A Bloom Filter and Matrix-based Protocol for Detecting Node Replication Attack

Weimin Xie, Liejun Wang, and Mingwei Wang
School of Information Science and Engineering, Xinjiang University, Urumqi, China
Email: xingjixieweimin@163.com, wljxju@xju.edu.cn, wang_walfred@sina.com

Abstract—A crucial problem in the security of wireless sensor network is that the nodes are subject to several kinds of attacks and threat like node replication attack, and it may bring serious consequence. To mitigate the damage brought by the node replication attack, in this paper, we propose a replication detection protocol that can discriminate the replicated nodes through detecting geographic fingerprints collision output from the bloom filter. If the receiver node found a conflicting location claim, that indicated there was an aggressor, then the node replication attack was detected. Moreover the usage of matrix decomposition ensured the security of the network, and also it provided the communication link verification.

Index Terms—Sensor Network; Matrix Decomposition; Bloom Filter; Fingerprint; Security

I. INTRODUCTION

A wireless sensor network (WSN) is a distributed ad hoc network constituted by a large number of tiny-size, low-cost, and resource-constrained sensor nodes. It is expected to serve for the military and civilian tasks, which are usually deployed in a harsh or unattended environment, such as environmental monitoring, target tracking, bridge inspection, agricultural irrigation and so on [1]. Since data is commonly transmitted over a wireless channel in WSN, the security of data communications becomes a critical issues for those WSN-based applications. Due to cost concerns, current sensor nodes generally lack hardware protection for tamper-resistance, and thus are vulnerable to node capture attacks if a node is captured, the adversary tends to replicate a counterfeited node and drop back to the network, such that he can eavesdrop, distort or even forge the data, thereby control the entire network [2]. Therefore, how to detect the node replication and find the forged illegal node is one of the fundamental tasks towards a secure WSN. Motivated by the serious threats from the node replication attacks, in this paper, we present a new algorithm aiming at node replication attack detection by leveraging bloom filter and matrix QR decomposition. Our scheme is designed on the basis of a common principle of the bloom filter, that is, the same node can’t have two locations. Thus, the nodes can calculate the fingerprint of itself by using the bloom filter, then, they exchange its fingerprint and Q information with their direct neighbor and cluster, and verify the fingerprint and Q information.

This paper is organized as follows. Section II we study the related research and analysis their shortage. In section III we introduce the network model, node replication attack, matrix QR decomposition and give the definition of a bloom filter. In section IV we detail our replication detection algorithm and the process of the detection. Section V provides performance analysis. Section VI concludes this paper.

II. RELATED WORKS

As a fundamental topic in WSN security, the node replication attacks attract many attentions from researchers to explore better schemes for the detection of the replicated nodes [3-8]. Unfortunately, all these schemes are inevitably constrained by other factors, such as time synchronization, geographic routing mechanisms [9], or other geographic partition methods. All of these factors bring additional computation burdens to the resource limited sensor nodes, which will directly result in a performance degradation of the detection accuracy. Besides, some other schemes detect replicated nodes by periodically scanning which makes the adversary have an opportunity to insert replicated nodes into the network between two scanning.

Since the node replication attack detection problem is a basic problem of WSNs, and researchers around the world have put forward some key schemes. The most simple scheme to solve node replication attacks is using centralized scheme in which each node joining the network should broadcast a signed location claim to its neighbors. At least one of its neighbors forwards this location claim to the base station or a central party. If the base station receives more than one location claims with the same identity but different locations, that means there is a conflicting location claims, then the base station detects node replication attacks and broadcasts a message to the whole network to revoke the replicated node. The centralized scheme achieves 100% detection of node replication attacks. However, the centralized scheme has some disadvantages in terms of communication and memory costs. Furthermore, nodes closest to the base station suffer from high communication overhead for routing packets to the base station. This decreases the lifetime of battery-powered sensor nodes.

Another straightforward scheme that also achieves 100% detection of node replication attacks is node-to-network broadcasting approach in which
neighbors of a joining node broadcast the location claim to the whole network and every node stores the location claim. Once any node receives conflicting location claims, it uses conflicting location claims as an evidence to revoke replicated nodes or send a report to the base station for handling the problem. This approach achieves distributed detection. However, similar to the centralized scheme, this scheme requires high communication and memory costs as well. This is impractical to the resource-limited WSNs.

Parno proposed the first work to address the node replication attacks [10]. They proposed two protocols: Randomized Multicast and Line-Selected Multicast. In Randomized Multicast, each node broadcasts a location claim to its neighbors. According to Birthday Paradox [11], at least one witness node is likely to receive conflicting location claims when replicated nodes exist in the network. In order to reduce the communication costs and increase the probability of detection, they proposed Line-Selected Multicast protocol. It use the fact that sensors act as routers to reduce the communication overhead of the previous protocol. In this protocol, nodes in the path from each neighbor forwarding the location claim of a node N, to the randomly chosen witnesses, stores the location claim of N. As consequence, each path from a neighbor of N to a witness node constitutes a line. If a conflicting location claim for a node N crosses a line, then the attack is detected.

Conti solve the crowded center problem by introducing an network-wide pseudo random number seed that is periodically renewed and must be known instantaneously to all nodes in the network [12]. The infrastructure for distributing such a pseudo-random number seed (such as a satellite or a broadcasting ground station) may not always be available.

Zhu proposed two more efficient distributed protocols for detecting node replication attacks: Single Deterministic Cell (SDC) and Parallel Multiple Probabilistic Cells (P-MPC) [13]. Both protocols need the sensor network to be a geographic grid, each unit of which is called a cell. In SDC each node’s ID is uniquely mapped to one of the cells in the grid. When executing detection procedure, each node broadcasts a location claim to its neighbors. Then each neighbor forwards the location claim with a probability to a unique cell by executing a geographic hash function [14] with the input of node’s ID. Once any node in the destination cell receives the location claim, it floods the location claim to the entire cell. Each node in the destination cell stores the location claim with a probability. Therefore, the clone nodes will be detected with a certain probability since the location claims of clone nodes will be forwarded to the same cell. The difference between SDC and P-MPC is the number of destination cells. In P-MPC the location claim is forwarded to multiple deterministic cells with various probabilities by executing a geographic hash function with the input of node’s ID. The rest of procedure is similar to SDC. Therefore, the clone nodes will be detected with a certain probability as well.

Bekara and Laurent-Maknavicious proposed a new protocol for securing WSN against nodes replication attacks by limiting the order of deployment [15]. Their scheme requires sensors to be deployed progressively in successive generations. Each node belongs to a unique generation. In their scheme, only newly deployed nodes are able to establish pair-wise keys with their neighbors, and all nodes in the network know the number of highest deployed generation. Therefore, the clone nodes will fail to establish pair-wise keys with their neighbors since the clone nodes belong to an old deployed generation [16].

From above, we can see that lots of approaches need much memory overhead or computation overhead.

III. PRELIMINARIES

A. Network Structure

The whole network is composed of base station, cluster head and sensor node. The network is divided into several clusters; each cluster consists of a cluster head and a set of sensor nodes. In this scheme, the clustered structure is used. The relationship is illustrated in figure 1.

Figure 1. The structure of the network

The base station takes charge of the whole network, and it cannot be copied. Each sensor node has limited resource such as limited computational, limited memory storage capacity, and very short radio transmission range. Cluster heads are equipped with higher resource than sensor nodes. We assume that the nodes are stationary after deployment. Neighboring nodes form wireless links and data communications between them are protected by preloaded key materials. Each node knows its own geographic location and its neighbors’ locations. Many localization schemes can be used to provide such location information [17, 18].

B. Threat Model

We assume that the deployment region is neither under the total control of friendly forces nor under the control of the enemy forces. In examining the security of a sensor network, we take a conservative approach by assuming that the adversary has the ability to surreptitiously capture a limited number of legitimate sensor nodes. We limit the percentage of nodes captured, since an adversary that can capture most or all of the nodes in the network can obviously subvert any protocol running in the network. Having captured these nodes, the adversary can employ arbitrary attacks on the nodes to extract their private information. For example, the adversary might exploit the unshielded nature of the nodes to read their cryptographic information from memory. The adversary could then clone the node by loading the node’s cryptographic
information onto multiple generic sensor nodes. Since sensor networks are inherently designed to facilitate ad hoc deployment, these clones can then be easily inserted into arbitrary locations within the network, subject only to the constraint that each inserted node shares at least one key with some of its neighbors. We allow all of the nodes under the adversary’s control to communicate and collaborate, but we make the simplifying assumption that any cloned node has at least one legitimate node as a neighbor.

C. Matrix QR Decomposition

If non-singular matrix \( K \) can be presented as a product of an orthogonal matrix \( Q \) and an upper triangular matrix \( R: K = QR \), then this process can be called QR decomposition \([19]\). The \( Q \) is an orthogonal matrix composed of row vector group (its columns are orthogonal unit vectors meaning \( Q^T Q = I \)), while the \( R \) is an upper triangular matrix composed of column vector group.

In our proposal, the QR decomposition can be used for node-to-node authentication (see section IV).

D. Bloom Filter

Bloom filters are compact data structures for probabilistic representation of a set in order to support membership queries. It was invented by Burton Bloom in 1970 \([20]\).

![Figure 2. A bloom filter with 4 hash function](image)

The idea (illustrated in Figure 2) is to allocate a vector \( v \) of \( m \) bits, initially all set to 0, and then choose \( k \) independent hash functions, \( h_1, h_2, h_3 \ldots h_k \), each with range \( \{1, \ldots, m\} \). For each element \( a \in A \), the bits at positions \( h_1(a), h_2(a), \ldots h_k(a) \) in \( v \) are set to 1 (A particular bit might be set to 1 multiple times.) Given a query for \( b \) we check the bits at positions \( h_1(b), h_2(b), \ldots h_k(b) \). If any of them is 0, then certainly \( b \) is not in the set \( A \). But this compact representation is the payoff for allowing a small rate of false positives in membership queries; that is, queries might incorrectly recognize an element as member of the set.

One prominent feature of bloom filters is that there is a clear tradeoff between the size of the filter and the rate of false positives. To summarize: Bloom filters are compact data structures for probabilistic representation of a set in order to support membership queries. If the location is regarded as an element of bloom filter, then it can be used to detect the node replication attack \([21]\).

IV. DETAILS OF DETECTION

In this section, we present our scheme for detecting node replication attacks in detail. The detection scheme consists of three phases: Generating initial information, computing a fingerprint for each sensor based on its social network, and then detecting clone attacks afterwards.

A. Generate the Initial Information

In order to detect the node replication attack, some initial information should be generated as the following steps.

Step 1: Generate symmetric matrix. \( N \) is the maximum number of nodes to be deployed in the cluster. The server generates a symmetric matrix with a size of \( N \times N \).

Step 2: Decompose the symmetric matrix. The QR decomposition is used to decompose the symmetric matrix generated in last step. \( Q \) is an orthogonal matrix, the row vector groups are orthogonal to each other. \( R \) is an upper triangular matrix. The generated symmetric matrix is shown in formula (1), where \( k_{ij} = k_{ji} \).

\[
\begin{bmatrix}
  k_{11} & k_{12} & \cdots & k_{1N} \\
  k_{21} & k_{22} & \cdots & k_{2N} \\
  \vdots & \vdots & \ddots & \vdots \\
  k_{N1} & k_{N2} & \cdots & k_{NN}
\end{bmatrix}
= \\
\begin{bmatrix}
  Q_{11} & Q_{12} & \cdots & Q_{1N} \\
  Q_{21} & Q_{22} & \cdots & Q_{2N} \\
  \vdots & \vdots & \ddots & \vdots \\
  Q_{N1} & Q_{N2} & \cdots & Q_{NN}
\end{bmatrix}
\times
\begin{bmatrix}
  R_{11} & R_{12} & \cdots & R_{1N} \\
  R_{22} & \cdots & R_{2N} \\
  \vdots & \vdots & \ddots \\
  R_{NN}
\end{bmatrix}
\]

(1)

Step 3: Every sensor node including the cluster chooses one row from the \( Q \) matrix and one column from the \( R \) matrix; both the row and column which are chosen by the node have a same number. For example, the node \( A \) chooses \( Q_i \) from \( Q \), then it should choose \( R_j \) from \( R \); node \( B \) chooses \( Q_j \) from \( Q \), then it should choose \( R_i \) from \( R \).

Step 4: Each node maintains two bloom filters, one is used to calculate the fingerprint of itself, while the other keeps its neighbor’s fingerprint. Bloom filter is a \( m \)-dimensional binary array. As shown in the figure 3, all \( m \) bits are initialized to zero.

![Figure 3. A bloom filter](image)
Step 5: After deployment, all the nodes calculate their location coordinate information. In this step, this localization schemes in literature [3, 4] can be used to provide such location information.

B. Generate the Fingerprint

After deployment, the nodes exchange their fingerprint and Q information with their direct neighbor and cluster, so every node keep their neighbors’ fingerprint. The fingerprint can be calculated by its localization using bloom filter. The process describe as follows:

The bloom Filter uses k independent hash functions, these hash functions will map the location fingerprint to the array \{1, ..., m\}.

For any location fingerprint \( x \), the bit in the array mapped by the i-th hash function will be setted to 1(1≤i≤k).

If one bit has been setted to 1 for more than one time, then it would only works for the first time, the followed will have no effect.

The node applied k-times hash function on its own geographical coordinates, it will produce an m-bit binary array, then the array is the fingerprint.

In figure 4, for example, there are two hash function selected the same position (fifth from the left), while the fingerprint array generated is \{010010101010\}.

\[
X_1 \quad X_2 \quad \cdots
\]

Figure 4. A fingerprint generated by bloom filter.

C. Detection Procedure

In this section, the detection processes are described in detail as follows. We assume that the nodes which will communicate with each other are node A and node B. Let’s assume A is the sender, while B is the receiver. Each message which sent by A contains its fingerprint generated by its bloom filter. When Node B receive the message, it will check the fingerprint of A, and then decides whether the verification pass. The verification processes is as follows:

For node B, after received the fingerprint information of A, it will check its bloom filter. If all the bits correspond to 1 in the fingerprint sent by A are all 1 in the Bloom filter of B, the verification can pass. Because according to the theory of bloom filter, as long as one of the bits is not 1, then the address fingerprint correspond to the ID was not saved in the bloom filter, that means B receives a new location, so the verification can’t be passed. Because one node with a certain ID cannot has different geographical location.

V. PERFORMANCE ANALYSIS

A. Node-Node Authentication

The proposed scheme can achieve node-to-node authentication. when B received the row information sent by A, then B continues calculate formula(2) and formula (3):

\[
h(B) = Q^b_j \times Q^A_i
\]

\[
K^j_{\mu} = R^b_j \times Q^A_i
\]

With the usage of QR decomposition, the Q is an orthogonal matrix, so any two row vectors of Q is orthogonal to each other. In our scheme, when node B receives the information from A, if \( h(B) \) is not zero, or \( K^j_{\mu} = R^b_j \times Q^A_i \) is not equal to the value of the corresponding position in matrix K, according to the principle that matrix Q is an orthogonal matrix, that indicates there may be copied nodes in the network. So, we ensure the safety further. This scenario uses the geographical location information and the row information of matrix Q to detect whether the node has been copied or not.

B. Communication and Computation Overhead

It is known that communication dominates the energy consumption of a sensor node. Hence, to reduce the energy consumption of networks, it should emphasis on reducing communication overhead. In the detection procedure section, we know that each message which sent by A contains its fingerprint generated by its bloom filter. That means the detection of replicating nodes occurs only when there are needs to send or forward the message. So their communication and computation overhead are lower than other schemes especially the schemes which detect the attack periodicity.

C. Bloom Filter False Positive Rate

In this paper, the bloom filters were used to generate fingerprint. One prominent feature of bloom filters is that there is a clear tradeoff between the size of the filter and the rate of false positives. As stated before that, a bloom filter of false positives means that if it answers that an element is in the set, the element may not be in the set. Observe that after inserting n keys into a filter of size m using k hash functions, the probability that a particular bit is still 0 is:

\[
p_0 = \left(1 - \frac{1}{m}\right)^n \approx e^{-\frac{n}{m}}
\]

The useful approximation comes from a well-known formula for calculating e:

\[
\lim_{x \to \infty} \left(1 - \frac{1}{x}\right)^x = e
\]

Thus the probability that a specific bit has been flipped to 1 is \( p_1 \):

\[
p_1 = 1 - \left(1 - \frac{1}{m}\right)^n \approx 1 - e^{-\frac{kn}{m}}
\]

The main design tradeoffs are the number of hash functions used (driving the computational overhead), the
size of the filter and the error (collision) rate. A false positive on a query of element \( x \) occurs when all of the hash functions applied to \( x \) return a filter position that has a 1.

In practice, good results have been achieved using MD5 and other hash functions [22]. We assume hash functions to be independent. Thus the probability of a false positive (the probability that all \( k \) bits have been previously set) is expressed in formula (4). Formula (4) is the main formula to tune parameters according to application requirements [21].

\[
p_{m} = (1 - p_{c})^{k} = \left(1 - \left(1 - \frac{1}{m}\right)^{\ln m}\right)^{k} \tag{7}
\]

It can be short for formula (8) With the usage of formula (5),

\[
p_{m} = \left(1 - e^{-\frac{\ln m}{m}}\right)^{k} \tag{8}
\]

To minimize the false positive rate, let \( p = e^{-\frac{\ln m}{m}} \), we could use calculus, and then know that the \( p_{m} \) is minimized when \( p = \frac{1}{2} \), so we have

\[
k = \frac{m}{n} \cdot \ln 2 \tag{9}
\]

Actually, only a small number of hash functions are used. The reason is that the computational overhead of each hash additional function is constant while the incremental benefit of adding a new hash function decreases after a certain threshold.

Formula (7) is the base formula for engineering Bloom filters. It allows, for example, computing minimal memory requirements (filter size) and number of hash functions given the maximum acceptable false positives rate and number of elements in the set.

D. Confidentiality and Storage Cost

With the usage of bloom filter and hash function for generating the node, the exposing of geographical location information to a wireless environment is avoided. R matrix information is stored in local storage, do not need to broadcast, it could enhance security.

In addition, for the receiving part, with the use of bloom filter, the storage costs of the location information is greatly saved, which is very important for the limited resources of the sensor networks.

VI. Summary

Sensor networks are vulnerable to node replication attacks. In this paper, we propose a distributed protocols for detecting these malicious attacks. Security issue is always a bottleneck problem which restricting the development of WSN. With the usage of bloom filter, this scheme can: 1, decreasing the storage cost of the location information. 2, there are little Communication and computation overhead. Because the detection of replicating nodes occurs only when there are needs to send or forward the message. 3, avoiding the exposure of localization information directly to the wireless environment. Because in the whole process of fingerprint generation, it is calculated locally. In addition, the matrix QR Decomposition and verification used in this scheme, also provides the network the orthogonally validation, and also improved the security performance of network. The proposed protocol also has limitations. As stated in network structure section, it cannot detect the replication attacks in a mobile sensor environment. This problem is to be solved in our future work. Our future work is to design replication detection protocol that work for both static and mobile sensors.

Acknowledgment

This work was supported by Natural Science Foundation of XinJiang under grant NO. 2013211A012. The authors would like to thank the anonymous reviewers for their constructive comments that helped to improve the quality of this paper.

References


Weimin Xie, born in Hengyang Hunan, June 1th,1988. A post graduate of the School of Information Science and Engineering, Xinjiang University, Urumqi, China, major in information processing. The research interests include wireless sensor networks and security.

Liejun Wang, born in 1975, professor, Master supervisor. Ph. D. graduated from Xi'an University of Posts and Telecommunications. As the corresponding author, he is now a teacher of the Information Science and Engineering, Xinjiang University. His research interest covers image processing and wireless sensor networks security.

Mingwei Wang, August 15th, 1988. A post graduate of the School of Information Science and Engineering, Xinjiang University, Urumqi, China, major in information processing. The research interests include wireless sensor networks and security.