Energy Efficient Key Management Protocols to Securely Confirm Intrusion Detection in Wireless Sensor Networks

Jibi Abraham and K S Ramanatha
Department of Computer Science & Engineering
M S Ramaiah Institute of Technology, MSRIT Post,
Bangalore - 560 054, Karnataka, India
Email: jibiabraham@msrit.edu, ksramanatha@msrit.edu

Abstract: Wireless Sensor Networks are prone to many security attacks. The most complex among them is the node compromise attack. Networks enhanced with services like aggregation and security require a different intrusion detection mechanism than the generally used solutions and there is a possibility of a compromised node producing false intrusion detection alarms. Therefore we need suitable mechanisms to detect intrusion and to securely confirm the same. We propose two major schemes: 1) a post-deployment key distribution based permanent key establishment scheme 2) two on-the-fly key establishment schemes. The simulation results and analysis show that on-the-fly schemes are better suited for intrusion confirmation in energy constrained sensor networks. A simple and practical intrusion defense scheme also is suggested.

Keywords: Sensor networks, intrusion detection, secure confirmation of intrusion, key management.

1 Introduction

Wireless sensor networks (WSN) are vulnerable to different types of attacks [9] and many of these attacks survive only in sensor networks because the deployment of the network might be in inaccessible terrain and providing costly tamper resistant hardware is not an expected feature. In a node compromise attack [1], an attacker seizes a node from the sensor network, connects this node to his laptop, extracts the stored data, puts new data/behavior and has control on that node. The node compromise attack can generate many other security attacks [9][10] and the effect of all these attacks finally leads to either unavailability of the network functionality or short network life.

In this paper we consider a network that is divided into many clusters. The cluster head in each cluster manages the cluster memberships, the distribution of the security key to members and aggregation of the collected data from the members as described in paper [7]. The data is encrypted at the cluster member by using a pair-wise key or cluster-wide key and decrypted at the cluster head. The cluster head performs aggregation on the collected data, encrypts the data using a different key belonging to the next hop or its parent cluster and the encrypted data is sent to the next node. Detection of attacks in these networks requires a specialized Intrusion Detection Scheme (IDS). Further, even after using the special IDS, a compromised node can always send false intrusion detection alarms about legitimate nodes in the network. Hence proper key management protocols are required to confirm intrusion after detection.

In this paper we propose an intrusion detection scheme as well as a scheme to securely confirm the detection. This is done in Section 3. Section 2 brings out rather limited attempts in this direction in the literature. The intrusion confirmation scheme requires establishment of pair-wise keys between the intrusion detector and the nodes on the data aggregation tree path. In Section 4 we bring out the unsuitability of key pre-distribution method for pair-wise key establishment [3][4][5] in malice prone sensor networks. We have modified and enhanced the grid-based polynomial pre-distribution scheme of [3] to design a new scheme that we have called post-deployment permanent key establishment scheme suitable for secure confirmation of intrusion. We conduct detailed evaluation to bring out deficiencies of such schemes, in general. We therefore propose two on-the-fly schemes which make use of our original work in [7] for the establishment of pair-wise keys. A comparison of all the three schemes is made to bring out the superiority of on-the-fly schemes in general and on-the-fly scheme using the Base Station in particular. In Section 5, a suitable intrusion defense scheme is suggested.

In this paper emphasis is placed on the key management protocols required to confirm intrusion and therefore on their implementation and simulation results. An analysis of the secured data thus collected needs to be made to determine whether the intrusion has really taken place or it is a case of a false alarm and for identifying the intruder in the case of intrusion. However such a confirmation analysis is not within the scope of this paper. Further, the intrusion detection as well as the defense mechanisms are only proposed. Their implementation and evaluation are left for future work.

2 Literature Survey

Even though a few watchdog mechanisms [1][2] are available in the literature, they are not suitable for the networks that support aggregation and security. A secure aggregation scheme is suggested in [6], where the data forging attack at a specific aggregator can be detected. Each node A has a pair-wise key K_{AS} shared with the Base Station S, stored as a pre-deployment knowledge. In the i^{th} data transmission phase, a leaf node A computes a temporary key $K_{i\text{-}AS} = E_{KAS}(i)$, sends its data reading R_A, node ID, ID_A and Message Authentication Code, MAC (R_A, $K_{i\text{-}AS}$) to its parent. Here $E_{KAS}(i)$ denotes the encryption of i using the secret key K_{AS}. The parent node B calculates the aggregation $Aggr$ of its children nodes' readings and sends $Aggr$, ID_B and MAC ($Aggr$, $K_{i\text{-}BS}$) to its parent. When the final aggregation result reaches the Base Station, the Base Station verifies the final aggregate and during data validation phase it broadcasts the temporary keys ($K_{i\text{-}AS}$, $K_{i\text{-}BS}$,...) to the network. Using the temporary keys received, an intermediate aggregator verifies the intermediate aggregation results. Even though the scheme achieves low data communication through aggregation, it is vulnerable because the intermediate aggregation can be easily tampered if a parent and a child node in their hierarchy are compromised. Also data validation step consumes lots of battery power by broadcasting the temporary keys and re-computing the MAC.

It is observed that schemes available in the literature for intrusion detection and its secure confirmation for sensor networks offering aggregation and security services are very limited and are inadequate. The newly proposed schemes of this paper detect intrusion posed by a single compromised node as well as intrusion from a set of cooperative nodes and avoid chances of any false intrusion detection alarms.

3 Detection and Confirmation of Misbehavior in Sensor Networks Offering Aggregation and Security Services

Some of the major misbehaviors that are resulted from a compromised node are explained first. In ***Spoofing*** attack, a node supplies falsified sensed data. In a ***Masquerade*** attack, a compromised node supplies false data as if the data gets originated at a non-compromised node. In ***Falsified Aggregate*** attack, a compromised cluster head computes the aggregate of the data supplied by its members and instead of relaying the actual aggregate results, it will transmit an altered aggregate to its parent. Many pair-wise key establishment protocol steps include message traversal between one or more intermediate nodes and there is a possibility of many of the intermediate compromised nodes breaching the protocol steps by playing **man-in-the-middle** attack.

We propose a new intrusion detection mechanism that avoids unnecessary buffering of bulk data received from many nodes. Usually multiple sensors trigger event detection in a sensor network and all the sensors may report the same detection data to the Base Station. Assume that every cluster is numbered and each data message includes the cluster number of the data origin, data source ID and the aggregated data value. When a data originated from a cluster reaches at a cluster head on the data traversal path at the first time, the cluster head CH_k at level k inserts a record including cluster number and the aggregate value into a Cache. Since there is not much possibility of wide deviation in values sensed by neighbor nodes for a specific event detection, when the data arrives from the same cluster at later times, if CH_k finds wide deviation in the stored aggregate value with the latest received aggregate, then it suspects the possibility of alteration attack at $CH_{k+1,}$ the next lower level.

After perceiving deviation, the cluster head that has detected intrusion intimates the Base Station and as part of the intrusion defense mechanism, the Base Station may instruct all the other nodes in the network to reduce communication towards the suspected node. But here comes possibility of another attack like a compromised cluster head sending false intrusion detection alarm about a legitimate neighbor cluster head to the Base Station. As a result of the defense mechanism, all the data communications towards the legitimate cluster head may get diverted to the compromised cluster head and this compromised cluster head may start misbehaving on the data. Hence it is always better to confirm the intrusion by directly contacting all the nodes on the data traversal path of the deviated aggregate value. This leads to designing an *Intrusion-Confirm* protocol.

The *Intrusion-Confirm* protocol consists of a request message and a reply message. Let the data originator be the source node S_i. To verify and confirm the occurrence of an attack, CH_k prepares an *Intrusion-Confirm* request message demanding to provide the value of the actual data supplied by cluster of S_i. This request message is sent to all the nodes on the aggregation tree path starting from CH_{k+1} to S_i. After collecting *Intrusion-Confirm* reply messages from all nodes, CH_k analyzes and confirms whether CH_{k+1} is a compromised node or not.

There can be a possibility of a sudden actual shift in the sensed data. In this situation the immediate cluster head at first notices the change. If any other cluster head other than the immediate cluster head perceive the value shift then that may be a case of an intrusion. Even if two cluster heads make a collaborative attack, since the aggregation corresponds to the data from the source cluster, attack can be detected at the higher level. The possibility of a compromised cluster head/member masquerading a legitimate cluster head/member by supplying false data can also be detected. The scheme requires data buffer to store the *Intrusion-Confirm* reply messages and data messages from many neighbour nodes only after the detection to confirm the intrusion.

The *Intrusion-Confirm* request and reply messages are to be secured because otherwise the malicious node in between can alter the request and reply messages by playing man-in-the-middle attack. Therefore there is a requirement of pair-wise key establishment from CH_k to all the nodes in the aggregation tree path from CH_{k+1} to S_i which is the main concentration of this paper.

4 Pair-wise Keys Establishment for Intrusion Confirmation

Schemes for pair-wise key establishment can be categorized into centralized and decentralized. In centralized schemes, the Base Station acts as a Key Distribution Centre to establish pair-wise keys after verifying the authenticity of each node, assuming that Base Station is physically guarded and can never get compromised. Decentralized schemes based on key pre-distribution [3][4][5], claim to be the most suitable for sensor networks. The nodes in the network co-operate among each other and establish the pair-wise keys. But in a malice prone network, any sensor node may get compromised at any time or fresh malicious nodes may also get added to the network at any time. All the key pre-distribution schemes actually permit any node to establish keys with other nodes without verifying the authenticity, if they have the appropriate pre-distribution material.

The grid-based polynomial pre-distribution scheme presented in [3] claims that the scheme has high probability to establish pair-wise keys, is tolerant to key capture and has low communication overhead. So we decided to try the possibility of using this scheme to establish the required pair-wise keys to confirm intrusion. The scheme utilizes t-degree bi-variate polynomials $f(x, y) = \sum_{i,j=0}^{t} a_{ij}\, x_i\, y_j$ over a finite field F_q, where q is a large prime number and satisfies the property that $f(x, y) = f(y, x)$. The scheme uses a 2-dimensional grid of size m X m, where $m = |\sqrt[2]{N}\,|$ and N is the number of nodes in the network. Each row i of the grid is associated with a row-polynomial $f_{i,\,r}\,(x, y)$ and each column i is associated with a column-polynomial $f_{i,\,c}\,(x, y)$. Each node has to be assigned a grid ID (i, j) that is a row and column intersection point in the grid so that the node stores the polynomial shares of i^{th} row and j^{th} column. A pair-wise key is established either as a Direct-Key or as an Indirect-Key. Two nodes can establish a Direct-Key if both of them share the same row or column polynomial. Otherwise, nodes need to find a key path such that any two adjacent nodes in the path can establish a Direct-Key and by utilizing the nodes on the key path an Indirect-Key is established.

The proposed protocol to confirm intrusion requires usage of pair-wise keys to securely contact the nodes on the data traversal path. If the polynomial based pre-distribution scheme is used after intrusion detection, there is possibility of man-in-the-middle attack posed by one of the compromised nodes included in the key path if an Indirect-Key has to be established. So the required pair-wise keys cannot be established on-the-fly, but they are to be established permanently in the initial phase of the network formation assuming that it not possible to compromise any node at the initial phase of network formation. Hence we have proposed a post-deployment permanent key establishment scheme.

4.1 Post-deployment Permanent Key Establishment scheme

The grid-based scheme in [3] is modified and enhanced as follows to design a post-deployment permanent key establishment algorithm:

- The basic work assumes that each node in the network is in communication range with all other nodes in the network. If a node cannot have Direct-Key with a second node, that node itself can decide the key path required to establish Indirect-Key. But this is not the situation in real time deployment of sensor networks, where not all nodes are in communication range with each other and they use multi-hop communication protocol to communicate each other. Sensor nodes may not be having enough resources to store the details about all the other nodes in the network and the key path discovery process has to be very similar to a route discovery used to establish a route between two nodes that may introduce substantial communication overhead. Hence the basic scheme is enhanced with the Base Station storing the details about all the nodes existing in the network. Whenever any node requests for a key path to another node, the Base Station returns it if a successful path is available.
- The basic work provides a method to convert node address to Grid-ID, which is called as *row-wise* assignment. It assigns consecutive addresses to the Grid-ID starting from the first row and the first column, row-wise. i.e. in the order of (0,0), (0,1), . . .,(0, *m*-1), (1, 0), . . ., (1, *m*-1) and so on up to (*m*-1, *m*-1) where *m* is the dimension of the grid. But if the network is designed for a larger value of *m* but lesser number of nodes is used in the actual deployment, not all row polynomials get well utilized in the scheme. An improved Grid-ID assignment called *Diagonal-wise* assignment is proposed, in which consecutive addresses are assigned in a diagonal way so that all the polynomials are equally well distributed.
- As there are chances of physical capture of nodes from the deployment location to extract the polynomial shares stored at the node, it is preferred to distribute the polynomial share after network deployment, as part of node authentication to the Base Station. The Base Station generates 2*m* number of polynomials immediately after its initialization. As soon as a node self-organizes and joins the network, it will register to the Base Station. The Base Station stores the details about each registering node in a Register and the details include a list of registered nodes with which this currently registering node can establish a Direct-Key. The Base Station returns the row and column polynomials to that node, if the registration is successful.

When a node gets connected to the network and recognizes its parent node, it establishes pair-wise keys with its parent and all the grandparent nodes on the aggregation hierarchical tree using the enhanced Grid-based scheme. The key establishment steps are given in Table1.

Table 1. Permanent Key establishment algorithm

Step	Description
1	Register the node ID to the Base Station and obtain the polynomial shares
2	If Direct-Key is feasible to its parent • From the matched row or column polynomial, derive the key. • Send a Direct-Key confirmation message to the parent. • Parent verifies the Direct-Key and returns the node ID of its parent, if the parent is not the Base Station.
3	If Indirect-Key is feasible to its parent • Request the Base Station for key path to the parent. • When the key path is obtained, generate a random pair-wise key. • Using the nodes on the key path, the encrypted pair-wise key is intimated to the parent. • The parent verifies the pair-wise key and returns the node ID of its parent, if the parent is not the Base Station.
4	If Direct-Key is feasible to the next grand parent • From the matched row or column polynomial, derive the pair-wise key. • Send a Direct-Key confirmation message to the grand parent. • The grand parent verifies the Direct-Key and returns the node ID of its parent, if the parent is not the Base Station.
5	If Indirect-Key is feasible to the next grand parent • Request the Base Station for key path to the grand parent. • When the key path is obtained, generate a random pair-wise key. • Using the nodes on the key path, encrypted pair-wise key is intimated to the grand parent. • The grand parent verifies the key and returns the node ID of its parent, if the parent is not the Base Station.
6	Repeat the steps 4 and 5 until no more grandparents exist for this node.

4.1.1 Protocol Evaluation

The protocol is implemented for TinyOS [11] using NesC language. RC5 algorithm is used for encryption/decryption operations. The simulations are done in TOSSIM [8] for various sized networks and results are taken. While the protocol certainly achieves the main purpose of secure confirmation of intrusion detection, it is beset with several drawbacks. Therefore, a very detailed evaluation is done to bring out the deficiencies of the scheme in real time implementation. The following observations are made:

- The maximum eligible payload size of any *TinyOS* message is 29 Bytes. The structure of the Indirect-Key establishment message used in the implementation, its field interpretations and memory requirement for each field are given in Table 2. It is clear that a maximum of 5 nodes can only be included in the key path to frame an eligible TinyOS message. During simulation of the proposed protocol, it is observed that the length of the key path formed was 2, 3 and 4 for a maximum of 250 nodes in the network. So if the Grid-based scheme is used for pair-wise key establishment with more than 250 nodes then there is a possibility of key establishment failure because it may not be possible to find a key path with a maximum of 5 nodes.

Table 2. Structure of Indirect Key Establishment Message

Message-field	Interpretation	Size (Byte)
Message-type	Key establishment with parent or grand parent	1
Key-path [5]	List of nodes on the key path	10
Key [16]	Encrypted pair-wise key, which has to be established	16
Path-length	Length of the key path	1
Current-node	Node in the key path where currently processing is done	1

- As the length of key path increases, the overhead towards computation and communication also increases. If the key path length is four, the Indirect-Key is established with the help of three intermediate nodes, requiring four times hop-to-hop transmissions, four times encryptions and four times decryptions. The energy expense towards pair-wise key establishment with the parent nodes and with the grand parent have been evaluated. The energy expense is directly proportional to the number of keys established and more energy is spent if the number of Indirect-Keys is more than the number of Direct-Keys. The evaluations are taken for both *Row-wise* and *Diagonal-wise* Grid-ID assignments. It is observed that when the networks size is less than 100, the number Direct-Keys are more in *Row-wise* assignment compared to *Diagonal-wise* assignment. But for larger size networks, *Diagonal-wise* scheme has more number of Direct-Keys. The percentage of Direct-Keys formed with parents the grand parents, compared to the total number of keys formed in the network is shown in Figure 1 and Figure 2 respectively. It is observed that for larger networks, the percentage of Direct-Keys is drastically reduced as the number of nodes in the network increases.

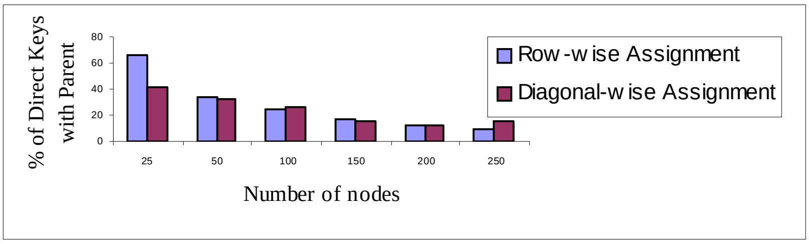

Fig. 1. Percentage of Direct-Keys formed with parents

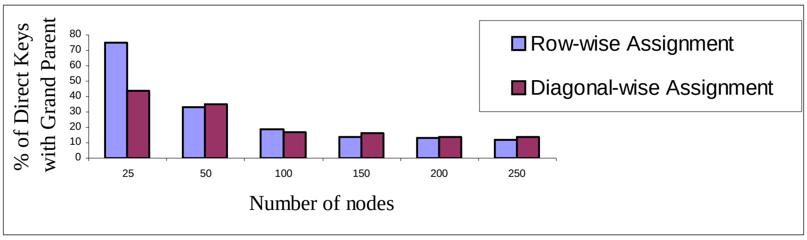

Fig. 2. Percentage of Direct-Keys formed with Grand parents

- The hierarchical trees formed during the simulations are with unequal degree for different sub trees. The maximum degree of these trees is between 6 and 8. As the number of nodes in the network increases, the number of levels in the trees is also increasing. The number of levels in the trees in the simulations with the *Diagonal-wise* assignment is given in Figure 3. The number of keys to be stored at the Base Station is same as total number of nodes in the network. But the number of keys to be stored in many nodes in the next few levels down the Base Station is also very near to number of keys stored at the Base Station. The Base Station can be a high-end node, but all the other nodes in the network are normal resource constrained sensor nodes. The number of keys being stored at the nodes in level 0, level 1 and level 2 in the hierarchical trees of various sizes are shown in Figure 4. As the number of keys needs to be stored at the nodes is high, the memory overhead of the protocol is very high.

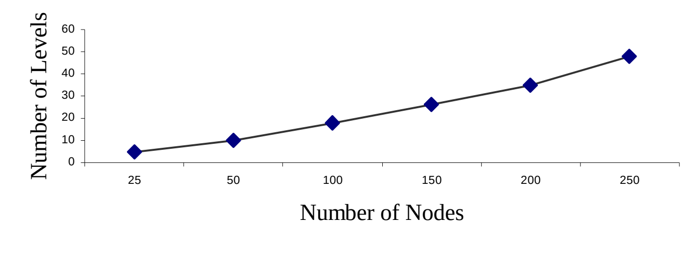

Fig. 3. Number of levels in the hierarchical trees with *Diagonal-wise* assignment

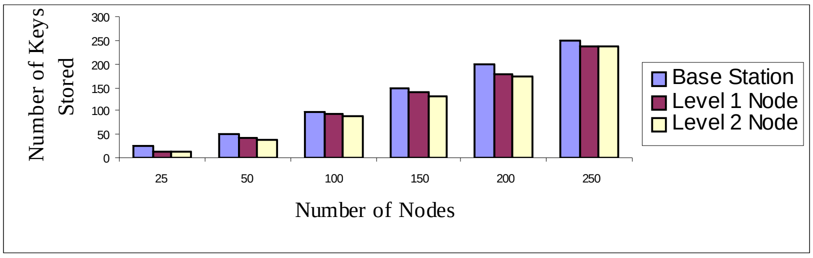

Fig. 4. Highest number of keys stored in various levels

- When a node requests for an Indirect-Key path, The Base Station finds out a key path from the details of list of registered nodes. Many times the Base Station could not find a successful key path because Grid-IDs are assigned either row-wise or diagonal-wise in increasing order of address of the nodes and node registration is done in a random order. So at a particular time, there is a possibility of unavailability of a common node to establish the key path, consequences to a failure in finding a key path and the requesting node is required to repeat the key path request until it could get one. The number of times the key path request has failed for the entire network is also evaluated and the details are given in Figure 5. As the number of nodes increases, the number of failures also increases.

The above protocol evaluation has brought out the deficiencies of permanent key establishment protocols, in general. But it has been observed that as the network size increases, the memory required for key storage increases and also the communication overhead is more due to massive Indirect-Key formation as it is clear from Figure 1 and Figure 2. Sensor networks are very much resource constrained and therefore the memory required for key storage has to be reduced. Further there is no likelihood that all the nodes in the network surely get compromised. Therefore most of the keys established through the permanent scheme do not get utilized as the network gets compromised sparingly in its lifetime. Hence it is preferred to build temporary on-the-fly secure sessions in order to confirm intrusion whenever it is detected.

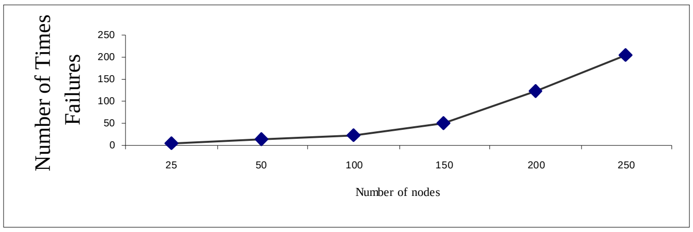

Fig. 5. Failure rate of Indirect Key Path Request

4.2 On-the-fly Secure Session Establishment schemes

Two simple on-the-fly schemes which are resistant to man-in-the-middle attack are proposed in this paper. The first scheme takes the help from the Base Station assuming nobody can compromise the Base Station. The second scheme uses multi-path communication. Since the keys generated for secure communication are used for only one session, their permanent storage in the memory is avoided.

4.2.1 Intrusion Confirmation protocol utilizing Base Station

The proposed scheme is a simple, energy efficient, low memory requirement scheme utilizing the Base Station and the pair-wise keys already established in the sensor networks between any node and the Base Station. These pair-wise keys are established using the protocol presented in our earlier work [7] in which a sensor node authenticates to the Base Station and establishes efficiently two pair-wise keys, one with Base Station and another with its cluster head.

The cluster head CH_k randomly generates a key $K_{CHk\text{-}Si}$ and encrypts the key using the pair-wise key between CH_k and the Base Station. The *Intrusion-Confirm* request message is prepared including the request type, encrypted $K_{CHk\text{-}Si}$ and digital signature and the message is sent to the Base Station. The Base Station decrypts the pair-wise key, re-encrypts it with the pair-wise key shared with S_i and forwards the request message to S_i. Once the request message reaches S_i, it decrypts the pair-wise key and prepares the *Intrusion-Confirm* reply message. The reply includes the actual value supplied by the node and its parent ID. The reply is encrypted using $K_{CHk\text{-}Si}$ and directly sent to CH_k, which decrypts the reply and stores the reply for further analysis. The *Intrusion-Confirm* request/reply protocol will continue with the parent of S_i until the replies are obtained from all the nodes in the aggregation tree path from S_i to CH_{k+1}. The protocol is given in Table 3. The digital signature generation and verification are included to preserve integrity and origin authentication. They use Elliptic curved based algorithm and the public keys which would have already been distributed as in our earlier work [12]. ECDSAG and ECDSAV denote the generation and verification of digital signature.

Table 3. Intrusion Confirmation through the Base Station

1 Cluster head CH_k when suspecting intrusion by CH_{k+1} on the data supplied from S_i, prepares *Intrusion-Confirm* request message.

- $K_{CHk\text{-}Si} = Genkey\ ()$; $(r, s) = ECDSAG$ (Request-type || S_i || $K_{CHk\text{-}Si}$, Pr_{CHk})
- $CH_k \rightarrow BS$: Request-type || S_i || $E_{KCHk\text{-}BS}$ ($K_{CHk\text{-}Si}$) || (r, s)

2 When *BS* receives the *Intrusion-Confirm* request from CH_k

- Verifies digital signature by *ECDSAV* (Request-type || S_i|| $K_{CHk\text{-}Si}$, r, s, Pu_{CHk})
- $(r, s) = ECDSAG$ (Request-type || CH_k || $K_{CHk\text{-}Si}$, Pr_{BS})
- $BS \rightarrow S_i$: Request-type || CH_k || $E_{KBS\text{-}Si}$($K_{CHk\text{-}Si}$) || (r, s)

3 When S_i receives the *Intrusion-Confirm* request from *BS*, prepares *Intrusion-Confirm* reply message.

- Verifies digital signature by *ECDSAV* (Request-type ||CH_k || $K_{CHk\text{-}Si}$, r, s, Pu_{BS})
- $S_i \rightarrow CH_k$: Reply-type, $E_{KCHk\text{-}Si}$ (Data || *Cluster_ Head* (S_i))

4. When Cluster head CH_k receives the *Intrusion-Confirm* reply from S_i

- Decrypts the reply and stores (Data, S_i) for further analysis.
- Continue steps 1, 2 and 3 for *Cluster_Head* (S_i) until (*Cluster_ Head* (S_i) $\neq CH_k$)

4.2.2 Intrusion Confirmation protocol using Multi-path

In multi-path communication based scheme, CH_k finds out all possible routes between CH_k and S_i. The scheme utilizes the pair-wise keys already established in the sensor network between any cluster member and its cluster head by using the protocol presented in [7]. CH_k randomly generates a session key and prepares an *Intrusion-Confirm* request message to include the type of request and the encrypted session key. The request message is forwarded through multi-paths to S_i in a secure way by utilizing the pair-wise keys. Using the session key the source node encrypts the Intrusion-Confirm reply and returns the replies back to cluster head CH_k by using multi-paths.

For example let one such path be: CH_1 - CH_2 - CH_3 - S_4. CH_4 randomly generates the session key $K_{CH1\text{-}S4}$ to be established, encrypts it with the pair-wise key between CH_1 - CH_2 and forwards the request message to CH_2. It decrypts the session key and re-encrypts it with the pair-wise key between CH_2 - CH_3. Finally the request will be reaching S_4. It decrypts the session key, prepares the reply by encrypting with $K_{CH1\text{-}S4.}$ Once all the *Intrusion-Confirm* request messages from different routes arrive at S_4, it sends a common reply message to CH_1 through separate paths using multi-path communication. When CH_1 receives the reply, decrypts it and stores the reply. A compromised node in one of the routes may try to influence the reply. Therefore that reply may be different from all the other replies forwarded through other routes. By proper analysis, the requested cluster head can conclude the occurrence of intrusion.

4.2.3 Comparison of Key Establishment Protocols

The basic characteristics of the three protocols used for key establishment towards intrusion confirmation proposed in this paper lead us to provide a comparison of the protocols as given in Table 4. It brings out the superiority of the on-the-fly schemes in general and the on-the-fly scheme using the Base Station, in particular.

Paradoxically itself there are possibilities of a compromised cluster head executing the intrusion confirmation protocol many times to drain the battery power in the sensor nodes in the aggregation tree path. Only a scheme that involves the Base Station can detect this attack since the highly resourced Base Station can store the relevant Intrusion-Confirm request messages and can make an analysis to determine the event and establish the culprit. This attack is hard to detect in the permanent key scheme and in the multi-path based scheme and requires further study.

Table 4. Comparison of Key Establishment Protocols

Parameter	Post-deployment Permanent key scheme	On-the-fly scheme using	
		Base Station	Multi-path
Key Establishment	Initial phase	On-the-fly	On-the-fly
Security against man-in-the-middle attack	√	√	√
Security against multiple times execution of intrusion confirmation	X	√	X
Intervention of Base Station	X	√	X
Communication overhead for key establishment	High	Low	Depends on number of paths
Communication overhead for Intrusion Confirmation	Very low	Low	Medium
Memory for additional key storage	High	Nil	Nil
Speed of response after attack	Immediate	Slight Delay	Moderate Delay
Probability of failure in key establishment	>0	= 0	= 0
Energy expense	High	Low	Low

5. Intrusion Defense Scheme

We suggest a simple and practical intrusion defense scheme. The cluster head analyzes the *Intrusion-Confirm* Replies and if intrusion is confirmed, it sends an alarm message called *Intrusion-Alert* to the Base Station. The structure of the message is as shown in Table 5. When the Base Station receives the alert message, it increments the alert counter for this suspected node. When the alert counter value reaches a threshold, it verifies the validity of the intrusion directly with the suspected node because a group of compromised nodes can also supply false intrusion detection alerts about a legitimate node. As a second line of defense to verify the claim, the Base Station sends an encrypted request to the suspected node to return the latest aggregated value supplied from the cluster of S_i. If intrusion is confirmed, it decides to lower the level of trust to the compromised node and intimates the new trust level to all the neighbors of the compromised node. When a neighbor node receives the trust level message, it reduces the communication towards the compromised node.

Table 5. Structure of Intrusion-Alert Message

CH_k	CH_{k+1}	Data supplied by CH_{k+1}	S_i	Data supplied by S_i

6. Conclusion and Future Work

A new watchdog mechanism to detect node compromise attacks is proposed for a network that provides aggregation and security services. Also to detect false alarming, we require an intrusion confirmation protocol, which directly contacts all the nodes in the required aggregation tree path in a secure way and collects the details. We have proposed mainly two schemes. The first scheme is a post-deployment permanent key scheme, which uses polynomial distribution method to establish the required pair-wise keys. But this requires huge memory for key storage and high communication overhead. So we have proposed two simpler and energy efficient on the fly schemes: one by involving the Base Station and another by using multi-path communication. We have also suggested a feasible intrusion defense mechanism. The future work includes implementation and evaluation of the intrusion detection, confirmation analysis and defense schemes.

References:

1. Vijay Subhash Bhuse, Lightweight Intrusion Detection: A Second Line of Defense for Unguarded Wireless Sensor Networks, PhD Thesis, Western Michigan University, 2007.
2. Krontiris Ioannis, Tassos Dimitriou and Felix C. Freiling, Towards Intrusion Detection in Wireless Sensor Networks, Proceedings of 13th European Wireless Conference, 2007.
3. D. Liu and Peng Ning, Establishing Pair-wise Keys in Distributed Sensor Networks, Proceedings of ACM Conference on Computer and Communication Security, 2003.
4. Eschenauer and Gligor, A Key-management Scheme for Distributed Sensor Networks, Proceedings of ACM Conference on Computer and communication Security, 2004.
5. Chan, H., Perrig, A. and Song D., Random Key Pre-distribution Schemes for Sensor Networks, Proceedings of IEEE Symposium on Research in Security and Privacy, 2003.
6. Lingxuan Hu and David Evans, Secure Aggregation for Wireless Networks, Proceedings of Symposium on Applications and the Internet Workshops, pp. 384, January 2003.
7. Jibi Abraham and K S Ramanatha, An Efficient Protocol for Authentication and Initial Shared Key Establishment in Clustered Wireless Sensor Networks, Proceedings of 3rd IEEE/IFIP International Conference on Wireless and Optical Communications Networks, April 11-13, 2006, Bangalore, India.
8. Philip Levis, Nelson Lee, Matt Welsh and David Culler, TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications, Proceedings of ACM Conference on Embedded Networked Sensor Systems, 2003, pp. 126-137.
9. Chris Karlof and David Wagner, Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures, Proceedings of IEEE International Workshop on Sensor Network Protocols and Applications, May 2003, pp. 113-117.
10. Sasha Slijepcevic, Miodrag Potkonjak, Vlasios Tsiatsis, Scott Zimbeck and Mani B Srivatsa, On Communication Security in Wireless Ad-Hoc Sensor Networks, Proceedings of 11th IEEE International Workshops on Enabling Technologies, pp. 139 – 144, 2002.
11. Jason Hill, Robert Szewczyk, Alec et. All., System Architecture Directions for Network Sensors, Proceedings of International Conference on Architectural Support for Programming Languages and Operating Systems, 2000, pp. 93-104.
12. Jibi Abraham and K S Ramanatha, A Complete Set of Protocols for Distributed Key Management in Clustered Wireless Sensor Networks, Proceedings of IEEE International Conference on Multimedia and Ubiquitous Engineering, pp. 920-925, April 2007, Korea.