scheme algorithm presented here has two flaws which affect the efficiency of the algorithm are discussed below.

Flaw 1: The total number pixels in the image should be an integer multiple of k to partition the image into non overlapping set of k pixels. It is observed during the implementation that the above discussed scheme failed to recover the image correctly if n is not an integer multiple of k.

Flaw 2: In the image sharing scheme, the size of parameters $(a_1, a_2, \ldots, a_k, b)$ is determined mathematically and each parameter requires $k * 8/(k + 1)$ bits. This sometimes creates a difficulty if number of bits is not an integer, which can happen quite often. For example if k =5 then number of bits comes to be $5 * 8/(5+1)=6.66$. The above mentioned flaws have been taken care in this work and proposed scheme along with the possible solutions to these flaws are discussed.

B. Blakley's Recovery Algorithm

The encrypted transmitted images one reconstructed at the receiver end by solving the equations formed by considering k) shares from total n) parameters. The reconstruction process is as follows (Rishiwal & Ashutosh, 2011).
Step 1: For preset values of k and n use the same threshold (k,n).

Step 2: Each shared image is partitioned into no overlapping sets of k pixels.

Step 3: Acquire k +1 parameters (a_1, a_2, ..., a_k, b) from k pixels.

Step 4: Find the intersection point of these k hyper-planes constructed by parameters.

Step 5: Store the coordinate of the intersection point in a k-dimensional space as k pixels in the reconstructed image.

Step 6: Repeat Step 2 through Step 5 to reconstruct the transmitted image.

The scheme cannot be generalized due to the flaws. Therefore there is a need to develop a new generalized scheme.

4.6 HOMOMORPHIC PROPERTIES OF BLAKLEY SECRET SHARING

Homomorphism is a concept related to functions. A function f is said to be \((\oplus, \otimes)\) homomorphic, if f satisfies \(f(x \oplus y) = f(x) \otimes f(y)\) for operations \(\otimes\) and \(\oplus\). A secret sharing scheme is a function, which maps secrets to the distributed shares. So, this research work discusses about homomorphism in the context of secret sharing schemes. The definitions of homomorphism for a secret sharing scheme were described (Benaloh 1987).

Definition: Let \(\oplus\) and \(\otimes\) be binary functions on elements of the secret domain S and of the share domain T, respectively. This work says that a \((t,n)\) threshold scheme has the \((\oplus, \otimes)\) homomorphism property (or is \((\oplus, \otimes)\)-homomorphic) if for all S, whenever
\[ d = F_S(d_{i_1}, \ldots, d_{i_t}) \]

\[ d' = F_S(d'_{i_1}, \ldots, d'_{i_t}) \]

\[ d \otimes d' = F_S(d_{i_1} \otimes d'_{i_1}, \ldots, d_{i_t} \otimes d'_{i_t}) \]

where \( d \) and \( d' \) denote shared secrets, \( d_{i_1} \otimes d'_{i_1} \) are shares of user \( i \) for secrets \( d \) and \( d' \) respectively, \( S \) is the coalition and \( F_S \) is the function used by coalition \( S \) for recovering the secret from their shares.

### 4.7 EXPERIMENTAL RESULTS

One of the very important issues in the MANET is a computation cost. Some of the mobile in the ad-hoc devices have restricted power, and cannot support jobs requiring heavy computation cost. By using Blakley Secret Sharing evaluation of communication, computation cost and large mobile nodes with existing schemes. With the development of this secret sharing computational cost of this proposed method is less when compared with standard AKM method. Almost all operands in proposed AKM reduce, resulting from each Proposed AKM share as \( 1/t \) faster than AKM. Furthermore, the computation cost of all operations can be reduced to \( 1/t \). The evaluation is carried out using Network Simulator 2.34 under Linux environment.

Proposed AKM inherits the AKM structure and transmissions between each node are (update) shares. Thus, the single message discussion needs to be transmitted showing significant improvement. The length of secret key \( k \), protected by the Blakley Secret Sharing, must be long enough, such as 2048 bits or more for some security issues. In Blakley Secret Sharing, \( k \) is the constant in \( a(x) \) equation. Table 4.1 shows the message length comparison. The length of all the shares...
\[ a(x_i) = \sum_{j=1}^{l-1} a_j x^j + k, 1 \leq i \leq n, \text{is bounded by } \|k\|. \]

Table 4.1 Message Length Comparison

<table>
<thead>
<tr>
<th>Methods</th>
<th>Length of Message (share) size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard AKM (Shamir SSS)</td>
<td>(</td>
</tr>
<tr>
<td>BLAKLEY SSS</td>
<td>2 \leq k \leq p</td>
</tr>
</tbody>
</table>

Figure 4.3 show the computation time for proposed Blakley AKM in processing group key initialization.

**Communication and Computation Cost / Time**

In the experiments, the cost is measured both by the number of messages and the computation time.

![Average computation time for proposed Blakley AKM](chart.png)

**Figure 4.3 Computation Time of Group Key Initialization**
Figure 4.3 shows the computation cost of group key initialization with group membership rate $p = 10\%, 20\%, \text{ and } 30\%$. As per number of nodes increases computation time is also increased. For example, when $p = 10\%$ and $n = 1500$, computation time is 33 seconds, and 110 seconds for $n = 2000$. With the increase of $p$ cost also increases quickly.

![Average computation time for proposed Blakley AKM](image)

Figure 4.4 Computation Cost/Time of Group Key Initialization

Figure 4.4 shows that the computation cost increases in group key initialization as the number of nodes increase for all two schemes (proposed Blakley AKM and standard AKM). Proposed Blakley AKM has a low cost and for group key initialization standard Blakley AKM has the low computation time under the same network settings.
Figure 4.5 Computation Cost/Time for Member Addition

Figure 4.5 shows the computation cost with $p = 10\%$ for different schemes in processing group rekeying at events of member addition. For all schemes, the computation cost of group key updating is relatively lower than group key initialization process. However, proposed Blakley AKM has the smallest computation cost/time, while standard AKM still produces the largest computation cost.

Figure 4.6 Computation Cost/Time for Member Leaving

Figure 4.6 shows the computation cost of group rekeying at the event of member leave for different schemes. The computation cost of group key
updating for a member leaving is higher than the member addition process but relatively less than the group key initialization process. While standard AKM still produces the largest computation cost, proposed Blakley AKM generates low cost. Figure 4.7 show the total number of messages for proposed Blakley AKM in processing group key initialization.

![Figure 4.7 Message Cost of Group Key Initialization](chart)

Figure 4.7 shows the message cost of group key initialization with group membership rate \( p = 10\% \), 20\%, and 30\%. As the number of nodes increase because of the increased number of group members as well as non-member forwarding nodes message cost increases. However, the increase rate is higher for larger \( p \). For instance, when \( n=1500 \), the message cost increases from 450 to 600 for \( p=10\% \) and 30\%, respectively.

Under the same network settings, standard AKM has the most messages cost while proposed Blakley AKM requires the medium amount of messages during group key initialization processing. It also shows that under the same network settings, standard AKM also have similar results with increasing number of nodes and \( p \) value. While standard AKM during the group key updating process at the event of member addition and under the
same network settings, standard Blakley AKM also have similar results with increasing number of nodes and $p$ value.

The results of the experiments are summarised below:

- Under the same network settings, proposed Blakley AKM has moderate computation cost that is lower than standard AKM for both group key initialization and member leave scenarios. For the member addition process among the two schemes proposed Blakley AKM has the least computation cost with less number of reconstruction of shares.
- With the increase of network nodes, the message cost of proposed Blakley AKM increases. The message cost also increases with the increase of group membership percentage $p$.
- Under the same network settings, proposed Blakley AKM has moderate message cost, higher than standard AKM. Proposed Blakley AKM has the least message cost for member addition and leaving processes among the key management schemes.

### 4.8 SUMMARY

Blakley's secret sharing schemes are proposed for the MANET is discussed. A new efficient sharing scheme is proposed. The effectiveness of the proposed scheme has been presented through analysis and experimental results. It is evident from the experimental results and analysis that proposed scheme is more efficient than the existing scheme and can transmit and recover more secret keys effectively. But, the secret which is shared is not verified accurately whether it is right or not. To address this issue, a verifiable secret sharing scheme is introduced in MANET and described in the next chapter.