

# Node Authentication Using BLS Signature in Distributed PKI Based MANETS

N Chaitanya Kumar, Abdul Basit, Priyadarshi Singh, V. Ch. Venkaiah, and  
Y. V. Subba Rao

School of Computer and Information Sciences, University of Hyderabad,  
Hyderabad-500046, India

**Abstract.** Authenticating a node in mobile ad-hoc networks is a challenging task due to their dynamic and resource constraint infrastructure. For this purpose, MANETS adopt two kinds of approaches Public key cryptography and identity based cryptography. In Public Key Infrastructure (PKI), Certificate Authority (CA) is responsible for key management. In order to adopt it to MANET, the job of the CA must be distributed. The master secret key is shared among the nodes of the MANET, to self-organize the network without a central authority. The key is shared based on Shamir secret sharing scheme with bi-variate polynomial to make the MANET fully self-managed by nodes. In this paper, we considered PKI based scenario and proposed a new scheme to authenticate a node using BLS signature which is light weight compared to the existing schemes thus making it suitable for MANET.

**Keywords:** Mobile ad-hoc network, bi-variate polynomial, secret sharing technique, threshold cryptography, BLS signature.

## 1 Introduction

MANET known as Mobile Ad-Hoc Network is a self-organized, dynamic and infrastructureless network[1]. MANET consists of mobile nodes that roam freely, every node has its own range of signal communication, other nodes within the range can interact and exchange messages. New nodes join and some other nodes may leave or some nodes fail to connect as they move out of the MANET network range[2]. The nodes in MANET are energy constrained, i.e., nodes are battery powered devices. There are many security threats to MANETS such as Denial of service, eavesdropping, interception and routing attacks[3] [4]. Public Key Infrastructure (PKI)[5] helps in securing communication using authentication and encryption through digital certificates and public key cryptography respectively. The distributed PKI approach is adopted in this paper so as to make the MANET network completely de-centralized.

Generally in a PKI environment, a certificate authority(CA) issues and manages the public key certificates of participating entities, the CA uses a master secret key  $s$  to sign the certificate. General PKI is not suitable for MANET as we cannot

assign the sole power of CA to a single node because of its dynamic and changing topology i.e., the node with CA functionality may break-down or move out of MANET range, which results in non-availability of CA. To achieve the distributed PKI environment for MANETS, we use a  $(t,n)$  threshold scheme[6][7][8], which helps in distributing CA power, i.e., we have to distribute the master secret key  $s$  to nodes of the MANET[9]. In our proposal, we discuss how a threshold number of nodes sign a certificate and the verification of the certificate can be done by any node using BLS signature scheme[10].

### 1.1 Attacks on MANETS[11]

In MANETS, there are two types of attacks- Passive and Active. Passive attacks capture valuable data in transit and active attacks cause huge damage to the network by disrupting the normal flow of the operations. Malicious nodes cause both active and passive attacks. A malicious node is the one, which does not authenticate itself to other honest nodes and misbehaves in the network. An honest node can also be compromised if it is under the control of the attacker. As the network comprises of layers of protocols, the attacks are specific to a layer and the security should also be implemented in the corresponding layer. Since the mobile nodes share a wireless medium, the messages transmitted can eavesdrop or fake messages may be injected at physical layer. Because of one-hop connectivity maintained among neighbors, the attacker can launch traffic analysis and traffic monitoring attacks. In network layer, the attacker exploits the routing algorithms to create routing hops and network congestion[4]. The attacker uses a compromised node to perform SYN flooding and denial of service(DOS) attacks at transport layer. The majority of attacks in the application layer are worm attacks, mobile viruses and repudiation attacks. Some attacks like denial of service and man-in-the-middle can be launched from several layers. This paper proposes node authentication using BLS signature, so that many of the attacks can be avoided.

### 1.2 Distributed PKI

Public key cryptography(PKC)[12] provides many security services like confidentiality, integrity, authentication, non-repudiation, encryption and digital signatures. Public key infrastructure(PKI)[5] manages digital certificates which are important in the deployment of public key cryptography. In PKI environment, Certificate authority(CA) issues and maintains the certificates of participating entities, the certificate contains the public key and the ID of the entity, the CA signs the certificate using the master secret key  $s$  and this certificate can be verified by the master public key  $PK$ . In MANETS we cannot adopt the same PKI, as the network is dynamic and infrastructure-less. So the role of the CA needs to be distributed to the nodes i.e., the master secret key  $s$  is to be shared among different nodes and the

master secret key can only be generated if atleast the threshold number of shares of secret are pooled together.

### 1.3 Threshold Cryptography

As MANET is a decentralized network, the master secret key ( $s$ ) of the PKI is distributed among the nodes using secret sharing schemes. One of the popular and most widely used secret sharing technique is the Shamir's secret sharing technique[8]. In this scheme, dealer distributes a secret  $s$  among  $n$  users. Each user receives it's share privately from the dealer. To reconstruct a secret, it uses  $(t, n)$  threshold access structure, where  $t$  out of  $n$  shares are required. Shamir's secret sharing scheme can be adopted in MANETS. Even the role of the dealer can be played by the nodes of MANET itself. This is achieved by using a bi-variate polynomial. This is discussed in section 3.1.

### 1.4 Related work

One common issue faced by MANET when applying cryptography is, how to distribute the role of CA or trusted authority, many proposals use secret sharing technique to distribute secret key  $s$  of CA or trusted authority to secure MANET. Zhou and Haas[6] were the first to propose distributed CA for MANETS. They used threshold cryptography to distribute the role of the Certification Authority (CA) in a PKI scenario among a set of selected servers. However, this proposal is not suitable for a purely ad-hoc environment as these selected nodes may not always be available. Kong et al.[13] adapted a similar idea to distribute trust among all the nodes. However, their specific RSA threshold scheme has been proved insecure[14][15]. Shamir secret sharing technique[8] is the most widely used secret sharing technique. We show that Shamir secret sharing technique along with the use of bi-variate polynomial helps to distribute the secret of CA among all nodes of MANET. In other works, bi-variate polynomials have already been used to dynamically allow new nodes joining the network without the need of any external trusted party. This technique is the result of inspiration from the original work of[16]. Anzai et al.[17] and Herranz et al.[18] constructed decentralized, flexible, dynamic group key distribution schemes by using polynomials in two variables. The goal is to generate common group secret keys. Saxena et al.[19] used similar technique to establish pairwise keys in a non-interactive way for a mobile ad-hoc scenario. Recently Daxing et al. [22] proposed aggregate signature algorithm for MANET using bilinear pairing and Hanaoka et al. [24] construct multi user setting signature with tight security based on BLS signature.

Our work is more related to the cryptographic techniques proposed for MANETs by Herranz et al. [18]. They proposed a fully self managed MANET and the ways

to authenticate communication among the nodes. Our paper proposes the node authentication in their set up using BLS signature proposed by Boneh et al.[10]. Our proposal reduces the size of keys used as it uses the bilinear pairing. This scenario is much suitable for MANET because its nodes are mostly resource constraint devices and they can not afford the heavy computational overhead required by larger keys.

## 2 preliminaries

### 2.1 Self-Organized PKI and Secret Sharing Technique

In self-organized PKI for MANETS, the role of PKI is completely distributed among the nodes of MANET using secret sharing scheme[8]. Blakley [7] and Shamir [8] were the first to introduce secret sharing techniques. In general a secret sharing scheme contains a dealer and a set  $U = \{u_1, u_2, \dots, u_n\}$  of  $n$  users. The dealer has a secret  $S$  and wants to distribute the share  $s_i$  of the secret corresponding to the user  $u_i$  privately. A valid subset  $u$  ( for :  $u \subset U$ ) of at least  $t$  number of users holding valid shares can reconstruct the secret  $S$ . The  $t$  is refereed as the threshold number and  $(t, n)$  is refereed to as the threshold access structure[8]. In our paper, we use Shamir's secret sharing technique that uses a  $(t, n)$  threshold access structure[8]. Shamir's secret sharing scheme uses  $(t, n)$  threshold access structures using polynomial interpolation. Let  $Z_q$  be a finite field with  $q > n$  and let  $S \in Z_q$  be the secret. The dealer picks a polynomial  $P(x)$  of degree at most  $t-1$ , where the constant term of  $P(x)$  is  $S$  and all other coefficients are selected from  $Z_q$  uniformly and independently at random. That is,

$$P(x) = S + \sum_{i=1}^{t-1} a_i * x^i$$

Every user  $u_i$  is publicly associated to a field element  $a_i$ . Distinct parties are mapped to distinct field elements. The dealer privately sends to user  $u_i$  the value  $[S]_i = P(a_i)$ , for  $i = 1, 2, \dots, n$ . Without loss of generality, we can assume that the set of parties willing to recover the secret  $S$  is  $P_1, \dots, P_t$ . The secret  $S$  can be obtained as  $\sum_{i=1}^t l_i * [s]_i$  where  $l_i = \prod_{j \neq i} \frac{a_j}{a_j - a_i}$  are the Lagrange coefficients. It is proven that any set of less than  $t$  parties obtain no information about  $S$ , that is, any secret is equally probable given their shares.

### 2.2 Bilinear Pairing and Related Assumptions[21]

Let  $G_1$  be a cyclic additive group generated by some element  $P$ , whose order is a prime  $q$ , and  $G_2$  be a cyclic multiplicative group of the same order  $q$ . Let  $a, b$  be elements of  $Z_q^*$ . We assume that the discrete logarithm problem (DLP) in both  $G_1$  and  $G_2$  are hard. A bilinear pairing is a map  $e : G_1 \times G_1 \rightarrow G_2$  with the following properties:

- Bilinear: For all  $S, T \in G_1$ ,  $e(aS, bT) = e(S, T)^{ab}$ .
- Non-degenerate: There exists  $S$  and  $T \in G_1$  such that  $e(S, T) \neq 1$ .
- Computable: There is an efficient algorithm to compute  $e(S, T)$  for all  $S, T \in G_1$ .

We have the following assumptions:

- The Decisional Diffie-Hellman problem (DDHP) in  $G_1$  should be easy.
- The DDHP in  $G_2$ , the computational Diffie-Hellman problem (CDHP) and the discrete logarithm problem (DLP) in both  $G_1$  and  $G_2$  should be hard.
- The inversion of the bilinear pairing be hard, i.e., the bilinear pairing inversion problem (BPIP) is defined as:
  - BPIP : Given  $S \in G_1$  and  $e(S, T) \in G_2$ , find  $T \in G_1$ .

### 2.3 BLS Signature[10]

This scheme was introduced by D. Boneh, B. Lynn, H. Schacham. It is based on Computational Diffie-Hellman assumption on certain elliptic curve. We discuss the Gap Diffie-Hellman Group where this signature scheme works.

**Gap Diffie-Hellman Groups (GDH Groups)** Consider a (multiplicative) cyclic group  $G = \langle g \rangle$ , with  $q = |G|$  a prime. There are three problems on  $G$ .

- Group Action: Given  $u, v \in G$ , find  $uv$ .
- Decision Diffie-Hellman : For  $a, b, c \in Z_q^*$ , given  $(g, g^a, g^b, g^c)$  decide whether  $c = ab$ .
- Computational Diffie-Hellman : For  $a, b \in Z_q^*$ , given  $(g, g^a, g^b)$ , compute  $g^{ab}$ .

The GDH group is defined as :

- $G$  is a  $\tau$ -decision group for Diffie-Hellman if the group action can be computed in one time unit, and Decision Diffie-Hellman can be computed on  $G$  in time at most  $\tau$ .
- The advantage of an algorithm  $A$  in solving the Computational Diffie-Hellman problem in a group  $G$  is  $Adv_{CDH_A} = Pr[A(g, g^a, g^b)] = g^{ab} : a, b \xleftarrow{R} Z_q^*$  Where the probability is over the choice of  $a$  and  $b$ , and the coin tosses of  $A$ . We say that an algorithm  $A$   $(t, \epsilon)$ -breaks Computational Diffie-Hellman in  $G$  if  $A$  runs in time at most  $t$ , and  $Adv_{CDH_A} \geq \epsilon$ .
- A prime order group  $G$  is a  $(\tau, t, \epsilon)$ -GDH group if it is a  $\tau$ -decision group for Diffie-Hellman and no algorithm  $(\tau, \epsilon)$ -breaks Computational Diffie-Hellman on it.

### Signature Scheme

- Setup of protocol:  
**Public information:** cryptographic hash function  $H : \{0, 1\}^* \rightarrow G_1$  and cryptographic bilinear map  $e : G_1 \times G_1 \rightarrow G_2$   
**Signer's public key:** generator  $P \in G_1, P_{pub} = sP$ , where  $s$  is the secret key and  $P_{pub}$  is the public key.
- Sign: For any message  $M \in \{0, 1\}^*$ , signature is computed as  $sig = sH(M)$
- Verify: Signature is only valid if the following equation holds.

$$e(P, sig) = e(P_{pub}, H(M))$$

- Proof:  $e(P, sig) = e(P, sH(M)) = e(sP, H(M)) = e(P_{pub}, H(M))$

## 3 Our proposal

This section is divided into four major phases namely Setup, Key Generation, Signature Generation Protocol and Signature Verification Protocol.

### 3.1 Setup

In this phase every node  $n_i$  receives partial share  $s_i$  of the MANET secret  $s$ . This is achieved using the following protocol.

- Let  $n$  be the number of nodes in the MANET,  $t$  be the threshold and  $k$  be the founding number of nodes.
- The founding number of nodes are such  $t \leq k \leq n$ .
- Every founding node chooses a bi-variate polynomial  $f_i(x, z)$ , symmetric in  $x, z$  and the max degree.
- Every node  $n_i$  computes  $f_{ij}(h(n_j), z)$  for all other founding nodes and itself,  $1 \leq i \leq k$ .
- Now every node secretly sends computed  $f_{ij}(h(n_j), z)$  to corresponding node  $n_j$ . Furthermore, node  $n_i$  includes the value  $y_i = f_i(0) * P$  in each of these messages.
- Finally every node has values received from other founding nodes and also it's own value  $f_{ii}(h(n_i), z)$  with it.  
 Then every node  $n_i$  computes  $f_i(z) = f(h(n_i), z) = \sum_{j \in k} f_{ji}(h(n_i), z)$ .
- Now every node  $n_i$  has partial secret  $s_i = f_i(0)$  and a secret equation  $f(h(n_i), z)$ .

The MANET secret function  $f(x, z) = \sum_{i \in n} f_i(x, z)$  and MANET secret key is  $s = f(0, 0)$  are safe and hidden. This secret information can only be reconstructed if and only if there are at-least  $t$  nodes having partial share of MANET secret. For a new node  $n_w$  trying to join the network, it has to request at-least  $t$  nodes for the values  $f_{iw}(h(n_i), h(n_w))$ . When  $t$  nodes accept the node  $n_w$  request, then they send  $f_{iw}(h(n_i), h(n_w))$  to node  $n_w$ . Now node  $n_w$  has  $t$  values and these values are used in Lagrange's interpolation to derive a secret polynomial corresponding to node  $n_w$ , Lagrange's interpolation is applied as follows:

–

$$f_w(z) = f(h(n_w), z) = \sum_{n_j \in n} \prod_{n_i \in n, n_i \neq n_j} \frac{(z - h(n_i))}{(h(n_j) - h(n_i))} * f(h(n_j), h(n_w))$$

- The partial secret of node  $n_w$  is  $f_w(0)$  and secret polynomial of node  $n_w$  is  $f_w(z)$  i.e.,  $f(h(n_w), z)$

### 3.2 Key Generation

After every node  $n_i$  has received a partial secret  $s_i$ , now the nodes run RSA key generation protocol. The protocol is responsible for generating a public ( $pk_i$ ) and private ( $sk_i$ ) key pair. The private key ( $sk_i$ ) is kept secret with the node  $n_i$  and public key ( $pk_i$ ) is made available to all other nodes. The public key  $pk_i$  is used to encrypt messages that are sent to node  $n_i$ , and the node  $n_i$  uses its private key  $sk_i$  to decrypt messages as well as to sign messages.

### 3.3 Signature Generation Protocol

Now every node  $n_i$  has two secret keys namely partial secret key of MANET  $s_i$  and individual secret key  $sk_i$ , partial secret key is used to partially sign a certificate and any  $t$  out of  $n$  nodes are required to sign a certificate to generate fully signed/valid certificate. When a node  $n_i$  wants to get a public key certificate, it asks its neighboring nodes to generate partial signature on the certificate linking  $n_i || pk_i$ . If the node  $n_i$  receives at-least  $(t - 1)$  partial signs, then the node itself can generate a partial sign using it's own partial share, now the node has  $t$  partially signed values, then it uses the following Lagrange's interpolation to generate a fully signed certificate.

- $p_i = H(m) * s_i$  where  $s_i$  is the individual share of each user and  $H(m)$  is the hash of message  $m$ .
- The final signature( $shm$ ) is computed as  $shm = \sum_{i \in t} p_i * L_i$ , where  $L_i$  is Lagranges Coefficient.  $L_i = \prod_{p_j \in t, j \neq i} \frac{(0 - h(N_j))}{(h(N_i) - h(N_j))}$

Now that every node obtains its certificate in the above described manner. Next we discuss the protocol to verify the certificate.

### 3.4 Signature Verification Protocol

Any node  $n_j$  can verify the certificate of node  $n_i$  by running the following protocol. Node  $n_j$  has the following information regarding node  $n_i$ :

- the signed certificate of node  $n_i$  ( $shm$ ).
- the public key of the MANET ( $PK$ ) and value  $P$ .
- ID of node  $n_i$  and public key of node  $n_i$  ( $N_i || pk_i$ ).

The node  $n_j$  uses BLS signature to verify the certificate:

- Verify  $e(shm, P) = e(H(m), PK)$  If true certificate is valid, else invalid.

### 3.5 Example

- **Setup**
- Let the initial set of nodes  $N_M = \{N_1, N_2, N_3, N_4\}$   
No.of Nodes = 4
- Public Parameters :  
An additive group  $G$  of prime order  $q = 4019$ .
  - The curve used is  $E(F_{4019}) : y^2 = x^3 + 1$
  - The Generator is  $P = E(3198, 578)$
  - Let  $t = 2$  (degree of polynomials) and  $k = 67$  (Field of Polynomials)
- An admissible bilinear pairing - Weil Pairing
- Two explicit collision resistant hash functions - HTP(Hash to Point) :  $\{0, 1\}^* \rightarrow G_2$  and HTR(Hash to Range) :  $\{0, 1\}^* \rightarrow G_1$  where HTP hashes the given message onto the elliptic curve group  $G_2$  and HTR hashes the given value to the group  $G_1$ .
- Each node chooses a random symmetric-bivariate polynomial in  $GF(67)$   
 $N1 = 3x^2z + 3z^2x + 8xz + 5z + 5x + 5$ ,  $N2 = 5x^2z + 5z^2x + 3xz + 8z + 8x + 9$   
 $N3 = 8x^2z + 8z^2x + 5xz + 3z + 3x + 6$ ,  $N4 = 2x^2z + 2z^2x + 4xz + 8z + 8x + 4$
- The implicit polynomial defined by all the nodes is  
 $F(x,z) = N_1 + N_2 + N_3 + N_4$   
 $= 18x^2z + 18xz^2 + 20xz + 24x + 24z + 24$
- The secret  $s$  of the MANET is  $F(0,0) = 24$ .
  
- Each node secretly sends to each of the other founding nodes the univariate polynomial  $F_{ij} = F_i(x, h(N_j))$ .
- The hash values of the nodes are  
 $h_{n1} = HTR('Node1', k) = 37$ ,  $h_{n2} = HTR('Node2', k) = 54$   
 $h_{n3} = HTR('Node3', k) = 25$ ,  $h_{n4} = HTR('Node4', k) = 17$
  
- Each node sends the following values to other Nodes :
- N1 also includes  $Y_1 = 5 * P = (152, 1437)$   
 $N_{11} = 44x^2 + 53x + 56$ ,  $N_{12} = 28x^2 + 6x + 7$   
 $N_{13} = 8x^2 + 3x + 63$ ,  $N_{14} = 51x^2 + 3x + 23$
- N2 also includes  $Y_2 = 9 * P = (409, 2266)$   
 $N_{21} = 51x^2 + 63x + 37$ ,  $N_{22} = 2x^2 + 10x + 39$   
 $N_{23} = 58x^2 + 59x + 8$ ,  $N_{24} = 18x^2 + 30x + 11$
- N3 also includes  $Y_3 = 6 * P = (3063, 3143)$   
 $N_{31} = 28x^2 + 18x + 50$ ,  $N_{32} = 30x^2 + 17x + 34$   
 $N_{33} = -x^2 + 36x + 14$ ,  $N_{34} = 2x^2 + 55x + 57$
- N4 also includes  $Y_4 = 4 * P = (3863, 2497)$   
 $N_{41} = 7x^2 + 13x + 32$ ,  $N_{42} = 41x^2 + 26x + 34$   
 $N_{43} = 50x^2 + 18x + 3$ ,  $N_{44} = 34x^2 + 51x + 6$



- Then all the nodes calculate their secret univariate polynomial from the recieved values.
- $S_1(x) = 63x^2 + 13x + 41$  ,  $S_2(x) = 34x^2 + 59x + 47$
- $S_3(x) = 48x^2 + 49x + 21$ ,  $S_4(x) = 38x^2 + 5x + 30$
- The public key,  $PK = s * P$   
 $= 24 * E(3198,578) = E(2651, 2267)$
- PK should also equal to  $Y_1 + Y_2 + Y_3 + Y_4$   
 $= E(152,1437)+E(409,2266)+E(3063,3143)+E(3863,2497) = E(2651, 2267)$
- Each node calculates its share from  $S_i(0)$ .  
 The shares of the nodes are -  $S_1 = 41, S_2 = 47, S_3 = 21, S_4 = 30$
- These shares can be verified by substituting hash value of the nodes in the following polynomial  $f(z) = F(0,z) = 24 * z + 24$
- If Node  $N_5$  wants to join the MANET, It should identify it self to 3 other nodes and request for acceptance.  $\{N_2, N_3, N_4\}$   
 $h_{n5} = HTR('Node5', k) = 27$
- $N_5$  receives the following values  
 $S_{25} = S_2(27) \text{ mod } 67 = 28$ ,  $S_{35} = S_3(27) \text{ mod } 67 = 22$   
 $S_{45} = S_4(27) \text{ mod } 67 = 62$
- $N_5$  computes its secret univariate polynomial by using Lagrange interpolation  
 $S_5(x) = 17 * x^2 + 18 * x + 2$
- **Key Generation**
- Each node computes its own key pair as follows :  
 Node 1 = [(89,649),(189,649)], Node 2 = [(17,321),(25,321)]  
 Node 3 = [(63,115),(7,115)], Node 4 = [(91,202),(11,202)]
- **Signature Generation**
- The share of each node in MANETs secret key is used as secret key for signature i.e  $S_1 = 41, S_2 = 47, S_3 = 21, S_4 = 30$
- Each node produces a certificate by linking Id with PK  
 $m_1 = 'Node1' + '89' + '649'$ ,  $m_2 = 'Node2' + '17' + '321'$   
 $m_3 = 'Node3' + '63' + '115'$ ,  $m_4 = 'Node4' + '91' + '202'$
- Then all the nodes exchange partial signatures to compute fully signed certificate.
- If Node 1 wants to compute its certificate ( $m_1 = 'Node1' + '89' + '649'$ ), then it requests Node 2, Node 3 and Node 4 for their partial signatures.
- Here  $P = E(3198,578)$ ,  $s = 24$ ,  $mpub = E(2651,2267)$  and  $s2 = 47, s3 = 21, s4 = 30$
- $hm_1 = HTP(m_1) = E(163, 1362)$
- The partial signatures of Nodes 2, 3 and 4 are  
 $p_2 = (hm_1) * s2$ ,  $p_3 = (hm_1) * s3$ ,  $p_4 = (hm_1) * s4$

- By lagranges interpolation we get the signature on the message1 as  $shm1 = E(2350, 3239)$ .
- **Signature Verification**
- Calculate  $e(hm1, m_{pub}) = 1365 * a + 2045$
- Calculate  $e(shm1, P) = 1365 * a + 2045$  this is equal to  $e(hm1, m_{pub})$
- Hence Verified
- Message communication after verification
- Public and private key pairs of each node
  - Node 1 =  $[(e1, n1), (d1, n1)] = [(89, 649), (189, 649)]$
  - Node 2 =  $[(e2, n2), (d2, n2)] = [(17, 321), (25, 321)]$
  - Node 3 =  $[(e3, n3), (d3, n3)] = [(63, 115), (7, 115)]$
  - Node 4 =  $[(e4, n4), (d4, n4)] = [(91, 202), (11, 202)]$
- Message (M) = 56
- If Node 1 wants to send a message to Node 3, then Node 1 Encrypts the message using Node 3's public key and sends to Node 3.
- $C = \text{Encrypt}(M, e3, n3) \quad C = \text{mod}(56, 115)^{63}$   
Encrypted Value C = 463
- Node 3 receives the Cipher value and Decrypts the message using Node 3 private key.
- $M = \text{Decrypt}(C, d3, n3) \quad m = \text{mod}(463, 115)^7$   
Decrypted Value M = 56

#### 4 Conclusion

In this paper, we proposed a new scheme of verifying a certificate in decentralized PKI based MANETS. In our scheme the nodes of the MANET holds a secret share and every node chooses its own public and private keys. The public key is associated with the node identity in the certificate. This certificate management is done using BLS Signature. Our scheme uses a bivariate polynomial to reduce the communication overhead. The same technique can be used in performing other functionalities of MANET like implementing threshold operations in sub group nodes communication and share verification etc.

#### References

1. F. Anjum and P. Mouchtaris, Security for wireless ad hoc networks. Wiley-Blackwell, Mar. 2007.
2. Vanesa Daza, Javier Herranz, Paz Morillo, Carla Rfols, Cryptographic techniques for mobile ad-hoc networks, Computer Networks, Volume 51, Issue 18, 19 December 2007.
3. Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (Mobicom 2002), September 2002.
4. Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet leashes: A defense against wormhole attacks in wireless networks. In Proceedings of IEEE Infocom 2003, April 2003.

5. S. Kent and T. Polk. Public-key infrastructure (x.509) (pkix) charter. <http://www.ietf.org/html.charters/pkix-charter.html>.
6. L. Zhou, Z.J. Haas, Securing ad hoc networks, IEEE Network 13 (6) (1999) 24-30.
7. G.R. Blakley, Safeguarding cryptographic keys, in: Proceedings of the National Computer Conference, American Federation of Information, Processing Societies Proceedings, vol. 48, 1979, pp. 313-317.
8. A. Shamir, How to share a secret, Communications of the ACM 22 (1979) 612-613.
9. Seung Yi and Robin Kravetso. Moca : Mobile certificate authority for wireless ad hoc networks. In The second annual PKI research workshop (PKI 03), Gaithersburg, 2003.
10. Dan Boneh, Ben Lynn, and Hovav Shacham (2004). "Short Signatures from the Weil Pairing". Journal of Cryptology. 17: 297-319.
11. Djenouri, Djamel, L. Khelladi, and N. Badache. "A survey of security issues in mobile ad hoc networks." IEEE communications surveys 7.4 (2005): 2-28.
12. Stallings, William (1990-05-03). Cryptography and Network Security: Principles and Practice. Prentice Hall. p. 165. ISBN 9780138690175.
13. H. Luo, J. Kong, P. Zerfos, S. Lu, L. Zhang, URSA: ubiquitous and robust access control for mobile ad hoc networks, IEEE/ACM Transactions on Networking 12 (6) (2004).
14. M. Narasimha, G. Tsudik, J.H. Yi, On the utility of distributed cryptography in P2P and MANETs: the case of membership control, in: Proceedings of ICNP203, 2003, pp. 336-345.
15. S. Jarecki, N. Saxena, J.H. Yi, An attack on the proactive RSA signature scheme in the URSA ad hoc network access control protocol, in: Proceedings of the SASN04, 2004, pp. 19.
16. C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, Perfectly-secure key distribution for dynamic conferences, in: Proceedings of Crypto92, LNCS, vol. 740, Springer-Verlag, 1993, pp. 471-486.
17. J. Anzai, N. Matsuzaki, T. Matsumoto, A quick group key distribution scheme with entity revocation, in: Proceedings of Asiacrypt99, LNCS, vol. 1716, Springer-Verlag, 1999, pp. 333-347.
18. V. Daza, J. Herranz, G. Sez, Constructing general dynamic group key distribution schemes with decentralized user join, in: Proceedings of ACISP03, LNCS, vol. 2727, Springer-Verlag, 2003, pp. 464-475.
19. N. Saxena, G. Tsudik, J.H. Yi, Efficient node admission for short-lived mobile ad hoc networks, in: Proceedings of ICNP05, 2005, pp. 269-278.
20. Singh Nidhi, Appala Naidu Tentu, Abdul Basit, and V. Ch Venkaiah. "Sequential secret sharing scheme based on Chinese remainder theorem." In Computational Intelligence and Computing Research (ICCC), 2016 IEEE International Conference on, pp. 1-6. IEEE, 2016.
21. Boneh, Dan, and Matt Franklin. "Identity-based encryption from the Weil pairing." Annual International Cryptology Conference. Springer Berlin Heidelberg, 2001.
22. Daxing Wang, Jikai Tang. "Efficient Aggregate Signature Algorithm and Its Application in MANET". in: International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering. vol. 7, No:11, 2013.
23. Adul Basit, N Chaitanya Kumar, V. Ch. Venkaiah, Salman Abdul Moiz, Appala Naidu, Wilson Naik "Multi-stage Multi-secret Sharing Scheme for Hierarchical Access Structure." In International Conference on Computing, Communication and Automation (ICCCA), 2017 IEEE International Conference.
24. Hanoka G, Shuldt J.C.N, "On signatures with tight security in the multi-user setting" (2017) in : Proceedings of 2016 International Symposium on Information Theory and Its Applications, ISITA 2016, art. no. 7840392, pp. 91-95.

## Authors

**N Chaitanya Kumar** received M.Tech from JNTU Hyderabad, and he did Bachelor degree in computer science. Currently, he is pursuing his PhD in Computer Science from the University of Hyderabad. His research interests include Information security, Cryptography in MANET.

**Abdul Basit** received Master of computer application from Jamia Hamdard University New Delhi. He did Bachelor of Science in Information technology from SMU Gangtok. Currently, he is pursuing his PhD in Computer Science from the University of Hyderabad. His research interests include Information security, Cryptography, Cyber security, and Algorithms.

**Priyadarshi Singh** received M.Tech from IIT(ISM) Dhanbad. He did Bachelor degree in Information Technology. Currently, he is pursuing his PhD in Computer Science from the University of Hyderabad. His research interests include Cryptography, Public key infrastructure.

**V. Ch. Venkaiah** obtained his PhD in 1988 from the Indian Institute of Science (IISc), Bangalore in the area of scientific computing. He worked for several organisations including the Central Research Laboratory of Bharat Electronics, Tata Elxsi India Pvt. Ltd., Motorola India Electronics Limited, all in Bangalore. He then moved onto academics and served IIT, Delhi, IIIT, Hyderabad, and C R Rao Advanced Institute of Mathematics, Statistics, and Computer Science. He is currently serving the Hyderabad Central University. He is a vivid researcher. He designed algorithms for linear programming, subspace rotation and direction of arrival estimation, graph colouring, matrix symmetriser, integer factorisation, cryptography, knapsack problem, etc.

**Subba Rao Y V** obtained his PhD from the University of Hyderabad. Currently, he is an Assistant Professor in the School of Computer and Information Sciences, University of Hyderabad. His area of interests includes Cryptography, Theory of Computation, etc.