Distributed Compromised Nodes Detection Scheme at First Stage for SurvSec Security Architecture

Mohamed Helmy Megahed
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada
mmega080@uottawa.ca

Dimitrios Makrakis
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada
dimitris@site.uottawa.ca

Hisham Dahshan
Communications Department
Military Technical College
Egyptian Armed Forces
Cairo, Egypt
hishamdahshan@yahoo.com

***Abstract*—SurvSec is a novel security architecture for reliable network recovery from base station BS failure of surveillance Wireless Sensor Network WSN in hostile environment. Compromised nodes detection is a very important security mechanism in surveillance WSN to detect compromised nodes before they destroy the security of the WSN. Node compromise attack is a multi-stage attack which consists of three stages: physically capturing and compromising sensor nodes; redeploying the compromised nodes back to network and compromised nodes rejoining the network. Only two protocols detect compromised nodes at first stage. The first protocol can be easily broken by targeting couple of nodes at the same time and the second protocol has high overheads and it is based on the distribution of one key list for all nodes which is not secure if one node is compromised. In this paper, a new compromised nodes detection algorithm that detects compromised nodes at first stage for SurvSec security architecture was proposed. The proposed scheme was based on four algorithms. First algorithm provided the network with key management. Second algorithm provided the network with secure localization. Third algorithm provided the network with secure clustering. Fourth algorithm built overlapped groups from clusters. Each cluster has a security manager (SM) and backup security manager (BKSM) to manage security issues. From the locations of nodes in the cluster, the nodes can form a group by sending and receiving from their right and left neighbours in the cluster. Each group forms overlapped group with its neighbour groups. The groups resemble interconnected rings in a chain and if attackers capture one group in the chain, the chain will be cut and its overlapped groups will discover the compromised group. Each node in the cluster sends an encrypted "Hello" message to its neighbours in the cluster every 15 seconds. If a node does not respond to the "Hello" message, this means it is compromised and its neighbours will send to the SM that the node is compromised and if the SM is compromised, its neighbours will send to the BKSM that the SM is compromised then to BS. The proposed protocol was designed to be resistant against large number of compromised nodes by collaborative work of attackers. Extensive simulation results were given to demonstrate the high detection rate of the proposed scheme besides the low overheads with high security level for the protocol.**

Keywords-Overlapped Groups; ***Node Compromise Attack***; ***First Stage***.

I. INTRODUCTION

A node compromise attack is a three stage attack. In the first stage, the attacker captures some sensor nodes from the network and then compromises these nodes. In the second stage, these compromised nodes are redeployed into the network. In the third stage, the attacker will use these compromised nodes to launch various security attacks. Much work has tackled the node compromise attack [1, 2]. However, all of them address the node compromise attack either in the second stage based on node redeployment detection [2] or in the third stage based on node misbehavior detection [3-6]. We believe that group of attackers will launch node compromise attack to jeopardize the whole network in few minutes. Therefore, early detection of node compromise attack can lead to high security level.

Several protocols have been proposed for detecting compromised nodes at the second and third stage. Some protocols rely on the assumption that compromised node will change its location or its signal strength will alter after it is compromised.

Xiaodong [7] made the first attempt to detect node compromise in the first stage. He described a new couple based compromised node detection protocol to build couples of sensor nodes in ad-hoc pattern to detect node compromise attack at the first stage. The nodes within the same couple can monitor each other. This protocol assumes each sensor node can detect that it is connected to a programming board during the attack. After that the node will send a message to its couple to identify the other node that it is compromised. This protocol cannot be used against collaborative work of attackers to compromise large number of nodes where attackers can collect the couples at the same time. Also, it is not secure to depend on a message from the compromised node to its couple indicating that it is compromised.

The protocol [8] requires high storage overhead for one key list for the whole network, high communication overhead to broadcast "Hello" message to all neighbours then receive the same message from the neighbours, and high energy cost. Also, if one node is captured, then the key list is known to the attacker and the protocol is no longer secure.

In this paper, we developed a new overlapped groups based node compromise detection scheme. Compared with previously reported schemes, the proposed scheme detects

 ISBN: 978-1-61208-207-3

the node compromise attack in the first stage against large number of attackers with low overheads and with high security level. Specifically, after sensor nodes are deployed, first, they share link keys using our key management scheme; second, they perform secure localization using our secure localization scheme; third, they perform secure clustering so that each cluster has SM and BKSM; fourth, they perform overlapped groups for compromise node detection. Each node sends and receives from its interconnected nodes in the cluster to form a group. Every group is overlapped with other groups to form overlapped groups.

A. Contributions

1- We designed a homogenous network utilizing security managers SM and backup security managers BKSM to implement the distributed security concept.
2- We proposed to combine our key management with our secure localization and our secure clustering to develop high security level for the network before forming the overlapped groups.
3- We proposed a novel compromised nodes detection scheme at first stage based on formation of overlapped groups with low communication overhead, storage overhead and power cost. Each node in a group monitors its neighbours in the group. Each group is overlapped with other groups.
4- We designed the protocol such that our network is a chain and each group in the network is a ring in the chain and rings are interconnected therefore, if one ring is compromised, its interconnected rings will discover this.

B. Outline of the Paper

Section 2 presents the related work. Section 3 describes the assumptions and threat model. Section 4 describes the four algorithms to develop our overlapped groups based compromised nodes detection scheme at the first stage. Section 5 presents the security analysis. Section 6 presents the performance analysis. Section 7 presents the simulation results. Section 8 presents the comparison with others works. Section 9 concludes the paper.

II. RELATED WORK

In this section, we present related work to our proposed scheme.

A. SurvSec Security Architecture

Surveillance Security (SurvSec) is a new designed security architecture for reliable network recovery from single BS failure of surveillance WSN with single BS [18]. SurvSec relies on a set of sensor nodes serve as SMs for management and storage of the security related data of all sensor nodes. SurvSec has three components: (1) Sensor nodes serve as SMs, (2) Data Storage System, (3) Data Recovery System.

SurvSec is used for securing surveillance WSN during the time between the BS failure and the new mobile BS deployment which is the perfect time for attackers to compromise many legitimate nodes then destroy the security of the whole network. Also, SurvSec describes how the new BS will verify the trustworthiness of the deployed WSN otherwise a new WSN must be deployed.

B. Compromised Nodes Detection Schemes

We need an effective security scheme to identify compromised nodes in a timely manner because compromised nodes in surveillance WSN represent uncovered areas. A node compromise attack involves three stages. From [1-6], the authors proposed many protocols to detect compromised nodes based on location, signal strength, reputation, weighted trust, intrusion detection and MAC layer misbehavior. However, these approaches are not effective since they can detect compromised nodes on the second or the third stage and they depend on node's misbehavior or node's location, which means a node may be compromised but behaves well until a programmed time. In [7], a couple based compromised node detection protocol at first stage is proposed to build couples of sensor nodes where the couple can monitor each other but this scheme cannot be used against collaborative work of attackers to compromise large number of nodes because attackers can collect the couples at the same time. Also, we cannot depend on a message from the compromised node to its monitored node. Two protocols [8] are proposed based on four messages. Each sensor node broadcasts a "Hello" message to his neighbors which receive this message and reply to it. If the node did not send for three times, it is marked as compromised and the compromised node neighbors flood the network with the node is compromised message. This protocol uses one key list for the whole network which is insecure in addition to large communication overhead, storage overhead, and high power cost.

Also, software-based attestation techniques [9-17] have been proposed to verify the contents of the code running on nodes where the node's free memory space is filled with incompressible random noise known to the attester.

For the detection in the second stage: In [2], Song et al. made the first attempt to detect compromise node in the second stage. They assume that an adversary will not be able to precisely deploy the compromised sensors back into their original positions.

For the detection in the third stage: In [1], Carl et al. demonstrate the case in which compromised nodes can be detected in the third stage and they show exactly what information can be obtained and how it can be used to disrupt, falsify data within, or eavesdrop on sensor networks. They suggest that sensor nodes in hostile environment would be desirable not to respond to the standard on chip debugging and if a node can detect its own movement by either accelerometers or GPS then it can preemptively delete important information stored in SRAM.

 ISBN: 978-1-61208-207-3

In [3], Kyasanur and Vaidya propose modifications to IEEE 802.11 MAC protocol to simplify misbehaviour detection. Once the sensor nodes are compromised, they will launch false data injection attack. Thus, several en-route filtering schemes [4, 5] have been proposed to drop the false data en-route before they reach the sink. Nevertheless, these schemes only mitigate the threats. Thus Ye et al. [6] propose a probabilistic nested marking scheme to locate colluding compromised nodes in false data injection attacks. Recently, several software-based attestation schemes for node compromise detection in sensor networks have been proposed [11].

III. NETWORK MODEL&THREAT MODEL

A. Network Model

We consider a hierarchical WSN consisting of a BS, sensor nodes which are grouped into clusters and beacon nodes equipped with GPS called beacons. Each node has a unique ID, unique location and unique certificate. The assumptions of model are as follows:

1- We assume sensor nodes are static and some nodes continuously store the detected security threats and all other security data related to nodes where these nodes are SMs.
2- Nodes in the cluster form a group and every group is overlapped with other groups.

B. Threat Model

We consider a group of attackers that try to uncover the keys of the network through capturing some nodes then redeploy the compromised nodes in the network again.

IV. PROPOSED SCHEME

The proposed scheme has four phases which are key management phase to distribute keys among nodes, secure localization phase to determine nodes locations, secure clustering phase to choose BKSM to revoke SM if it is compromised, and forming overlapped groups phase for the overlapped groups based compromised nodes detection protocol at first stage. The proposed scheme has four types of sensors: SMs, BKSMs, initiators and sensor nodes.

A. Key Management Phase

We propose a novel hybrid and dynamic key management protocol utilizing our novel scheme of certificates shared verification to eliminate the needs for high end sensor nodes HSNs which have high power for intensive calculation of public key operations. High end sensor nodes are the best targets for the attackers in the hostile environment. Our proposed key management scheme has two steps which are: key predistribution and key establishment. HSN is the nodes cluster head.

i. Key Predistribution

The key predistribution step consists of acquiring the sensors certificate from the certificate authority CA. ECC is used in this protocol to perform security functions on sensors with limited computing resources. The protocol uses the elliptic curve implicit certificate scheme [19] instead of X.509 because of the resulting low storage overhead, low communication overhead, which is a dominant factor for low bit transmission channels in WSN.

The certificate generation processes for any sensor node U is performed offline before it joins the network.

1- CA selects a random integer q_{CA} as its static private key, and computes the public key $Q_{CA}= q_{CA} X P$. q Multiplied by P.
2- To obtain a certificate and private-public key pair, the sensor U randomly selects a key pair (q_U, Q_U) where $Q_U= q_U X P$ and sends Q_U and q_U to CA. U sends its public and private key to CA so that CA can verify the pair. CA is not on the network, so it works off line.
3- CA verifies U's identity and private-public key pair.
4- The implicit certificate for U is the concatenation of CA's public key Q_{CA}, the device identity ID_U, the U public key Q_U and the certification expiration date t_U , i.e., the certificate is (Q_{CA}, ID_U, Q_U, t_U) signed by the CA private key using ECDSA.

ii. Key Establishment

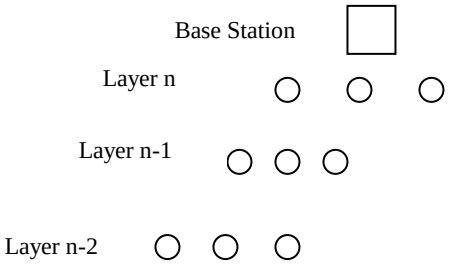


Fig. 1 Network Topology

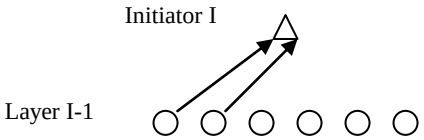


Fig. 2.a Certificates Verification using Initiator for 2 nodes

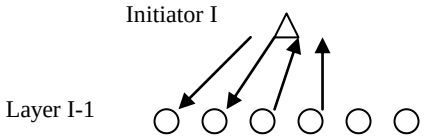


Fig. 2.b Certificates Verification using Initiator for 2 nodes

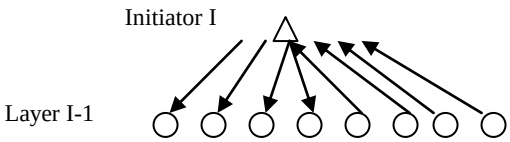


Fig. 2.c Certificates Verification using Initiator for 4 nodes

Figure 1 shows the network topology and Figure 2 shows the certificates shared verification process for one

layer using algorithm 1 that uses initiator nodes to start the process of key establishment protocol. We assume that there are nodes named as security managers SMs and these nodes are located every two layers. We assume that there are nodes named as initiators every predefined number of nodes such as 30, 20 or 10 nodes to start the operation of key management process. Initiator node verifies certificates of first two nodes then it sends the certificates of the second two nodes to the verified first two nodes then it sends certificates of other four nodes to the verified four nodes. Algorithm 1 is efficient in terms of distribution of power consumption among sensor nodes in the cluster and it can be used with all SMs in their clusters. The nodes under the initiator are ordinary nodes and the nodes under the ordinary nodes are SMs and so on until we reach another initiator.

Each node in HSN model performs four times certificate verification for three beacon nodes and for HSN certificates. With the same number of certificates verification at each node, we developed our proposed certificates shared verification scheme. Each node in our scheme verifies four certificates only with the cost of increasing the communication overhead with four messages for every node. These verifications are: first verification for SM certificate, two verifications for two nodes in the cluster, and one verification for beacon node certificate.

We explain our scheme in the form of an algorithm.

Algorithm 1: Initiator nodes to start key management process

1: I → n : { I (Q_{CA}, ID_{SM}, Q_{SM}, t_{SM}) }

Each initiator node broadcasts its certificate to its underneath nodes at layer n to verify it. The nodes at layer n verify the certificate of the initiator. The initiator node is a SM.

2: n → I : { n (Q_{CA}, ID_U, Q_U, t_U) }

The initiator node receives the certificates of its underneath nodes for verification. We assume there are n nodes underneath the initiator node. First, the initiator node verifies the certificate of the first two nodes.

3: I → $n_{1,2}$: { share link keys }

The initiator node shares link keys with node 1 and node 2 as steps from 4 to 9.

4: n : selects (k), calculates (d_U), encrypts (d_U)

Each node underneath initiator I at layer n selects a k-bit random number c_U of 160 bits to produce its link key contribution with the BS. Each node at n calculates the value of $d_U = H(c_U \| ID_U)$ where H is a cryptographic hash function. Each node at n encrypts d_U with I public key Q_I. To encrypt and send a message d_U to I, each node at n chooses a random positive integer x and produces the ciphertext C_m consisting of the pair of points which are: $C_m = (x\,P,\ d_U + x\,Q_I)$. We add the number d_U to both values of the point $x\,Q_I$.

5: n → I : { C_m }

Each node underneath I at layer n sends its encrypted link key contribution with the I which is C_m.

6: I : decrypts (C_m), selects (k), calculate (d_{BS}), encrypts (d_{BS})

I decrypts C_m for every node at n. I multiplies first point in the pair by I's private key and subtracts result from second point: $d_U + x\,Q_I - q_I\,(x\,P) = d_U + x\,(q_I P) - q_I\,(x\,P) = d_U$.

I selects a k-bit random number c_I of 160 bits for each node near I to produce its link key contribution with nodes near I. I calculates the value of $d_{BS} = H(c_I \| ID_I)$ for every node near I where H is a cryptographic hash function.

I encrypts d_{BS} for every node near I using symmetric key encryption under key d_U, generating value $y = E_{du}\,(ID_I \| d_{BS})$.

7: I → n : { y }, {hash {K}}

I sends y, the encrypted link key contribution of I, to every node near I. I generates the link key with every node near the I at n by calculating $K = H\,(d_u \| ID_U \| d_I \| ID_I)$ then $H(K)$ where H is a cryptographic hash function. I sends $H(K)$ of every node at n to its participant to achieve correctness.

8: n : decrypts (y), calculates (K)

Every node at n decrypts the received message y using symmetric key encryption under key d_U to obtain the value d_I.

Every node at n generates the link key with I by calculating the $K = \mathrm{H}(d_u \| ID_U \| d_I \| ID_I)$.

9: n → I : {z}

Every node at n calculates $z = \mathrm{H}(K)$ and sends z to I. I checks if $z = H(K)$. If yes, the link key is established correctly. Otherwise, the protocol is terminated.

10: I → $n_{1,2}$: { $n_{3,4}$ (Q_{CA}, ID_U, Q_U, t_U) }

The initiator node sends to node 1 and node 2 underneath the certificates of node 3 and node 4 for verification.

11: $n_{1,2}$ → I : { valid certificates or invalid certificates }

Node 1 and node 2 send to the initiator node two messages indicating that certificates of nodes 3 and 4 are valid or not.

12: I → $n_{3,4}$: { share link keys }

The initiator node shares link keys with node 3 and node 4 as steps from 4 to 9 in algorithm 1.

13: I → $n_{1,2,3,4}$: { $n_{5,6,7,8}$ (Q_{CA}, ID_U, Q_U, t_U) }

The initiator node sends to node 1, node 2, node 3 and node 4 underneath the certificates of node 5, node 6, node 7 and node 8 for verification and nodes 1, 2, 3, 4 respond with valid certificate or not.

14: I → $n_{5,6,7,8}$: { share link keys }

The initiator node shares link keys with node 5, node 6, node 7 and node 8 as steps from 4 to 9 in algorithm 1.

Finally, the process of the initiator continues to verify all

of its underneath nodes then its underneath nodes use algorithm 1 to share link keys with their underneath nodes and so on.

1. The second layer after the initiator is SMs and so on until the initiator layer because initiators are defined every 30 or 20 or 10 nodes.
2. After the SMs and the sensor nodes establish link keys, they determine their locations using our proposed secure localization scheme with certificates shared verification.

B. Secure Localization Phase

A number of secure localization algorithms [20] have been reported. Different researchers have different strategies to categorize them. These strategies can be divided into direct and indirect localization, centralized and distributed localization, range-based and range-free localization, absolute and relative localization. We propose to get the location information to form the group from the followings approach:

The indirect approaches of localization were introduced to overcome some of the drawbacks of the GPS-based direct localization techniques while retaining some of its advantages. In this approach, a small subset of nodes in the network, called the beacon nodes, are equipped with GPS receivers to compute their location. Beacon nodes send beams of signals providing their location to all nodes in their vicinity. Using the transmitted signal containing location information, nodes compute their location. Each node needs three beacon nodes to locate its position.

Our proposed scheme depends on SM and certificates shared verification for secure localization. We assume that each cluster has three beacon nodes. Sensor nodes in the cluster send the beacon nodes certificates to SM then SM sends these certificates to its underneath nodes for verification to insure one verification time for beacon nodes certificates for the whole cluster. This is done because Verification power is 1000 times more than communication power [21]. SM assures that certificate verification for beacon nodes is done only once for the whole cluster to reduce the power of verification. Each node needs to verify three beacon nodes with total of 3n verifications but with certificate shared verification this is done once. SMs are clusterheads.

Algorithm 2: Secure Localization

1: $Beacons_{1,2,3} \rightarrow SM_n$: {$Beacons_{1,2,3}$ (Q_{CA}, ID_B, Q_B, t_B) }

The beacon nodes near BS broadcast their certificates and locations to SMs near BS. We need three beacon nodes to locate the position.

2: $SM_n \rightarrow BS$: { $Beacons_{1,2,3}$ (Q_{CA}, ID_B, Q_B, t_B) }

The SMs near BS at layer n send the certificates of the beacon nodes to BS for verification.

3: $BS \rightarrow SM_n$: {valid certificates of $Beacons_{1,2,3}$ }

BS sends to SMs at layer n that beacon nodes certificates are valid.

4: $SM_n \rightarrow Beacons_{1,2,3}$: { $Key_{1,2,3}$ }

Every SM at layer n shares a link key with the three beacon nodes in four steps.

5: SM_n : calculates (x, y) position

Every SM at layer n calculates its position.

6: $Beacons_{1,2,3} \rightarrow$ n-1 : {$Beacons_{1,2,3}$ (Q_{CA}, ID_B, Q_B, t_B) }

The beacon nodes near BS broadcast their certificates and locations to nodes at layer n-1.

7: n-1 $\rightarrow SM_n$: { $Beacons_{1,2,3}$ (Q_{CA}, ID_B, Q_B, t_B) }

The nodes at layer n-1send the certificates of beacon nodes to SMs at layer n for verification. If the beacon nodes certificates are previously verified, it is ok but if there are new beacon nodes certificates, then SMs at layer n send the new beacon nodes certificate to BS for verification.

8: $SM_n \rightarrow$ n-1 : { $Key_{1,2,3}$ }

Every SM at layer n sends its link keys with the beacon nodes to its connected nodes at layer n-1.

9: n-1 : calculates (x, y) position

Every node at layer n-1 calculates its position.

10: $Beacons_{4,5,6} \rightarrow SM_{n-2}$: {$Beacons_{4,5,6}$ (Q_{CA}, ID_B, Q_B, t_B) }

The beacon nodes near SMs at layer n-2 broadcast their certificates and locations to SMs at layer n-2.

11: $SM_{n-2} \rightarrow SM_n$: { $Beacons_{4,5,6}$ (Q_{CA}, ID_B, Q_B, t_B) }

The SMs at layer n-2 send the certificates of the beacon nodes to SMs at layer n for verification.

12: $SM_n \rightarrow$ n-1 : { $Beacons_{4,5,6}$ (Q_{CA}, ID_B, Q_B, t_B) }

The SMs at layer n send the certificates of the beacon nodes to nodes at layer n-1 for verification.

13: n-1 $\rightarrow SM_n$: { valid certificates of $Beacons_{4,5,6}$ }

The nodes at layer n-1 send to SMs at layer n that beacon nodes certificates are valid.

14: $SM_n \rightarrow SM_{n-2}$: { valid certificates of $Beacons_{4,5,6}$ }

The SMs at layer n send to SMs at layer n-2 that beacon nodes certificates are valid.

15: $SM_{n-2} \rightarrow Beacons_{4,5,6}$: { $Key_{4,5,6}$ }

Every SM at layer n-2 shares a link key with the three beacon nodes in four steps.

16: SM_{n-2} : calculates (x, y) position

Every SM at layer n-2 calculates its position.

17: $Beacons_{4,5,6} \rightarrow$ n-3 : {$Beacons_{4,5,6}$ (Q_{CA}, ID_B, Q_B, t_B) }

The beacon nodes near nodes at layer n-3 broadcast their certificates and locations to nodes at layer n-3.

18: n-3 $\rightarrow SM_{n-2}$: { $Beacons_{4,5,6}$ (Q_{CA}, ID_B, Q_B, t_B) }

The nodes at layer n-3 send the certificates of beacon nodes to SMs at layer n-2 for verification. If the beacon nodes certificates are previously verified, it is ok but if there are new beacon nodes certificates, then SMs at

 ISBN: 978-1-61208-207-3

layer n-2 send the new beacon nodes certificate to SMs at layer n for verification.

19: SM_{n-2} → n-3 : { $Key_{4,5,6}$ }

Every SM at layer n-2 sends its link keys with the beacon nodes to its connected nodes at layer n-3.

20: n-3 : calculates (x, y) position

Every node at layer n-3 calculates its position. **Finally**, lower layer SMs send certificates of beacon nodes to higher layer SMs for verification.

1. Certificates shared verification for beacon nodes certificates between SM and its underneath nodes will reduce setup time and reduce computations complexity at the cost of increasing only four messages.
2. Certificates verification for beacon nodes is done only one time not multiple times at each node underneath the SM to reduce computations complexity.
3. Sensor nodes underneath SM will use the shared keys between the SM and the beacon nodes which will reduce the setup time, computations and storage overhead.
4. After the SMs and the sensor nodes determine their locations, they form secure clustering.

C. Secure Clustering Phase

SMs can form secure clustering [22] with their nodes underneath and SM can choose BKSM to replace it if the SM is compromised.

Algorithm 3: Secure Clustering

1: BS → n : {req SM_msg }

BS sends to nodes near BS at layer n that these nodes are SMs using its shared symmetric key with these nodes.

2: SM_n → n-1 : { adv cluster_msg }

Every SM at layer n sends an encrypted advertise message to nodes at layer n-1 to form a cluster.

3: n-1 → SM_n : { join cluster_msg }

Every node at layer n-1 sends an encrypted message to its SM at layer n to join the cluster.

4: SM_n → n-1 : {choose BKSM }

The SM at layer n chooses BKSM according to maximum connectivity between the BKSM and the nodes in the cluster where BKSM must be connected to all nodes in the cluster.

5: n-1 → n-2 : { req SM_msg }

The nodes at layer n-1 send to nodes at layer n-2 an encrypted message that these nodes are SMs.

6: SM_{n-2} → n-3 : { adv cluster_msg }

Every SM at layer n-2 sends an encrypted advertise message to nodes at layer n-3 to form a cluster.

7: n-3 → SM_{n-2} : { join cluster_msg }

Every node at layer n-3 sends an encrypted message to its SM at layer n-2 to join the cluster.

8: SM_{n-2} → n-3 : {choose BKSM }

The SM at layer n-2 chooses BKSM according to maximum connectivity between the BKSM and the nodes in the cluster where BKSM must be connected to all nodes in the cluster.

9: $BKSM_{n-2}$ → n-3 : { BKSM (Q_{CA}, ID_{BKSM}, Q_{BKSM}, t_{BKSM}) }

The BKSM at layer n-2 sends its certificate to the nodes at layer n-3 where SM at layer n-2 verifies this certificate. **Finally**, the steps of forming the secure clustering are performed until the lower layer of SMs.

1. Our proposed secure clustering scheme assumes hybrid key management protocol to achieve high security level.
2. Our proposed scheme chooses BKSM to solve the problem of compromised SM and to sign the message of revoked SM.
3. Our scheme achieves secure clustering in four messages.

D. Forming Overlapped Groups Phase

Each node in a cluster sends its location to its SM. From the nodes locations at the SM, the SM starts the process to form a group. Assume each cluster has n nodes and the SM builds the overlapped group from the nodes in the cluster as shown in algorithm 4.

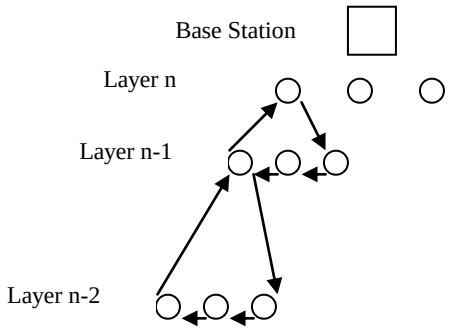


Fig. 3 Overlapped Groups Formation

Figure 3 shows the overlapped groups formation. Algorithm 4 represents forming a group from seven nodes which are labeled from n to n-6. The SM sends a message to its nearest node containing the sequence of sending and receiving messages in the cluster to form a group according to each node neighbours. The SM chooses a group key for the cluster and sends it to all nodes in the cluster. A group is a cluster.

Algorithm 4: Forming Overlapped Groups

1: SM_n → n, n-6 : { join group_msg }

SM at layer n sends an encrypted message to node n and node n-6 to form the group and the message contains the interconnections of all nodes in the cluster to form the group. The used key is the group key between the SM and the nodes in the cluster. The sent message includes what every node is connected to in the cluster to form a closed loop.

2: n → n-1, SM_n : { join group_msg }

 ISBN: 978-1-61208-207-3

Node n sends an encrypted message to node n-1 and SM to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus one. The sent message is the join group message.

3: n-1 → n, n-2 : { join group_msg }

Node n-1 sends an encrypted message to node n and node n-2 to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus two.

4: n-2 → n-1, n-3 : { join group_msg }

Node n-2 sends an encrypted message to node n-1 and the node n-3 to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus three.

5: n-3 → n-2, n-4 : { join group_msg }

Node n-3 sends an encrypted message to node n-2 and the node n-4 to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus four.

6: n-4 → n-3, n-5 : { join group_msg }

Node n-4 sends an encrypted message to node n-3 and the node n-5 to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus five.

7: n-5 → n-4, n-6 : { join group_msg }

Node n-5 sends an encrypted message to node n-4 and the node n-6 to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus six.

8: n-6 → n-5, SM_n : { join group_msg }

Node n-6 sends an encrypted message to node n-5 and the SM at layer n to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus seven.

Finally, the "Hello" message is sent from one node to two neighbour nodes in the cluster and the two nodes respond to the "Hello" message. If the node is compromised, it will not send the "Hello" message and therefore, the recipient nodes will mark it as compromised and they will send to the SM to revoke that node. If the SM is compromised, its monitored nodes will send to the BKSM to revoke the SM.

1- Our proposed compromised nodes detection scheme is based on the overlapped groups to discover the compromised group. If a node is compromised in a group, it will be detected by its neighbours who will send to SM that this node is compromised. If a SM is compromised, its neighbour nodes will send to the BKSM to revoke it.

2- Each node sends at first time with key K then next time with key K+n+1 and next time with key K+2n+1 and so on.

3- Each node sends a "Hello" message and receive two messages from its neighbours in 15 seconds.

4- Each group forms an overlapped group with its upper group and its lower group.

5- We designed the compromised nodes detection protocol at first stage such that our network resembles a chain and each cluster in the network forms a group and each group is a ring in the chain and rings are interconnected therefore, if one ring is compromised, its interconnected rings will discover this.

V. SECURITY ANALYSIS

Security analysis of our protocol focuses on resilience to node compromise attack, collusion attack and impersonation attack.

A. Compromised Node Attack

1- If an attacker compromises one regular node, therefore, the probability of insecure link is $P_{insec} = 1/N$ where N is the number of nodes at the network. For n compromised regular nodes the probability of insecure links is $P_{insec} = n/N$.

2- If the attacker compromises one SM, therefore, the probability of insecure links is $P_{insec} = (n_s + 4) /N$ where n_s is the number of nodes in the cluster of the SM. For n compromised SMs the probability of insecure links is $P_{insec} = n (n_s + 4) /N$.

3- Our proposed key management assumes compromised node detection at the first stage and compromised nodes revocation. Therefore, SM will revoke the regular compromised node and the BKSM will revoke the SM to eliminate the insecure links.

B. Collusion Attack

Two nodes can collude when they share their keys with each other. Our designed protocol is resistant to collusion attack because each sensor node communicates only with a SM therefore; compromised nodes cannot discover themselves.

C. Impersonation Attack

Each node has a certificate to join the key management process and to join the network. This prevents the attacker from impersonating any legitimate node. Also, knowing the public key of the SM will not reveal the private key for the SM because this needs the attacker to solve the elliptic curve discrete logarithmic problem ECDLP which is a hard problem.

 ISBN: 978-1-61208-207-3

VI. PERFORMANCE ANALYSIS

The performance analysis is measured in computation complexity, communication complexity and storage complexity. We assume that the network is secure during setup time which depends on number of initiators.

A. Computation Complexity

SM generates a group key and sends it encrypted with the shared link key with every node in the cluster to use it in the process of compromised nodes detection. Each sensor node decrypts the message sent with group key with its shared link key with the SM.

Our scheme has lower computation overhead than the scheme that uses couples to detect compromised nodes at the first stage. Our scheme has the same computation overhead compared to the scheme uses distributed compromised nodes detection at first stage. Our scheme has low computation overhead to generate the group key and to send it encrypted to all nodes in the group.

B. Communication Complexity

Communication complexity is the number and size of packets sent and received by a sensor node. In our protocol, the number of messages sent is one message every 15 seconds and there are two messages received every 15 seconds with total of three messages sent and received every 15 seconds to establish the compromised nodes detection protocol. Our scheme has lower communication overhead than the other two schemes that detects compromised nodes at the first stage.

C. Storage Complexity

Storage complexity is the amount of memory units required to store security credentials. Each sensor node stores the group key with the SM and other nodes in the cluster. Our scheme has the same storage overhead as the scheme uses couples to detect compromised nodes at the first stage but it has lower storage overhead than the scheme uses distributed compromised nodes detection at first stage. Our scheme needs to store only one key which is the group key between the SM and the nodes in the group

VII. SIMULATION RESULTS

We built a model for the proposed design and we implemented a simulator in MATLAB that can scale to thousand of nodes. In this simulator, sensors can send and receive data from each other's. The simulation verifies the correctness and the feasibility of our security architecture. It is our future work to implement SurvSec in some sensor network testbeds with all its ingredients. Our simulation scenarios include N nodes distributed randomly. We choose N as 1000 sensor nodes.

The followings are the built models for simulation:

1- Network setup model for the overlapped groups.
2- Compromised nodes detection protocol.

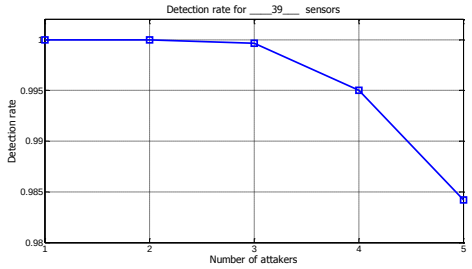
In the simulations, these parameters are given as follows:

1- The number of sensor nodes *n* is varied from 39 to 1000 sensor nodes.
2- The interval of beacon information is set to 15 seconds.
3- The time of an adversary to successfully compromise a sensor node is varied from 30 seconds to 60 seconds.

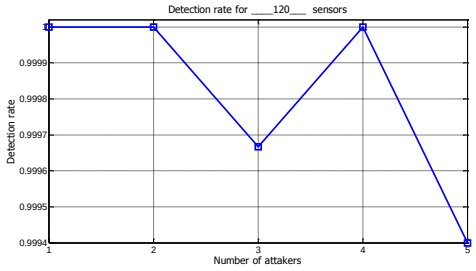
In this section, we evaluate the detection rate under different n.

The detection rate is equal to the detected compromised sensor nodes over all compromised nodes.

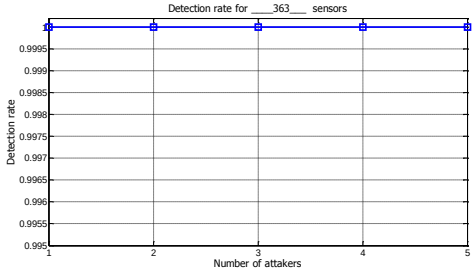
In the proposed adversary model, we assume that an adversary can simultaneously compromise *k* sensor nodes, where $k<n$.



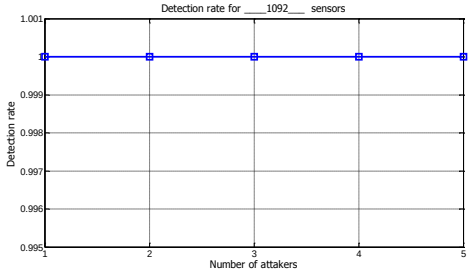
(a) $n = 39$, $k = 5$, number of compromised nodes is 5.



(b) $n = 120$, $k = 10$



(c) $n = 363$, $k = 15$



(d) $n = 1092$, $k = 25$

Fig. 4. Detection rate varies with number of compromised nodes under different n =39, 120, 363, 1092, Interval = 15 Sec.

 ISBN: 978-1-61208-207-3

Thus, we first evaluate the detection rate under different parameters *n*, *k* and beacon interval and the results are shown in Figure 4. From Figure 4, we can see the detection rate does not increase linearly with *k*. When *n* = 363 or *n* = 1092, the detection rate reaches the maximum. Due to this observation, when the number of sensor nodes increase, we found that the proposed scheme has high resiliency against node compromise attack by collaborative work of attackers at the same time for large hierarchical WSN.

VIII. COMPARISON WITH OTHER WORKS

Now, we compare between our proposed model and previous works that detects compromised nodes at first stage.

TABLE 1, COMPARISON BETWEEN OUR MODEL AND OTHER MODELS.

	Property	**CAT [7]**	**Distributed Detection [8]**	**Our Model**
1	Detect compromised nodes for group of attackers	No	Yes	Yes
2	Detection rate	Less than 100%	Near 100%	Near 100%
3	Communication overhead	14 messages every 15 sec for beacon every 2 sec High overhead	At least 6 messages every 15 sec moderate overhead	At least 3 messages every 15 sec low overhead
4	Computation overhead	Low	Low	Low
5	Storage overhead	Low to store one key	High to store key list	Low to store one key
6	Setup time	Low	Low	Low
7	Power cost	High	High	Low

Our proposed model can be used against collaborative work of attackers to compromise large number of nodes at the same time. Also, the detection rate is near 100%. Our model has low communication overhead and low computation overhead and low storage overhead. Our model has low power cost since it sends and receives only three messages every 15 sec which is lower than the other two schemes. Key predistribution time is equal to the time that is needed from the initiator to distribute keys with its underneath nodes because initiators work separately.

IX. CONCLUSION

In this paper, we proposed the overlapped groups-based compromised nodes detection scheme to early detect the node compromise attack in the first stage. Concretely, the simulation results showed that by building groups among neighboring sensor nodes in a local area, physical node compromise attack can be detected immediately. Also, the simulation results showed that the proposed detection scheme has high detection rate. This work is an initial work to form overlapped groups for detecting compromise attack at the first stage and we do not expect that the proposed scheme will solve all the problems in the node compromise nodes attack. Our future work will continue to build more overlapped groups to early detect compromise nodes attack.

REFERENCES

[1] C. Hartung, J. Balasalle, and R. Han, "Node compromise in sensor networks: the need for secure systems," in *Technical Report CU-CS-990-05, Dept. of Comp Sci, Univ of Colorado at Boulder*, Jan 2005.

[2] H. Song, L. Xie, S. Zhu, and G. Cao, "Sensor node compromise detection: the location perspective," in *IWCMC'07*, Honolulu, Hawaii, USA, Aug. 2007.

[3] P. Kyasanur and H. Vaidya, "Detection and handling of mac layer misbehavior in wireless networks," in *IEEE DSN*, 2003.

[4] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by hop authentication scheme for filtering of injected false data in sensor networks," in *IEEE Symposium on Security and Privacy'04*, 2004.

[5] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, "Toward resilient security in wireless sensor networks," in *ACM MobiHoc'05*, 2005.

[6] F. Ye, H. Yang, and Z. Liu, "Catching moles in sensor networks," in *IEEE ICDCS'07*, Jun, 2007.

[7] Xiaodong Lin, "*CAT: Building Couples to Early Detect Node Compromise Attack in Wireless Sensor Networks*", IEEE "GLOBECOM" 2009.

[8] Wei Ding, Yingbing Yu, and Sumanth Yenduri, "Distributed First Stage Detection for Node Capture", IEEE Globecom 2010.

[9] A. Seshadri, M. Luk, E. Shi, A. Perrig, L. van Doorn, and P. Khosla, "Pioneer: verifying integrity and guaranteeing execution of code on legacy platforms," in *SOSP*, Oct. 2005.

[10] D. Spinellis, "Reflection as a mechanism for software integrity verfication," in *ACM Trans. Inf. Syst. Secu.*, Vol, 3, No, 1, 2000.

[11] Yi Yang, Xinran Wang, Sencun Zhu, and Guohong Cao, "*Distributed Software-based Attestation for Node Compromise Detection in Sensor Networks*", 26th IEEE International Symposium on Reliable Distributed Systems, IEEE 2007.

[12] Taejoon Park, and Kang G. Shin, "*Soft Tamper-Proofing via Program Integrity Verification in Wireless Sensor Networks*", IEEE Transactions on Mobile Computing, Vol. 4, No. 3, May/June 2005, IEEE 2005.

[13] Xiaojiang Du, "Detection of Compromised Sensor Nodes in Heterogeneous Sensor Networks", IEEE "ICC" 2008.

[14] Arvind Seshadri, Adrian Perrig, Leendert van Doorn, and Pradeep Khosla, "*SWATT: SoftWare-based ATTestation for Embedded Devices*", In IEEE Symposium on Security and Privacy (2004), IEEE Computer Society 2004.

[15] Tamer AbuHmed, Nandinbold Nyamaa, and DaeHun Nyang, "*Software-Based Remote Code Attestation in Wireless Sensor Network*", IEEE "GLOBECOM" 2009.

[16] Jun-Won Ho, Matthew Wright, and Sajal K. Das, "*ZoneTrust: Fast Zone-Based Node Compromise Detection and Revocation in Sensor Networks Using Sequential Analysis*", 2009 28th IEEE International Symposium on Reliable Distributed Systems, IEEE 2009.

[17] J. Deng, R. Han, and S. Mishra, "*Secure Code Distribution in Dynamically Programmable Wireless Sensor Networks*", In Proc.

 ISBN: 978-1-61208-207-3

International Conference on Information Processing in Sensor Networks, pp. 292–300, ACM 2006.

[18] Mohamed Megahed, and Dimitrios Makrakis, "SurvSec: A New Security Architecture for Reliable Network Recovery from Base Station Failure of Surveillance WSN", 2nd International Conference on Ambient Systems, Networks and Technologies, ANT 2011

[19] Rene Struik and Gregg Rasor, "Mandatory ECC Security Algorithm Suite", IEEE 2002, Wireless Personal Area Networks, March 2002.

[20] C. Savarese, J. Rabay and K. Langendoen. "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks". USENIX Technical Annual Conference, Monterey, CA, June 2002.

[21] Krzysztof Piotrowski, Peter Langendoerfer and Steffen Peter, "How Public Key Cryptography Influences Wireless Sensor Node Lifetime", Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks, ACM 2006.

[22] Leonardo B. Oliveira, Hao C. Wong, M. Bern, Ricardo Dahab, and A. A. F. Loureiro, "SecLEACH – A Random Key Distribution Solution for Securing Clustered Sensor Networks", Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications NCA 06, IEEE 2006.

 ISBN: 978-1-61208-207-3