A Scheme for Key Revocation in Wireless Sensor Networks

Subhankar Chattopadhyay & Ashok Kumar Turuk

Dept. of Computer Science and Engg., National Institute of Technology Rourkela, Rourkela India-769008
E-mail: subho.atg@gmail.com & akturuk@nitrkl.ac.in

Abstract **- Key revocation is a challenging task in wireless sensor networks. There can be many nodes in a sensor network sharing the same key(s) in their key ring. If any of those nodes in the network gets compromised then the keys of those nodes having same keys will also be revealed. Nodes sharing keys with the compromised node should revoke the shared keys as soon as possible. In this work we proposed key revocation scheme based on voting procedure. We have shown that all the keys of a compromised node can be successfully revoked from the entire network.**

Index Terms**—Wireless Sensor Networks, Key revocation, Secret sharing, Sensor node..**

I. Introduction

Wireless sensor network has a wide variety of applications in military and civilian areas. Sensor nodes are deployed in a battlefield to detect enemy intrusion. They are used to measure temperature, heat, sound, pressure, magnetic and seismic fields. They are also used in industry for machine health monitoring, waste water monitoring etc. In certain applications such as in military, the communication needs to be secure. A shared key among the sensor nodes is required for secured communication. Cryptographic key management in sensor networks is an uphill task. Key pre-distribution is regarded as a promising key distribution procedure in sensor network because both the symmetric and public key cryptography are difficult to implement in sensor networks. A node in a sensor network may get compromised. As a result, the keys stored in the node gets revealed to the adversary affecting the secured communication. These keys are also shared by many nodes in the network. Unless the compromised keys are revoked from all the nodes in the network, the secure communication is affected. Key revocation is one of the important phases in key management. In key revocation, compromised keys are removed from all the nodes in the network. Key revocation may be distributed or centralized. In centralized key revocation the base station decides about the compromised nodes and only the base station participates in the key revocation process. In distributed process, nodes co-operate among themselves to detect a compromised node and participate in the key revocation process.

Centralized key revocation has a single point of failure. Distributed key revocations are faster but difficult in implementation. Nodes that participate in the key revocation process itself may be a malicious node.

In this paper we proposed a distributed key revocation algorithm. Our proposed scheme is an improvement over Chan et. al. [1] [2]. Rest of the paper is organized as follows. In Section II we discuss few key revocation schemes. Our proposed scheme resembles with that of Chan et. al. [1] [2] whose shortcomings are discussed in Section III. The network model and assumptions is discussed in Section IV. Proposed key revocation scheme is presented in section V. Analysis of the proposed scheme is done in Section VI. Finally few conclusions are drawn in Section VII.

II. Related works

Key revocation problem was first addressed by Eschenaur and Gligor [3]. They proposed a centralized approach to key revocation in which a controller node broadcast a message to each node in the network informing about the compromised keys. In their scheme a signature key is sent to each node prior to the broadcast message. A distributed approach to key revocation was first proposed by Chan et. al. in [2] and which was extended by them in [1]. Their proposed scheme can revoke the compromised node but may not fully revoke the compromised keys from the network.

Wang et. al. [4] proposed a key updating technique in which the compromised keys are made obsolete. In their scheme, a session key is broadcast at the beginning of each session to update the keys in each node so that compromised nodes do not get this session key and hence their keys becomes obsolete. Park et. al. [5] proposed the idea of dynamic session to reduce the life time of a compromised node in the network. Moore et al [6] proposed a suicide strategy to revoke compromised nodes in the network.

In their strategy, in order to revoke a compromised node, a legitimate node has to die. This incurs an overhead.

III. Problems with Chan et. Al. [1], [2] key Revocation mechanism

In this section we discuss the problems associated with Chan et. al.'s [1], [2] scheme.

1) It is possible to remove a compromised node from the network, however it may not be possible to remove all its keys from the network. We give a scenario to support our claim. Let us consider two nodes, u and v, sharing a common key, say k_1. Suppose they are deployed far away from each other in the network such that they are not in the communication range of each other. Let there exists a node w which is in the communication range of v but not in the communication range of u and they share the common key, k_1. Nodes v and w can discover the common key k_1 between them and can communicate via this key k1. Suppose node u gets captured then the key k_1 gets revealed. The neighbors of u do not share the key k_1 with u. Therefore, they are unaware of the fact that the key k_1 has been compromised. Hence, v and w will not be informed. Now the adversary can use the key k_1 and decrypt all the messages between v and w. In the preset scenario compromised keys are not removed completely from the network in Chan et. al. scheme, compromising the network security.

2) Sybil attack [7] is possible in their proposed scheme. A compromised node removed from the network is known only to the neighbor. Rest of the network is not aware of the node that is revoked from the network. Therefore, a clone can be deployed elsewhere in the network; resulting in a Sybil attack.

3) Path keys established through the compromised node is not revoked.

4) Each node has to store votes for all its neighbors before the deployment. For this to happen we need to know the network topology before the deployment which is not always possible.

IV. Network model and assumptions

In this section we discuss the network model and assumptions considered for the proposed key revocation scheme.

We assume the network is divided into hexagonal regions such that there will not be more than k number of nodes in each region. Each region have a unique id $< i, j >$ where i and j are the row and column of that region. Regions are further divided into two types: basic region and non-basic region. A region $< i, j >$ is called a basic region *if* $[i\%2 = 0 \;\&\&\; j\%2 = 0 \;\&\&\; i\%4 \neq 0] \,\|\, [i\%4 = 0 \;\&\&\; j\%2 = 1]$. Other regions are called non-basic regions. A non-basic regions will have atmost two basic regions in their neighborhood. Basic regions have a single unique trivariate polynomial, whereas a non-basic region will have atmost two trivariate polynomials assigned to them drawn from each of its neighborhood basic region. $f_1(x, y, z), f_2(x, y, z), ...f_n(x, y, z)$ are trivariate polynomials where highest degree for x, y, z will be $7k$. Figure-4.1 shows the division of network into two types of regions.

Each node will have a share of the polynomial(s) of it's region along with a unique hash function h(). If $f_1(x, y, z)$ is the polynomial for a region $< i, j >$, then for a node with node id u, the polynomial shares $f_1(u, y, z)$ is called authentication polynomial, $auth^1_u(y, z)$ and $f_1(x, u, z)$ is called the verification polynomial, $verf^1_u(x, z)$ for the node u.

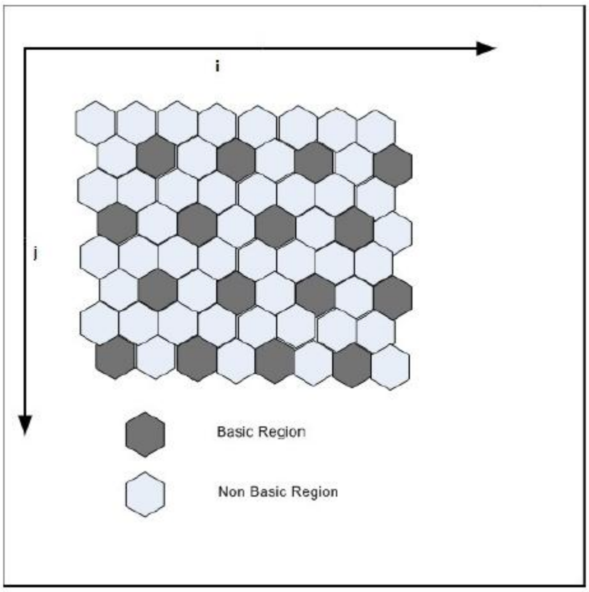


Fig. 1. Division of network into basic and non-basic region

We made the following assumptions :

1) Nodes have a unique identifier.

2) Each node have more than t numbers of neighborhood where t is the minimum number of neighbors who must agree to revoke a node.

3) Number of compromised node in a node's neighborhood is less than t.

4) Base station is not prone to compromise.

5) Nodes have built in intrusion detection system.

6) Each node maintains a two-hop neighborhood information.

7) Each node stores all the intermediary nodes for each path key formed.

8) The diameter of a region is greater than 2r where r is the radius of any node's communication range.

9) Each node maintains two lists; a Blacklist which contains the list of compromised nodes and a Suspected list which contains the list of suspected neighbor nodes along with the accuser.

Lemma 1 : *If two nodes are neighbors of a same node then they will be situated either in the same or in the neighboring regions.*

Proof : When two nodes are in the communication range of each other then they are in the neighborhood of each other. To become the neighbor of a node, distance between the two nodes should be less than or equal to r where r is the radius of node's communication range. Two nodes which are at a distance greater than r from each other can not be neighbor of each other. We have assumed that the diameter of a region to be greater that 2r. Therefore, two nodes which are situated in non-neighboring regions, the distance between them will always be greater than 2r. Hence, those two nodes can never be neighbors of a same node. Therefore we can conclude that nodes which are neighbors of a same node lies either in same region or in the neighboring regions.

V. PROPOSED SCHEME

In this section we proposed two revocation schemes. First scheme is described in Subsection-4.3.1 and second in Subsection-4.3.2.

A. *Scheme I*

The proposed scheme consists of four phases. They are : Setup, Voting, Revocation and Removal. Action taken in each phase is described below.

1) Setup : In this phase, the network is divided into regions. The Blacklist and Suspected list at each node is set to empty.

2) *Voting :* In this phase a node vote against its suspected neighbors. If a node u wants to vote against one of its neighbor node, say, v, then u will prepare a message M containing the node id of the victim node v. This message is sent to all the neighbors of victim node v, and to the *base station*. Let w be one of the neighbors of v. Then according to Lemma 1, the nodes u and w are either from the same region or from the neighboring regions. Therefore, they have share of a unique polynomial. Let the polynomial be $f_i(x, y, z)$. Then u have $auth_u^i(y, z) = f_i(u, y, z)$ and w have $verf_w^i(x, z) = f_i(x, w, z)$. The node u will send the message M along with a single value $p = auth_u^i(w, h(M))$, *i.e.*, $p = auth_u^i(y, z)$ at $y = w$ and $z = h(M)$. After receiving the message, the node w will check the message's authentication. It will compute the value $q = verf_w^i(u, h(M))$, *i.e.*, $q = verf_w^i(x, z)$ at $x = u$ and $z = h(M)$. If $q = p$ then the message is authenticated. Node w updates its Suspected list by inserting the id of node v into the Suspected list if the list does not contain the id of node v. The name of the accuser that is node u in the present scenario is also inserted into the list.

Each node maintains a counter on the number of votes registered against each of its neighbors. A node for which the number of votes registered against it crosses the threshold parameter, t, then that node is put under Blacklist.

When a node x puts a node y in the Blacklist, it performs the following actions:

a) Stop communicating with y.

b) Delete all the keys it shares with y.

c) Delete all the path keys formed through y.

3) *Revocation :* The *base station*, on receiving t number of votes against a node, will prepare a key revocation message containing the node id of the compromised node and the compromised keys. Then it will broadcast this message, along with the authentication polynomials corresponding to polynomials assigned to each region. The *base station* will broadcast $\sum f_i(base.id, y, h(M_1))$ where $base.id$ is the id of the base station, and $f_i(x, y, z)$ is the set of all the trivariate polynomials for $1 \leq i \leq n$ where n is the number of basic regions.

4) *Removal :* After receiving the key revocation message from the *base station*, a node will first check the message authenticity to ensure that it has come from the *base station*. A node l with region's polynomial $f_k(x, y, z)$ will compute $f_k(base.id, y, h(M_1))$ where $y = l$. It will also compute its own verification polynomial $verf_l^k(x, z)$ where $x = base.id$ and $z = h(M_1)$. If the above two values are equal, then the message is authenticated. Then the node l will delete all the keys mentioned in the message and put the victim node v in the Blacklist. Since, the list of compromised nodes exists in each node, this can prevent sybil attack and node replication attack.

B. *Scheme II*

This scheme also consists of four phases like Scheme-I.

This differs from Scheme-I in the presence of monitor nodes in each region. Monitor nodes are more secured than the normal nodes and communicate directly among themselves. Actions performed at each step is explained below :

1) Setup : Action in this phase remain same as that in Scheme-I.

2) Voting : The mechanism of voting remains same as that in Scheme I. However, the vote is sent to the monitor node of the accuser's region. On receiving a vote, monitor node checks whether the suspected node belongs to its region or not. If not, then it will send this voting information to the monitor node of victims region. The monitor node of victim's region will update its Suspected list. When the number of votes reaches a threshold parameter t registered against a node, then the corresponding monitor node will inform other monitor nodes about the compromised node along with the keys that has been compromised.

3) Revocation : In this phase, the monitor node prepare a message containing the node id of the compromised node and the compromised keys. This message is sent to all the nodes in the monitor node's regions along with an authentication value for each node.

   For example, if the monitor node is m, sending an authentication message M_3 to node d then it will send an authentication value $f_k(m; d; h(M_3))$ where $f_k(x; y; z)$ is a tri-variate polynomial corresponding to the region of m.

4) Removal : After receiving key revocation message from their corresponding monitor node, a node checks its authenticity to ensure that it has come from its monitor node. If the node is l1 then it will compute its own verification polynomial *ver* f_l (x; z) where $x = m$ and $z = h(M_3)$. If this value is equal to the authentication value sent by the monitor node m, then the message is authenticated. Then the node, l, deletes all the keys contained in the message and put the accused node in the Blacklist.

VI. Analysis of the Proposed Scheme

Our proposed scheme is an improvement over Chan et. al. scheme. It differs from their scheme in the following ways :

1) We have divided the network into hexagonal regions.

2) The idea of sessions is not used in our proposed scheme.

3) We have used trivariate polynomials whereas their voting procedure was based on secret sharing of bivariate polynomial.

4) We have used the concept of monitor node was also not present in their scheme.

In this section we analyze the proposed key revocation mechanism.

1) compromised nodes can not collude and revoke a node as there can not be more than t numbers of nodes in the neighborhood of a node.

2) To compute a trivariate polynomial, the adversary has to capture all the nodes having a share of that polynomial. Forgery of a vote is not possible as there will not be more than k number of nodes in each region.

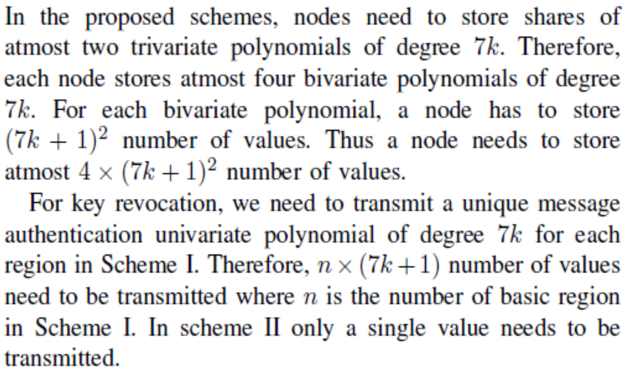
3) Votes are verified so that no false voting results in to revocation of a legitimate node. Neighbor nodes will not be able to authenticate the false vote by an adversary. Therefore, no revocation or updation of Suspected list will occur.

4) A listener can not replay a vote to generate additional votes.

5) The neighbor nodes do not broadcast the revocationmessage to the entire network. Thus, it is not vulnerable to denial of service attack.

6) As all the path keys constructed by the compromised nodes are removed after the revocation, the adversary can not affect the network computing the path keys at later stage.

7) The proposed scheme is resistant to Sybil attack or any other kind of replication attack.



In the proposed schemes, nodes need to store shares of atmost two trivariate polynomials of degree $7k$. Therefore, each node stores atmost four bivariate polynomials of degree $7k$. For each bivariate polynomial, a node has to store $(7k + 1)^2$ number of values. Thus a node needs to store atmost $4 \times (7k + 1)^2$ number of values.

For key revocation, we need to transmit a unique message authentication univariate polynomial of degree $7k$ for each region in Scheme I. Therefore, $n \times (7k + 1)$ number of values need to be transmitted where n is the number of basic region in Scheme I. In scheme II only a single value needs to be transmitted.

Next, we calculate the computation needed for authentication of key revocation. In Scheme I, during verification, each node gets a univariate polynomial $f(y)$ of degree $7k$. Then, they compute the value of the polynomial $f(y)$ at y = node *id*. This computation needs $(7k + (7k-1) + (7k-1) + ... + 1)$ number of multiplications = $7k(7k+1)/2$ number of multiplications and $7k$ number of addition. In Scheme II only a single value is transmitted, hence this computation is not required. Next step of verification is similar for both the schemes. Each node needs to compute its verification polynomial. Verification polynomial is a bivariate polynomial of degree $7k+1$ and it is of the form $x^m(a_1y^m + a_2y^{m-1} + a_3y^{m-2} + ... + a_{m+1}) + x^{m-1}(b_1y^m + b_2y^{m-1} + b_3y^{m-2} + ... + b_{m+1}) + ...$ where $m = 7k+1$.

Each term within the bracket is a univariate polynomial which needs $7k(7k+1)/2$ multiplications and $7k$ additions. Each of these values again needs to be multiplied by x of different power. Total number of multiplications is $= 7k(7k+1)/2 \times (7k+1) + 7k(7k+1)/2$
$= 7k(7k+1)(7k+2)/2$

Number of additions is $= 7k(7k+1) + 7k$
$= 7k(7k+2)$

We have shown the comparison of two of our schemes in Table 4.1.

VII. Conclusion

We have proposed two key revocation model for wireless sensor network in this paper. In the Scheme I we overcome the problems of the existing algorithm and Scheme II have been introduced in order to further reduce the communication cost of Scheme I. In future, any other mechanism can be used for voting technique so further reduce the storage cost and time to revoke a compromised node. Also future improvements can be made in terms of reducing the computational and communication cost.

TABLE I
COMPARISON BETWEEN KEY REVOCATION SCHEME I AND SCHEME II

	Scheme I	Scheme II
Storage Requirement in each node	$(7k+1)^2$	$(7k+1)^2$
Number of values to be transmitted for authentication	$(7k+1)\times$ number of regions	1

Message authentication cost	polynomial computation:
	multiplications=$\frac{7k(7k+1)}{2}$
	additions=$7k$
	polynomial verification:
	multiplications=$\frac{7k(7k+1)(7k+2)}{2}$
	additions=$7k(7k+2)$
		polynomial computation:
		multiplications=0
		additions=0
		polynomial verification:
		multiplications=$\frac{7k(7k+1)(7k+2)}{2}$
		additions=$7k(7k+2)$

References

[1] Haowen Chan, Virgil D. Gligor, Adrian Perrig, and Gautam Muralidharan. On the distribution and revocation of cryptographic keys in sensor networks. IEEE Trans. Dependable Sec. Comput., 2(3):233–247, 2005.

[2] Haowen Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy, page 197, Washington, DC, USA, 2003. IEEE Computer Society.

[3] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In CCS '02: Proceedings of the 9th ACM conference on Computer and communications security, pages 41–47, New York, NY, USA, 2002. ACM.

[4] Yong Wang, Byrav Ramamurthy, and Xukai Zou. Keyrev: An efficient key revocation scheme for wireless sensor networks. In ICC, pages 1260–1265. IEEE, 2007.

[5] Chul-Hyun Park, Yi-Ying Zhang, In-Tai Kim, and Myong-Soon Park. Dls: Dynamic level session key revocation protocol for wireless sensor networks. In Information Science and Applications (ICISA), 2010 International Conference on, pages 1–8, 2010.

[6] Tyler Moore, Jolyon Clulow, Shishir Nagaraja, and Ross Anderson. New strategies for revocation in ad-hoc networks. In Frank Stajano, Catherine Meadows, Srdjan Capkun, and Tyler Moore, editors, ESAS, volume 4572 of Lecture Notes in Computer Science, pages 232–246. Springer, 2007.

[7] John R. Douceur. The sybil attack. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, IPTPS, volume 2429 of Lecture Notes in Computer Science, pages 251–260. Springer, 2002.

❖❖❖