Using a Random Perturbation-based Scheme for Establishing Authentic Associations in Wireless Mesh Network

Walaa El-salhy
Department of Mathematics
Faculty of Science
Ain Shams University

Fayed F. M. Ghaleb
Department of Mathematics
Faculty of Science
Ain Shams University

Ahmed M. Khedr
Department of Mathematics
Faculty of Science
Zagazig University

ABSTRACT
Establishing an Authentic Association among entities in Wireless Mesh Networks WMN is a nontrivial problem and the architecture of WMN is relatively new and lacks a robust secure scheme .In this paper, we develop a Polynomial Based scheme which provides pair-wise connectivity, low communication, marginal storage overhead and high scalability while making on the fly Authentic Association feasible by using random perturbation based (RPB) scheme. New Proposed scheme is not only observed to be resilient against both traffic analysis and node capture attacks but also it is more secure , only requires a small storage space and has a little communication overhead.

General Terms
Security, wireless Mesh Network, Authentic Association.

Keywords
Authentication, scalability, Mesh Routers, Bi-variate polynomial, Mobile client, Hash function.

1. INTRODUCTION
Wireless mesh networks (WMNs) is a network consisting of two types of nodes, Mesh Routers (MRs) and Mesh Clients (MCs) where MRs maintain arbitrary ad-hoc connectivity automatically between them[1] . As known in [2] WMNs have become increasingly important due to their deployment flexibility while keeping the cost at a fairly low level. In this scenario, the network traffic has to be propagated for authentication from the MCs to the AAA Server (authentication, authorization and accounting) through MRs in a multi-hop fashion. we need to enhance a robust scheme at [2] that could protect the integrity of propagating message in the WMN. Such networks suffer from threats like false handoffs and fake registrations and the network is also prone to attacks like Traffic analysis and Node Capture attack.

This makes requirement of having a robust WMN scheme more desirable with a little communication overhead. A polynomial based scheme was employed to be both secure and consistent and to keep the communication alive. We need to enhance it and achieve secure communication by generating keys in a real-time manner. In our proposed scheme, we use bivariate polynomial and use perturbed polynomial which make the system harder to break .by them we enabled to compute common secure key by independent calculation on each individual entity , this scheme is more secure, highly scalable and supporting very large network .
The rest of the paper is organized as follows: Related work is explained in Section 2. Section 3 provides details of our scheme proposal. Section 4 gives Performance Evaluation and Security Analysis and Section 5covers our concluding remarks.

2. RELATED WORK
We are already aware of schemes in a mobile ad hoc network (MANET) and wireless sensor network (WSN) which enable secure communication among entities. Various key distribution mechanisms have been proposed to establish secure links between sensor nodes in wireless sensor networks which leads to make the system efficient and robust such as [3],[4],[5],[6],[7],[8],[9]and[10]. in [11] also introduce an enhanced approach to his previous work. Based on [11] Amit Gaur et al.[2]proposed scheme, using bivariate polynomials that capable of computing a common seed by independent calculation on each individual entity.

3. PROPOSED ENHANCED APPROACH FOR POLYNOMIAL BASED SCHEME
In this paper we consider The WMNs. Based on the previous work in [2] we enhanced a distributed scheme for establishing secure connection between pair of entities on the fly. The following secrets are specified and computed by the central server to be preloaded to the entities:

1. Common key (K) which would be used for encrypting all the communication between entities prior to establishment of a secured symmetric key.
2. Constructs a set (S) of legitimate IDs .
3. A set of Bivariate Polynomials Functions $F_{i,j,k}(x,y)$ $(where \quad 0 \le i < h, \quad 0 \le j < m, 0 \le k < m)$ selected randomly from a three-dimensional matrix of polynomials and computes them with ID of every node (a univariate polynomial of $F_{i,j,k}(x,y)$ for the node ex. $F_{i,j,k}(u,y)$ for node u .
4. Constructs a set (Φ) of perturbation polynomials.
5. Generates perturbed polynomials then Preloads every node with it ex. Assigns node u with $g_{(i,j,k)_u}(y)$

6. Hashing function $H_f(\ )$ for hashing secrets for more security.

For The Procedure of Our Enhanced Approach there are four phases used in our approach, namely, Bivariate Polynomial specification phase, Generation and Predistribution of Perturbed Polynomials phase, Authenticated association phase and Secure path generation phase.

3.1 Bi-variate Polynomial specification phase

In this phase a polynomials are randomly selected from a matrix of polynomials and given to the entity by using Yi-Cheng's scheme [10] [11]of distribution a bivariate polynomial $F_{i,j,k}(x,y)$ of degree p which is defined as :

$$F_{i,j,k}(x,y)=\sum_{d,e=0}^{p} a_{d,e}x^d y^e \qquad (1)$$

where the coefficients $a_{d,e}$ are randomly chosen over a Finite Field $Gf(q)$ where q is a sufficiently large prime number and i,j,k are the indices for the position of polynomial in a three-dimensional space as mentioned in [2]. Also, the polynomial satisfies the following property:

$$F_{i,j,k}(x,y)=F_{i,j,k}(y,x) \qquad (2)$$

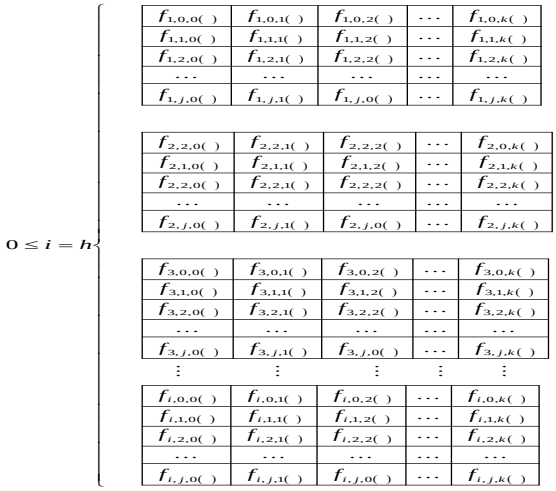


Fig.1 Illustrating Distribution of Bivariate Polynomial functions in a Three Dimensional Matrix.

As mentioned in [2] , Once a large number of polynomials have been generated, a limited number of polynomials are randomly selected and placed in a three-dimensional matrix. This matrix can be viewed as i number of two dimensional square matrices each with degree p, where i vary between 0 and h and p varies between 0 and m. Hence, the total number of selected polynomials

$$TNP = h \times m \times m \qquad (3)$$

We can see from Figure 1, a bivariate polynomial $f_{i,j,k}(x,y)$ is at the position i,j,k in the polynomial matrix.

From h total matrices RS are randomly selected such that:

$$RS \geq \left\lceil \frac{(h+1)}{2} \right\rceil \qquad (4)$$

Where $\lceil\ \rceil$ is Ceiling Function which returns the smallest integer above a non-integer number (For example, $\lceil 3.2 \rceil = 4$, $\lceil 5 \rceil = 5$).This condition makes sure that two sets share at least one common matrix between them.

Then the authority chooses a unique ID (e.g., u) to each node before deploying it into the network .The authority also computes polynomials for every node. In particular, for the node of $ID\ u$, the polynomials will be in the form:

$$f_{i,j,k}(u,y)=\sum_{e=0}^{p} B_{u,e}y^e \qquad (5)$$

where

$$B_{u,e}=\sum_{d=0}^{p} a_{d,e}u^d \qquad (6)$$

3.2 Generation and Predistribution of Perturbed Polynomials phase

we first list some notations based on [8] and introduce the concept called perturbation polynomials.

3.2.1 Notations

Following is a list of notations used in presenting the basic idea of a random perturbation-based (RPB):

- q, l: q is a prime number $(q>2)$, and l is the minimal integer such that $2^l > q$. Thus, every element in field $Gf(q)$ can be represented by l bits.
- S: a set of legitimate IDs for sensor nodes. In this paper, we let $S \subset \{0,\cdots,q-1\}$.
- r: a positive integer such that $2^r < q$
- Φ: a set of perturbation polynomials (to be defined in Section 3.2.2).
- $F_{i,j,k}(x,y)$ a symmetric polynomial, in which the degree of x and y are both p
- $g_u(y)(u \in S)$: a p -degree univariate polynomial that is preloaded to node (with $ID\ u$) before it is deployed.

3.2.2 Perturbation Polynomials

In RPB, [8] we introduce the concept of perturbation polynomial, which is defined as follows:

Given a finite field $Gf(q)$, a positive integer r (2r <q), and a set of node IDs S ($S \subset \{0,\cdots,q-1\}$), a polynomial set Φ is a set of perturbation polynomials regarding r and S if any polynomial φ(.) ∈ Φ has the following limited infection property:

$$\forall u \in S, \varphi(u) \in \{\ 0,\cdots,2^r-1\ \}.$$

The above definition ensures that the value of a perturbation polynomial will not be greater than 2^r-1; i.e., it has at most r bits.

3.2.3 Basic idea of this phase

Using the RPB scheme[8] to generate a perturbed share, which is the sum of the original share and a perturbation polynomial with the limited infection property. The authority picks system parameter r and constructs a set of perturbation polynomials Φ regarding S and r. The procedure for constructing S and Φ are detailed in Section 4.4 at [8], then the authority randomly picks a polynomial φ(y) from Φ and generates $g_{(i,j,k)}(y)$ to every node for example the authority preloads node u with

$g_{(i,j,k)_u}(y) = f_{i,j,k}(u,y) + \varphi_{(i,j,k)_u}(y)$. Note that node u is only given the coefficients of $g_{(i,j,k)_u}(y)$ so it cannot find out the coefficients of either $f_{i,j,k}(u,y)$ or φ. Adding perturbation polynomials makes it harder to break the symmetric polynomials. This is because the adversary cannot obtain the original shares of polynomial $f_{i,j,k}(x,y)$.
A row and column are randomly selected from each matrix, and all the perturbed functions present in that particular row and column are distributed to the entity.

$g_{i,0,0}(\,)$	$g_{i,0,1}(\,)$	$g_{i,0,2}(\,)$	$g_{i,0,3}(\,)$	$g_{i,0,4}(\,)$
$g_{i,1,0}(\,)$	$g_{i,1,1}(\,)$	$g_{i,1,2}(\,)$	$g_{i,1,3}(\,)$	$g_{i,1,4}(\,)$
$g_{i,2,0}(\,)$	$g_{i,2,1}(\,)$	$g_{i,2,2}(\,)$	$g_{i,2,3}(\,)$	$g_{i,2,4}(\,)$
$g_{i,3,0}(\,)$	$g_{i,3,1}(\,)$	$g_{i,3,2}(\,)$	$g_{i,3,3}(\,)$	$g_{i,3,4}(\,)$
$g_{i,4,0}(\,)$	$g_{i,4,1}(\,)$	$g_{i,4,2}(\,)$	$g_{i,4,3}(\,)$	$g_{i,4,4}(\,)$

Fig.2 Selection of perturbed Polynomials from a single matrix.

The total number of perturbed functions $(TpFn)$ distributed to each entity is given by: $TpFn = Rs \times ((m-1)+m)$

$= Rs \times (2 \times m - 1)$ (8)

In this phase the perturbed polynomials have been prepared for the distribution. In Figure 3 there is an i^{th} matrix that is common for two entities. For the first entity u 3^{rd} row and 4^{th} column has been selected and for second entity v 4^{th} row and 2^{nd} column has been selected. We can clearly observe that between two matrices, functions with index $(i,2,1)$ and $(i,3,3)$ are in common or in other words any two entities can have at least two functions that will be shared with each other. This makes sure that any two entities can establish AA between the two once they contain functions from the central server.

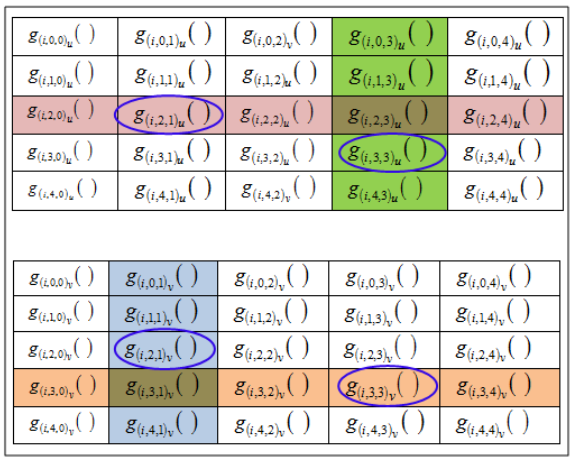


Fig.3 Illustrating Common Functions for u & v

3.3 Authenticated Association (Keys Establishment) phase

We now show how any two (entities) nodes (say u and v). Assume that the two nodes discovered common one matrix of perturbed functions and discovered two related common perturbed functions there are three steps:

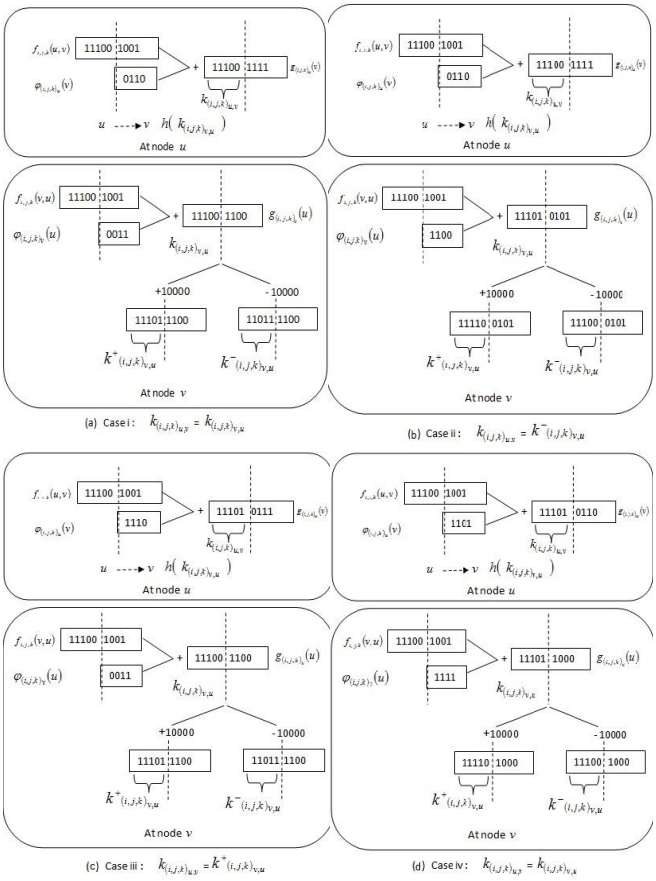
- Step 1: Node u evaluates the functions e.g. $g_{(i,j,k)_u}(y)$ at $u = v$, and represents the evaluation result in l binary bits.
- Step 2: It uses the most significant $l-r$ bits of common perturbed functions $g_{(i,j,k)_u}(y)$ denoted as $k_{(i,j,k)_{u,v}}$, as the key.
  - The number of the keys is $2 \times m - 1$
- Step 3: Node u sends $h(k_{(i,j,k)_{u,v}})$ to node v, where $h(.)$ is a secure hash function such that any node overhearing cannot derive $k_{(i,j,k)_{u,v}}$.

After receiving $(2 \times m - 1)$ $h(k_{(i,j,k)_{u,v}})$ node v goes through the following steps to construct three keys denoted as $k_{(i,j,k)_{u,v}}$, $k^{-}{}_{(i,j,k)_{u,v}}$ and $k^{+}{}_{(i,j,k)_{u,v}}$:

- Step 1: node v evaluates $g_{(i,j,k)_v}(u)$, $g_{(i,j,k)_v}(u) + 2^r$ and $g_{(i,j,k)_v}(u) - 2^r$.
- Step 2: Each evaluation result is represented in l binary bits and its most significant $l-r$ bits is computed and assigned to the $k_{(i,j,k)_{v,u}}$ $k^{+}{}_{(i,j,k)_{v,u}}$ or $k^{-}{}_{(i,j,k)_{v,u}}$ respectively.

These steps will be repeated with all related common perturbed functions.
The maximum number of related common perturbed functions = m and the minimum number = 2 ,As stated in Theorem1 at [8], one of $k_{(i,j,k)_{v,u}}$ $k^{-}{}_{(i,j,k)_{v,u}}$ and $k^{+}{}_{(i,j,k)_{v,u}}$ that are computed by node v must be the same as $k_{(i,j,k)_{u,v}}$ that is sent by node u .Examples:
Some examples are shown in Fig. 3 to illustrate the property stated in Theorem 1. In these examples, the prime number is 457 (i.e., (111001001)2), l is 9, and r is 4 that is $f_{i,j,k}(u,v)$, $f_{i,j,k}(v,u)$, $g_{(i,j,k)_v}(u)$ and $g_{(i,j,k)_u}(v)$ can be represented by l = 9 binary bits ; $\varphi_{(i,j,k)_u}(v)$, $\varphi_{(i,j,k)_v}(u)$ can be represented by r = 4 binary bits , for the examples in fig. 3, $k_{(i,j,k)_{v,u}}$ $k^{-}{}_{(i,j,k)_{v,u}}$ and $k^{+}{}_{(i,j,k)_{v,u}}$ are the most

significant $l-r$ bits of $g_{(i,j,k)_u}(v)$, $g_{(i,j,k)_v}(u)$, $g_{(i,j,k)_v}(u)-2^r$ and $g_{(i,j,k)_v}(u)+2^r$.



(a) Case i: $k_{(i,j,k)_{u,v}} = k_{(i,j,k)_{v,u}}$ (b) Case ii: $k_{(i,j,k)_{u,v}} = k^{-}{}_{(i,j,k)_{v,u}}$

(c) Case iii: $k_{(i,j,k)_{u,v}} = k^{+}{}_{(i,j,k)_{v,u}}$ (d) Case iv: $k_{(i,j,k)_{u,v}} = k_{(i,j,k)_{v,u}}$

Fig.4 Examples of Using RPB to Generate Pairwise Keys .

We can see four cases regarding the matching between these keys:

- ✓ Case i: As shown in fig.4 (a) if neither $f_{i,j,k}(u,v)$ + $\varphi_{(i,j,k)_u}(v)$ nor $f_{i,j,k}(v,u)$ + $\varphi_{(i,j,k)_v}(u)$ generates a carry from bit $r-1$ to bit r we have $k_{(i,j,k)_{u,v}} = k_{(i,j,k)_{v,u}}$.
- ✓ Case ii: As shown in fig.4 (b), if only $f_{i,j,k}(u,v)$ + $\varphi_{(i,j,k)_u}(v)$ generates a carry from bit $r-1$ to bit r we have $k_{(i,j,k)_{u,v}} = k^{-}{}_{(i,j,k)_{v,u}}$.
- ✓ Case iii: As shown in fig.4 (c), if only $f_{i,j,k}(u,v)$ + $\varphi_{(i,j,k)_u}(v)$ generates a carry from bit $r-1$ to bit r we have $k_{(i,j,k)_{u,v}} = k^{+}{}_{(i,j,k)_{v,u}}$.
- ✓ Case iv: As shown in fig.4 (d) if both $f_{i,j,k}(u,v)$ + $\varphi_{(i,j,k)_u}(v)$ and $f_{i,j,k}(v,u)$ + $\varphi_{(i,j,k)_v}(u)$ generates a carry from bit $r-1$ to bit r we have $k_{(i,j,k)_{u,v}} = k_{(i,j,k)_{v,u}}$.

Based on Theorem 1 at [8], node v can find out $k_{(i,j,k)_{u,v}}$ by computing $h(k_{(i,j,k)_{v,u}})$, $h(k^{+}{}_{(i,j,k)_{v,u}})$ and $h(k^{-}{}_{(i,j,k)_{v,u}})$ and comparing them with $h(k_{(i,j,k)_{u,v}})$ sent by node u as follows:

- $h(k_{(i,j,k)_{u,v}}) = h(k_{(i,j,k)_{v,u}}) \Rightarrow$
  $k_{(i,j,k)_{u,v}} = k_{(i,j,k)_{v,u}}$ (9)
- $h(k_{(i,j,k)_{u,v}}) = h(k^{+}{}_{(i,j,k)_{v,u}}) \Rightarrow$
- $k_{(i,j,k)_{u,v}} = k^{-}{}_{(i,j,k)_{v,u}}$ (10)
- $h(k_{(i,j,k)_{u,v}}) = h(k^{-}{}_{(i,j,k)_{v,u}}) \Rightarrow$
  $k_{(i,j,k)_{u,v}} = k^{-}{}_{(i,j,k)_{v,u}}$ (11).

We indicate to a common key with $sk_{(i,j,k)w}(w=1,2,\ldots,2xm-1)$

Comparing our scheme $(EARPB)$ with (PBS) scheme [2] we can see that applying the perturbation method reduces the size of the generated pairwise key because some bits of the polynomial evaluation results are cut off .

3.3.1 Pairwise key Generation

Once the secret keys have been distributed, the entities can start the operation of their network. To communicate with other entities, they need to establish AA. So, in a handshake message, the following information is exchanged by encrypting a common key K and later decrypted at receiver by the same key K as all nodes share this master key, indices of their functions. For example, we have two entities u and v and let's assume, $k_{(1,2,3)_{u,v}}$, $k_{(2,3,3)_{v,u}}$ denoted by $sk_{(1,2,3)1}$, $sk_{(2,3,3)2}$ are the two keys they share with each other which are obtained after exchanging their indices. Finally to have AA one-way hashing function $H_f(\,)$ is applied to all common keys to generate a unique secure symmetric key $(ussk)$ for communication between two entities.

$ussk = H_f(sk_{(1,2,3)1} \oplus sk_{(2,3,3)2})$

In general

$$ussk = H_f(sk_{(i,j,k)1} \oplus sk_{(i,j,k)2} \oplus sk_{(i,j,k)3} \oplus \ldots\ldots\ldots) \quad (12)$$

3.4 Secure Path Generation phase

We now consider the establishment of a secured authenticated path from individual entity to AAA Server in a WMN. In Figure 5, MR_i is for i^{th} Mesh Router, MC_i is for i^{th} Mesh Client, IGW_i is for i^{th} Internet Gateway, AA_i corresponds to i^{th} Authentic Association between two entities and $(AA_i <-> AA_i)$ corresponds to pair-wise association between the two entities. Consider a case when network is in operation and AA_1 and AA_6 (Authentic Association between

IGW_1 and AAA and Authentic Association between MR1 and IGW1 respectively) are already established, as previous stage ensures authenticated association between any two entities that are part of the network.

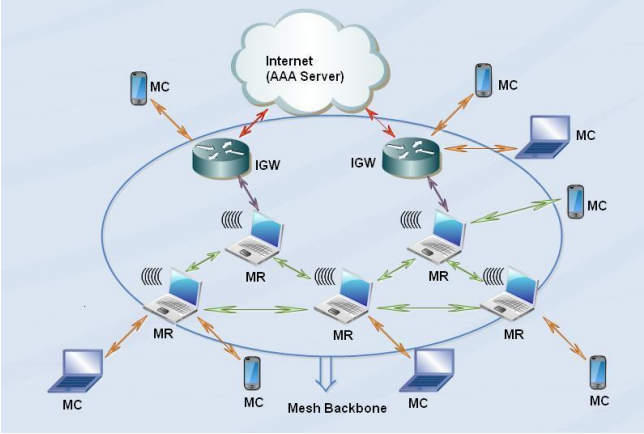


Fig.5 Example AAA in a WMN

4. Performance Evaluation and Security Analysis

In this section, we analyze our approach .We observe that our enhanced approach guarantees a shared key between any two entities as there exist at least two common perturbed polynomials so it can achieve the network connectivit

4.1 Resilience against Node Capture Attack and Traffic analysis attack

Node Capture attack is a serious threat in these WMNs where captured node can reveal the secrets of non-captured nodes and increase their vulnerability by attacks. The problem associated with [3],[4] is that different pair of nodes might use same key for securing communication in the network which means compromised nodes can reveal information regarding non-captured nodes. Whereas in our scheme like [11],[2] every pair of entities generates a unique key for communication by independent calculations, capture of one entity does not affect the security between non-captured nodes just like in [11]. Also, at no point, any information is exchanged regarding the functions distributed to the entities and the only information that is exchanged is Ids of the entities and indices of functions which are transmitted in encrypted form in a handshake message. These make our scheme quite secure against traffic analysis attack. Another point to be noted is that the use of one sided hash function to generate key by hashing all the seeds makes this scheme even more secure

5. Concluding Remarks:

For WMNs where mobility of MCs poses a considerable problem in providing secure associations, EARPB provides an effective solution by carrying out independent calculations to agree on a common key along with having low communication and storage overheads. Our proposed scheme is highly scalable by supporting a network of large number of entities, despite storing very few functions on each individual entity. It also provides security against attacks such as Traffic analysis attack and Node capture attack in an effective manner. Since we can have communication between any pair of entities irrespective of their position and mobility.

6. ACKNOWLEDGMENTS

Walaa El-salhy ackowledges help and support of Dr.Fayed F. M. Ghaleb &Dr.Ahmed Khedr .Special thanks to faculty of science Ain Shams University .

7. REFERENCES

[1] Nandiraju, N., et al., Wireless Mesh Networks: Current Challenges and Future Directions of Web-In-The-Sky. Wireless Commun., 2007. 14(4): p. 79-89.

[2] Gaur, A., et al., Polynomial based scheme (PBS) for establishing Authentic Associations in Wireless Mesh Networks. J. Parallel Distrib. Comput., 2010. 70(4): p. 338-343.

[3] Eschenauer, L. and V.D. Gligor, A key-management scheme for distributed sensor networks, in Proceedings of the 9th ACM conference on Computer and communications security2002, ACM: Washington, DC, USA. p. 41-47.

[4] Chan, H., A. Perrig, and D. Song, Random Key Predistribution Schemes for Sensor Networks. IEEE SYMPOSIUM ON SECURITY AND PRIVACY, 2003: p. 197-215.

[5] Blom, R., An optimal class of symmetric key generation systems, in Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques1985, Springer-Verlag New York, Inc.: Paris, France. p. 335-338.

[6] Du, W., et al., A pairwise key predistribution scheme for wireless sensor networks. ACM Trans. Inf. Syst. Secur., 2005. 8(2): p. 228-258.

[7] 7. Blundo, C., et al., Perfectly-Secure Key Distribution for Dynamic Conferences, in Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology1993, Springer-Verlag. p. 471-486.

[8] Zhang, W., et al., A random perturbation-based scheme for pairwise key establishment in sensor networks, in Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing2007, ACM: Montreal, Quebec, Canada. p. 90-99.

[9] Liu, D.g. and P. Ning, Location-Based Pairwise Key Establishments for Relatively Static Sensor Networks. ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03), October 31, 2003 George W.Johnson Center at George Mason University, Fairfax, VA, USA., 2003.

[10] Cheng, Y. and D.P. Agrawal. Efficient pairwise key establishment and management in static wireless sensor networks. in Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on. 2005.

[11] Cheng, Y., et al., Enhanced Approach for Random Key Pre-Distribution in Wireless Sensor Networks in international Conference on Communication, Networking and Information Technology , 2008.