Secured Modified Bloom's based Q-composite Key Distribution for Wireless Sensor Networks

Shruthi. P & M. B. Nirmala

Department of Computer Science and Engineering, Siddaganga Institute of Technology,
Tumkur, Karnataka, India
E-mail : pshru35@gmail.com, nirmalamb@gmail.com

***Abstract* – The security issue in a wireless sensor network (WSN) has been drawing considerable research attention in recent years. However Key management, a basic security service, becomes the core design for various security services, such as encryption and authentication. To increase the connectivity of each key in a large-scale WSN and to enlarge its maximum supportable network size and to provide good resilience various algorithms were proposed Although several methods for supporting performance have been proposed in the last few years, it is still far from a broad use in several applications. Hence in this paper, we propose a method based on random pair wise distribution scheme called q-composite modified bloom's distribution scheme to provide good network resilience and substantially support a large network.**

Keywords – WSN, random pair wise distribution scheme, resilience, q-composite modified bloom's distribution scheme.

I. INTRODUCTION

The development of wireless sensor networks has become an important research topic in recent years due to critical applications such as emergency response, medical monitoring, military tracking, energy management and pollution monitoring. A wireless sensor network would poses low bandwidth and computing power, limited memory and energy resources. Large scale of sensor nodes are being prone to failure, without a central device, hence a sensor network can be easily assaulted or compromised by adversaries because it is often deployed in unattended Environments. To enhance its security, researchers have come up with a number of security services, including key management. A key management protocol for WSNs should be simple and light due to limited processing power, battery life, communication bandwidth And memory space of the sensor nodes.

Currently, there are three general key agreement schemes: trusted-server or arbitrated protocol, self-enforcing, and key pre-distribution scheme (Du, Ding, Han, and Varshney, 2003). The trusted-server scheme requires a trusted server to establish shared-session keys between nodes and is prone to directed attacks against the central point of weakness. Another key agreement scheme is the self-enforcing scheme, which depends upon asymmetric protocols and algorithms. The only practical scheme for key distribution in large sensor networks is key pre-distribution, where key information is installed in each sensor node prior to deployment. Typically, two solutions have been used for this, i.e. a single mission key where all nodes carry a master secret key or a set of separate n – 1 keys, each being a pair wise set that is privately shared with another sensor node. Imagine a scenario in which a sensor network is deployed for homeland security. If there is no authentication and access control over this sensor network, an intruder can potentially give a false command to the sensor nodes and turn them into sleep mode without detection of abnormal activities. Security protocols are rooted on secret keys which are pre-shared among the members in the network. Members in the network use the secret key to authenticate other members or encrypt the sensitive data that is transmitted in the air. However, using the same secret key on every wireless link will significantly increase the chance of crypto-analytic attacks. Keys that are different for different links are called link-dependent keys or session keys. Session keys have higher security assurance because they vary over time and space, and thus are preferred for use on wireless links. Master-key-based schemes are those in which every node shares a single preinstalled Master-key. Session keys used on different wireless links can be negotiated, for example, a simple three way handshaking and authentication protocol [13] based on the Master-key. This type of key management scheme has the underlying assumption that the sensor nodes are tamper proof and the master-key which is stored inside each node cannot be retrieved by the adversary [12][13]. Once the master-key is prone to attack, the adversary can use it to break the security of the entire network.

Another issue to consider in sensor network security is the design of protocols to bootstrap the establishment of a secure communications infrastructure from a collection of sensor nodes which may have been pre-initialized with some secret information but have had no prior direct contact with each other. We refer to this problem as the bootstrapping problem. A bootstrapping protocol allows the nodes deployed at a later time to join the network securely. The difficulty of the bootstrapping problem stems from the numerous limitations of sensor networks. some of the more important ones include the inability to utilize existing public key cryptosystems (since the expensive computations involved could expose the power-constrained nodes to a denial-of-service attack), the inability to pre-determine which nodes will be neighbors after deployment, and the inability of any node to put absolute trust in its neighbor (since the nodes are not tamper resistant and are vulnerable to physical capture).hence from the above discussion we can find many disadvantages of the traditional master key based key distribution scheme which poses a severe security issues in WSN's. The Q-Composite Random Key Pre-distribution Scheme requires that two nodes have at least q common keys to set up a link [12]. As the amount of key overlap between two nodes is increased, it becomes harder for an adversary to break their communication link. The Q-Composite Scheme achieves security under small scale attacks while being vulnerable under large scale attacks and is useful where large scale attacks are easily detected. There are two limitations that we need to care when we are discussing about the Q-composite key distribution scheme the primary one is that each node in the network captures a certain number of keys in its node which are obtained from its key pool. In that case if size of the network is small there will no problem in the network however this will not be the case if we are concerned with the large networks .memory and overhead problems may pose a severe issue though the good resilience is provided. Hence therefore an alternative key distribution scheme has to be developed to overcome the above limitations. In this paper we verify the various key distribution schemes and show how the q-composite modified bloom's distribution scheme overcomes the limitations and provide good scalability for the network.

II. Related Works

A. *Master key based key pre-distribution*

A single key is used for all the nodes in the network. All the secured data processing n sensor nodes depend on a single secret key. All the nodes in the entire network use only a shared secret key.

Broadcast session key negotiation protocol (BROSK) is based on single master key which is pre-deployed to sensor nodes. A pair of sensor nodes (S_i, S_j) exchanges random values. They use master key Km to establish session key.

$$K_{i,\,j} = PRF\ (K_m|RN_i|RN_j)$$

Each sensor uses one unit of memory to store the master key. It is possible to derive all link keys once the master key is compromised; therefore the scheme has very low resilience. Lightweight key management system [Dutertre et al. 2004] proposes a solution with slightly better resilience where more than one master key is employed. It assumes in a WSN,group of sensor nodes are deployed in successive generations of size θ. Each sensor node stores a group authentication key bk1 and a key generation key b_{k2}. If two sensor nodes Sa and Sb are from the same generation, they authenticate each other by using the authentication key b_{k1}. They exchange random nonce values RNa and RNb, and establish the session key

$$K_{a,\,b} = PRF\ (b_{k2}|RN_A|RN_B)$$

It is possible that nodes are from two different generations. A sensor node Sa, of an old generation i, stores a random value RN_a and a secret $Sa_{,j}$ for each new generation j. Secret $Sa_{,j}$ is used to authenticate sensor nodes from new generation j. Node Sb of new generation j can authenticate itself by generating the secret $S_{a,j} = PRF(gk_j\ |RN_A)$ given RN_a. Secret gk_j is only known to nodes of new generation j. Once authenticated, both parties use $Sa_{,j}$ as the key generation key to generate the pair-wise key $Ka_{,b}$. If there are g such generations, each sensor needs at most 4 + 2g units of memory to store the keys. Resilience of the scheme is still low because an adversary only needs to compromise the secrets bk_1, bk_2 and gk_j of Generation j to compromise all the links of nodes in generation j. Furthermore, adversary may log the messages flowing in the network to process later when the required credentials are compromised completely.

B. *Eschenauer and Gilgor's Key Pre-Distribution Scheme*

Eschenauer and Gilgor (Eschenauer and Gilgor, 2002) proposed a random key pre-distribution scheme based on probabilistic key sharing and utilization of a simple shared-key discovery protocol for key distribution, key revocation, and node re-keying. Prior to a sensor network deployment, each sensor node receives a key ring with a randomly chosen subset of keys from a large key pool. Upon deployment and network initialization, sensor nodes will be able to establish a secure and direct communication link provided that a shared key exists between one or more pairs of sensor nodes. Due to the random distribution of keys to each sensor node, it is probable that a shared key may not be available, necessitating an intermediary node with a common key between the two sensor nodes to establish a common session key.

Eschenauer and Gilgor found that to establish "almost certain shared-key connectivity for a 10,000-node network, a key ring of only 250 keys randomly selected from a 100,000 pool has to be pre-distributed to every sensor node." It consists of three phases: key pre-distribution, shared key-discovery, and path-key establishment. The key pre-distribution phase occurs prior to sensor node deployment. During this phase, a large pool of P random keys and their key identifiers are generated. Each sensor node receives a subset of randomly chosen k keys and their associated key identifiers plus a shared key with a trusted controller node that stores all keys and associated identifiers for every network node. The shared-key discovery phase occurs during an initialization period where each sensor node attempts to discover its neighbors with which its shares a common key(s). This can be done by the broadcast of each sensor nodes key identifier list in plaintext or a list α, EKi (α), i = 1… k. Decryption with a proper key of EKi (α) would meet the challenge and establish a shared key with the broadcasting node. The path-key establishment phase is used to assign a path-key to selected sensor-node pairs within a defined communication range that do not share a common key but are connected by two or more links created during the shared-key discovery phase. An intermediary node generates the path-key with a shared key between two or more unconnected link nodes.

III. EXISTING SYSTEM

A. *q-composite key distribution schemes*

In the master key scheme, any master key is the base to which the security mechanism holds for. We propose a modification to the existing scheme where q common keys (q > 1) are needed, instead of just one. By increasing the amount of key overlap required for key-setup, we increase the resilience of the network against node capture.

The operation of the q-composite keys scheme is not similar to that of the existing scheme; it differs in the size of the key pool S and the fact that multiple keys are used to establish communications instead of just one. In the initialization phase and in a random distribution scheme, we pick a set S of random keys out of the total key space, where $|s|$ is computed. For each node, we select m random keys from S (where m is the number of keys each node can carry in its key ring) and store them into the node's key ring. In the key-setup phase, each node must discover all common keys it possesses with each of its neighbours. This can be accomplished with a simple local broadcast of all key identifiers that a node possesses. While broadcast-based key discovery (BROSK) is straightforward to implement, it has the disadvantage that a casual eavesdropper can identify the key sets of all the nodes in a network and thus pick an optimal set of nodes to compromise in order to discover a large subset of the key pool S. A more secure, but slower, method of key discovery could utilize client puzzles such as a Merkle puzzle [14]. Each node could issue m client puzzles (one for each of the m keys) to each neighboring node. Any node that responds with the correct answer to the client puzzle is thus identified as knowing the associated key. After key discovery, each node can identify every neighbor node with which it shares at least q keys. Let the number of actual keys shared be q, where q >q'. The main tedious method in the q-composite distribution scheme is generation of the key pool size and the generation of the key rings. Hence the total number of ways to choose two key rings with i keys in common is the product of the various terms, i.e.

$$\boldsymbol{p(i)} = \frac{\binom{|s|}{i}\binom{|s|-\mathrm{i}}{2m-i}\binom{2(\mathrm{m-i})}{m-i}}{\binom{|s|}{i}} \tag{1}$$

As evaluating the q-composite scheme we found that though it provides a good resilience for the network but it poses major shortcomings such as memory and overhead also. In addition it does not provide satisfactory security for large communication network For example when q = 2, P = 0.33, m = 200: for this criteria if 50 nodes get compromised, the additional communication link compromised will be 4.74 in Q composite, where as 9.52 in EG scheme. But the drawback is if the large number of nodes compromised, there is a possibility of compromising the entire communication link and finally it does not provide scalability also.

B. *Bloom's key distribution schemes*

Bloom's key distribution method [15] allows any pair of users in the system to find a unique shared key. According to this method, a network with N users and a collusion of less than t+1 users cannot reveal the keys which are held by other users. Thus the security of the network depends on the chosen value of t, which is called Bloom's secure parameter (t<<N). Larger value of t leads to greater resilience but one needs to be careful in choosing a high value because that increases the amount of memory required to store key information. During the initialization phase, a central authority or base station first constructs a (t + 1)×N matrix P over a finite field GF(q), where N is the size of the network and q is the prime number. P is known to all users and it can be constructed using Vandermonde matrix. It can be shown that any t+1 columns of P are linearly independent whenn_i, i=1, 2…N are all distinct.

$$P=\begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ n & n^2 & n^3 & \ldots & n^n \\ n^2 & (n^2)^2 & (n^3)^3 & \ldots & (n^n)^2 \\ n^t & (n^2)^t & (n^3)^t & \ldots & (n^n)^t \end{bmatrix}$$

Then the central authority selects a random (t + 1) × (t + 1) symmetric matrix S over GF (q), where S is secret and only known by the central authority. An N × (t + 1) matrix A = (S. P) T is computed. Because S is symmetric, it is easy to see

$$K=A.P=(S.P)^T.P = P^T.S^T.P = P^T.S=(A.P)^T=K^T$$

User pair (i, j) will use Kij, the element in row i and column j in K, as the shared key. Because Kij is calculated by the i-th row of A and the j-th column of P, the central authority assigns the i-th row of A matrix and the i-th column of P matrix to each user i, for 1, 2….. N. Therefore, when user i and user j need to establish a shared key between them, they first exchange their columns of P, and then they can compute Kij and Kji, respectively, using their private rows of A. The t-secure parameter guarantees that no compromise of up to t nodes has any information about Kij or Kji.

In the Bloom's scheme [15] for any two nodes to generate a common key, each node should store column of public matrix and row of the calculated secret matrix. Since every sensor node is provided with limited memory and energy, it will be difficult to store both the row and column in the sensor memory for a large network.

IV. PROPOSED SCHEME

The proposed method we use here is modified Bloom's scheme [15]. The reason that we don't prefer bloom's is that in the original Bloom's scheme all the computations involved in generating the keys are based on Vandermonde matrix which is a public matrix (P) and known to even the adversaries. Here, to make sure that any t+1 columns of P are linearly independent i.e., to generate unique keys, all the values in the matrix are chosen to be distinct. However, for large values of t, number of rows in the matrix increases and which in turn corresponds to a greater value in the columns because the column values increases in a geometric series. In the Bloom's scheme [15] for any two nodes to generate a common key, each node should store column of public matrix and row of the calculated secret matrix. Since every sensor node is provided with limited memory and energy, it will be difficult to store both the row and column in the sensor memory for a large network. To reduce the computation and memory overhead in Bloom's scheme, instead of using Vandermonde matrix [16] we propose the use of non-binary Hadamard matrix as the public matrix.

In this paper we propose modified bloom's q-composite key distribution scheme .The main aim is to provide good scalability for the network, provide less overhead and memory consumptions and finally to obtain a good resilience in the network.

A. Basic terminology

A key space is defined as a pair of matrices in this project. From a key space, a set of candidate secrets are generated. These candidate secrets form a pool of candidate secrets N. For the QC scheme, a key space corresponds to a single symmetric key. Often, we use the terms, "key space" and "pool of candidate secrets", interchangeably. A key pool S consists of a set of key spaces. A key ring is a set of secrets associated with each node, or one may consider as memory usage of a node assigned for storing secrets. For each node, a set of secrets will be randomly picked from the candidate-secret pools generated from key spaces in a key pool and stored into the node's memory to form a key ring. We assume only one secret can be chosen from each key space for each node. Shared secrets are shared keys or keying materials derived from the same key space. In addition to the concepts of the cryptographic keys, we also define the following terms. We use the terms, a neighbor and a neighboring node to represent a node that is within communication range of a node of interest. It is unnecessary for a neighboring node to establish an encrypted communication link to a node of interest. On the other hand, an adjacent node is the neighboring node that establishes an encrypted communication link to a node of interest. Finally, network size represents the number of nodes in a network. Since the MBBQ scheme is the generalized framework, each model is specified by the parameters. To investigate the performance tradeoffs of the MBBQ scheme, we define the following parameters. Memory usage M is the amount of memory assigned for a key ring, or one may consider as the amount of memory assigned for storing secrets on each sensor node. For all the schemes, the memory usage to store secrets are calculated based on the equation, M = m × (λ+1), where m is the key ring size, which is the number of secrets in a key ring. Required key overlapping q is the required number of shared secrets to establish an encrypted communication link. Key ring size m is the number of secrets stored on a key ring. The value of the key ring size varies depending on the model in conjunction with the value of λ under the restriction that M = m × (λ + 1). λ denotes the threshold of the secure property provided by the Bloom's scheme. λ also indicates the number of secrets generated from each key space since we assume that the number of secrets |N| is λ + 1, where N represents the candidate-secret pool and |N| is the size of the candidate-secret pool. Thus, each key space generates λ + 1 secrets. Network size n is the number of nodes in a network.

B. Modified bloom's-Based Q-Composite Scheme

In this section, we introduce the modified Blom-based q-composite scheme (MBBQ), a framework for random key pre-distribution schemes the basic scheme explores the key pool size |S| to study the random key pre-distribution scheme. The QC scheme reinforces the basic scheme in resiliency against the node capture attack by using the greater key overlapping q. The Bloom's scheme only uses a single key space to establish pair wise keys. The Blom-based scheme (BB) extends the Bloom's scheme by introducing the multiple key spaces. The MBBQ scheme is built upon the BB scheme but requires that q pairs of secrets must be generated from the common key spaces to establish a link instead of only one key space. As we can observe, the dimensions of the research area have not been exhaustively covered by the previously proposed random key pre-distribution schemes. The MBBQ scheme covers all the three dimensions simultaneously. First of all, the MBBQ scheme is obviously equivalent to the BB scheme when q = 1 since the difference is the required key overlapping to establish an encrypted link. Interestingly, the MBBQ scheme is equivalent to the basic scheme when q = 1, m = M, and λ = 0, and equivalent to the QC scheme when q > 1, m = M, and λ = 0 since each key space corresponds to a single symmetric key when the generator matrix G is a square (λ + 1) × (λ + 1) Hadamard matrix which is a secret matrix obtained from the total number of node values in the network. The central authority stores each row of a matrix in the node memory with corresponding index. Finally user pair (i, j) can compute the key by generating the Hadamard matrix and multiplying the secret row stored in the node with column of the Hadamard matrix corresponding to the node index with which it want to communicate. In our proposed system the main utility of modified bloom's can be generalized as being the generator of the secrets for the prescribed nodes of with the λ value. To make more elaborate about the proposed system our main target is handling the security issue, to precisely explain the various dimensions of the MBBQ scheme initially we affirm that the model $\mathrm{M}_{1,\mathrm{M},0}$ is not as resilient as the model $\mathrm{M}_{1,\mathrm{M},\lambda}$ where and $0 < \lambda < M$, because the models $\mathrm{M}_{1,\mathrm{M},\lambda}$ where 0 < m < M and $0 < \lambda < M$, provide the λ-secure property. The effect of λ -secure property of the models$\mathrm{M}_{\mathrm{q},\mathrm{M},\lambda}$, where $0 < m < M$ and $0 < \lambda < M$, indicates that a certain number of nodes must be captured before the portion of a network is compromised. Thus, the fraction of communication links compromised remains low until a certain number of nodes are captured. On the other hand, the models $\mathrm{M}_{\mathrm{q},\mathrm{M},0}$allow compromising the portion of a network even with one captured node. This behavior of the models$\mathrm{M}_{\mathrm{q},\mathrm{M},\lambda}$ where $0 < m < M$ and $0 < \lambda < M$, is favorable to protect against the node capture attack because it provides a network administrator with more chances to detect the node capture attack before a portion of a network is compromised. For this reason, we consider that the models $\mathrm{M}_{\mathrm{q},\mathrm{M},\lambda}$where $0 < m < M$ and $0 < \lambda < M$, are more resilient than the models $\mathrm{M}_{\mathrm{q},\mathrm{M},0}$ although some models $\mathrm{M}_{\mathrm{q},\mathrm{M},\lambda}$where $0 < m < M$ and $0 < \lambda < M$, are more insecure than the models$\mathrm{M}_{\mathrm{q},\mathrm{M},0}$ in terms of the fraction of the compromised communication links once a certain number of nodes are captured and compromised. Furthermore, the models $\mathrm{M}_{1,\mathrm{M},\lambda}$ with largerλ, where $0 < m < M$ and$0 < \lambda < M$, become more resilient against node capture of the models $\mathrm{M}_{1,\mathrm{M},\lambda}$, where $0 < m < M$ and $0 < \lambda < M$. as we can observe from, the model $\mathrm{M}_{1,\mathrm{M},\lambda}$with the largerλ, where $0 < m < M$and $0 < \lambda < M$, requires the larger number of captured nodes before the portion of the network is compromised. Thus, the model $\mathrm{M}_{1,\mathrm{M},\lambda}$ with larger λ, where $0 < m < M$ and $0 < \lambda < M$, is more tolerable against node capture. Intuitively, it is because a node with larger λ has the larger -secure property, and reveals less information to the adversary since the corresponding key ring size m is the smaller. Thus, a larger number of nodes are required to obtain a certain amount of secret information.

Therefore, the MBBQ scheme as the generalized framework helps us to provide the broader view of the random key pre-distribution schemes. In addition, the evaluation and comparison of the different schemes can be performed more systematically. In general, a model of the MBBQ scheme is determined by the value of q, m, and λ. Thus, to facilitate the representation of MBBQ models, let $M_{q,m,\lambda}$ be the model that has parameters q, m and λ for the given key ring size M.

C. Connectivity analysis

Local connectivity is a part of connectivity analysis. Local connectivity p_{local} is the probability of two neighboring nodes establishing an encrypted communication link. This metric is useful to predict the connectivity of any two nodes. For a moment now let us consider the key pool size is |S|, and each node has m secrets on its key ring, the local connectivity of the MBBQ scheme is $\mathbf{P=1-}\sum_{i=0}^{q-1}\frac{\binom{|s|-m}{m-i}\binom{m}{i}}{\binom{|s|}{m}}$

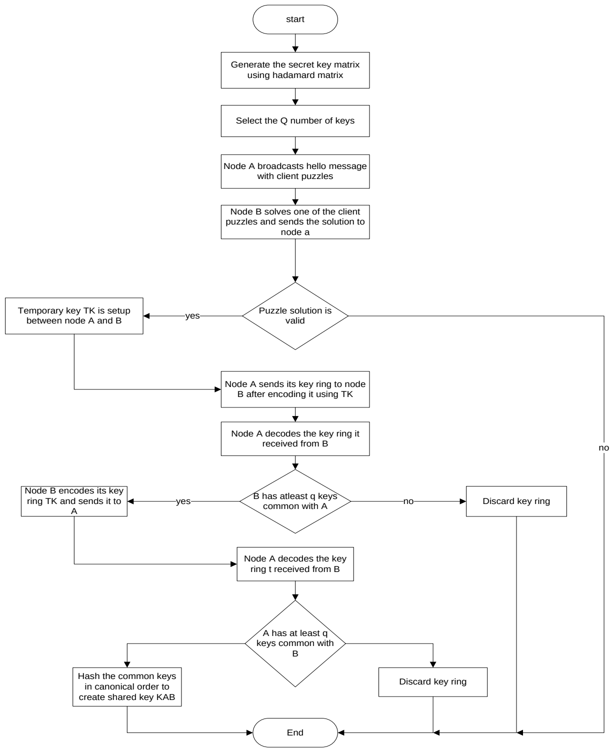
We make the following general observation. First, increasing the required key overlapping q reduces the required key pool size to achieve a certain local connectivity. When the local connectivity is too low, the required key overlapping can be decreased to increase the local connectivity. Second, when the local connectivity is too low, the key ring size can be increased to increase the local connectivity.

Now let us consider the impact of the memory usage M when the key pool size is fixed such that the local connectivity becomes 33% when the memory usage M is 200. From this, we have the following findings. First, the increment of memory usage increases the local connectivity. Increasing the memory usage allows a sensor node to increase the key ring size. Accordingly, a node can store a larger number of secrets. Thus, a node has a higher chance of having the shared secrets with neighboring nodes. Second, the models$\mathrm{M}_{q,m,\lambda}$, where 0 < m < M and 0 < λ < M, with greater q achieve higher local connectivity than the same models with smaller q after the crossing point. In addition, the models$\mathrm{M}_{q,m,\lambda}$, where 0 < m < M and 0 < λ < M, with smaller m, thus greater λ, achieve higher local connectivity than same models with greater m and smaller λ after the crossing point.

D. Scalability Analysis

As the metric to evaluate the scalability of the MBBQ scheme, we use the supportable network size. The supportable network size is the maximum number of sensor nodes in a network that a scheme can support. Since the sensor network is a large-scale network with hundreds or thousands of nodes, the supportable network size is an important factor to ensure that the deployed model is capable of supporting the maximum network size. The supportable network size can also be calculated from different concerns. When we limit the number of allowed duplicated secrets t for security reasons, increasing the memory usage does not always increase the supportable network size. We observe that the corresponding supportable network sizes are almost the same level when q are the same in spite of the different key pool sizes except the models$\mathrm{M}_{q,M,0}$. Second, the models with the smaller required key overlapping q is more scalable than the models with greater required key overlapping q when the local connectivity is fixed. Furthermore, the model $\mathrm{M}_{1,M,0}$ (which corresponds to the basic scheme) is the most scalable followed by the model$\mathrm{M}_{q,M,0}$, where q > 1 (which corresponds to the QC scheme), and the model$\mathrm{M}_{q,M,\lambda}$, where 0 < m < M and 0 < λ<M (which corresponds to the BB scheme). Then, the model$\mathrm{M}_{q,M,\lambda}$, where q > 1, 0 < m < M, and 0 < λ < M, is the least scalable. Furthermore, the model$\mathrm{M}_{q,M,\lambda}$, where 0 < m < M and 0 < λ < M, is more scalable than the model$\mathrm{M}_{q,M,\lambda}$, with q > 1. This is because the corresponding supportable network size are almost the same level when q are the same level, and because the model with smaller required key overlapping q is more scalable than the models with greater required key over lapping q. Thus, the model$\mathrm{M}_{1,M,\lambda}$, where 0 < m < M and 0 < λ <M, and the model$\mathrm{M}_{1,M,0}$, have almost the same supportable network size. However, the compared model is$\mathrm{M}_{q,M,0}$, with q > 1. Consequently, the model$\mathrm{M}_{q,M,0}$, with q > 1 is less scalable than the model$\mathrm{M}_{1,M,0}$, Thus, the model$\mathrm{M}_{q,M,\lambda}$, where 0 < m < M and 0 < λ < M, is more scalable than the model $\mathrm{M}_{q,M,0}$, with q > 1.

E. Flow chart of the modified bloom's q-composite key distribution scheme [11]



V. RESULTS AND DISCUSSIONS:

A. *Memory evaluation*

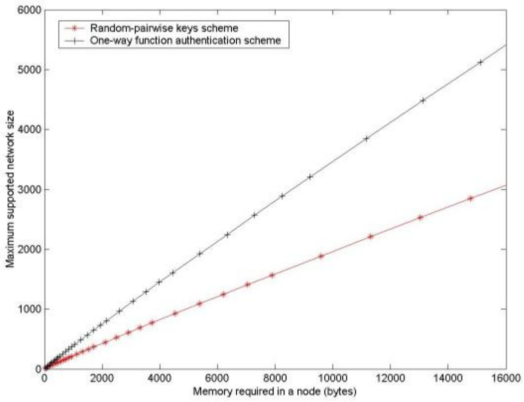


Fig. 1: Plot of memory vs. the network size

B. *Scalability evaluation*

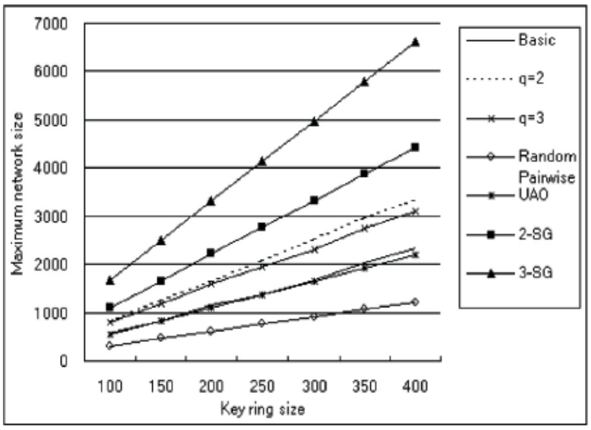


Fig. 2: Key ring size vs. the network size

C. *Resilience of the network*

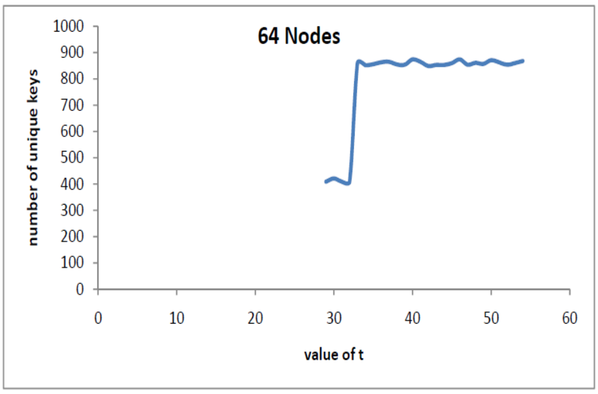


Fig. 3: Number of unique keys for a network of 64 nodes

VI. CONCLUSION

As it is evident from the above theory modified bloom's q-composite distribution scheme offers a very good solution for memory, overhead and scalability issues when compared to the other distribution schemes and provides very good resilience even for the large networks.

VII. REFERENCES

[1] "Po-Jen Chuang*, Tun-Hao Chao and Bo-Yi Li", Scalable Grouping Random Key Predistribution in Large Scale Wireless Sensor Networks.

[2] "Alan price, Kristie kosaka, samir chatterjee", A Secure Key Management Scheme for Sensor Networks.

[3] "Takashi Ito, Hidenori Ohta, Nori Matsuda, and Takeshi Yoneda", A Key Pre-Distribution Scheme for Secure Sensor Networks Using Probability Density Function of Node Deployment.

[4] "Seyit A. C¸Amtepe and B¨ Ulent Yener", Key Distribution Mechanisms for Wireless Sensor Networks: a Survey.

[5] "Haowen Chan, Adrian Perrig, Dawn Song", Random Key Predistribution Schemes for Sensor Networks.

[6] "Manoj R, N.Dhinakaran", A Survey of Key Predistribution Schemes for Key Management in Wireless Sensor Networks.

[7] "Bo-Cheng Charles Lai, David D. Hwang, Sungha Pete Kim, Ingrid Verbauwhede", Reducing Radio Energy Consumption of Key Management Protocols for Wireless Sensor Networks.

[8] "D. Manivannan and P. Neelamegam", WSN: Key Issues in Key Management Schemes-A Review.

[9] "Huirong Fu, Satoshi Kawamura, Chengzhi Li", Blom-based Q-composite: A Generalized Framework of Random Key Pre-distribution Schemes for Wireless Sensor Networks.

[10] "Rohith Singi Reddy", Key Mangament In Wireless Sensor Networks Using A Modified Blom Scheme.

[11] "Jeegar brahmakshatriya", Implementation of Pair-Wise Key Pre-distribution Schemes on Wireless Sensor Devices.

[12] "D. W. Carman, P. S. Kruss, and B. J. Matt", "Constraints and approaches for distributed sensor network security," NAI Labs Technical Report #00-010, Sept. 1, 2000.

[13] "Menezes, P. C. van Oorschot, and S. A. Varstone", Handbook of Applied Cryptography, CRC Press, 1997.

[14] "R. Merkle". Secure communication over insecure channels. Communications of the ACM, 21(4):294–299, 1978.

[15] "R. Blom", "An optimal class of symmetric key generation systems," in Proc. Of EUROCRYPT '84, pages 335-338, 1985.

[16] "W. H. Press, B.P. Flannery, S. A. Teukolsky and W. T. Vetterling", "Vander monde Matrices and Toeplitz Matrices." 2.8 in Numerical Recipes in FORTRAN.