Journal of Computer and Communications, 2014, 2, 17-22
Published Online May 2014 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2014.27003

Scalable Trust-Based Secure WSNs*

Amar Agrawal, Ruizhong Wei

Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada
Email: rwei@lakeheadu.ca

Received January 2014

Abstract

In this paper, we consider the scalable of wireless sensor networks with trust-based security. In our setting, the nodes have limited capability so that heavy computations are not suitable. So public key cryptographic algorithms are not allowed. We focus on the scalability of the network and proposed new testing algorithms and evaluation algorithms to test new nodes added, which give them reasonable values of trust. Based on these algorithms, we proposed new components for trust management system of wireless sensor networks.

Keywords

Trust-Based Security, Wireless Sensor Network, Trust Management

1. Introduction

Wireless Sensor Networks have various Real World applications like battlefield monitoring, forest fire detection, landslide detection and various other monitoring tasks. Sensors in a WSN are very small devices consisting of limited energy, computational power, memory and range. Along with these limitations a WSN has to work efficiently. Scalability is one of the many advantages that WSN has to offer. If we need to Scale the network we simply add a node to the existing network and it starts to function. But this isn't easy as it sounds. Node authentication plays an important role in this network. There is a threat of introducing malicious nodes into a network to carry out an attack or probably bring the whole network down. Malicious nodes can provide false information which in turn will defeat the main purpose of setting up a WSN. Trust in such networks play a very important role here. Our paper mainly focuses on the Scalability part. When a new node is introduced into the network we need to be able to trust the new node. For this we have proposed a method to calculate trust of a new node by performing a series of tests on the node. This will help us determine the behavior, intention and honesty of the node. Assuming this pattern we can determine the initial trust value of the node. This kind of system eradicates the possibility of node hijacking and stealing sensitive data like the key used for authentication.

*Research supported by NSERC discovery grant 239135-2011.

How to cite this paper: Agrawal, A. and Wei, R.Z. (2014) Scalable Trust-Based Secure WSNs. *Journal of Computer and Communications*, **2**, 17-22. http://dx.doi.org/10.4236/jcc.2014.27003

There are different types of WSN architectures out there like hierarchical model, but we will be using the conventional architecture of a WSN which consists of all nodes having equal priority. The main reason for choosing this model is ease of scalability and rapid deployment. Conventional model does not require higher energy level Cluster Heads, so the hassle to integrate this node into a WSN is eliminated. We also need to manufacture only one type of Sensor. Our paper mainly focuses on rapid scalability of a WSN. Some of the applications are in Time Critical scenarios where deployment needs to be quick and hassle free. For example, we need to deploy a node in the emergency of a forest fire or we need to deploy a sensor network on the battlefield. There are existing digital signature authentication techniques in use but can be compromised if they fall in the wrong hands. Keys can be stolen from a sensor easily and then a hostile node with the same key can then be introduced into the network. Also some digital signatures use public key style algorithms that is not suitable for our hardware.

One possible node authentication technique is observing the behavior of the node for a given time frame. We can run a couple of tests to see how the new node behaves. In this way we can conclude if the node is friendly or hostile. A hostile node will fail the test since it will probably drop packets, alter messages, delay messages, etc. However, this test will be a rigorous test which may consume a large amount of energy of the new node. So the new node cannot set a time frame in which it acts honest since the test will drain all its energy which may lead to a dead node. In this paper, we propose a new method to authenticate new nodes for a WSN. The proposed algorithm uses an efficient hash function and the testing message is simple and short. This method can be a useful component of a trust management system for a scalable trust-based secure WSNs.

The rest of this paper are arranged as follows. In Section 2 we briefly review some related work. Section 3 introduces the model of our system and some techniques we will use in our algorithm. Section 4 gives our algorithms. Finally, Section 5 gives a brief summery of this paper.

2. Related Work

The paper [1] contains information on behavior evaluation of nodes in order to calculate trust for all nodes in the network. This paper describes that a general behavior evaluation process consists of four steps: expectation definition, actual output observation, difference calculation, and normalization of various task-specific evaluation. Out of these observations, behavior evaluation in routing and behavior of data processing is observed. Behavior evaluation in routing consists of tracking how successfully the packets were delivered. This is measured by two discrete values good or bad. While data processing consists of analyzing the quality of data reported by the node. These observations are then combined by two proposed combination frameworks, Bayesian Inference and Dempster-Shafer Theory of Evidence.

Another mechanism of determining misbehaving nodes using Watchdog has been explained in the paper [2] and [3]. A Watchdog mechanism has been proposed in these papers to locate misbehaving nodes by studying its behavior. As the name suggests the sending node listens to its neighboring node which is responsible for sending packet to the next node. It can detect if the packet has not been sent or the packet has been tampered. In watchdog a buffer has been implemented that holds recently sent packets. This buffer is then compared with overhead packets to check if they match. If a node remains in the buffer for a long time then it is assumed to have failed in forwarding the packet. Watchdog's weaknesses are that it might not detect a misbehaving node in the presence of ambiguous collisions, receiver collisions, limited transmission power, false misbehavior, collusion, and partial dropping. For the watchdog to work properly, it must know where a packet should be in two hops.

In the paper [4], the authors propose implementation of a separate set of designated supervising nodes. This node will solely be responsible for monitoring traffic of nodes within their range. It will thus study the behavior and data related operations of nodes and make their evaluations available to other sensors within the network. Their proposal combines certificate-based and behavior-based trust evaluations.

All the above work are based on a flat WSN architecture which is not scalable.

[5] proposed a hierarchical dynamic trust management protocol for cluster-based WSNs, considering both social trust and QoS trust. [6] and [2] discused trust based security management and a survey of sensor network security is given in [7].

3. System Model

We consider a general WSN, which consists of a powerful base station (or sink) and randomly deployed sensor

nodes. The power and resource of the nodes are limited. Sensor nodes are monitoring or collecting information over a field. Data are forwarded to the base station along the network. The security of the network are based on trust management schemes as those in [5] [8]. Because of the environment of the field changing, the WSN needs extension or change. So new nodes will be deployed and these new nodes need to join the network. The new nodes are also randomly deployed.

In this paper, we will not consider how to initial the network and how to establish trust-based security when the network formed. Rather, we focus on how to handle the change of the network while still keep the trust-based security. We will propose some method to establishing trust relationships between the new nodes and the existing nodes. So basically, our method will increase the scalability of the trust-based secure WSNs. In our model, the hardware of the nodes are lightweight. So it is not suitable for a node to perform complicated computations, which is a usually property of trust-based secure WSNs.

3.1. Trust Parameter

Along with the authentication protocol we need a mechanism to calculate trust value. The parameter that will used to decide if the node is malicious or not is the Trust Parameter. We will adapt the basic idea of the trust parameter as proposed by Fenye Bao, Ing-Ray Chen, MoonJeong Chang, and Jin-Hee Cho in [5]. The parameters are as follows:

$$T_{ij}(t) = w_1 T_{ij}^{intimacy}(t) + w_2 T_{ij}^{honesty}(t) + w_3 T_{ij}^{energy}(t) + w_4 T_{ij}^{unselfishness}(t) \quad (1)$$

The value of T_{ij} denote the trust value that node i evaluates towards node j at time t and w_i are weights in the range of $[0,1]$, where $w_1 + w_2 + w_3 + w_4 = 1$. This parameter consists four trust components. For different applications, the weight can be adjusted to fit different purposes.

The four components are as follows:

1. Intimacy: It is basically the number of interaction node i has with node j over the maximum interactions of node i with other neighboring nodes over a period of time.

2. Honesty: This is the experience that node i has experienced with node j by direct observation. Bad experiences may involved delay in transmission, packet dropping, interval and other factors. If node j exceeds the number of dishonesty threshold, it will be considered a dishonest node.

3. Energy: This parameter is used to analyze the amount of energy a node has mostly denoted as a percentage of the amount left. This determines if node j has enough energy to perform the required operation.

4. Unselfishness: This parameter determines if node j is being selfish such as not performing reporting, data forwarding and sensing functions faithfully. Node i will use direct observations and preferably latest experience with node j to calculate this parameter.

If the node i is not the neighboring node (1-hop node) of node j then it will use its past experiences from its neighboring nodes.

To calculate the value of any component X of the above four, the following equation is used:

$$T_{ij}^{X}(t) = \begin{cases} (1-\alpha)T_{ij}^{X}(t-\Delta t) + \alpha T_{ij}^{X,direct}(t) & \text{if } i \text{ and } j \text{ are 1-hop neighbors;} \\ avg_{k \in N_i}\{(1-\gamma)T_{ij}^{X}(t-\Delta t) + \gamma T_{kj}^{X,recom}(t)\}, & \text{otherwise.} \end{cases} \quad (2)$$

where Δt is the trust update interval, $T_{ij}^{X,direct}(t)$ is based on direct observations, N_i consists of nodes which are i's 1-hop neighbor, and $T_{kj}^{X,recom}(t)$ is the recommendation from node k towards node j.

The above is called peer-to-peer trust evaluation in [5]. The details of the evaluations are omitted here. The readers are refereed to [5].

In general, we will use the following trust parameter:

$$T_{ij}(t) = \sum_{X \in Com} w_X T_{ij}^{X}(t). \quad (3)$$

Here Com is the set of trust components which depends on applications and

$$\sum_{X \in Com} w_X = 1.$$

For example, in [2], $Com = \{intimacy, honesty, energy, unselfishness\}$.

3.2. One-Time Password

A one-time password (OTP) system was published as RFC 2289 (which is a revision of RFC 1938). This system uses a standard hash function such as MD4, MD5 or SHA-1. To create a one-time password, server sends a challenge message to user. Then the user chooses a secret pass-phrase which consists at least 10 characters. The pass phrase is concatenated with the seed. The result of the concatenation is passed through the secure hash function N times, where N is specified by the user. The resulting digest is the one-time password record. The next one-time password to be used is generated by passing through the secure hash function $N-1$ times.

To authenticate the user, the server passes the password through the secure hash function once and compares the result with the stored previous OTP. If the result of the operation matches the previous OPT, the authentication is successful and the accepted one-time password is stored for future use. In this way, a pass-phrases can be used for $N-1$ times.

The security of this system depends on the hash function's one-way property. The seed used here enables the user to use the same secret pass-phrase for different machines.

In our application, we use the basic idea of OTP for our purpose, but not as password. Since we just need the one-way property of the hash function, we use relatively efficient hash function MD5 denoted as h.

In a new node s, the following messages are installed before deployment:

- U_s: the unique node ID of the new node s.
- N_s: an integer which denotes the times of hash. It is not necessary that all the nodes have different values of N. Usually we can set a range for N_s, e.g., $150 \le N \le 200$.
- P_{U_s}: a random string having at least 10 characters.

These values are also stored in the base station securely.

Each node also has an MD5 algorithm.

4. The Scalable WSNs

The scalability of WSNs mostly depends on how the network evaluate the trust of newly added nodes. We can divide the initialization of the trust for the new nodes into two categories. One is using public key based cryptography. This method requires more on hardwares. Some kind of Trusted Platform Module crypto-processor chips are proposed (e.g., see [9]-[11]).

This paper focus on more constrained devices, where public key systems are not suitable. Velloso et al in [12] have a brief discussion about the scalable ad hoc networks. In this paper, we investigate some detailed efficient algorithms for the scalable WSNs without using public key based cryptography.

Suppose we already have a WSN which uses trust-based security. Since the environment changed or some other reasons, more new nodes are deployed, which are supposed to join the existing network. We need to keep the trust-based security of the network after the new nodes joining.

4.1. New Nodes Testing

We propose the following basic algorithm to do the new nodes testing. In what follows, i, j denote nodes (existing nodes or new nodes), s denotes a new node, S denotes the base station or sink.

1. S broadcasts a new_node_join code. This broadcast lets the nodes in the WSN know that new nodes are joining the network.
2. Each new node s calculates a value H_s which equals to $N_s - 1$ times hash of $U_s \| P_{U_s}$ and updates N_s to $N_s - 1$.

$$H_s = \underbrace{h(h(\cdots h}_{N_s-1}(U_s \| P_{U_s})\cdots)).$$

3. s sends (U_s, H_s) to its neighbors.
4. When a node i received (U_s, H_s), it was recorded to R_i.
5. S broadcasts (U_s, C_s) for all the new nodes s, where C_s equals to to N_s times hash of $U_s \| P_{U_s}$

$$C_s = \underbrace{h(h(\cdots h}_{N_s}(U_s \| P_{U_s})\cdots)).$$

6. When node i receives (U_s, C_s), i checks if U_s is in the record. If it is, then i compute $h(H_s)$

where H_s is from the R_i. i then compare the resulting value with C_s. If two values are the same, then i adds one credit for s. i updates the record C_s to H_s for future use.

7. The steps 2, 3 and 6 are repeated for n times, where n is a predefined integer or broadcasted by S.

A malicious node m may perform the following attack. After receiving (U_s, H_s), m sends out (U_s, H'), where H' is some random string. The purpose of m is letting the neighbor of s think that s is not authenticated at the next round.

To avoid that kind of attack, node i does not update the record R_i if the values in step 6 are not equal. The record is updated only if the value H_s is accepted.

4.2. Evaluate Neighbors

Now we need some algorithms to let a new node estimate the trust value of its neighbor nodes, especially for the old nodes. One simple method is to use the above algorithm. We can change the above algorithm so that the old nodes are not distinguish from new nodes. One disadvantage of that method is that the station needs to broadcast information for every node. If the existing network is small, then that method is fine. However, if the network is big, then the station will broadcast big amount of messages. Note that the broadcast normally is done by the nodes in the network forwarding to the neighbors. So this method will consume a lot of energy of the network.

We outline the following algorithm for a new node to evaluate the neighbors.

1. s sends a eva_neighbor_req code to its neighbor nodes.
2. s sends a list L_s of nodes, for which s wants to evaluate the trust.
3. Optionally, s can send out a request of multi-hops. In this case, s can specify how many hops s wants to forward.
4. When a node i received L_s, it sends $\left(i, j, T_{ij}(t)\right)$ to s for all $j \in L_s$. i also forward L_s to the neighbor nodes, if requested.
5. s calculates $T_{sj}(t)$ after receiving all the feed back as follows: Suppose the smallest value it received is a and the largest value is b. Then s checks to see the majority of the received values are at $[a,(a+b)/2]$ or $[(a+b)/2,b]$. Let Maj be these majority values. If the distribution of the values are evenly, then Maj contains all the values. Then

$$T_{sj}(t) = avg_{i \in Maj}\{T_{ij}(t)\}.$$

The purpose of using majority in step 5 is to ignore possible malicious nodes sending fake evaluations.

The above "majority principle" can also be used in general trust evaluation. When some formulas of Section 3.1 are used, we can just use the majority values.

For simplicity, we just divide the value of $[a,b]$ into two parts. Actually, we can divide $[a,b]$ into 3 or 4 equal parts and using one of these parts as Maj.

4.3. Evaluate Trust for New Nodes

The algorithm in subsection 4.1 gives some authentication information for new nodes, but not the value of trust. The results may give some positive value to some components in Com.

We will divide the components in Com into three parts. One part is full positive so the default values are full. Example for that kind of components is *energy* for new nodes. The second part is average so the default value is average. Example for this is *unselfishness*. The third part of components consists of the components related to node authentication which will depend on the results of the algorithm of Section 4.1.

Here the evaluation of trust for new nodes means the initialization of the trust of new nodes. After initialization, the new nodes are joint the network and the normal process of trust computations are performed.

5. Conclusion and Future Work

In this paper, we proposed new algorithms for the purpose of improving scalability of WSNs which depend on trust-based security. The main calculation of the algorithm is perform a simple hash function. The communications are short strings. So the algorithms are efficient in both time and space. For new nodes adding, the sink only needs to perform one network wide broadcast. Therefor the network energy consume of the scalable is also efficient.

The proposed algorithms can be one component of the trust management system for WSNs.

In the future, we will further investigate the existing trust management systems and find out how to combine our component to the system and how to further improving. Some detailed implementations are also to be done in the future. Simulations can also used for the purpose of evaluation of our proposal.

Our current work is based on simple setting of WSNs. But the method is not difficult to be modified for other kind architectures, such as hierarchical structures. One possible future work is detailing the modification of the method for fitting other architectures.

References

[1] Huang,L., Li, L.and Tan,Q. (2006) Behavior-BasedTrust in Wireless Sensor Network.*Proceedings of the 2006 international conference on Advanced Web and Network Technologies, and Applications*, 214-223.

[2] Marti, S., Giuli, T.J., Lai, K. and Baker, M. (2000) Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. *Proceedingsof the* 6*th Annual International Conference on Mobile Computing and Networking*, 255-265.

[3] Ganeriwal, S., Balzano, L.K. and Srivastava, M.B. (2008) Reputation-Based Framework for High Integrity Sensor Networks.*ACM Transactions on Sensor Networks* (*TOSN*), **4**, Article No. 15. http://dx.doi.org/10.1145/1362542.1362546

[4] Aivaloglou, E. and Gritzalis, S. (2010) Hybrid Trust and Reputation Management for Sensor Networks. *Wireless Networks*, **16**, 1493-1510. http://dx.doi.org/10.1007/s11276-009-0216-8

[5] Bao, F., Chen, I.-R., Chang, M.J. and Cho, J.-H. (2012) Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection.*IEEE Transactions on Network and Service Management*, **9**, 169-183. http://dx.doi.org/10.1109/TCOMM.2012.031912.110179

[6] Lopez, J., Roman, R., Agudo, I. and Fernandez-Gago, C. (2010) Trust Management Systems for Wireless Sensor Networks: Best Practices. *Computer Communications*, **33**, 1086-1093. http://dx.doi.org/10.1016/j.comcom.2010.02.006

[7] Chen, X., Makki, K., Yen, K. and Pissinou, N. (2009) Sensor Network Security: A Survey. *IEEE Communications Surveys Tutorials*, **11**, 52-73. http://dx.doi.org/10.1109/SURV.2009.090205

[8] Cho, J.H., Swami, A. and Chen, I.R. (2011) A Survey on Trust Management for Mobile and Ad Hoc Networks.*IEEE Communications Surveys Tutorials*, **13**, 562-583. http://dx.doi.org/10.1109/SURV.2011.092110.00088

[9] Hu, W., Corke, P., Ship, W. C. and Overs, L. (2009) secFleck: A Public Key Technology Platform for Wireless Sensor Networks. In: Roedig, U. and Sreenan, C.J., Eds., EWSN 2009, LNCS5432, 296-311.

[10] W. Hu, *et al.* (2010) Toward Trusted Wireless Sensor Networks.ACM Transactions on Sensor Networks, **7**, 1-25. http://dx.doi.org/10.1145/1806895.1806900

[11] Yissoff, Y.M. and Hashim, H. (2010) Trusted Wireless Sensor Node Platform. *Proceedings of the World Congress on Engineering* 2010, London, 774-779.

[12] Velloso, P.B., *et al.* (2010) Trust Management in Mobile ad Hoc Networks Using a Scalable Maturity-Based Model. *IEEE Transactions on Network and Service Management*, **7**, 172-185. http://dx.doi.org/10.1109/TNSM.2010.1009.I9P0339