Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 1203–1208

DOI: 10.15439/2015F169
ACSIS, Vol. 5

Trust Security Mechanism for Marine Wireless Sensor Networks

Walid Elgenaidi, Thomas Newe
University of Limerick,
Optical Fibre Sensors Research Centre
Department of Electronic and Computer Engineering
Limerick, Ireland
Email: {walid.elgenaidi, thomas.newe }@ ul.ie }

Abstract—**To provide a strong security service in Wireless Sensor Networks (WSNs), cryptographic mechanisms are required. Generally these security mechanisms demand intensive use of limited resource, such as memory, and energy to provide a defense against attacks. Monitoring the behavior of nodes and detecting risks according to these behaviors, and then taking decisions based on these measurements generally requires the use of a trusted Key Management scheme. In this paper we compare two existing security key management schemes that were designed for use in mobile ad hoc networks: "An overlay approach to data security in ad-hoc networks" authored by Jorg Liebeherr, Guangyu Dong, and "A hierarchical key management scheme for secure group communications in mobile ad hoc networks" authored by Nen-Chung Wang, Shian-Zhang Fang. Then a Hybrid Security Key Management Mechanism designed for use in the marine environment is proposed. This scheme focuses on reducing the memory storage of keys, using a leader node that is responsible for both the node joining and the node revoke processes. This security mechanism is implementing in real time on the Waspmote sensor platform.**

I. Introduction

Design of smart security solutions for wireless sensor networks for specific fields such as marine environments is a big challenge. Since smart security protocols must be designed to have efficient and flexible key distribution systems to prevent attacks while conserving energy [1], [5]. This work aims to provide a secure technique using both Symmetric and Asymmetric key algorithms. Our designed scheme seeks protection against attacks by providing the standard security services such as confidentiality and authentication in addition to addressing the re-keying process between adjacent nodes, as well as reducing the number of stored keys. Therefore the sensor nodes are configured in a point-to-point topology which is suitable for marine coastal WSN systems.

The system proposed in [2] uses public/private keys, pre-message keys and requires that offline signed certificates are stored in each node. Authentication between nodes is performing without coordination with other nodes; this makes trust revocation difficult for 'bad' nodes. Each node maintains a single symmetric key that it shares with its current neighbours in the network topology. Furthermore, a public key is required in the authentication of new neighbours, and every node requires its neighbour's public key for use with the RSA public key algorithm.

The system proposed in [3] offers key management for secure group communications using a two-layer structure. The selected cluster head constructs and transmits a group key to all nodes. This scheme uses symmetric keys for subgroup keys and communication keys. The Diffie–Hellman "DH" key exchange scheme is utilized to achieve secure key transmission between subgroups. These schemes are discussed in section II below.

Section III presents a comparison between the neighbourhood key and hierarchical key management schemes for secure group communications. Section IV proposes a new hybrid key management scheme and section V concludes.

II. Trust Key Management Techniques for Wireless Sensor Networks

There are essential practices for developing a good trust management system for WSNs and for the management of the necessary cryptographic keys [3]. In [2] Jorg Liebeherr, Guangyu Dong presented a key management and encryption scheme, called the neighbourhood key method, that ensures integrity and confidentiality of application data in overlay networks. The neighbourhood key method avoids network wide re-keying operations and payload data re-encrypting at each hop.

A. Updating and Exchanging Neighbourhood Keys

The solutions presented in this scheme [2] are orthogonal to the problem of secure routing, which seeks protection against attacks to routing protocols. Each node has its own certificate and this certificate has been signed by a trusted third party using X.509 Version 3. These certificates, which include secret keys are exchanged between neighbours to use in encrypting or signing messages of

The authors would like to acknowledge the support of the SFI funded MaREI (Marine Renewable Energy Ireland) Centre and Ministry of Higher Education and Scientific Research - Libya

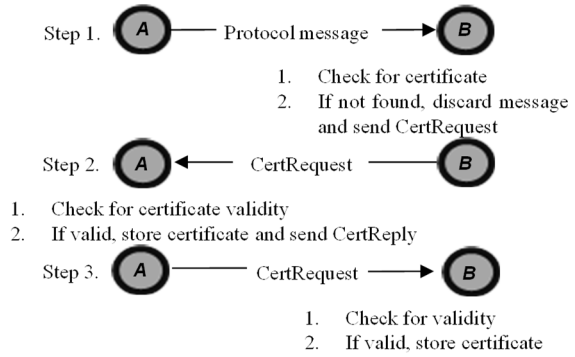

Fig.1. Authentication of nodes

authenticated nodes. Encryption of data and the signing of hashes in each node are done with a single symmetric key called a"neighbourhood key". The neighbourhood key is shared with its current authenticated neighbours in network.

A joining node must generate a new neighbourhood key and send it to all of its authenticated neighbours in order to maintain confidentiality in the network.

Therefore every node must use a public key algorithm to encrypt the new neighbourhood key with the public keys of all the authenticated neighbours that are stored in the node during the authentication process as shown in "Fig. 1,". In this phase, nodes update keys only with current neighbours. However updating and exchanging a new neighbourhood key is executed whenever the set of authenticated neighbours are changed or the specified maximum lifetime of the current neighbourhood key is expired. The security issues are exacerbated during failures in reconstruction of the network topology when one or more nodes join and leave the network at the same time.

The neighbourhood Scheme prevents nodes against a DoS attack from a malicious adversary by implementing an integrity test, and also the allowed frequency of transmitted Key Request messages is limited.

B. Constructing and Transmitting Keys Using Cluster Head

Nen-Chung Wang, and Shian-Zhang Fang [3] introduced a hierarchical key management scheme for secure group communications in a mobile ad hoc network. They proposed a new approach with a two-layer structure whereby a cluster head manages information between nodes in the layers as shown in "Fig. 2". Node with the largest weight value in each level is selected to be a cluster head [4]. The key transmission operation between nodes in the same level subgroup and with nodes in other level subgroups is coordinated by the cluster head.

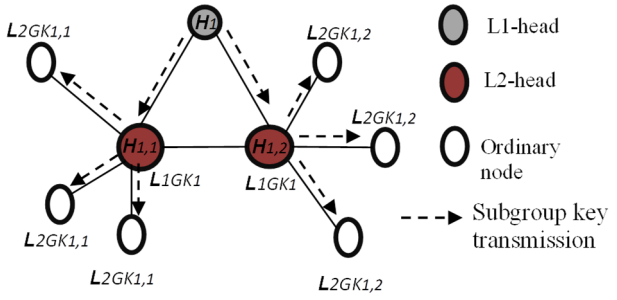

Fig.2. Subgroup key transmission operation between nodes

Level 1 subgroup "L1-subgroup" contains all of the nodes in the subgroup and the level 2 subgroups "L2-subgroup" are selected based on their positions. The node with the largest weight value in every L2-subgroup will be selected as the level 2 cluster head "L2-head" to manage the other nodes of the L2-subgroup.

The Diffie–Hellman "DH" scheme is used for secure transmission between nodes in subgroups, and each subgroup has its own subgroup key. L1-head generates the communication keys that are used between the different subgroups. The encryption and decryption operation during data transmission in different subgroups is only through subgroup keys, which means that packets are transmitted through the cluster heads.

The level 2 cluster head "L2-head" is responsible for a new node joining its subgroup. It initiates the generation of a new subgroup key "K_{LXGK_S}" after a new node joins a subgroup.

A node leaving of subgroup [4] falls under three cases: the leaving of ordinary nodes, the leaving of L2-heads and the leaving of L1-heads. These cases for each scheme are explained in the section III.

III. Security Analysis and Comparison between Neighbourhood and Hierarchical Key Management Schemes

In this section, the neighbourhood key and hierarchical key management scheme for secure group communications are compared under the headings; Security Keys, Node joining process, and Node leaving process.

A. Security Keys

In both schemes the packet during data transmission is encrypted using different cryptographic algorithms depending on the packet function; AES for symmetric encryption, RSA for public key encryption, and Diffie-Hellman scheme to generate keys. Two symmetric keys "128 bits" in the neighbourhood scheme are generated in every node, one is shared with all authenticated neighbours and the other is used to encrypt the payload of the message.

$$\left(\{M\}_{k_s}, \{k_s\}_{k_{nj}}\right) \quad (1)$$

Where "M" is message, "k_s" is source key, and "k_{nj}" is neighbourhood key of node j.

In order to reduce the delay that is incurred by decrypting and re-encrypting the message between forwarded nodes, a node only needs to re-encrypt the source key "k_s" with its own neighborhood key before transmission. However the hierarchical key management scheme uses

three security keys to deliver data between nodes in a network. Symmetric subgroup keys "K_{LXGK_S}" are used for transmission between all nodes that fall under the L1-head subgroup. Additionally, secure data delivery in different subgroups is achieved through symmetric communication keys "k_c" and "k_{DH}" which only belongs to the source node and the destination node. "k_{DH}" is used for the first encrypted packet transmitted.

Example: Assume node A in subgroup X would like to send data to node C in subgroup Y:
In node A:

$$\{\mathrm{M}\}_{\mathrm{K_{DH}}} \quad (2)$$

Packet will encrypt and decrypt with "K_{LXGK_S}" when transmitted through nodes in the same subgroup. At the first forwarding node in the subgroup Y:

$$\{\{\mathrm{M}\}_{\mathrm{K_{DH}}}\}_{\mathrm{K_C}} \quad (3)$$

After receiving the packet, the first forwarding node in the subgroup Y will decrypt the packet with "k_C", then encrypt the decrypted packet with "K_{LYGK_S}" and then send the packet to node C

$$\{\{\mathrm{M}\}_{\mathrm{K_{DH}}}\}_{\mathrm{K_{LYGK_S}}} \quad (4)$$

Where M is message, "k_{DH}" is the Diffie-Hellman generated key, and "K_{LYGK_S}" is the symmetric, Y subgroup key.

The decryption and encryption steps are repeated until the destination node receives this packet.

All nodes in a subgroup have their own public and private keys. In case of any change in subgroup members, the L1-head will encrypt the regenerated Symmetric subgroup keys "K_{LXGK_S}" with the public key of each node before sending it to the nodes via the L2-heads in its subgroup.

B. Node Joining Process

In the neighbourhood key method the authentication process relies on public key certificate that are signed by an offline trusted third party [7][8]. Also nodes can perform authentication with new nodes independently without any coordinate from any other nodes. A new node sends a join request including its own signed certificate to existing nodes. Certificates between nodes will be exchanged after the received node has verified the certificate of the new node. Once the certificates are exchanged, the nodes will exchange symmetric neighbourhood keys using the RSA algorithm. Whenever a node receives request messages from a node for the first time it must update its neighbourhood key store.

Since rebuilding and redistributing of a new neighbourhood key to all nodes is required each time a node joins or leaves the network, the network may take a long time to stabilise. This issue will worsen when many nodes join and leave the network at the same time.

The benefit of the hierarchical key management scheme is mainly based on its hierarchical structure. When a new node joins a subgroup, rekeying is not a global operation. The L2-subgroup head just regenerates the L2-subgroup key "K_{LXGK_S}" for this subgroup, which can be relatively few nodes.

C. Node leaving process

When an authenticated neighbour has not sent a message for a long time in the neighbourhood key scheme it is assumed to have left the network and a new neighbourhood key must be generated and transmitted to its authenticated neighbours. Whereas in the hierarchical key management scheme the leaving of a node falls under three scenarios;

- For the leaving of an ordinary node, the level 2 cluster head regenerates the L2-subgroup key.
- In the case of the L2-head leaving the subgroup. The node with the largest weight value of the remaining ordinary nodes in the subgroup will be selected to be the new L2-head.
- The third case is the leaving of L1-heads; L2-head with the largest weight value of the L2-heads in the subgroup will selected to be the new L1-head.

IV. Proposed Trust Security Mechanism for Marine Wireless Sensor Networks

This section outlines the proposed smart security technique for Wireless Sensor Network nodes suitable for use in marine coastal environments. The scheme addresses issues highlighted above in [2] and [3] such as the rekeying process and the number of stored keys required in each node. This scheme uses the advanced encryption standard "AES-128" and the public key cryptosystems "RSA-1024" to allow the secure transmission of data between nodes in the network. The pre-distribution of keys is currently been used is the scheme, whereby keys are allocated to all sensor nodes before deployment and securely transferred between nodes using a master key.

A. General Outline of the Scheme

Each node keeps its own symmetric key, called an adjacent key "k_{ni}" that it shares only with its two neighbours in the network "Fig. 3". Also each node must have

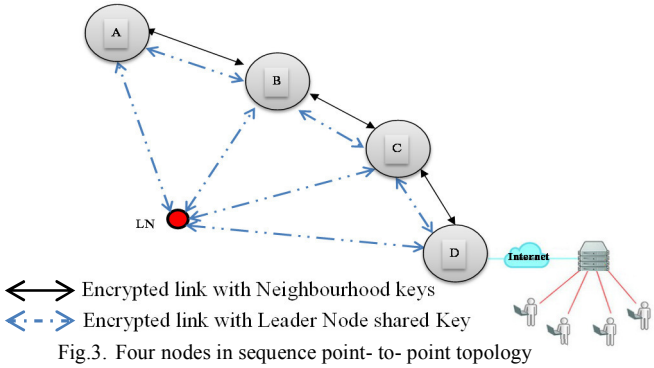

Fig.3. Four nodes in sequence point- to- point topology

a symmetric master key called the leader node key"k_{LN}" that is generated by the master node called Leader Node. The RSA algorithm is used for key management in updating the leader node key.

Each node performs authentication and revocation in coordination with a leader node. In the authentication process a new joining node simply sends a request-to-joint command to any ordinary node included with its Identification. Then the ordinary node sends the received Identification to the leader node. The leader node will verify the Identification based on stored certificates in a trust database of its network members. Once a new node is authenticated from the leader node, adjacent keys are securely exchanged between nodes. These keys are encrypted with the leader node key"k_{LN}".

One of the most important aspects of our security mechanism is the process of revocation and rebuilding the network topology when a node leaves. The leader node monitors the behaviour of all nodes in the network through broadcasting a 'hello' message, and all nodes must reply with a response message. If any node does not respond to the 'hello' message, the leader will revoke this node and rebuild the network via one of its authenticated neighbours.

As already mentioned, the leader node uses the public key of the authenticated neighbours to securely share a new symmetric leader node key and the identification of the revoked node.

B. *Key Maintenance and Revocation Process in Proposed Technique*

When a node leaves the network, it should not be able to decrypt the future encrypted traffic [6]. The leader node monitors all nodes' activities continuously in the network and every node maintains contact with the leader node. In case of any node not responding the leader node will remove this node from its member list and reconnect its neighbours to keeps the network functioning as shows in "Fig. 4".

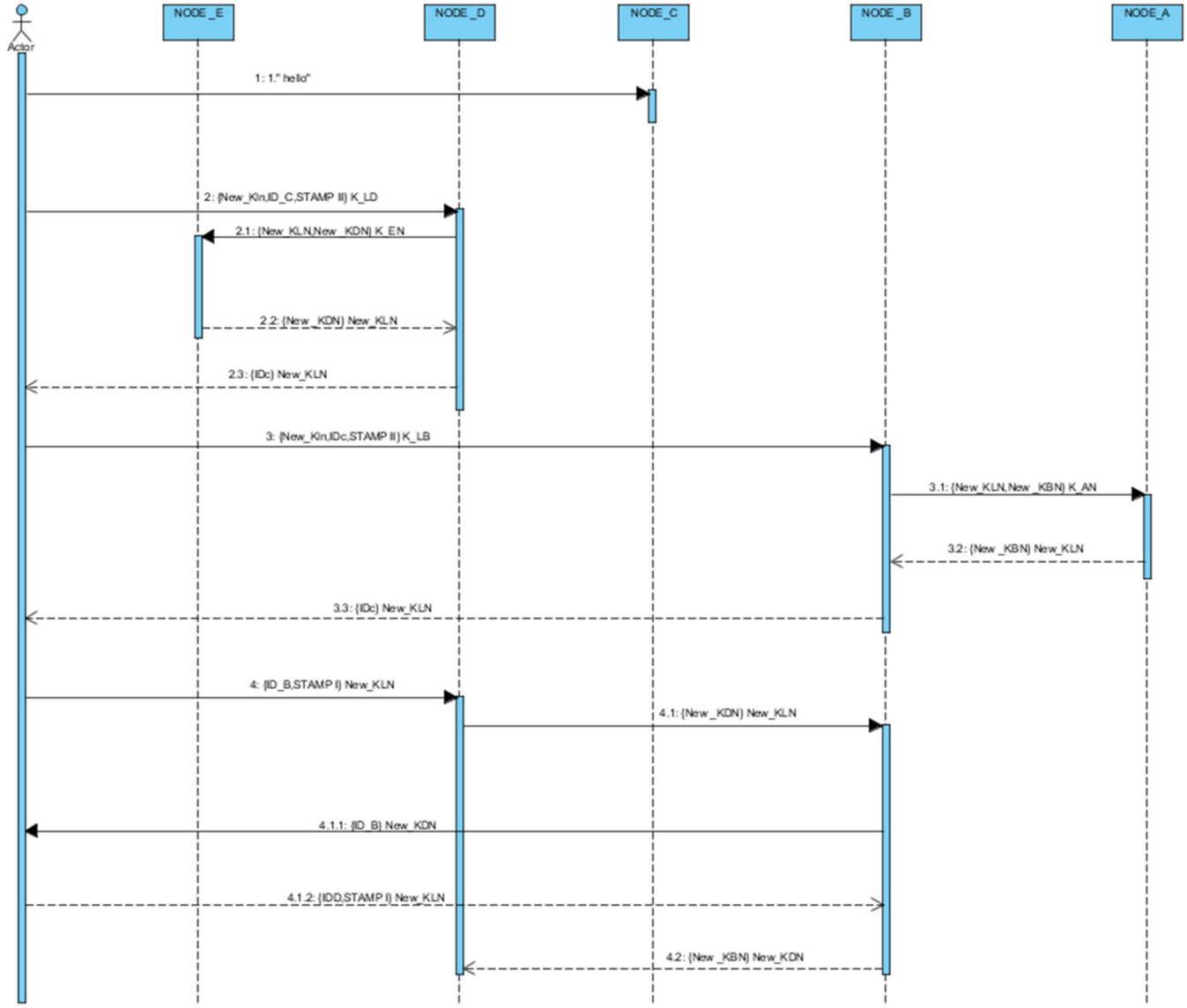

Fig.4. Key maintenance and revocation process: sequence diagram

In "Fig. 4" assume that node C does not response to the 'hello' message and nodes B and D are its two neighbouring nodes.

Notation Used:

New_K_{LN}: New shared secret key between Leader and nodes "128bits".

IDx: A unique Identification of node x.

$STAMP_{II}$: Part of message means the included ID is revoked.

New_K_{DN}: New shared secret key between D and its neighbours "128bits".

New_K_{BN}: New shared secret key between B and its neighbours "128bits".

$STAMP_{I}$: Part of message means the included ID is authenticated.

K_D: Public Key of D.

K_B: Public Key of B.

Messages Exchanged:

1: "Hello"

2: $\{New\_K_{LN}, ID_C, \text{STAMP}_{II}\}_{K_D}$.

2.1: $\{New\_K_{LN}, New\_K_{DN}\}_{K_{EN}}$.

2.2: $\{New\_K_{LN}\}_{NEW\_K_{DN}}$.

2.3: $\{ID_C\}_{NEW\_K_{LN}}$.

3: $\{New\_K_{LN}, ID_C, \text{STAMP}_{II}\}_{K_B}$.

3.1: $\{New\_K_{LN}, New\_K_{BN}\}_{K_{AN}}$.

3.2: $\{New\_K_{LN}\}_{NEW\_K_{BN}}$.

3.3: $\{ID_C\}_{NEW\_K_{LN}}$.

4: $\{ID_B, \text{STAMP}_I\}_{NEW\_K_{LN}}$.

4.1: $\{New\_K_{DN}\}_{NEW\_K_{LN}}$.

4.1.1: $\{ID_D\}_{NEW\_K_{LN}}$.

4.1.2: $\{ID_D, \text{STAMP}_I\}_{NEW\_K_{LN}}$

4.2: $\{New\_K_{BN}\}_{NEW\_K_{DN}}$.

C. *Description of key maintenance:*

After the relationships among the nodes are re-established, all nodes must send a response to the 'hello' message to the leader node. The leader node knows the certificates of all the nodes in the network. Assuming that node C does not response to the 'hello' message, then the LN must remove it from the network. Due to the point-to-point topology the nodes that are positioned before and after the revoked node C must be securely reconnected. The scenario for the revocation of node C "Fig. 4" is described below.

Step 1: Node C does not response to 'hello' message.

Step 2: After the LN has verified all node responses, the leader node will send an encrypted message to node D. This message includes a new leader node master key "New_K_{LN}" and the Identification of the revoked node C. Node D will update and share its adjacent key "New_K_{DN}" and new leader node key "New_K_{LN}" with authenticated node E, its neighbour. Node D will confirm this step by sending Identification of the revoked node "C" encrypted with" New_K_{LN}" to leader node.

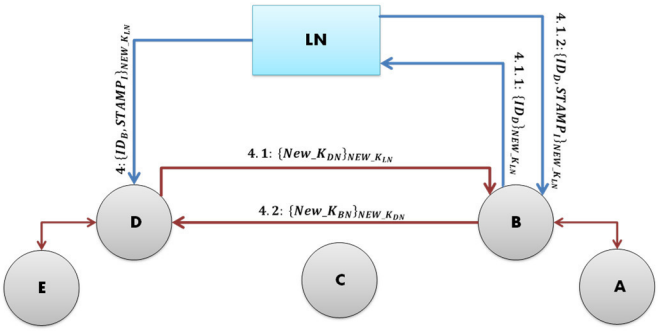

Fig.5. Rebuilding the network topology after node C revocation

Step 3: After receiving the revoked message of node C, node B will update its adjacent key and share both "New_K_{BN}" and" New_K_{LN}" with its authenticated node A.

Step 4: leader node will coordinate the authentication process between B and D as shown in "Fig. 5". Initially, leader node will send the Identification of B and $STAMP_I$ to node D in order to reconfigure the network. Then nodes D and B will mutually exchange their shared keys using the master key" New_K_{LN}".

D. *Advantages*

This approach provides a number of advantages in comparison with [2] and [3]. Firstly, each node in [2] performs authentication independent of and without coordination with other nodes. An exchange and verification of certificates between neighbours in the network occurs only when needed. However management of the authentication process in [3] is occurring via the L2-head and the L1-head.

In our scheme the authentication process is coordinated by the leader node LN. Ordinary nodes store only information of their neighbours which leads to reducing the number of keys stored in every node and also the security risks involved in storing large number of network keys. Furthermore, when a new node joins the network, authenticated nodes do not need to regenerate their keys if they are not a neighbour of the new node. After the new node is verified by LN, it will exchange adjacent keys with its neighbours "Fig. 5". This increases the life of the node as well as lifespan of the entire WSN as it only communicates with its closest nodes. The second advantage is that the encryption and decryption operation during data transmission in [3] is occurring through" K_{DH}", subgroup keys and communication keys. Therefore, it has a longer transmission time than our scheme, which encrypts and decrypts only the part of the source key when a message is forwarded. The third and important advantage of this

security mechanism is updating the shared key, where only current neighbours of a revoked node will regenerate their symmetric keys. The leader node will distribute a new master key through the trustworthy nodes. In the network, ordinary nodes are deployed in a line topology, and the distance between every two neighbours is around 1500m. In order to cover a range of up to 7000m, in this scheme, we have used XBee-802.15.4-Pro/2.4GHz integrated with Waspmote. This advantage will lead network to securely reconnect in case of three neighbour nodes are revoked.

V. Conclusion

The overall objective of this work is to design a smart security technique for Wireless Sensor Network nodes that can successfully operate in marine coastal environments. We address some potential drawbacks of two existing key management schemes that would be considered suitable and combine their advantages. These protocols used symmetric-key and public-key based key transport protocols for the provision of authentication between nodes. However, both schemes require updating all shared keys whenever the membership in the network changes. The time required to build and distribute new keys will lengthen the time it takes to establish a stable topology in comparison with our proposed scheme which restricts key update to the neighbours of the leaving node. An implementation of the technique is currently being performed on the Waspmote sensor platform and it is hoped that some measurements will be available for the conference presentation.

VI. References

[1] S. Babu, A. Raha, and M. Naskar, "Trust Evaluation Based on Node's Characteristics and Neighbouring Nodes' Recommendations for WSN" Wireless Sensor Network. vol. 6, August 2014, pp. 157-172, http://dx.doi.org/10.4236/wsn.2014.68016.

[2] J. Liebeherr, and G. Dong, "An overlay approach to data security in ad-hoc networks", Ad Hoc Networks, Elsevier, 5th July 2006, pp.1055-1072, http://dx.doi.org/10.1016/j.adhoc.2006.05.017.

[3] V. Wang, and S. Fang," A hierarchical key management scheme for secure group communications in mobile ad hoc networks",Ad Hoc Networks, Elsevier, 23 January 2007, pp. 1667-1677, doi:10.1016/j.jss.2006.12.564.

[4] S. Dhurandher, and G .Singh,. 2005. "Weight based adaptive clustering in wireless ad hoc networks". IEEE Personal Wireless Communications, New Delhi, India, Jan. 2005pp. 95 – 100, doi: 10.1109/ICPWC.2005.1431309.

[5] J. Lopez, R. Roman, I. Agudo, and C. Fernandez-Gago, "Trust Management Systems for Wireless Sensor Net- works: Best Practices". Computer Communications, Elsevier vol. 33, June 2010, pp. 1086-1093, doi: doi:10.1016/j.comcom.2010.02.006.

[6] K. Chauhan, and S. Amit, Singh, "Securing Mobile Ad hoc Networks: Key Management and Routing," AdHoc Networking Systems, (IJANS). vol. 2, April 2012.

[7] Y. C. Hu, and A. Perrig, "A survey of secure wireless ad hoc routing," IEEE Security and Privacy, vol. 2, pp. 28- 39, March 2004, doi: http://doi.ieeecomputersociety.org/10.1109/MSP.2004.1.

[8] Y. Sun, Z. Han, and Liu, K.J.R. "Defence of Trust Management Vulnerabilities in Distributed Networks". IEEE Communications Magazine, vol. 46, February 2008, pp. 112-119, doi: 10.1109/MCOM.2008.4473092.

[9] X.B. Zhang, S.S. Lam, H. Liu, "Efficient group rekeying using application-layer multicast" IEEE Distributed Computing Systems, June 2005, pp. 303- 313, doi: 10.1109/ICDCS.2005.27.