Enhanced Key Management Scheme in Heterogeneous Sensor Networks with Mobile Nodes

Priyanka Ahlawat
Computer Engineering Department
National Institute of Technology, Kurukshetra (India)
mannpammy@gmail.com

Abstract. Wireless Sensor Networks (WSN) consists of a large number of small resource constrained sensors having high security requirements. Establishing a secure communication is an open and difficult issue in WSNs. Key management plays a crucial role in providing a secure communication. WSNs are used for variety of applications and are deployed in hostile, open & unattended environment making them vulnerable to various attacks such as node capture attack, node replication attacks, traffic analysis etc. One of the prominent attacks is node capture attack. In proposed scheme, we mainly deal with resiliency against node capture in heterogeneous sensor networks consisting of Fixed node (FN) and mobile sensor node (MSN). FN are more powerful than MSN. In this scheme one way hash function is applied on the keys that are stored on MSN by using hash chain class. Thus when keys are captured by attacker, only its derived versions are revealed. An attacker cannot get backward values of the key. Resilience against node capture of basic key management scheme is enhanced by applying lightweight hash chaining technique.

Keywords: wireless sensor network (WSN), heterogeneous Networks, node capture, hash chaining technique, mobile sensor node (MSN), fixed node (FN), resilience, key management scheme.

I. INTRODUCTION

A WSN (Wireless sensor network) is collection of sensors whose size can range from few hundred to a few hundred thousand sensors. They does not rely on any predeployed network infrastructure, thus they communicate via an adhoc wireless network [1]. WSN architecture often contains one or more BSs that provide centralized control. The purpose of such networks is to gather information by sensors and forward to base stations (BS). Such networks have number of benefits because of their low cost and rapid deployment. WSNs are used for variety of applications such as ocean and wildlife monitoring, manufacturing, and healthcare and many military applications [1, 2]. They are often deployed in unattended, open and hostile environments and thus subjected to various attacks. Establishing secure communication is open and difficult issue in WSN. Key management plays a very important in establishing secure communication in WSN[3]. While designing key management scheme several constraints are to be considered Evaluation metrics for key management techniques can be classified into security, efficiency, flexibility metrics [4, 5]. Security metrics include node revocation, forward and backward secrecy, collusion resistance and...
resilience. Performance metrics include memory, processing, bandwidth, energy. Mobility, scalability and key connectivity are flexibility metric [5]. WSNs can be classified according to capacity of sensor resources in to two homogenous and heterogeneous networks. Homogenous networks are composed of sensors having identical resources whereas heterogeneous networks is composed of sensors having different resource capacities[2].There are three classes of key management schemes –key distribution scheme, key agreement scheme and key redistribution scheme[4,6,7].

Contributions of the work can be summarized as follows

- We have reviewed various key management schemes and concept of hashing for improving resiliency
- We have applied the light weight chain techniques of [8] to enhance the resiliency of key management scheme supporting node mobility in heterogeneous networks in [3].
- We have shown that our proposed key scheme performs better in terms of resiliency than basic key management scheme [3] & scheme in [9] other key management scheme keeping the same memory storage and computational overhead on mobile sensor nodes.

The paper proceeds as follows. Section II describes the related work regarding existing key management schemes. Section III describes the networks model along with the assumptions that we have considered. Section IV presents our proposed scheme. Section V gives the analysis of scheme and compare with other existing scheme finally followed by conclusion in section VI.

II. RELATED WORK

Key management is major building block for providing security infrastructure in WSN. A number of key management is proposed for heterogeneous WSN. Eschenaur and Gligor proposed a basic random key predistribution scheme (E-G scheme) for homogenous networks [9]. Chan et al proposed a variant of basic E-G scheme where instead of sharing one key, nodes have to share at least q keys to establish a connection [10]. If they share less than q keys, communication will not occur. This scheme has better resilience than basic E-G scheme but scalability is an issued. Du et al [11] and Traynoreet. al [10] proposed AP (asymmetric predistribution) scheme which has better security with less storage space. BimingTian et al proposed a key management scheme for heterogeneous sensor networks using keyed hash chain where different types of keys are established and renewed [13]. It has better security and performance than E-G basic scheme. Different hashing techniques used to enhance the resiliency of key management scheme are discussed in [2, 4, 14, 16, 17, 19].

III. NETWORK MODEL AND ASSUMPTIONS

We have taken the model in Figure 1 based on model described in [3, 12]. In the network model shown in Figure 1, base station (BS), FNs and MSNs are entities to form a heterogeneous network structure. BS is an infrastructure and assumed to be secure and will not be compromised. FNs are equipped with high power batteries, large storage, powerful computation and wide radio range. FNs can communicate with MSNs through a communication key and a public/private key pair with BS & other FNs They is assumed to be secure and will not be compromised. They need to be equipped with tamper resistance hardware. MSN has limited power, small memory space, low processing capability and short radio communication range. They can change their positions within a given environment. They are more numerous than FNs. They can be compromised by an adversary. FNs can communicate with BS and other FNs and MSNs within its radio
coverage area which is similar to basic key management scheme of [3].

IV. PROBLEM AND PROPOSED KEY MANAGEMENT SCHEME

WSN are used for variety of applications and are usually deployed in unattended, open and hostile environment. They are vulnerable to various attacks such as node capture attack, node replication attacks, traffic analysis etc. One of the prominent attacks is node capture attack. An attacker can easily capture secret information stored in sensor nodes. One way to resist the node capture attack is to store less information on sensor nodes but it leads to decreased key salability. This can lead to more sharing through path key establishment which is very energy consuming process. In an unbalanced predistribution[3], if mobile node is captured, all keys are revealed to attacker but as it has less number of keys, it will reveal less information than balanced predistribution scheme[9]. In order to further enhance the resilience, hash functions are introduced [2, 8, 15-19]. A cryptographic hash function $H(\cdot)$ is function that is considered to have following properties:

1. Preimage resistant given the output $K$ ‘it should be hard to find any $K$ such that $K'=H(K)$.
2. Collision resistant; it is hard to find two different messages $K_1$ and $K_2$ such that $H(K_1)=H(K_2)$.

From the study of basic key management in [3] and hashing techniques discussed in [8], we propose our enhanced key management scheme. In this scheme we apply a lightweight hash class[8] to keys stored in mobile sensor nodes. Thus when keys are revealed, only its derived versions are revealed. An attacker cannot get backward values of the key. It will guarantee same secure connectivity coverage with increased computation. This scheme mainly deals with mobile sensor node capture. We further assumed that derived versions of the key are having same identifier as original key.

Notations used:
- $\mathbb{P}, \mathbb{A}, \mathbb{C}$: original key pool, Authentication key pool, communication key pool respectively
- $|S|, |K|$ are key ring size of FN and MSN.
- $K_p, K_{pl}$: network public key & network private key.
- $K_{R1}$ and $K_{R2}$ are two disjoint key pool containing secret codes code1 and code2 respectively

A. Key predistribution phase

Before deploying the nodes, BS generates a large key pool similar to basic key management scheme[3]. From this key pool, two disjoint sub key pool are generated. These two sub key pools are authentication pool and communication pool. Authentication pool is used to generate authentication keys and communication pool is used to generate the communication keys between FN and MSN. BS randomly selects authentication keys from authentication pool and assigned to FN along with their key identifiers. FN is assigned a key ring (containing authentication keys) of size $|S|$. Information about all mobile nodes along with their associated key identifiers list is also stored in FN. For communication with other FN and BS, public/private key pair is used. A hash function is also stored in them along with algorithm to compute the encryption key and communication key. BS also assigns two distinct key ring of equal size namely $K_{R1}$ & $K_{R2}$ generated from communication key pool. These key rings contain secret information for generating communication key. A kprt & value of ‘L’ is also stored in them by BS. For MSN key ring assignment, we apply the hashing techniques discussed in [8], we assumes key ring size of MSN is of even size and Keys are distinct and ordered. The key ring $K$ is divided into two subset namely lower subset and higher subset. Keys belonging to lower subset are hashed $(L \mod L)$ times and keys belonging to higher subset are hashed $(L-1 \mod L)$ times where I =node identifier, L=parameter. To reduce hash computation, we use a parameter L which will bound the number of hash computations. Thus we have

- $K=$ lower set keys and higher set keys
  - Lower set keys = $(h^{1 \mod L}(k_1))\ldots (h^{1 \mod L}(k_{K/2}))$
  - Higher set keys = $(h^{L-1 \mod L}(K_{s1}))\ldots (h^{L-1 \mod L}(K_{s2}))$

Where $k_1, k_2,\ldots , k_K$ are the distinct and ordered (according to their key identifier) keys. Keys assigned to mobile nodes are divided into two subset to prevent smart node capture attack where an attacker capture lower identifier nodes to get upward key or a larger fraction of communication is compromise [8].

B. MSN authentication phase

After node is deployed, authentication is performed between FN and MSN. Once a FN is selected based on signal strength, bandwidth quality etc. MSN request to join or to start communication with a FN., it sends its MSN ID. When request for authentication is received by FN, then it will search for keys in its authentication
key ring S (MS identifiers as well key identifiers are stored in FN by BS) when shared key identifiers are found, it will compute the encryption key. FN checks whether shared key identifier to lower subset or higher subset. When it belongs to lower subset it will be hashed I MOD L times otherwise hashed L-1- I MOD L times. FN will generate it an encryption key. If key match= 1 then Encryption key = $h^I \text{ MOD } L \text{ or } L-1- I \text{ MOD } L (K_{\text{shared}}(\text{FN-MSN}))$ Else Encryption key = $h^I \text{ MOD } L \text{ or } L-1- I \text{ MOD } L (K_{\text{shared}}(\text{FN-MSN}) || \ldots \ h^I \text{ MOD } L \text{ or } L-1- I \text{ MOD } L (K_q_{\text{shared}}(\text{FN-MSN}))$ (for q shared keys)

In case no shared key is found between FN and MSN, FN will authenticate it through BS. BS is having all keys of MSN, then BS will generate encryption key & send to FN along with key IDs used to generate the encryption.

C. Communication Key establishment phase

To secure communication between FN and MSN, FN will generate communication key, encrypt by encryption key generated as discussed above and send to MSN along with key IDs used for generating encryption key. After authentication, FN will generate communication key from the two key ring of same size randomly assigned from communication pool. This communication key is encrypted by an encryption key and is sent to MSN along with key IDs. After this BS sends the encryption key along with the key identifiers used for that key to the FN. FN will use that key to encrypt the communication key and send to the MSN along with key identifiers.

Communication key = $(\text{code}1 + \text{code2})^q \text{ MOD } (\text{code2})$ Where $q$= total number of key shared between FN and MSN or total number of key identifiers used for computing encryption key similar to communication key generation in basic key management scheme in [3].

D. MSN leaving old FN and joining new FNs

During movement of MSN, it may move from range of one FN to another FN. Once new FN is selected based on link quality, availability and bandwidth. MSN sends previous FN ID to new one so that new FN can directly communicate with previous FN if no shared key is found, thus reducing the broadcast overhead and can authenticate with previous method. After authentication BS will add it its member list and inform BS about new node. This will help to avoid node replication attacks [3]. New FN generates a communication key for MSN. This is similar to basic key management scheme in [3]. The proposed key management scheme is explained as follows

Key predistribution phase:
1. BS generates a large key pool P.
2. BS randomly selects two disjoint sub key pools A (authentication key pool) & C (communication key pool.
3. FN keys assignment
   (i) It randomly selects keys (= |S|) from key pool A and assign (original keys) to fixed node along with their key IDS.
   (ii) It also store the information about all MSNs IDs & their associated key IDs list, public/private key pairs for communication with BS and other FNs in FN.
   (iii) It store two distinct key ring KR1 & KR2 (randomly selected from key pool C) of equal size containing secret code1 & code2 respectively.
4. MSN keys assignment
   (i) BS randomly selects keys (=|K|) from A
   (ii) Divide the selected keys into two sets. Lower set having lower identifiers keys and vice versa.
   (iii) For keys selected above
   (iv) Apply $k=h^I \text{ MOD } L \text{ or } L-1- I \text{ MOD } L(k) & \text{ store in MSN.}$

Node authentication phase
5. MSN sends its ID to FN to start communication.
6. FN search for MSN’ keys in its key ring
   (a) If key match =0
      Then FN asks BS for encryption key
   Else if key match =1

612
Encryption key = $h^{MODL} or L^{1-MODL}(K_{Shared (FN-MSN)})$
Else key match= ‘q’
Encryption key= $(h^{MODL} or L^{1-MODL}(K_{Shared (FN-MSN)}))h^{MODL} or L^{1-MODL}(K_{Shared (FN-MSN)})$
…………….. $h^{MODL} or L^{1-MODL}(K_{Shared (FN-MSN)})$

7. FN generates nonce, encrypt by encryption key & send to MSN along with key IDs.
8. MSN compute the decryption key and decrypt the nonce

Communication key establishment phase

9. After authentication, FN generate a communication key communication key = (code1+code2)$^{n}$ MOD code2
10. Encrypt it by encryption key & sends to MSN.
11. MSN decrypt it & can start communication.

MSN leaving & joining new FN
12. MSN send its ID to new FN along with ID of previous FN.
13. New FN search for key match.
   (a) If key match= ‘0’
       Then ask Previous FN for encryption key.
       If previous FN has no key match, Ask BS for encryption key. Else generate the encryption key and communication key discussed above

V. ANALYSIS OF PROPOSED KEY MANAGEMENT SCHEME

A. Resilience against node capture

The resilience of basic key management scheme in [3] is enhanced by using hash functions discussed in [8].
The scheme is highly resistant using lightweight hash chaining techniques that conceal the keys preloaded in
MSNs. We analyze resiliency of the proposed key management scheme against node capture attack. FNs are
equipped with tamper resistant hardware so they are secure. Secret information cannot be revealed. Due to
cost constraints, MSNs are not tamper resistant and hence can be easily captured by an adversary. In the
proposed key management scheme, we deal with MSN capture it is noted that MSNs keys are divided into
two subsets very similar to technique discussed in [8]. Lower subset containing keys having lower key
identifiers and higher subset containing higher key identifiers. Lower subset keys are hashed $I^{MODL}$ times
whereas higher subset keys are hashed $(L-1-I^{MODL})$ times to resist smart node attack. Resilience of
scheme is measured as the fraction of total network communication that are compromised when n MSN are
captured by an adversary after their deployment. When a mobile node is compromised, similar to analysis in
[8], we compute the resilience of the proposed scheme

Probability of hashing l times $= \frac{1}{\lambda^{l}}$ where $l=I^{MODL}$ (1)

When a compromised key is hashed for (L-1-I) times, then key having same identifier can be disclosed if it
hashed 1 (or L-1-L) or more times as we get only forward hash values. Thus probability that a key having
same identifier is disclosed when node I is captured

$$= ((\sum_{i=0}^{L-1} \left(\frac{L-i}{L}\right)) + (L - (L - 1 - l))/L$$

Thus probability that a key having same identifier is disclosed when a node is compromised

$$= \sum_{l=0}^{L-1} (L - l + 1 - l)/L$$

$$= \sum_{l=0}^{L-1} (L + 1)/L$$

$$= (L + 1)/L$$

Probability of key compromised when a node is compromised

$$p' = \frac{R(L+1)}{L+1}$$

The fraction of uncompromised keys = $(1 - p')$

When ‘n’ MSNs are captured, fraction of uncompromised keys = $(1 - p')^n$

Thus probability that a given key is known

Putting value from Eq. 3 we get
This will be the probability of a given key known to an adversary when \( n \) MSNs are captured. The resilience of key management scheme directly depends on number of MSNs captured. It is similar to analysis in [8].

\[
= (1 - (1 - \frac{\mu(L+1)}{2A/L})^n)
\]

We plot in Figure 2 fraction of uncompromised links being compromised depending on number of MSNs captured. The figure 2 clearly shows that resilience against the MSN capture is better using our scheme than the basic key management scheme in [3] and E-G scheme in [9]. In the analysis we fixed \( |K|=400 \) (E-G Scheme), \( |K|=200 \), \( |P|=10000 \), \( L=10 \) for other schemes.

**B. Storage capacity**

Storage memory required for this key management include storing \( K \) key pairs (keys (hashed), key identifiers) where keys are selected randomly from authentication key pool. Memory cost for a MSN = \( K \) *(key size +key identifier size) MSNs store public keys of other FNs for global connectivity. FN contains two secret codes \( M1 \& M2 \) taken from communication key pool \( C \) for generating communication key for MSNs. Suppose two key rings contains 200 keys each they can generate 40000 communication keys. If using E G scheme they have to store 40000 keys thus storage space is reduced foe this scheme. So storage capacity of proposed key management scheme (for MSNs) is same as basic scheme in [3].

**C. Communication overhead**

There is a reduced communication overhead for this scheme as compare to basic E-G scheme. Mobile node sends only its node identifier rather than sending list of key identifiers for the shared key discovery. However incase FN is unreachable, MSNs exchange key identifiers to other MSN for a temporary connection. For MSN to MSNcommunication the number of bits transferred will be \( m*\)key identifiers size incase node identifier is neglected. For communication between FN and MSN only MSN ID is sent similar to basic scheme in [3].

**D. Connectivity**

It has same connectivity as basic key management scheme discussed in [3].

**E. Impact of value of \( L \) on proposed key management scheme.**

The parameter \( L \) is introduced to reduce the number of hash computations on FNs. Similar to analysis in [8], we found that larger value of \( L \) will increase the resiliency of scheme with added computation. and is shown in Fig 3.
VI. CONCLUSION

Key management plays a crucial role in security of WSNs. Self enforcing schemes are heavy weight schemes for resource constrained sensors. In this paper we propose a resilient key management scheme supporting node mobility in heterogeneous sensor networks. The proposed key management scheme is made more resilient by applying a simple light weight hash chaining technique. Compared to existing key management scheme discussed in [3, 9], the proposed scheme is more resilient against MSN capture. This scheme is also resistant to smart node attack problem.

REFERENCES