Key Management and Security Planning in Wireless Ad-Hoc Networks

Hemanta Kumar Kalita
Innovation Lab – Performance Engg
TATA Consultancy Services
Gateway Park, MIDC, Road no 13,
Andheri(E), Mumbai

Avijit Kar
Dept of Computer Science & Engg
Jadavpur University, Kolkata

ABSTRACT

Wireless Sensor Network is a class of wireless ad hoc networks. Self organization is an important phase of a wireless sensor network. After deployment, sensors nodes are required to self organize themselves to form a network of their own. Security is an important aspect while forming the network. Only the authorized nodes should be allowed to join the network. For implementing security, such as confidentiality, integrity and authentication, keys are needed. How keys are handled is described in key management approach. Key management encompasses generation, storage, distribution, re-key and retirement of a key. It is observed that majority of the attacks are carried out during the key distribution phase itself. Hence, a proper key management is utmost important for implementing fool proof security in a Wireless Sensor Network. Depending upon the type of keys used, for example, symmetric or asymmetric key, key management techniques vary. In this paper an approach to key management for asymmetric key based security schemes of Wireless Sensor Network is proposed. The approach considers entire life of a 'key' and subsequently proposes algorithms/schemes for key (pre) distribution, re-key and revocation. We also discuss security planning steps for wireless sensor network in this paper.

General Terms

Algorithms, Performance, Design, Reliability, Experimentation, Security, Standardization

Keywords

WSN, Security, Key Management, PKI, Digital Certificate, Digital Signature

1. INTRODUCTION

Wireless Ad hoc Networks application can be divided into three classes: mobile ad hoc network (MANET), wireless mesh network (WMN) and Wireless Sensor Network (WSN). In this paper we consider the issue of key management and security planning in WSN. Henceforth, both the terms ad hoc networks and WSN will be used interchangeably in this paper.

Use of asymmetric key in Wireless Sensor Network (WSN) is not new. In the literature we find many such schemes which use asymmetric keys. In our background section we mention some of this scheme. Authors of this paper also propose an asymmetric key based data communication scheme for self-organized WSN in [14]. Since sensor node is constraint by low power and computational resource like memory etc hence the study shows that use of digital certificate for public key and its validation by usage of third party Certification Authority is not feasible in WSN. This is a typical scenario where we advocate use of asymmetric keys in WSN and in the meanwhile cannot go for use of digital certificate. Essentially, the public key used in WSN is termed as certificate less public key. As we know in asymmetric key based scheme there is a pair of keys called private key and public key. Private Key remains private to the user and the public key is made public. Any sender can send a message by encrypting it with recipient's public key and only the receiver can decrypt the message with its private key. On the other hand a sender can sign its message with its private key and send the digital signature (encrypted hash) along with the message. The receiver can validate the authenticity of the message by decrypting the digital signature with the sender's public key, generating a hash from the message using same hash function and comparing calculated hash with the decrypted hash.

What are the problems in using certificate less public key? To understand this we have to go through the Public Key Infrastructure used in the Internet. Public Key Infrastructure (PKI) defines assertion of identity of an entity (or person) that have not been met previously through the use of certificate containing identifying information and the entity's public key [9]. The certificate is more properly called X.509 certificate. PKI accomplishes this through the use of mutually trusted Certification Authority (CA). In Figure 1 we elaborate on how an entity A trusts certificate of another entity B. Note that certificate validation as defined in RFC 5280 is not trivial and the scope of the paper does not permit a detail review on this.

Since we are not using digital certificate for verification and validation of public keys in WSN, therefore steps 1-7 as shown in Figure 1 can't be conducted in WSN. Thus, onus of trusting B simply lies with A. If two sensor nodes share its public keys to each other it can establish a secure channel where data is transmitted in encrypted format. However, problem lies in the trust model. Both of the nodes do not know authenticity of each other by merely having its certificate less public key since there is no mutually trusted third party Certification Authority to verify its authenticity. In this situation any adversary node can spoof others certificate less public key claiming its own. Also, without proper authentication any adversary can join the network without any hassles and become an active member. A strong adversary may claim itself as the base station and there will not be any way to verify this for a new node.

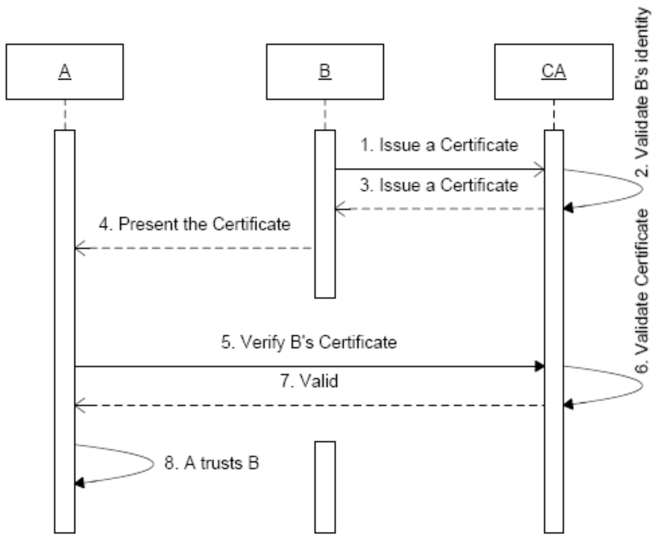

Figure 1 Authentication of B's certificate in PKI

From the above discussion it is clear that certificate less public key can't be used in WSN unless it is properly authenticated by some other means. Thus, any sender and receiver have to authenticate it before accepting each other's certificate less public key for data communication. This is a key distribution problem. In this paper we propose a solution to this key distribution problem with the help of a key management framework.

Remainder of this paper is divided into five sections. In Section 2, we discuss background or related work. In Section 3 we propose our key management algorithms for WSN. In Section 4 Security Planning for WSN is discussed and in Section 5 we provide result analysis of our proposed approach. Finally in Section 6 we conclude the paper.

2. RELATED WORK

[3] shows how to map from two classes of combinatorial designs to deterministic key distribution mechanisms. [8] proposes two schemes for public key authentication in wireless sensor network: a naive scheme and a memory efficient scheme. In the naive scheme authors propose each node to have hash value of all the other nodes' public key. In the memory efficient scheme authors proposes to use Markle tree, a hash tree used for optimization of authentication of public keys. [10] proposes Sizzle, a small footprint implementation of an HTTPS stack that brings the well established security properties of SSL to the embedded Internet. [4] and [5] consider distributed and hierarchical wireless sensor networks where unicast, multicast and broadcast type of communications can take place. [2] proposes design of an elliptic curve coprocessor suitable for 8 bit system typically used in the low-end node of the sensor networks. [13] proposes C4W, an identity-based public key infrastructure specially designed for wireless sensor networks (WSNs). [6] proposes a novel, self-organizing key management scheme for large-scale and long-lived WSNs, called Survivable and Efficient Clustered Keying (SECK). [7] presents an overview to key management and key distribution approaches for application in wireless sensor networks and categorizes key management solutions. [1] surveys well known security issues in WSNs and studies the behavior of WSN nodes that perform public key cryptographic operations. [11] proposes security architecture based on pseudo-inverse matrix. [12] presents secFleck, a Trusted Platform Module (TPM) based Public Key platform for sensor networks.

3. PROPOSED KEY MANAGEMENT

Key management deals with the generation, storage, (pre) distribution, re-keying (update), and revocation of keys. Secure methods of key management are extremely important for overall security of a system. Once a key is randomly generated, it must remain secret to avoid unfortunate mishaps (such as impersonation). In practice most attacks are aimed at the key management level, rather than at the cryptographic algorithm itself. In this section we describe our proposed key management approach for a secure framework of wireless sensor network.

3.1 Key Distribution Centre

A key distribution centre (KDC) is responsible for generation, storage, distribution and renewal of keys. KDC is supposed to follow the security policies approved and enforced. KDC works with the base station. It may be embedded inside the base station or operate from behind the base station.

3.2 Key Pool Generation

Whether using a secret-key cryptosystem or a public-key cryptosystem, one needs a good source of random numbers for key generation. The main features of a good source are that it produces numbers that are unknown and unpredictable by potential adversaries.

3.2.1 Random Number Source

Random numbers obtained from a physical process are in principle the best, since many physical processes appear truly random. One could use a hardware device, such as a noisy diode; some are sold commercially on computer add-in boards for this purpose. Another idea is to use physical movements of the computer user, such as inter-key stroke timings measured in microseconds. Techniques using the spinning of disks to generate random data are not truly random, as the movement of the disk platter cannot be considered truly random. By whichever method they are generated, the random numbers may still contain some correlation, thus preventing sufficient statistical randomness. Therefore, it is best to run them through a good hash function before actually using them. Another approach is to use a pseudo-random number generator fed by a random seed.

3.2.2 Key Generation Algorithm

The algorithm for key generation may use node id (For example, mac id) as seed to the randomizer for generating a key. For generation of public/private key pair, we suggest use of Elliptic Curve Cryptography. It is found that ECC has same level of security as in RSA with far lesser key size. For example, a 160 bit ECC key provides same level of security of a 1024 bit RSA key.

3.3 Definitions

One hop key K_1: K_1 is used for authentication of a new node by a trusted node in the wireless sensor network before exchanging each other's public key. One hop key K_1 can be used for communication between two nodes only and is common for the entire network. For a new node K_1 pre-loaded and is a futuristic key, meaning it is active in a particular period in future. Other characteristic of this key is it is re-keyed frequently.

Hook Key hkk_{node}: This key is specific and unique to a node. A Join Request message is encrypted using hkk_{node}. This is necessary for two reasons. (i) Base station can authenticate a new node by comparing the hook key from its list; (ii) a new node does not know the public key of other nodes as well as the base station. Also, the public key of the new node is not known to other nodes initially. hkk_{node} is supplied to a node before deployment (pre distributed).

Base station Public Key: Base station public key is secret to the trusted node of the wireless sensor network and supplied only to the authenticated new node. Periodically it is changed and announced.

Base station Private Key: Base station's private key is known only to the base station and given before deployment.

Node Public Key: Node's public key is secret to the trusted node of the wireless sensor network and announced after successful authentication of a new node.

Node Private Key: Node's private key is known only to the specific node and given before deployment.

Multi Cast Key: It is used by base station for multi casting group message such as announcement of a new node’s public key.

3.4 Key Pre-Distribution

Key Pre-distribution involves how keys are distributed to its recipient securely during commissioning. Note that – key pre-distribution is slightly different than key distribution. In case of key pre-distribution keys are transferred to the device at the time of device commissioning, whereas key distribution involves distribution of keys during runtime. In our framework of secure wireless sensor network, a key distribution centre needs to pre-distribute four keys to each field device (read sensor node). They are – one hop key, hook key, and private/public key pair. Also, base station is pre-distributed with four keys. They are – One Hop Key, its Private/Public Key pair and Multicast Key.

3.5 Key Distribution

Key distribution in runtime involves two level of authentication of a new node’s credential and in the mean while (i) exchange of public keys by a new sensor node with a neighbor connected to the sensor network and (ii) obtaining base station’s multicast key and public key by a new node.

3.5.1 Between Sensor Node

In this section we propose a key distribution scheme for exchanging public keys by the sensor nodes in WSN. As mentioned earlier the PKI we propose for WSN is certificate less, which means the public keys can't be verified for its authenticity with the help of Certificate Authority as is done in PKI. We solve this problem with the help of symmetric key, K_1 at the node level. In our proposed certificate less PKI scheme, public keys of two nodes are made public to each other only after verifying each others’ identity. This is necessary since a node can't verify the authenticity of the public key using third party. We propose how securely a node can obtain the public key of another node.

It is assumed that public keys of two nodes are not known to each other initially and our aim is to distribute one's public key to other node. A new node, A is required to authenticate itself to its nearest neighbor, D which is already connected to the sensor network and hence a trusted member of the network. For this A uses one hop key, K_1 pre-distributed to it during commissioning. Note that a new node can use K_1 in a pre-specified time only. On successful authentication, D accepts A's public key as valid one and gives back its public key to A. At this stage, both A and D can start communication. In Figure 2 we describe our proposed schemes for public key distribution between two sensor nodes.

3.5.2 Between Base station and Sensor Node

In this section we discuss a scheme for a new node to obtain the multicast key and public key of the base station. A new node, A after obtaining public key of D is still unknown to the rest of the network including the base station. Also, A does not have the public key of the base station as well as the multicast key and hence, it can't start communication to the base station unless D forwards its join request to the base station. Therefore, A needs to authenticate itself to the base station with the help of D by using the Hook Key, hkk_A pre-distributed. Since A can communicate to D, so it supplies its Join Request message encrypted with Hook Key, hkk_A and signed with its private key to D. D, already a trusted member of the network verifies the integrity of the message with A’s public key. D then forwards the encrypted Join Request to the base station by encrypting it with base station's public key and authenticating it with its own private key. On successful authentication, D supplies public key of the base station and the multicast key to A and confirms that to the base station. Base station then announces the public key of A to the network only to the node falling in the A’s Key path. This means announcement of public key of the new node A is done to all the nodes which fall on the path from D to the base station.

In Figure 3 we describe our proposed schemes for obtaining public key and multicast key of the base station by a new node.

3.6 Re-Keying

Re-keying or renewal of key is important part of any secure system.

3.6.1 Re-keying of One hop key

One hop key K_1 is a symmetric key common to all nodes in a WSN. If K_1 is renewed, then any new device can't join with the old K_1. It has to get new K_1 from the commissioning engineer out-of-band. This is needed to avoid node capture attack, where a rogue client may use credentials of the compromised node to join the network.

3.6.2 Re-keying of Hook Key

Hook Key is renewed for avoiding node capture attack. Note that if a node knows public key of any other node, then it will be able start communication with that node. Similarly, if a node knows public key of the base station, then it will be able to communicate with the base station. Hence, if a node goes off from the network for the time being and after some time starts communicating again, then it will not have to rejoin again. This is basically due to the fact that its public key is known to the other nodes of the network. However, during that time if base station's public key is renewed then base station will reject its message. In that case it will have to rejoin the network again with the help of its nearest neighbor trusted node by supplying the Hook Key to the base station. Thus, Hook key is very essential when a sensor node goes out of the network and needs to join again.

When a sensor node is active in the network and its Hook key expires according to the security policy then a new hook key for the sensor node is sent by encrypting with the sensor nodes public key and signing the message with the base station’s private key.

```
A:
$M \leftarrow K_A^{pub} + Nonce\ (timestamp)$
$e :\ K_1 \times M \rightarrow C$
$h : C \rightarrow H$
$e :\ K_1 \times H \rightarrow \Delta$
$A \rightarrow D : C + \Delta$
D:
$d :\ K_1 \times C \rightarrow K_A^{pub} + Nonce$
// Authentication
$e :\ K_1 \times (K_A^{pub} + Nonce) \rightarrow C'$
$h : C' \rightarrow H'$
$d :\ K_1 \times \Delta \rightarrow H$
If $(H = H')$ then
    Supply D’s public key to A by encrypting the
    message  with A’s public key and authenticating the
    message with D’s private key. Sends ‘nonce+1’ along
    with the message.
Else
    Reject
```

Figure 2 Public Key exchange by a new node

3.6.3 Re-keying of Public/Private Key pair

For security reason, even private/public key pair of the sensor nodes also needs to be renewed. However, this should be done by commissioning engineer using hand held device (out-of-band). No over-the-air renewal is recommended for (re) distribution of public/private key pair of a sensor node. Once renewed, a node needs to authenticate itself (two levels). Upon successful authentication base station announces the public key of the newly joined node to other nodes.

3.7 Key Revocation

The Key revocation is the process of removing keys from operational use prior to their originally scheduled expiry, for reasons such as node capture or a node is not part of the network for a long period. Key revocation has greater importance in public key cryptography but its significance cannot be ignored in symmetric cryptography as well. This is particularly important because the devices which are no longer part of the network should not have network-specific secrets. In our scheme we are having three device specific keys: hook key and public/private key pairs. One hop key is only network specific. When a node is found to be compromised, the device should self-destruct or otherwise keys should be automatically deleted from its memory. On the other hand base station prepares a key revocation list containing ID of the compromised node and multicast it to the members of the network. Meanwhile one hop key and public key of the base station is renewed.

```
A:
$M \leftarrow JoinRequest + Nonce\ (timestamp)$
$e :\ hkk_A \times M \rightarrow C$
$e :\ K_D^{pub} \times C \rightarrow C'$
$h : C' \rightarrow H$
$e :\ K_A^{pri} \times H \rightarrow \Delta$
$A \rightarrow D : C' + \Delta$
D:
D extracts C of A from $C'$, verifies integrity of the message
and forwards C to base station by encrypting with the base
station’s public key and authenticating with D’s private key.
$D \rightarrow B : C'' + \Delta'$
B:
Base station B now verifies $hkk_A$ from the message and says
‘Yes’ or ‘No’ to D.
D:
If  B’s response is ‘Yes’ then
    D gives the public key $K_B^{pub}$ and multicast key of
    Base station to A. And, confirms this to base station.
Else
    D rejects A.
B:
If  B gets the confirmation from D then
    B sends a multicast message to all nodes in the key
    path announcing A’s public key. The multicast
    message is encrypted using multicast key and
    signed using B’s private key.
```

Figure 3 A obtaining Public Key and Multicast Key of B

4. SECURITY PLANNING

Aim of this section is to provide basic guidance in developing a security plan for a site where WSN is to be deployed. Basic approach of security plan consists of the following steps:

1. Identify what you are trying to protect
2. Determine what you are trying to protect it from
3. Determine how likely the threats are
4. Implement measures which will protect your assets in a cost effective manner
5. Review the process continuously and make improvements each time a weakness is found

In Figure 4 we show security planning steps for WSN.

Risk Analysis involves determining what you need to protect, what you need to protect it from, and how to protect it. It is the process of examining all of your risks, then ranking those risks by level of severity. *Security Infrastructure* encompasses

- Personnel Security for reducing risks of human error, theft, fraud, or misuse of facilities
- Organizational Security control that addresses the need for a management framework that creates, sustains, and manages the security infrastructure
- Physical and Environmental Protection
- Data security planning to reduce the risk associated with the unauthorized access, disclosure, or destruction of data
- Network Security to ensure the secure operation of network assets through the use of appropriate layered protections
- Access Management to address an organizations ability to control access to assets based on business and security requirements
- Application Security to develop software applications based on industry best practices and include information security throughout the software development life cycle
- Responsibilities regarding information security are to be unambiguously allocated and detailed

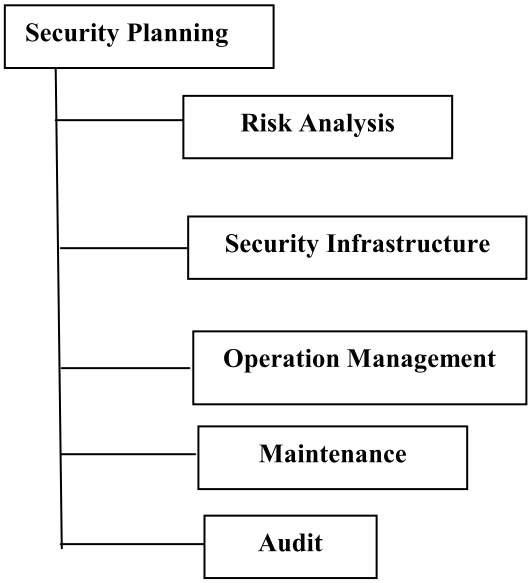

Figure 4 Security Planning

Operation Management includes change management, asset management, media handling and disposal, data and program backup, security monitoring and logging and incidence response. *Maintenance* phase of security planning takes care of security maintenance such as tracking relevant exploits, conformance maintenance, education, business continuity, disaster recovery plan, and security management. Tasks of security management are to

- Provide ongoing management support to the security process
- Serve as an alternative channel for discussion of security issues
- Develop security objectives, strategies, and policies
- Discuss status of security initiatives
- Obtain and review security briefings from the Information System Security Officer
- Review security incident reports and resolutions
- Formulate risk management thresholds and assurance requirements
- Yearly review and approval of the Information Security Policy
- Yearly review and approval of the ISMS

Finally, *Audit* phase of security planning is required for a review of the implementation of the information security infrastructure.

4.1 Security Policy

The security policy is basically a plan, outlining what the organization's critical assets are, and how they must (and can) be protected. A good security policy is comprised of several factors. Such as –

- *Usability*. A security policy is of no use to an organization or the individuals within an organization if they cannot implement the guidelines or regulations within the policy. It should be concise, clearly written and as detailed as possible in order to provide the information necessary to implement the regulation.
- *Acceptability*. A good security policy also takes into account the existing or implicit rules in use. A security policy should in no way impede or interfere with the business. Rather, it should enhance the process, providing confidence in the security of the day to day operations.
- *Enforceability*. It must be enforceable with security tools where appropriate, and with sanctions where actual prevention is not technically feasible. Firewalls, intrusion detection systems, anti-virus applications are some of the tools that can be used to apply the policies in the business environment.
- *Compliance to Laws*. Local, state laws should also be considered when creating the security policy.
- *Auditability*. A security policy should also specify what auditing processes will be put in place to verify compliance, and the punitive actions that may be taken in the event of non-compliance of any of the stipulated regulations.
- *Confidentiality*. The interests of employees, third organizations and the business goals of the organization should always be considered in a security policy.
- *Consensus*. When creating a security policy, it is a good approach to have drafts reviewed by representatives from different departments, such as IT managers, legal and human resources personnel and executives.

- *Create Standards*. Purchasing decisions can be influenced by security policies, as products will need to address security as outlined within the document. Therefore, a good security policy will help to create standards for software, hardware and other supporting network equipment.
- *Corrective Actions*. Security policies will also help to clarify what actions should be taken, and the people to be notified, in specific situations.
- *Provide Guidance*. Security policies that are well thought out and inclusive will always help in providing guidance and directives for policies in other areas.
- *Living Document*. Finally, a security policy is a living document, and as such, in order to be effective should be reviewed and updated on a periodic and regulated basis.

A good security policy has the following components:

- Parameters. Explain why the security policy is being implemented at the site.
- Risk Assessment. Identify assets and threats to the assets.
- Actual Policies. Define roles and responsibilities of the security department. Propagate security awareness throughout the organization. Plan for
    - backups and business continuity
    - physical security
    - access controls
    - authentication and encryption
    - auditing, reviews and compliance

5. RESULT ANALYSIS

In the previous section we propose Key Management scheme for a secure WSN. We introduce several algorithms/schemes covering the area of key generation, key distribution, re-keying, and key revocation. Re-keying of keys are done at the run-time for reasons such as key expiry, node capture, or when it is detected that a node is not connected to the network for long period of time. It is like regular operation of WSN such as data collection and acknowledgment. Foot prints of re-keying algorithms on the overall operation of the WSN are relatively very low as compared to key distribution. Again, key revocation is a process involving re-keying or otherwise involves design of sensor node for self-destruct capability. Hence in our analysis algorithms related to re-keying and key revocation are not considered. We simulate the algorithms related to key distribution and discuss the simulation results.

5.1 Comparison

We compare our proposed key distribution scheme with other such scheme in Table 1.

Table 1 Comparison of Key Distribution Scheme

TinyPK[15]	CA's public key pre-deployed to SN. Communication between external entity and sensor node only. No node to node communication. Needs CA. Challenge-response in plain text.
Markle Tree based[16]	Markle Tree root, hash of logN public keys, a node's public/private key pair pre-loaded. A node needs to sends its id, public key and H hash values as proof for public key authentication. Challenge-response in plain text
Pseudo Inverse Matrix based[11]	Negotiates secret key between a sensor node and base station. For node to node communication receiver needs to have decryption key from base station. Challenge-response in plain text.
Our Scheme	Four keys: futuristic one hop key, hook key and public/private key pair pre-loaded to a sensor node. Two level mutual authentication for a new node to accept its public key by a trusted node in the network. Layer wise encryption, authentication. Challenge response is in cipher text. Key path compression: Each trusted node possesses only a subset of public keys of other node.

5.2 Estimation of Average Energy Overhead

In the following we estimate additional packets (overhead) needed to transmit and receive initially for key distribution. From this we estimate average energy, E_{avg} depletion (overhead) due to the key distribution.

5.2.1 Estimation for A

A new node 'A' needs to send and receive the following packets:

- A supplies its public key using its credential (One Hop Key) to D: 1 packet (send)
- D then supplies its Public key to A or drops the request based upon verification of the credential: 1 packet (receive)
- A supplies the Hook key to D using the public key of D: 1 packet (send)
- D then supplies the public key of B to A: 1 packet (receive)
- Same is confirmed to D: 1 packet (send)
- B then broadcasts/multicast the public key of A to all the nodes in the Key path: 1 packet (receive)

Thus, total no of packets sent and received by a new node, A: 6.

5.2.2 Estimation for D

Note that, for authenticating a new node a trusted member, D is involved and D, in that process needs to send and receive several packets. If D needs to authenticate only single new node, then D needs to send and receive following packets:

- D verifies the credential and accepts the Public key of A knowing that it is indeed of A only: 1 packet (receive)
- D then supplies its Public key to A or drops the request based upon verification of the credential: 1 packet(send)
- A supplies the Hook key to D using the public key of D: 1 packet (receive)
- D encapsulates Hook Key of A with the public key of Base station (B) and signs it with its own private key and sends it to Base station: 1 packet (send)
- B receives the message, verifies the Hook Key of A and sends the verification result to D: 1 packet (receive)
- D then supplies the public key of B to A: 1 packet (send)

- Same is confirmed to B: 1 packet (receive), 1 packet (send)
- B then broadcasts/multicast the public key of A to all the nodes in the Key path: 1 packet (receive)

Total no of packets sent and received by a trusted node, D: 9.

5.2.3 Estimation for Other Intermediate Node

An intermediary node falling in the key path between D and B needs to receive and forward authenticating packets coming from D to B and from B to D:

- Any intermediate node knows Public key of D and hence can verify the authenticity of the message. Note that to reduce the size of the message, public key of any signing node is not transmitted along with the message: 1 packet (send), 1 packet (receive)
- B receives the message, verifies the Hook Key of A and sends the verification result to D: 1 packet (send), 1 packet (receive)
- D then supplies the public key of B to A. Same is confirmed to B: 1 packet (send), 1 packet (receive)
- B then broadcasts/multicast the public key of A to all the nodes in the Key path: 1 packet (receive)

Total no of packets sent and received by an intermediary node (between D and B): 7.

For the nth new node, A to join, (n-1)th node will be authenticator node, D and remaining nodes (n-2) will be the intermediary nodes. Hence, total numbers of packets transmitted and received for the nth new node to obtain the keys are:

$$6 + 9 + (n - 2) * 7 = 1 + 7n \cong 7n$$

A new node can't obtain the keys from the network if its nearest neighbor is not the trusted member of the network. In other words, a new node has to wait till its nearest neighbor becomes a trusted member of the network.

Initially, there will be only one new node, A trying to get the keys from base station, B. In that case, B plays both the roles of D and B. Then, there will be 2nd, 3rd, 4th node, etc. And, finally nth new node will get the keys from the network. From this observation, we can infer the worst case scenario for packets sent and received in the network for establishing security credentials.

In worst case, there will be X number of packets sent and received by the network for establishing security credentials, where

$$X = 7(1 + 2 + 3 + 4 \dots \dots + n) = \frac{7n(n+1)}{2}$$

While estimating energy consumption if we see overall scenario, the total units of energy consumed will be $E = XC$ where C is cost in terms of energy of a sensor node to send or receive a packet.

For simulation, we distribute E evenly to the whole network. In other words, E is divided to all the sensor nodes of the network. Therefore, while getting keys each sensor node depletes E_{Avg} units of energy, where

$$E_{avg} = \frac{E}{n} = \left(\frac{X}{n}\right)C = \left[\frac{7n(n+1)}{2n}\right]C = \left[\frac{7(n+1)}{2}\right]C$$

In actual scenario it may not be E_{Avg}. As some nodes may be used repeatedly as authenticator (read D) and some nodes will not even get a single chance. For example, the 'leaf node'. Since our aim is to estimate and compare over all 'life' of the network and also we considered worst case scenario, therefore the simulation of the proposed key distribution algorithms with these assumptions will give a fair idea about the behavior of the algorithms.

5.3 Simulation

For simulation we modify and enhance the Wireless Sensor Network simulator v1.1, an open source tool. The enhancements are done implementing our proposed algorithms in C# and integrating it with the WSN simulator.

5.3.1 Comparison of Total Residual Energy

In Figure 5 we compare residual energy of the WSN while not using and while using key distribution. We observe that Key distribution depletes a good amount of energy of the network. However, as we understand it is due to the fact that it makes the network secure.

5.3.2 Comparison of Total Alive Nodes

Similarly, in Figure 6 we compare no of alive nodes of the WSN while not using or using key distribution.

6. CONCLUSION

In this paper we propose our key management approach for an asymmetric key based security scheme for WSN and analyze our key management schemes. In our scheme, public key of the base station is known only to the authenticated devices of the network. Also, public key of a new node is verified by the one hop key to prevent flooding attack. We also provide two level authentications (by using one hop key, and hook key) before giving a new node the public key of the base station so that any rogue client is barred from joining the network. We also discuss how security planning needs to be done in WSN.

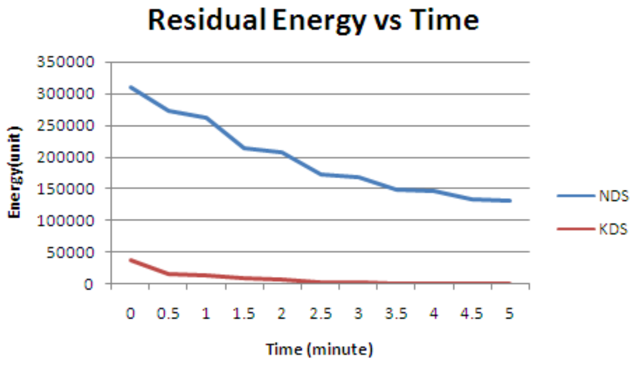

Figure 5 Residual Energy with and without Key distribution

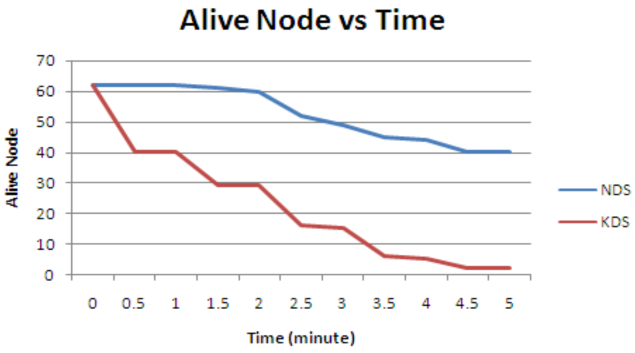

Figure 6 Alive Node with and without key distribution

7. REFERENCES

[1] Amin, F., Jahangir, A. H., and Rasifard, H. Analysis of public-key cryptography for wireless sensor networks security. World Academy of Science, Engineering and Technology 41 (2008).

[2] Bertoni, G., Breveglieri, L., and Venturi, M. Power aware design of an elliptic curve coprocessor for 8 bit platforms. Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW_S06) (2006).

[3] Camtepe, S. A., and Yener, B. Combinatorial design of key distribution mechanisms for wireless sensor networks.

[4] Camtepe, S. A., and Yener, B. Key distribution mechanisms for wireless sensor networks: a survey. Tech. Rep. TR-05-07, Rensselaer Polytechnic Institute, Computer Science Department, Lally 310, 110 8th Street, Troy, NY 12180-3590. 2005.

[5] Chang, J.-H., and Tassiulas, L. Maximum lifetime routing in wireless sensor networks. IEEE/ACM Transactions on Networking (TON) Volume 12, Issue 4 (2004).

[6] D Chorzempa, M. W. Key management for wireless sensor networks in hostile environments. Master's thesis, Virginia Polytechnic Institute and State University, 2006.

[7] Dressler, F. Key management in wireless sensor networks.

[8] Du, W., Wang, R., and Ning, P. An efficient scheme for authenticating public keys in sensor networks. MobiHoc_S05, May 25 U27, Urbana Champaign, Illinois, USA (2005).

[9] Goulet, W. Summarizing pki certificate validation, 2009.

[10] Gupta, V., Wurm, M., Zhu, Y., Millard, M., Fung, S., Gura, N., Eberle, H., and Shantz, S. C. Sizzle: A standards-based end-to-end security architecture for the embedded internet. SMLI TR-2005-145 (2005).

[11] Haque, M. M., Pathan, A.-S. K., Hong, C. S., and Huh, E.-N. Asymmetric key-based security architecture for wireless sensor networks.

[12] Hu, W., Corke, P., Shih, W. C., and Overs, L. secfleck: A public key technology platform for wireless sensor networks.

[13] Jing, Q., Hu, J., and Chen, Z. C4w: An energy efficient public key cryptosystem for large-scale wireless sensor networks. vol. Oct. 2006 Page(s):827 - 832, Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on.

[14] Kalita, H. K., and Kar, A. A new algorithm for end to end security of data in a secure self organized wireless sensor network. Journal of Global Research in Computer Science ISSN 2229-371X 1, 3 (November 2010).

[15] R. Watro, D. Kong, S. fen Cuti, C. Gardiner, C. Lynn, and P. Kruus. Tinypk: Securing sensor networks with public key technology. In 2nd ACM workshop on Security of adhoc and sensor networks SASN, 2004.

[16] X. Du and H.-H. Chen. Security in wireless sensor networks. IEEE Wireless Communications, 2008.