An Asynchronous Node Replication Attack in Wireless Sensor Networks

Jianying Zhou, Tanmoy Kanti Das, and Javier Lopez

Abstract Applications of wireless sensor network (WSN) are growing significantly, and many security protocols meant for WSN have been proposed. One of the unique problems of WSN is that the sensor nodes are not tamper resistant as the main attraction of deploying WSN is its low cost. *Node replication attack* exploits this weakness to launch an attack, in which cryptographic secrets from the compromised sensor nodes are used to create duplicate sensor nodes in large number. Then these sensor nodes are placed in critical locations of the WSN to mount attacks. Several protocols were proposed to defend WSN against the replication attack, and one of the promising among them is *distributed detection protocol* presented by Parno et al. at IEEE S&P 2005. However, we show in this paper that their distributed detection protocol is vulnerable to an *asynchronous* node replication attack. Further, we modify the protocol to make it secure for *dynamic* WSN supporting node mobility.

Keywords: Wireless Sensor Network Security, Node Replication Attack, Distributed Detection Protocol.

1 Introduction

Wireless sensor networks are quickly gaining popularity due to the fact that they are potentially low cost solutions to a variety of real-world challenges [1] and they provide a means to deploy large sensor arrays in a variety of con-

Jianying Zhou
Institute for Infocomm Research, Singapore, e-mail: jyzhou@i2r.a-star.edu.sg

Tanmoy Kanti Das
e-mail: dastanmoy@gmail.com

Javier Lopez
Computer Science Department, University of Malaga, Spain, e-mail: jlm@lcc.uma.es

ditions capable of performing both military and civilian tasks. However, due to inherent constraints of resources (computing, communication, and storage), security in WSN poses different challenges than traditional network/computer security [9, 15].

The security threats to WSN and the countermeasures have been studied intensively in the recent years [2]. Many of the identified attacks are generic in nature and may not work under every operational considerations. Similarly, the proposed defenses against these attacks are also generic in nature. Thus there is a need to analyze those countermeasures in different operational scenarios to verify their effectiveness. In this paper, we study the defenses against the node replication attack under certain operational conditions. We show that under these operational conditions the proposed defenses are inadequate to thwart our *asynchronous* node replication attack as defined in this paper.

The rest of this paper is organized as follows. In the next section, we survey the related work on WSN security. In section 3, we review the existing techniques for detection of node replication. In section 4, we present a new node replication attack which defeats the detection mechanism. In section 5, we suggest several modifications to make the existing distributed detection protocol secure. Section 6 concludes the paper.

2 Related Work

Most of WSNs consist of off-the-shelf low cost hardware without any tamper-proof capabilities. Thus sensor nodes are vulnerable to abuse and compromise. In fact, it is pointed out in [7] that MICA2 sensor nodes can be compromised within a minute. Thus compromising a node for mounting different kinds of attack is a practical threat to any WSN. Particularly the threat is really serious where sensor nodes are deployed in adversarial conditions. The attacker can capture nodes, replicate at will and place the duplicate nodes in critical network locations to cause maximum disruption in the network operation. It should be noted that a single captured node is enough to mount these sort of attacks. Use of tamper-proof hardware is the most easy and effective solution of the problem. However, tamper-proof hardware is costly and it is simply not economical to deploy such sensor networks. So we have to look for other avenues to resist this kind of attack.

Researchers are looking into security of WSN from different perspective. Lots of attacks and countermeasures are proposed in the existing literature [3, 4, 16, 18, 19]. McCune et al. [10] highlighted the *Denial-of-Message* (DoM) attack in which a set of nodes act maliciously and prevent broadcast messages to reach certain section(s) of the WSN. They also proposed Secure Implicit Sampling algorithm to detect such attacks. Newsome et al. [13] proposed the defenses against *Sybil attack* in which a single node takes on multiple identities

to deceive other nodes. Hu et al. [8] presented an algorithm, known as packet leashes, to defeat the *wormhole attack* in which the attacker captures message bits at one location and replays them in another location. Karlof et al. [11] discussed several attacks and countermeasures against routing protocols in WSNs.

To effectively resist any attack, we have to understand the goals and motivation of the attacker first. Let us consider a WSN deployed in hazardous and adversarial conditions like in a battle field. In this situation, any adversary's goal may include any/all of the following.

- Find out the network topology.
- Learn about the data collected by the sensor nodes.
- Inject false data to mislead the enemy.
- Bring down the network if possible.

It is assumed that it is not possible for the attacker to physically remove all sensor nodes. However, the attacker can capture a few of them. Thus to achieve his goals, the attacker may look for other options. One of the options available to him is to capture as many nodes as possible and turn them into malicious nodes. One technique that can be used is to quickly replicate the captured nodes and insert the duplicate nodes in strategic locations within the WSN to achieve any/all of the above goals.

In this paper, we consider a competitive scenario where two rival WSNs are deployed. The primary goal of both networks is the same: to observe some physical phenomenon. The secondary goal is to mount attacks on the rival network. Example of this type of competitive environment exists in battle field monitoring systems. Here nodes of a WSN collaborate to mount attacks on the other WSN. Attacks may be as simple as to read the secret data from the other network or more severe ones which incapacitate the attacked network to function. Under this operational scenario, we investigate the effectiveness of available defense mechanisms. We assume that the adversary can only capture a few nodes and the number of captured nodes are insignificant compared to the total number of deployed nodes. We show that the existing protocols are insufficient to prevent a new node replication attack effectively in *dynamic* WSN supporting node mobility. Our attack is different from the existing node replication attacks as at no point of time the number of malicious nodes directly involved in mounting the attack is greater than the number of captured nodes. Thus, one may argue this attack cannot be termed as node replication attack. However, the number of nodes directly involved in the attack over a period of time is much much higher than the number of captured nodes. Thus we consider this attack as a *variant* of the classical node replication attack.

3 Detection of Node Replication Attack

There are several techniques available in the existing literature to resist the *node replication attack* described above. They can be broadly categorized into three different categories: *localized* approach, *centralized* approach, and *distributed* approach [14].

In localized detection technique, neighbours of a node vouch for this node's location by voting [13]. However, this approach cannot detect the distributed node replication where the replicated nodes are more than two hops away. Other method that can reliably detect the node replication is based on centralized approach. Here each node sends its neighbours' claimed location information to the base station for verification. This method can effectively detect the node replication attack but nodes near the base station bear the burnt of excessive communication. Also nodes near the base station are subject to subversion by the attacker as failure of these nodes cripples the WSN. Thus distributed approach where all nodes in the WSN share the burden of detection, is the most preferred solution.

Parno et al. [14] presented a distributed detection and prevention mechanism for node replication attack. The *randomized multicast protocol* described in [14] is based on birthday paradox. In this protocol, each node sends its location claim having format $< ID_\alpha, l_\alpha, \{H(ID_\alpha, l_\alpha)\}_{K_\alpha^{-1}} >$ to its immediate neighbours. Here ID_α, l_α are id and location of node α, respectively; $K_\alpha,\ K_\alpha^{-1}$ are public and private keys of the node α, respectively; $H(M)$ is the hash of message M, and $\{M\}_{K_\alpha^{-1}}$ indicates α's signature on M. Upon receiving a location claim from its neighbour α, a node verifies signature of α on it and with probability p, it selects g random locations within the network. Then it forwards the location claim to the witness nodes near the selected locations using *geographic routing* [12, 17]. Similarly, neighbours of the replica α' also forward the location claim to the witness nodes. Based on the birthday paradox, it can be assumed two nodes with the same id but different location will have at least one common witness who will flood the network about the conflicting location claims. This will in turn exclude all the malicious nodes having id ID_α from the network. It was found that if a network consists of $n = 10000$ nodes and if g=100, average degree of each node $d = 20$ and $p = 0.05$ is the probability that a neighbour will forward a location information, then the probability of detecting a single node replication is 63%. If the node is replicated twice the probability of detection is greater than 95%. The communication cost of the protocol is of the order of $O(n^2)$.

A more efficient version of the randomized multicast protocol is *line selected multicast protocol* [14]. As we know, to send a piece of information from node α to β, the information should travel through several intermediate nodes as nodes in a WSN not only act as a data collection unit but also act as a router. When a location claim travels from node α to β, all the intermediate nodes in the path are aware about that particular location claim. Thus ever

a conflicting location claim crosses the path, these intermediate nodes can detect the conflict and inform others about it. This is the basic idea behind the line selected multicast protocol. It was found that the expected number of intersections c, between x randomly drawn lines (i.e. paths) within the unit circle is given by [14]

$$E(c) = x(x-1)\left(\frac{1}{6} + \frac{245}{144\pi^2}\right)$$

The exact protocol is as follows [14].

1. Let $r = p \cdot d \cdot g$, where p is the probability that a neighbour will forward a location information and d is the average degree of each node.
2. Each location claim from a node α is forwarded to r nodes by α's neighbours.
3. All the intermediate sensor nodes through which these claims travel to reach their intended recipient also store these claims in their buffer. And these intermediate nodes thus act as additional witnesses.
4. After receiving a location claim from node α', the witness node checks for the existence of similar claims among the claims already in its buffer. Here, by similar we mean that the two claims have the same id. If a similar claim α with a different location already exists, the witness node informs all other nodes about them. Consequently, α and all α's are excluded from the network.

The advantage of line selected multicast over randomized multicast comes from the fact that nodes along the path through which the claims travel also act as witnesses. It was found during Monte-Carlo simulations that if we have two paths for α originating from α's neighbour and two for α', then the probability of intersection is 56%. This probability increases to 95% if we have five such paths instead of two. Communication overhead for the entire network is $O(n\sqrt{n})$, assuming that the length of each path is $O(\sqrt{n})$. Under the similar assumption, each node requires to store $O(\sqrt{n})$ location claims.

Both the protocols described above require a loose notion of synchronization for proper detection of node replication attack. The frequency of execution of detection protocol depends on several conflicting requirements like detection efficiency, storage and communication costs etc.. In one variant of the protocol, detection algorithm runs after fixed interval T and it takes t unit of time for detection algorithm to complete. Note that $T >> t$. After the execution of the detection algorithm, all the nodes only remember the identities of revoked nodes. However, there is a lacuna in this approach. Node replication can be mounted between the two detection passes. To overcome this, all nodes also remember the list of its valid neighbours detected during previous run of detection protocol and all nodes refuse to communicate with a node unless it participates in a detection pass. In another approach, time is divided into T epochs consisting of k time slots. During each time slot, a

fixed number of nodes announce their location and the standard protocols are followed thereafter. All the protocols presented here assume that each node got at least one legitimate neighbour. Otherwise, masked replication attack can be mounted, though one can easily defeat the masked replication attack with the use of pseudo-neighbour(s) [14].

4 Asynchronous Node Replication Attack

As described earlier that we are considering a competitive environment where more than one WSN is deployed. This type of situation may be available in military applications like battle field or border monitoring system. Each of the WSN has a specific set of goals and one important goal is to mount attacks on the enemy network to prevent it from functioning normally or read data from the enemy network. Cryptographic information retrieved from the captured node(s) most of the time is used for mounting such an attack. One of the widely used techniques to mount an attack based on information retrieved from the captured nodes is node replication, and here we study the protocols used to detect such an attack.

In the classical node replication attack, it is assumed that the adversary will replicate a captured node and will insert large number of duplicate nodes for malicious purpose. All the duplicate nodes are present simultaneously in the network. Distributed detection mechanism succeeds because of the presence of nodes with the same id in different locations. Classical node replication attack is very straightforward and the defense mechanism is also simple. We like to investigate the effectiveness of such defense mechanisms against an *asynchronous node replication attack* where the number of nodes actively mounting the attack at any specific point of time is not greater than the number of captured nodes, but over a period of time the total number of nodes actively participating in mounting the attack is far greater than the total number of captured nodes.

Let us denote the original WSN as "blue" network and the WSN deployed by the adversary as "gray" network. Besides the nodes actively participating in mounting an attack, all other nodes in gray network passively participate in mounting an attack, i.e., they only help active nodes by providing network resources from gray network.

4.1 Basic Strategy

Here we consider that the attacker only possesses a single captured node and blue network is running *distributed node replication detection* protocol. Let us discuss how we can mount a node replication without being detected. We

will generalize this strategy later to consider that multiple captured nodes are available to the attacker. The main idea behind our attack is to use the captured cryptographic secrets by different nodes of the gray network during each detection pass. At any point of time the number of nodes mounting the attack is equal to the number of captured nodes. However, over a period of time, the total number of nodes directly used to mount the attack is much higher than the number of captured nodes. Let us present the attack in a stepwise manner (see Figure 1).

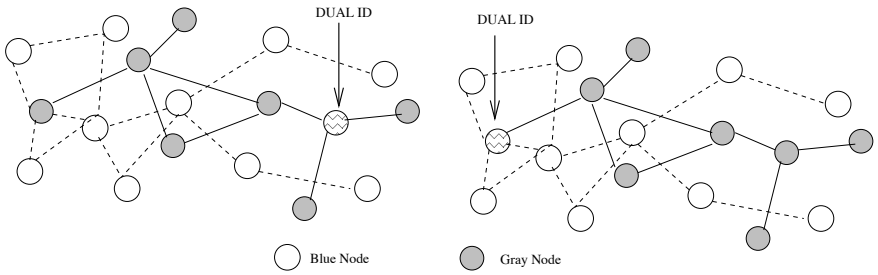

Fig. 1 Dual ID node moves in blue network

1. Deploy the gray sensor network over the same area as that of the blue network. One of the sensor node δ in the gray network holds dual id consisting of the captured id from the blue network and its own id in the gray network.
2. We assume that the nodes in the gray network can securely communicate among themselves, which means authenticity, integrity and confidentiality of message exchanged among gray nodes are guaranteed.
3. The node δ with dual id acts as the gateway between the gray and blue networks. However, the blue network is unaware of it.
4. In each detection pass of the blue network, different nodes from the gray network act as the gateway. At the beginning of the detection phase in the blue network, the nodes in the gray network decide upon a node γ, which will take over the role of the gateway δ. This node then gets all the cryptographic information from the previous gateway and participates in the detection phase as a legitimate node in the blue network. Also the previous gateway δ_p forgets the id used to communicate with the blue network. However, δ_p continues to be a part of the gray network. This way δ evades detection, as well as learns about the blue network topology and other sensitive information about the blue network. To an outside observer monitoring the blue network, it would appear that δ is changing location each time a detection pass runs. However, it would not be possible under a distributed detection environment to detect such a malicious behavior by δ.

Let us now analyze the situation from the blue network's point of view. It seems that everything runs normally. In each detection pass a node with valid id ID_δ is moving to different part of the network. Due to the distributed nature of the detection mechanism it cannot even detect this type of random disappearance/appearance of node δ from different part of the blue network each time the detection phase runs, thus unsuspecting the blue network falls pray to the attack. However, one question remains, what the attacker achieves by mounting the attack. His gain is as follows.

- Though the attacker is able to compromise only one node, he can discover the entire topology of the blue network.
- The attacker can learn about the traffic pattern of the blue network.
- The attacker can identify the nodes which are critical for network-wise communication and whose failure may partition the network.

The attack becomes more critical if the number of captured nodes becomes more than one. Then several δs can collaborate to mount more severe attacks.

4.2 Collaborative Strategy

Now we consider a more realistic scenario where an adversary captures more than one sensor node. Let the number of captured nodes be m and the nodes using the captured identity is denoted by $\delta_0, \delta_1, \ldots, \delta_{m-1}$. Note that, the number of captured nodes are insignificant compared to the total number of deployed nodes. Otherwise, if the adversary controls most of the nodes, he can control the network with ease. In the previous subsection, we have assumed that the adversary is in possession of only one captured node, thus can mount the attack through only one node. Here, we consider more than one captured node is available and they can collaborate to mount the attack. Though the basic strategy remains the same, in the present case, $\delta_0, \delta_1, \ldots, \delta_{m-1}$ can mount coordinated attack. Also, δ_i, δ_j may not be within the radio range of each other but they can communicate with each other using both blue and gray networks. Thus when δ_i, δ_j communicate through the gray network, the communication remains secret from the blue network. Let us present the attack in a stepwise manner (see Figure 2).

1. Let the total number of captured nodes be m. At the beginning of each detection pass, the gray network selects m number of nodes. Selected nodes hold dual id, one for the blue network and the other for the gray network.
2. Let us denote the nodes having dual id as $\delta_0, \delta_1, \ldots, \delta_{m-1}$. Different set of nodes from the gray network may perform the role of dual id nodes during each detection pass. Newly captured ids can also be added to increase the number of captured ids. In reality the attack can be mounted with a single captured id and one can add new captured ids as when available. Thus the attack is dynamic in nature.

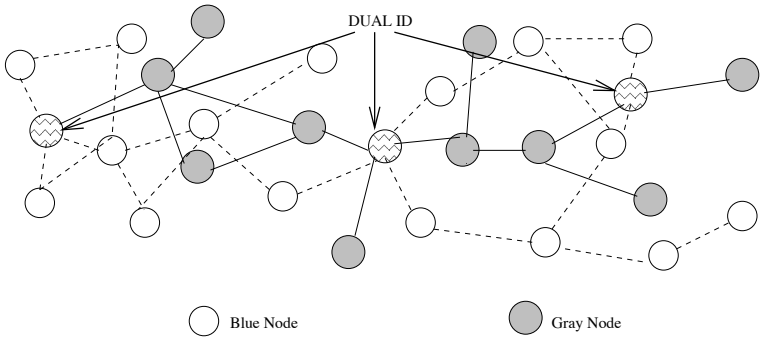

Fig. 2 Attack using several nodes

3. As part of two networks these nodes with dual id can communicate between them through both networks. Say, δ_i needs to communicate something to δ_j without giving any hint to the blue network, it routes that message through the gray network. On the contrary, when they want to communicate something which all nodes in the blue network should be aware, they route the message through the blue network.
4. Consider that a message is received by the node δ_i. According to the routing algorithm it faithfully forwards the message to node β. However, δ_i also sends the message to all other dual id nodes through the gray network. All those dual id nodes upon receiving the message, replay it there. Consequently they mount a *wormhole attack* on the blue network.
5. The gray network can inject bogus data through all δ_i in a coordinated manner to defeat the data aggregation algorithm. We will discuss this in detail afterward.

We assume that the central base station is not responsible for monitoring and supervising operations of the blue sensor network, collective efforts of the sensor nodes are responsible for smooth operation of the network. In fact this is the basis of the distributed node replication detection algorithm of [14]. A very potent attack that can be mounted on the blue network is *coordinated false data injection*. Consider that in the area A_i, certain important event is observed by both blue and gray networks, and the gray network wants to mislead the blue network about the event. All that gray network will do is to turn all the gray nodes in that area (assuming that the number of gray nodes in the area is less than the total number of captured nodes) into dual id nodes. And all the dual id nodes will send false report about the event to the blue network. This may mislead the data aggregation algorithm in the blue network if the ratio of dual id nodes and good nodes in the region A_i is close to 1. If the ratio is greater than 1, then the possibility of misleading the blue network is very high. Afterward the gray network will redeploy the dual id nodes as when required. Note that no physical movement of actual sensor nodes is required, only the captured ids are distributed properly by the

gray network to mount the attack. Similarly, we can place dual id nodes in any critical location of the network without any physical movement of nodes. Thus mounting *denial of service* or *denial of message* attack becomes very easy.

5 Prevention

Before we discuss the prevention mechanism, let us first highlight the weaknesses of the distributed detection approach which was exploited to mount a successful attack. The underlying protocol over which the distributed detection is used to prevent the node replication attack, allows nodes to move from one place to other. In addition to this, distributed detection does not take into account the possible movements of nodes. We exploit this lacuna to mount the attack. Thus to prevent the node replication attack presented in the previous section, either the sensor nodes should not be allowed to move or the protocol to prevent the node replication attack should take into account the mobility of the sensor nodes.

Let us now consider that the sensor nodes remain stationary throughout the entire life of the network, i.e., movements of sensor nodes are not permitted. So the topology of sensor network remains static. Due to the restriction on sensor node movement, the attack presented in the previous section can be resisted as explained below.

- During the attack, cryptographic secrets of captured nodes are sent from one gray node to another. This is done to relocate dual id nodes from one place to another without any physical movement of gray nodes. It appeared to the blue network that certain nodes are moving from one place to another. As the movement of nodes is no longer allowed, the blue network will not allow a node to re-join the network after it changes its location. This effectively thwarts the attack as the attacker cannot insert dual id nodes anywhere in the network. Dual id node can only be inserted at the fixed location, i.e., at the original location of the captured node. Also this cannot be termed as node replication attack either.
- Consequently it is no longer possible to learn about the entire network topology using a small number of captured nodes which reduces the effectiveness of the attack.
- Another important aspect of the proposed attack was to mislead the data aggregation algorithm by injecting coordinated false data. This is done by increasing the concentration of dual id nodes in a particular region. However this is not possible if nodes are to remain stationary, so the attack cannot work. Thus, the restriction on node movement will be able to thwart the attack successfully.

However question remains how practical it is to stop mobility altogether. Some applications may require movement of sensor nodes for operational purpose and restricting the movement of nodes may prove undesirable. After all, the ad-hoc nature of the network is one of the main attractions of deploying such a WSN. Thus a stationary network may be able to prevent the attack, it may not be practical always. On the other hand, if we are to allow the movement of sensor nodes, then it is very difficult to differentiate between the malicious node movements and legitimate node movements and the problem is somewhat equivalent to intrusion detection problem. Also sensor nodes do not have much processing power to make such a complicated decision and it may require the involvement of base station at some point of time.

Another possibility is use of base station to monitor the network. When a sensor network receives a location claim from a new sensor node it forwards the location claim with probability p to the base station. This "new" sensor node may be recent addition to the network or it may be a node moved in from a different neighbourhood. Thus the base station always remains aware about the movement or addition of sensor nodes in the network. Also the base station can take appropriate action if it suspects any foul play by any malicious sensor node. However as pointed in [14], the centralized approach also has its drawbacks. Thus we need to have hybrid approach where we combine the distributed detection approach with centralized monitoring to have a secure protocol. The basic philosophy behind the hybrid approach is to allow the base station take decision regarding the nature of sensor node's movement, i.e., the base station makes the distinction between the normal and malicious node movements. However, the base station never participates in the detection process. Let us first present the extended protocol for distributed detection of node replication and movement in detail.

5.1 Distributed Detection of Node Movement

Here we present the possible modifications required to make the protocol for distributed detection of node replication secure. In the modified protocol, before the end of a detection phase, each sensor node checks whether it received location claims from all of its neighbours. If it was found that location claim from a particular neighbour α was missing, it follows a similar protocol after it receives a location claim. Only difference is that it now indicates a missing sensor node and its previously known location. The modified protocol is as follows.

Just before the end of execution of distributed detection protocol [14], each node β checks whether it heard from all the nodes it heard during the previous detection phase. If it finds that any node α is missing, then with probability p, it selects nodes present in g random locations in the network to forward α's previous location claim. This missing node alert has the format

$< M, \{ID_\alpha, l_\alpha, \{H(ID_\alpha, l_\alpha)\}_{K_\alpha^{-1}}\}, \{H(ID_\alpha, l_\alpha, \{H(ID_\alpha, l_\alpha)\}_{K_\alpha^{-1}})\}_{K_\beta^{-1}} >$. Here M indicates that it is the notification for a missing node. ID_α is the ID of α and l_α is the previous location of α. Nodes are loosely time synchronized, so one can include a time-stamp in the location claim and in the "missing node alert" to prevent any kind of replay attack.

After receiving this missing node notification by a witness node, it verifies all the required signatures to satisfy himself that it is not a forged notification. Then it checks existence of any location claim from α. If there exists a location claim from α in its buffer and it indicates a change of location, the witness node takes "appropriate" action. However, if there is no change in location of α, the witness node simply deletes the missing node notification. A missing node alert may be triggered by the loss of location claim sent by α to β due to communication error. In such a situation witness nodes find that there is no change in location and delete the alert. This eliminates the probability of false positives. Here we have described the required modifications over the randomized multicast protocol. One can also easily modify the line selected multicast protocol in a similar fashion.

After detecting a node movement, it is required to differentiate between a legitimate movement and a malicious movement to decide upon the course of remedial actions. There is no easy solution to this problem as sensor nodes themselves are resource constraint and lack processing power to analyze and decide upon whether present movement is malicious or normal. Another option is to allow limited movement, i.e., every node must stay within a predefined region and if nodes stray beyond the region, they are simply removed from the network. By removal, we mean that no node will communicate with them. However the best possible solution to node movements would be to inform the base station and let the base station analyze the movement to determine whether it is a normal or malicious movement. Thus the protocol no longer remains distributed as it requires the involvement of base station and we call it hybrid protocol.

Security Analysis

It was pointed out earlier that d is the average degree of each sensor node. Thus when a sensor node α changes its location and if all of α's neighbours detect it properly, then $p \cdot d \cdot g$ nodes receive those missing node alerts. Similarly for the current location of α, the number of witness nodes will be $p \cdot d \cdot g$. If there is a common witness between these two different sets of witnesses, then we can detect the movement of the node α. Thus according to the birthday paradox if there is a collision, we can detect the movement. The probability of collision is given by [14, 5]

$$P_c \geq 1 - e^{\frac{-p^2 \cdot d^2 \cdot g^2}{n}}$$

Using the similar setup of [14], i.e., if $n = 10000$, $g = 100$, $d = 20$, and $p = 0.05$, one can detect the movement of sensor node with the probability 63%. And if $p = 0.1$, then one can detect the sensor node movement with probability 98%. Thus the probability of successful detection of node movement is quite high.

5.2 Hybrid Protocol

One drawback of distributed detection of node movement is that it cannot detect the node movement if a node α (i) first refuses to send its location claim in a detection round, (ii) then it moves to a different location and (iii) joins a new neighbourhood in the next detection round. This way α avoids the conflict between the current location claim and the previous location claim (i.e., missing node alert). Hybrid protocol avoids this drawback by involving the base station in the detection process. Hybrid protocol consists of distributed detection of node replication/movement and centralized decision making. Involvement of base station is required to differentiate between the normal and malicious node movements. It is assumed that the base station always maintains a list of all nodes present (including those who had left the network) in the network with their claimed location.

Let us now point out the required modifications over the distributed detection protocol. After receiving the claim from a neighbour α, with probability p it forwards the location claim to g witness nodes. Now, if it is the first time that the neighbour heard from α, the list of witness nodes must include the base station. Thus even without presence of any conflict the base station can also detect both node movement and replication on its own. In the distributed detection of node movements, nodes take appropriate action once they detect movement of sensor nodes. In the hybrid protocol this "appropriate" action is forwarding the two claims, which proves the change of location, to the base station for further action. Upon receiving the claims the base station has to decide whether the present movement is malicious or not. This is somewhat equivalent to behavior-based intrusion detection [6]. Information retrieved from past behaviors constitutes the normal behavior and any major deviation causes an intrusion alert. Thus any unexpected action (i.e., movement not seen before) of the sensor node may cause the base station to revoke it. Note that, addition of base station in the witness list thwarts the attack discussed before.

One drawback of the hybrid system is that the revocation part is deterministic. It is known that after the detection of node movement by any witness node, the claims will be forwarded to the base station for remedial action. At this point a powerful attacker may try to block those messages from reaching the base station. Under these circumstances the protocol looks vulnerable. To overcome this vulnerability, we modify the protocol in the following manner.

After detecting the movement of node α, the witness node broadcasts the location claim and missing node alert to all the nodes in the network including the base station with the request of revocation of node α. Now the base station also receives those messages and analyzes the movements of α. If the movement is in the expected line then it broadcasts a message to reinstate the node α. This protocol is suitable for those networks where node moves occasionally. Otherwise, communication overhead will be too high.

6 Conclusion

In this paper, we presented a detailed analysis of the distributed node replication detection protocol [14]. One of the main motivations behind the development of distributed node replication detection protocol is that it is more secure and robust compared to centralized approach. However, we showed that the protocol is vulnerable against an *asynchronous node replication attack*. Also the communication overhead of the entire network which is of the order of $O(n\sqrt{n})$ for line selected multicast is quite high. We modified their protocol and proposed a *hybrid approach* consisting of distributed detection and centralized monitoring to make it secure even in *dynamic* WSN supporting node mobility.

Acknowledgements The authors are grateful to the anonymous referees of IFIP SEC'08 for their helpful comments. This work is partially funded by the EU project SMEPP-033563.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "A Survey on Sensor Networks". *IEEE Communications Magazine*, 40(8):102-114, 2002.
2. J. Baek, E. Foo, H. C. Tan, and J. Zhou. "Securing Wireless Sensor Networks - Threats and Countermeasures". Book Chapter in *Security and Privacy in Wireless and Mobile Computing*, Troubador Publishing, 2008.
3. M. Cagalj, S. Capkun, and J. P. Hubaux. "Wormhole-Based Antijamming Techniques in Sensor Networks". *IEEE Transactions on Mobile Computing*, 6(1):100-114, 2007.
4. H. Chan, A. Perrig, and D. Song. "Secure Hierarchical In-Network Aggregation in Sensor Networks". *2006 ACM Conference on Computer and Communications Security (CCS'06)*, pp. 278-287, 2006.
5. T. Cormen, C. Leiserson, R. Rivest, and C. Stein. "Introduction to Algorithms". MIT Press, 2001.
6. D. Denning. "An Intrusion Detection Model". *IEEE Transactions on Software*, Vol. SE-13, No. 2, pp. 222-232, 1987.
7. C. Hartung, J. Balasalle, and R. Han. "Node Compromise in Sensor Networks: The Need for Secure System". *Technical Report CU-CS-988-04*, Department of Computer Science, University of Colorado at Boulder, 2004.

8. Y. C. Hu, A. Perrig, and D. B. Johnson. "Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks". *2003 IEEE INFOCOMM*.
9. J. Lopez and J. Zhou (editors). "Wireless Sensor Network Security". *Cryptology & Information Security Series*, Vol. 1, IOS Press, 2008.
10. J. M. McCune, E. Shi, A. Perrig, and M. K. Reiter. "Detection of Denial-of-Message Attacks on Sensor Network Broadcasts". *2005 IEEE Symposium on Security and Privacy (S&P'05)*, pp. 64-78, May 2005.
11. C. Karlof and D. Wagner. "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasure". *AdHoc Networks*, Vol. 1, Issues 2-3, pp. 293-315, Elsevier, September 2003.
12. B. Karp and H. T. Kung. "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks". *2000 ACM Annual International Conference on Mobile Computing and Networking (MobiCom'00)*, pp. 243-254, 2000.
13. J. Newsome, E. Shi, D. Song, and A. Perrig. "The Sybil Attack in Sensor Networks: Analysis & Defenses". *2004 ACM International Symposium on Information Processing in Sensor Networks (IPSN'04)*, pp. 259-268, April 2004.
14. B. Parno, A. Perrig, and V. Gligor. "Distributed Detection of Node Replication Attacks in Sensor Networks". *2005 IEEE Symposium on Security and Privacy (S&P'05)*, pp. 49-63, May 2005.
15. A. Perrig, J. Stankovic, and D. Wagner. "Security in Wireless Sensor Networks". *Communications of the ACM*, 47(6):53-57, Special Issue on Wireless Sensor Networks, 2004.
16. A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. "SPINS: Security Protocols for Sensor Networks". *Wireless Networks*, Vol. 8, pp. 521-534, 2002.
17. S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenkar. "GHT: A Geographic Hash Table for Data-Centric Storage". *2002 ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, September 2002.
18. D. Wood and J. A. Stankovic. "Denial of Service in Sensor Networks". *IEEE Computer*, Vol.35, No. 10, 2002.
19. F. Ye, H. Luo, S. Lu, and L. Zhang. "Statistical En-route Filtering of Injected False Data in Sensor Networks". *IEEE Journal on Selected Areas in Communications*, 23(4):839-850, April 2005.