An Efficient Group Key Distribution Security Scheme in Wireless Sensor Networks

S.Prema[*], S.Nagaraj [**]

[*] II-M.E-CSE, Selvam College of Technology, Namakkal.
[**] Assistant Professor-CSE, Selvam College of Technology, Namakkal.

Abstract- An Unattended Wireless Sensor Network (UWSN) collects the sensing data by using mobile sinks (MSs). It differs from the traditional multi- hop wireless sensor networks in which unbalanced traffic makes the sensors close to the base station deplete their power earlier than others. An UWSN can save the battery power and prolong the network lifetime. Unfortunately, MSs would be given too much privilege when acting as the collecting base station, which will cause security concern if Replicated. Besides, UWSNs are usually deployed in unreachable and hostile environments, where sensors can be easily Replicated. Thus, their security issues should be carefully addressed to deal with node compromise. In this paper, we present a novel key management scheme to secure UWSNs. We employ the Blundo symmetric polynomial mechanism to guard against the newly Replicated nodes in a period while utilizing the periodic key updating based on the reverse hash chain to block the Replicated nodes and revoke the Replicated MSs if failing the authentication. We show that our scheme is robust against node Replicated attacks and carry out comparison analysis on the intrusion-tolerance ratio, communication and computing overhead.

Index Terms- wireless sensor networks; security; key management; mobile sink.

I. Introduction

An Unattended Wireless Sensor Network (UWSN) is a kind of hybrid wireless sensor network consisting of mobile sinks (MSs) and static sensing nodes [1], which periodically collects sensed data from static sensors by using the mobile sinks. In such a UWSN, the static nodes are composed of a large number of low-power sensors with limited communication, computing and storage capacity, and are usually deployed in uninhabited environments, such as military sensing, invasion detection, commercial security, and etc. While in the traditional wireless sensor networks, a large number of sensors are required to send the collected data to a designated base station by multi-hop communications, resulting in excessive routing burden to the nodes close to the base station. In UWSNs, on the other hand, it is the MSs that bear most of the communication overheads by moving around and collecting data. The deployment of mobile nodes could reduce the power cost in the multi-hop data transmissions in static WSNs and balance the network energy consumption, and hence prolong the network lifetime. Unfortunately, the introduction of the mobile nodes also brings some security concerns, which will be addressed in this paper.

In some earlier studies, MSs are assumed to have the same capability as the base station, and the efficiency of data collection is the major design consideration [2, 3]. However, security is an unavoidable issue in UWSNs because of the unattended nature and hostile environments. To secure a UWSN, the key management has been investigated lately. It is observed that the key management schemes for conventional WSNs cannot be directly applied in UWSNs because of the participation of MSs. Meanwhile, the battery power in a sensor is more limited than that of a node in a mobile ad hoc network (MANET), thus the key management scheme designed for MANETs is also not effective in UWSNs [4]. Moreover, some key management schemes [5] for UWSN often provide MSs with excessive privileges, even without considering the pinpoint attacks on MS.

High privileges given to MS may cause the security problems. Song, et al. [6] investigated the revocation problem of MS in their key management scheme, which synthesized Blundo symmetric polynomials [7] and Merkle Tree [8]. However, the routing for sensing data collection in their scheme was set up for a single path and a single task, and the MS identity was bond to such a unique path and task. Combined with symmetric polynomial key pool scheme [9] and random key pre-deployment scheme [10], Rasheed et al. [5] proposed a new key management scheme for a UWSN, in which the employment of symmetric polynomial key pools can enable the bonding relation of MS with the moving routing and data collection task.

However, it assigns too important role to the MS. If the MS fails or is Replicated, 90% of the pre-distributed random keys will be useless, which results in very low probability of establishing security channels between nodes, and the resulting scheme was degenerated to basic Blundo scheme. Moreover, they have not taken the security of MS into account.

To address the security over UWSNs, in this paper, we present a novel Modernize key management scheme, in which the symmetric polynomial is used to generate shared keys between nodes and enables the threshold security feature of the network, while the reverse hash chain is utilized for key update or revocation to effectively restrict the privileges of MSs and at the same time for the identity generation of the newly-joined MSs.

II. NETWORK MODEL

A UWSN is a WSN with MSs used to help static BS collect data. Here, we use the general WSN network structure based on group deployment. For simplicity, we study the key management scheme under the network model with one MS.

A. Network Structure

We assume that the common nodes (Nodes) and the Cluster heads (CHs) pre-loaded with the same symmetry polynomial are distributed based on group deployment [11]. According to the principle of group deployment, it is easy for the common nodes to find the nearest sink as their cluster head in two-dimensional space.

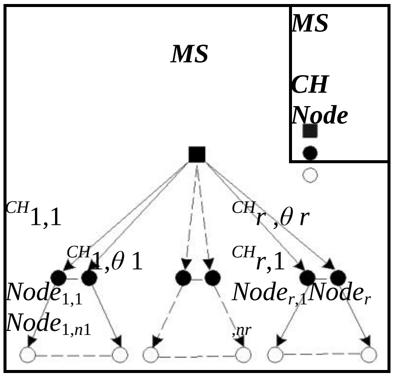

Figure 1. Three-layer UWSN network structure

When the MS joins the WSN, the network forms a three-layer structure, consisting of the MS, CHs and Nodes as shown in Figure 1. Here we assume that MS and the CHs have strong computing power, communication capability and high storage. In this cluster-based UWSN, CHs shall wait for the arrival of the MS after collecting the sensed data.

B. Adversarial Model

The network is assumed to be unattended, with the MS periodically collecting the sensed data and at the same time managing and maintaining the network, which includes security management such as key management, the revocation of Replicated nodes, etc. In the network, the adversary can launch external attack by monitoring the transmitted information or even internal attack by using physical capture to get all the secret information of arbitrary nodes.

We do not assume the MS is equipped with costly hardware security protection, which means that the adversary can obtain all confidential information stored in the MS if captured, but is costly for the adversary to perform physical capture attack. The adversary cannot capture and compromise more than t static nodes in a certain time period T_1 or capture and replicate the MS in time period $T_2(T_2>T_1)$.

III. MODERNIZE INTRUSION-TOLERANT KEY MANAGEMENT

In our Modernize intrusion-tolerant key management (ITKM) scheme, key distribution consists of three phases, namely, key material pre-distribution, shared-key generation, and key update as well as MS revocation.

TABLE I. NOTATION

notation	Meaning
A-B	Node A and B
$f_{A\text{-}B}(ID,y)$	The shared binary symmetric polynomial between A and B
t	The degree of Binary symmetrical polynomial
S_i	The (i+1)th hash value of the reverse hash chain
a	The number of symmetric polynomials pre-loaded in MS
b	The number of Sinks in each group

A. Key Material Pre-distribution

The BS chooses a random symmetric bivariate polynomial $f(x,y)$ of degree t with coefficients over a finite field F_q, where q is a prime number large enough to accommodate a symmetric key:

$$f(x,y) \square \sum_{0\le m,n\le t} a_{mn} x^m y^n \quad (a_{mn} \square a_{nm})$$

which is also called Blundo symmetric polynomial. BS pre-loads the same Blundo symmetric polynomial for MS and CHs while loading CHs and Nodes with different one referred to as $f_{MS\text{-}Node}(ID,y)$ and $f_{CH\text{-}Node}(ID,y)$. BS randomly selects an initial value S_n and calculates the reverse hash sequence $\{S_i\}_{i\square 0}^{n}$:

$$S_{i-1} \square h(S_i)(1\le i\le n) \quad \square\square\square$$

In which, $h()$ is a collision-free one-way hash function and $\{S_i\}_{i\square 0}^{n}$ is called the reverse hash chain [12] who is generated through the formula (2).

Similarly, BS chooses a random value r_{MS} and then uses formula (3) to calculate the reverse hash sequence with $MS_n = r_{MS}$.

$$MS_{i-1} \square h(MS_i)(1\le i\le n) \quad \square\square\square$$

At the beginning of the network deployment, BS pre-loads MS and CHs with Blundo symmetric polynomial $f_{MS\text{-}CH}(ID,y)$ and hash value S_0, in which the ID in $f_{MS\text{-}Node}(ID,y)$ represents the node identity and S_0 represents the first value of the reverse hash chain. Meanwhile, BS pre-loads the CHs and Nodes with Blundo symmetric polynomial $f_{CH\text{-}Node}(ID,y)$ and hash S_0.

B. Key Agreement Protocol for Peer Nodes (KAP)

Since the network consists of MS, CHs and Nodes, there are two types of key agreement: one between homogeneous nodes (between the CHs or between Nodes) and one between heterogeneous nodes (between the MS and CHs or between CHs and Nodes) . According to the network model, the MS does not directly communicate with Nodes, and so we do not need to consider key agreement between the MS and Nodes.

Let us first consider the key agreement between peer nodes *u* and *v* which are pre-loaded with Blundo symmetric polynomial $f(u,y)$ and $f(v,y\ )$ respectively. Let the current hash values stored in *u* and *v* are S_i and S_j, respectively. If there is an attack or network failure, some nodes may fail to update their hash values, which will result in $S_i \neq S_j$. The node with the latest hash value will identify the other communication party subjected to malicious attack. Otherwise, they can establish the shared key as shown in Figure 2.

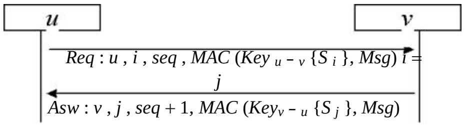
u
v
Req : u , i , seq , MAC (Key u – v {S i }, Msg) i = j
Asw : v , j , seq + 1, MAC (Keyv – u {S j }, Msg)

Figure 2. Key agreement protocol between peer nodes

Here, the node *u* firstly sends a request message *Req* to *v*

$$Req = \{u\ ,i\ , seq\ , MAC\ (Key_{u-v}\ \{S_i\}, Msg)\} \qquad (4)$$

where *u* also represents the node identity, *i* the identity of current hash value, *seq* the sequence number of transmission between node *u* and *v*. *Msg* is the current message $Msg = u \mid i \mid seq$, and $Key_{u\text{-}v}\{S_i\}$ the shared key calculated by *u* independently through (5).

$$Key_{u-v}\{S_i\} = h(f(u,v), S_i) \qquad (5)$$

Upon receiving *Req*, *v* firstly compares *i* with *j*. Only when $i = j$ can *v* calculate $f\ (v,u)$ and $Key_{u\text{-}v}\{\ S_i\}$. According to the symmetry property of Blundo symmetric polynomial, we have

$$f\,(v,u) = f\,(u,v) \qquad (6)$$

Because $i = j$, we know that $S_i = S_j$. From the definition of $Key_{v\text{-}u}\{S_j\}$ and (6), we have

$$Key_{u-v}\{S_i\} = Key_{v-u}\{S_j\} \qquad (7)$$

Hence, we obtain

$$MAC(Key_{v-u}\{S_j\}, Msg) = MAC(Key_{u-v}\{S_i\}, Msg) \qquad (8)$$

Therefore, *v* can verify the MAC through $Key_{v\text{-}u}\{S_j\}$ in *Req* message and complete the node and message integrity authentication. For legitimate node *u*, *v* will reply with *Asw*

$$Asw = \{v, j, seq+1, MAC(Key_{v-u}\{S_j\}, Msg)\} \qquad (9)$$

Upon receiving *Asw*, *u* will carry out the same procedure to verify the legitimacy of *v*.

When $i \neq j$, there should be at least one node failing to update its hash value, which will be considered unsecured or Replicated and, as a result, cannot pass the authentication or obtain (7), even if they do have the same Blundo symmetric polynomial and calculate the result in (6). This is because the one-way character of hash function and the node cannot calculate S_i through S_j and can not calculate the right $Key_{v\text{-}u}\{S_i\}$. Therefore, though *v* has $f(u,v)$, S_j and $MAC(Key_{u\text{-}v}\{S_i\},u \mid i \mid seq{+}1)$, it cannot forge $MAC(Key_{u\text{-}v}\{S_i\},v \mid i \mid seq{+}1)$ in *Asw*.

The above analysis shows that due to the updating of the hash values in the reverse hash chain, even if a node has a valid network identity, when it fails to obtain the current hash value, it cannot impersonate legitimate nodes and continue to engage in legal network communication.

However, as wireless sensor nodes may be subject to physical capture. When an attacker captures *k* nodes, the attacker can combine *k* Replicated nodes to expand the impact of the attack. Blundo et al. [7] showed that the symmetric polynomial (1) is *t* intrusion-tolerant when $k \leq t$ because the attacker cannot obtain the parameter a_{mn} of (1). Thus, even if *k* nodes collude, they still cannot find the polynomial $f(ID,y)$ pre-loaded in the nodes and launch node impersonation attack.

Finally, we want to point out that by using information entropy, we can show that the following two results analytically. Due to page limit, we omit the proofs here.

Theorem 1. In the key agreement protocol, any two nodes *u* and *v* are able to calculate their shared key $Key_{u\text{-}v}\{S\}$ when they are pre-loaded with the same hash value *S* and Blundo symmetric polynomial as shown in (1).

Theorem 2. The key agreement protocol and key update protocol are both *t* intrusion-tolerant.

C. Key Update Protocol for Heterogeneous Nodes (KUP)

BS needs to send MS the latest hash value S_{i+1}, through which BS could carry out the update of the secret keys. As it moves, the MS will communicate with CHs and securely send CHs the latest hash value S_{i+1} to replace the current one, namely, S_i. Meanwhile, the MS needs to notify corresponding Nodes that the current hash value has changed. Because key updates between the MS and CHs is similar to that between CHs and Nodes, so we let *M* be the upper node such as the MS in MS-CH or the CH in CH-Node, and let *u* be the low-level node such as the CH in MS-CH or the Node in CH-Node. Next, we will describe key update between *M* and *u*.

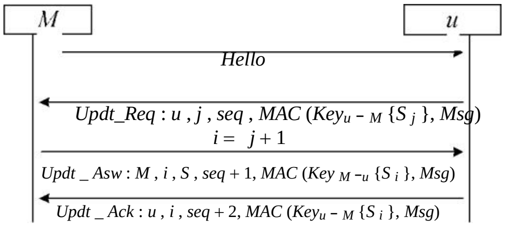
M
u
Hello
Updt_Req : u , j , seq , MAC (Keyu – M {S j }, Msg)
i = j + 1
Updt _ Asw : M , i , S , seq + 1, MAC (Key M –u {S i }, Msg)
Updt _ Ack : u , i , seq + 2, MAC (Keyu – M {S i }, Msg)

Figure 3. Key update protocol between heterogeneous

nodes

Assume the current hash value stored in M and u is S_i and S_j respectively, between which $i = j+1$ must hold. For $i < j+1$, u needs refuse to update its hash value and conclude that M might be a malicious node. If $i > j+1$, it indicates that the node u missed to update of the latest hash value, and will be marked as a potential malicious node by M. Notice that a malicious node M cannot calculate $S_i = h^{-(i-j)}(S_j)$ through $h^{-1}()$ because of the collision-freeness of one-way hash function. However, the malicious node u , who missed one round of key updating, may still want to join the network again through key updating. To prevent this, we design the KUP as shown in Figure 3.

Let M be pre-loaded with $f(M,y)$ and hash value S_i . Let u be pre-loaded with $f(u,y)$ and hash value S_j. M notifies u that it comes to update the hash value and conducts key agreement through *Hello* message. Otherwise, M and u can communicate with each other using their shared key established through KAP, which does not belong to KUP.

As shown in Figure 3, M first broadcasts *Hello* message to inform u that it comes for key updating. Upon receiving the *Hello* message, u sends the update request message *Updt_Req* to update its key materials.

$$Updt\_Req = \{u , i , seq , MAC (Key_{u-M} \{S_i \}, Msg)\} \quad (10)$$

The elements u, i, seq, and Msg in (10) are similarly defined as in (4) and the updated shared key $Key_{u\text{-}M}\{S_i\}$ between u and M is calculated through equation (5).

When M gets message *Updt_Req* with $j \neq i$-1, M records u as a potential malicious node as discussed before and then quits the update process; Otherwise, when $j = i$-1, M calculates $S_j = h(S_i)$ to obtain S_j and determines whether $MAC(Key_{M\text{-}u}\{S_i\}, Msg) = MAC(Key_{u\text{-}M}\{S_j\},Msg)$ through processes (5), (6) and (7). After passing the verification, M will send u the message

$$Updt\_Asw = \{M ,i, S, seq + 1, MAC(Key_{M-u} \{S_i \}, Msg)\} \quad (11)$$

From which M can send S_i to u securely with $S = E_{key_{M-u}\{S_{i-1}\}}\{S_i\}$ .Upon receiving *Updt_Asw*, u calculates $S_i = D_{key_{u-M}\{S_j\}}(S)$ and verifies the legitimacy of S_i because $S_j = h(S_i)$. Then u calculates the new shared key $Key_{M\text{-}u}\{S_i\}$ and uses it for message authentication as well as node authentication. Finally, u sends *Updt_Ack* message (12) to confirm the establishment of the shared secret key between M and u.

$$Updt\_Ack = \{u,i, seq + 2, MAC(Key_{u-M} \{S_i \}, Msg)\} \quad (12)$$

Through the above analysis, we observe that the Blundo symmetric polynomial and reverse hash chain are effectively combined in our KUP. Using the shared key calculated from the pre-loaded key materials between heterogeneous nodes, our KUP can carry out key update and realize securely shared keys establishment and authenticated node revocation.

As a final remark, the MS may be subject to capture. To fight against this attack, effective node revocation and authentication node update are necessary. BS firstly generates a number of identities for MS as shown in (3). All CHs are pre-loaded with MS_0 firstly. After a period of time, the current identity of MS is supposed to be MS_i. If MS_{i+1} is needed, BS will pre-load MS_{i+1} with the relevant symmetric polynomial $f(M_{i+1},y)$ and a new hash value. In this case, MS_{i+1} can send CHs its identity through *Hello* message and CHs can authenticate MS through $MS = h(MS_{i+1})$. If $MS_{\ i+1}$ is verified, CHs will store it as a legitimate one. Finally, we conclude that the use of reverse hash chain to generate the MS identity, can not only reduce unnecessary storage overheads, but also is resistant to the collusion attacks.

IV. PERFORMANCE ANALYSIS

In this section, we carry out performance for our scheme.

Network Intrusion Tolerant Rate

For the sake of discussion, we assume that the numbers of symmetric polynomials pre-loaded in the MS and cluster heads are a and b, respectively, there are b cluster heads in one subarea, and the total number of subareas is not more than t. Suppose that there are n nodes in each subarea, then the total number of nodes in the network is $N = a{\cdot}t{\cdot}n / b$. Our key agreement scheme can resist collusions among no more than t nodes in one subarea. We define the intrusion tolerance rate as the secure links probability shown in (13):

$$p_{\mathrm{Tol}}(x) = \frac{L_{\mathrm{All}}(x) - L_{\mathrm{Replicated}}(x) - L_{\mathrm{Ind\_Replicated}}(x)}{L_{\mathrm{All}}(x) - L_{\mathrm{Replicated}}(x)} \quad (13)$$

where x is the number of nodes Replicated by the adversary through physical capture attack, $L_{\mathrm{All}}(x)$ is the total number of links in the network, $L_{\mathrm{Replicated}}(x)$ is the number of the Replicated links and $L_{\mathrm{ind\_Replicated}}(x)$ is the number of links that are potentially unsecure because of the leaked secrets from the directly captured nodes.

It is clear that, if $n < t$, no matter how many nodes are Replicated, they will not affect the communications security among normal nodes. If $n > t$, the attacker could compromise more than t nodes in one subarea, leading to completely security breach in the subarea. With some mathematical manipulations, we obtain the network intrusion tolerance rate as in equation (14).

$$p_{\mathrm{Tol}} = \begin{cases} 1 & 0 \leq n \leq t+1 \\ g(x) & t+1 < n \leq N \end{cases} \quad (14)$$

where $g(x)$ is given by:

$$g(x) = \frac{\left(\frac{N}{n} - \frac{x}{t+1}\right)\binom{n}{2} + \binom{n-[x \bmod (t+1)]}{2}}{\left(\frac{N}{n} - \frac{x}{t+1} - 1\right)\binom{n}{2} + \frac{x}{t+1}\binom{n-t-1}{2} + \binom{n-[x \bmod (t+1)]}{2}}$$

In Figure 4, we show the network intrusion tolerance rate for n < 101, n = 200 and n = 400, when the network parameters are a = 2, b = 2, t = 100 and N = 10000. We also compare our scheme with those in [5], [6] and [9].

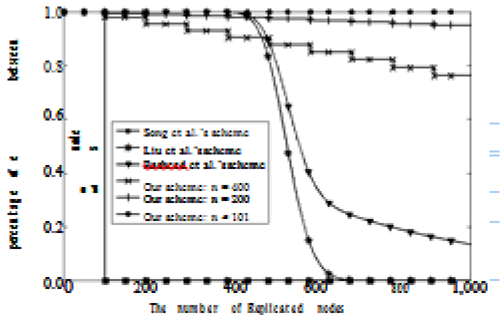

Figure 4. Network intrusion tolerant rate

We observe that, in [6], when the number of Replicated nodes exceeds the threshold t, the entire network will be completely controlled by the attacker. The scheme in [9] can guarantee that when the number of Replicated nodes is not more than 400, the whole network is secure.

B. Network Overhead

Liu et al. [9] point out that the communication overhead of Blundo symmetric polynomial approach is the same as RC5-MAC when the polynomial degree is t = 100 and node identity is 16 bits. Thus, the computation overhead of Blundo symmetric polynomial is 8 times of RC5-MACs when t = 100 and the node identity is 64 bits. From [13], the overhead of computing a MAC is equivalent to the overhead of transferring 1 byte information. It is assumed that the identity of our MS is 64 bytes and each remaining nodes 2 bytes.

We assume that the network is divided into m blocks as done in [6] and k times hash computation are needed to generate a random key in [5]. In addition, we assume that all encryption operations are based on RC5, while the message authentication and hash operations are CBC-MAC based on RC5. Table II shows the overhead of computation and communication in relevant schemes. In Song et al.'s scheme [6] the shared key generation process contains the computing of Merkle tree root, MS identity, symmetric polynomials and *MAC*s, which is equivalent to $11+\log_2^{m}$ bytes data transmission. In [5], the computation overhead consists of the computing of symmetric polynomials and random key and *MAC*s, which is equivalent to k+10 bytes data transmission. In our KAP there are 1 symmetric polynomial communication, 2 hash operations and 2 *MAC*s operations, which is equivalent to 12 bytes communication overhead. However, our KUP contains 1 Blundo symmetric polynomial, 4 hash, 3 *MAC*s, and 1 RC5 operation, which is equivalent to 16 bytes communication overhead. Results are summarized in table II.

TABLE II. COMPUTING AND COMMUNICATION OVERHEADS

Scheme	Computation overhead (RC5-MAC)	Communication overhead (Byte)
Song[6]	$11+\log_2^{m}$	$4\log_2^{m} + 23$
Rasheed[5]	k +10	64
KAP	12	32
KUP	16	78

Song et al.'s scheme needs to send and receive the identities of communicating parties, the authentication on message of MS, *seq* and the *MAC*s. However, Rasheed et al.'s scheme is more complex, requiring ensuring the communicating parties to get two kinds of shared keys. In our KAP, 2 node identities, the identity of current hash value i, 2 *seq* and 2 *MAC*s are needed to transmit, while in KUP, 3 node identities, 3 hash values identification, 3 *seq*, 3 *MAC*s and one encrypted hash value are needed to transmit.

V. CONCLUSION

In this paper, we propose an Modernize and intrusion-tolerant key management scheme for UWSN which uses the Blundo symmetric polynomial and the reverse hash chain technology for key agreement and key updating to prevent Replicated nodes from communicating with static nodes. We conduct detailed analysis and compare the performance of our scheme with related outcomes in terms of intrusion tolerant rate, computing and communications overhead and show the superiority of our proposed schemes.

REFERENCES

[1] D. Ma and G. Tsudik, "Security and privacy in emerging wireless networks," *IEEE Wireless Communications*. 2010, vol. 17(5), pp. 12–21.

[2] G. Anastasi, M. Conti and M. Di Francesco, "Reliable and Energy-Efficient Data Collection in Sparse Sensor Networks with Mobile Elements," *Performance Evaluation*, 2009, vol. 66(12), pp. 791– 810.

[3] J. Rao and S. Biswas, "Network-assisted sink navigation for distributed data gathering: Stability and delay-energy trade-offs," *Computer Communications*, 2010, vol. 33(2), pp. 160–175.

[4] A. El-Mousa and A. Suyyagh, "Ad hoc networks security challenges," *Int. Multi-Conf. on Systems Signals and Devices*. 2010, vol. 5, pp. 144–147.

[5] A. Rasheed and R. Mahapatra, "Key predistribution schemes for establishing pairwise keys with a mobile sink in sensor networks,"

IEEE Trans. on Parallel and Distributed Systems, 2011, vol. 23(1), pp. 176–184.

[6] H. Song, S. Zhu, W. Zhang and G. Cao, "Least privilege and privilege deprivation: Toward tolerating mobile sink compromises in wireless sensor networks," *ACM Trans. on Sensor Networks*, 2008, vol. 4(4), pp. 1–30.

[7] C. Blundo, A.D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences,"

Advances in Cryptology-Crypto'92, 1992.

[8] R.C. Merkle, "A certified digital signature," *In CRYPTO*, 1989, vol. 435, pp. 218–238.

[9] D.Q. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," *Proc. 10th ACM Conf. Computers and Comm. Security*, 2003, pp. 52–61.

[10] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," *In Proc. of the 9th ACM Conf. on Computer and Communication Security*, 2002, pp. 41–47.

[11] [illegible]

[12]

AUTHORS

First Author – S.Prema II-M.E-CSE, Selvam College of Technology, Namakkal, Premaselvaraj90@gmail.com

Second Author – S.Nagaraj (Assistant Professor-CSE), Selvam College of Technology, Namakkal.