DETECTION OF ON-OFF ATTACK BASED ON PREDECTABILITY TRUST IN WIRELESS SENSOR NETWORK

[1]AMOL R. DHAKNE, [2]P.N. CHATUR

[1,2]Government College of Engineering,Amravati, Maharashtra, India
E-mail: [1]dhakne.amol5@gmail.com, [2]prashant_chatur@rediffmail.com

Abstract— Now days, Trust management schemes are widely used in decision making process for access control, secure routing and Intrusion detection. Sometimes unintentional errors are possible, trust management must consider some redemption schemes so that nodes will be able to recover trust again. It is possible for malicious node that it can falsify the misbehavior as unintentional temporary error to misguide redemption scheme and due to which malicious node can get more chances to attack system by disturbing redemption scheme. Existing trust management schemes that consider redemption scheme to recover trust fail to distinguish between temporary errors and malicious behaviors where attackers behave well and bad alternatively. This paper presents disadvantages of existing redemption scheme describes a trust management and redemption scheme that can distinguish between temporary errors and malicious activities with some design.

Index Terms— On-Off Attack, Wireless Sensor Network, Predictability Trust.

I. INTRODUCTION

Trust is an important but complex concept in social science. Trust helps people to make decisions in unpredictable circumstances by reducing the uncertainty. Research on trust management schemes, which manage trust and decide policies, has emerged as a challenging issue. Trust management schemes aim to improve collaboration between the entities in a distributed system by predicting future behaviors of peers based on their previous behaviors. A trust management scheme typically does this using the following steps. First, each node observes and stores the neighbouring nodes' behaviors. Second, each node collects and stores the warnings or reports from other nodes about its neighbouring nodes. Third, each node calculates the trust based on the behaviour information collected and stored for each neighbouring node. Last, based on the trust and the policies that use the trust, each node decides the best node or group of nodes with which to collaborate.

In some systems, trust management schemes allow trust redemption in order to allow a node to regain the trust of its neighbours. For instance, a Wireless Sensor Network (WSN) [1] is composed of sensor devices that have constrained resources and unreliable radio for wireless communication. Thus, there exists a possibility that unintentional temporary errors might occur. When a node performs a bad behavior, like a dropped packet, it could be considered malicious even if the behavior was temporary and unintentional. When the trust management scheme isolates the node from the network based on the security policies, the node may not be used again even after it returns to normal behavior. This is a waste of the system resources, thus it would reduce the system efficiency.

By allowing a redemption scheme, the system may avoid a faulty detection, that is, the erroneous identification of anode as malicious. A redemption scheme provides further opportunities to these nodes by recovering the trust based on subsequent good behaviors or based on time elapsing.

Unfortunately, existing redemption schemes are vulnerable to an On-off attack, which is specifically designed to disrupt the trust management and redemption schemes. By behaving well and badly alternatively, the On-off attack aims to make the trust management scheme consider a bad behavior as a temporary error. Thus, the malicious node would remain active and would have more opportunities to attack the network. Moreover, there may be circumstances under which an On-off attacker should be allowed to remain in the system. That is, if the cost of removing the attacker is higher than the cost that the attack imposes on the system, then it may be better to leave it alone.

This paper presents a trust management scheme that uses a new kind of trust, called Predictability Trust, which is able to predict future trust values based on past behaviors with an efficient and flexible design.

II. TRUST MANAGEMENT SCHEME

We define trust as the probability that an object performs a given action as expected. A trust management scheme manages the trust by integrating the notions of credentials, access control, security policy, availability, and authentication. Distributed trust based intrusion detection approaches are also possible with trust scheme [2]. By using the integrated information, a trust management scheme can be used to aid an automated decisionmaking process for an access control policy. Trust can be evaluated in a variety of ways. Direct observation evaluates neighboring nodes by observing their behavior. For example, in a WSN, a node is able to detect malicious neighbors by monitoring how many packets were forwarded to the next node. Moreover, if a source node compares the contents of the packets,

it can detect fabrication or modification [3]. With indirect observation nodes publish their direct observations to their neighboring nodes to warn about malicious nodes or to report recovered nodes that were previously evaluated as malicious. In a WSN, a warning message from other nodes will exclude the malicious node from the network. On the other hand, recovery reports from other nodes can allow nodes to rejoin the network [4], [5], [6], [7].

A) Trust Redemption Scheme

Because unintentional temporary errors may occur a redemption scheme is required to allow an untrusted node to recover its trust value. Redemption schemes can be classified in two ways. Behavior Based Redemption (BBR) recovers trust based on subsequent behaviors. Time Based Redemption (TBR) recovers trust periodically. If both BBR and TBR are used together, we refer to this as Combined Redemption. In the following subsections, we classify existing trust models by these redemption schemes.

III. ON-OFF ATTACK

A smart attacker may attempt to disturb a trust redemption scheme by behaving well and badly alternatively so that trust is always redeemed just before another attack occurs. This type of attack is referred to as an On-off attack [8]. Most trust redemption schemes fail to effectively discriminate between an On-off attack and temporary errors, especially when the majority of the attacker's behavior is good. Therefore, an attacker may be able to remain active in the system by disguising the attacks as temporary errors. In general, if the malicious node performs n good behaviors and m bad behaviors alternating, we refer to this as an nG-mB On-off attack. For example, 4G-1B attack node means the node behaves well 4 times and behaves badly 1 time alternatively. This can also be called a 80%G-20%B On-off attack. An On-off attack does not work alone, but associates with another type of attack.

For example, a malicious node in a WSN may associate a selective forwarding attack with the On-off attack where most of the time it forwards all packets but occasionally it drops most or all of the packets. There are two states [8], [9] in an On-off attack; the On state and the Off state. An On state is referred as an attack state. When a malicious node is in the On state, the node attacks the target nodes with the associated attack. An Off state is referred to as a normal state. When a malicious node is in an Off state, it behaves normally. When the ratio of the Off state to On state is high, the trust management system may have difficulty detecting the malicious behaviors. There is a trade-off for the attacker between remaining active in the system and performing highly efficient attacks. The higher the Off-to-On ratio, the longer the attacker can remain in the system, but the less efficient the attack.

A) On-Off Attack Defense

There exist trust management schemes that address Onoff attack in various types of networks, such as Traditional Networks [10], Cognitive Radio Networks [11], Peer-toPeer Networks [12], Ad-hoc Networks [13], and Wireless Sensor Networks [14], [15]. However, no trust management scheme is able to discriminate between temporary errors and On-off attacks. In [16] and [12], the authors propose evaluating neighboring nodes by using direct and indirect trust values. The solution described in [16], uses a compilation of direct and indirect evaluations to reduce the trust of an On-off attacker to be lower than the trust of other neighboring normal nodes, so the network will detour around the On-off attack nodes in the system. However, in the simulation, they did not consider when a source node was surrounded by malicious nodes and had only one normal path to traverse. Therefore, since there were many good options to take, the On-off attacker did not have an opportunity to continue its attack. In our work, we have considered a harsher WSN environment in which a node may be surrounded by malicious nodes and only have one possible normal path to the base station. The scheme described in [12], also used an integration of direct and indirect evaluation to reduce the trust of an Onoff attack to be smaller than a threshold. The trust management scheme was tested in a simulation against an Onoff attack with 50 good behaviors and 50 bad behaviors, thus with an Off-to-On ratio of 1:1. This is a rather simple On-off attack that can be easily detected because it is hard to disguise the bad behaviors as temporary errors and because it has many bad behaviors in succession. Moreover, since both of these trust management schemes employ TBR, there still exists the risk to recover the trust of an On-off attack node to greater than the threshold. This scheme was not tested against more sophisticated On-off attacks with lower Off-to-On ratios. Our work considers various Off-toOn ratios. In fact, it allows the system designer to choose the allowable ratio in the system

IV. PREDICTABILITY TRUST

In this section, we present a new efficient and flexible trust management scheme that detects and defends against On-off attacks. This trust management framework relies on two key concepts: Predictability Trust and Dynamic Sliding Windows. Predictability Trust works with some other type of trust to detect On-off attacks. It uses sliding windows to keep track of previous behaviors so that it can determine how quickly to redeem trust.

A) Predictability Trust Calculation

Predictability Trust (PT) is computed based on how well a node's behavior meets expectations. For example, if a node's current forwarding trust is 0.9, we predict that this node will forward at least 90% of

the packets that are sent through it. Then in the next round, if this node forwards more than 90% of its packets, it meets the prediction, and is considered to have conducted a Good Predicted Behavior (GPB). If the node forwards fewer than (90−Δ)% of its packets, it does not meet the expectation, and is considered to have conducted a Bad Predicted Behavior (BPB). Here, Δ is a tunable parameter depending on the application scenarios. In our experiments, we set Δ=0.1. We will count the number of GPBs and BPBs conducted by node i (denoted by GPBi and BPBi respectively). The PT of node i is computed as in (1), using a beta reputation system Bayesian formulation [17].

$$PTi = \frac{GPBi+1}{GPBi+BPBi+2} \quad (1)$$

The PT of node i describes whether the current trust can accurately predict the node's future behavior and whether a node's behaviour is consistent with his past behavior. A low PT indicates that (a) the current Overall Trust value is less "trusted" and therefore should be lowered, and (b) this node's behavior is not consistent and therefore should be suspected for On-off attacks, which requires an adjustment in the trust redemption.

B) Individual Behavior Trust

A node can be monitored for its various behaviors, such as data forwarding and responding to requests from neighbor nodes [18]. For each behavior, a trust value is calculated, describing whether this node honestly conducts this behavior. We call these trust values as Individual Behavior Trust values. Let Tik denote the individual behavior trust value for the kth type behavior of node i. Many methods have been developed to compute Tik, based on the history of node i in terms of conducting behavior k [11], [19], [20], [21].

C) Overall Trust

Based on the individual behavior trust, we define Overall Trust to evaluate if a neighboring node is malicious or not. The Overall Trust is calculated from the individual behavior trust, using the methods such as these in [18], [21].The concepts of PT can work with any methods that calculate individual behavior trust or Overall Trust.

The first way that PT is used is in setting the maximum value of the Overall Trust using (2)

$$OTi = CTi \times PTi \quad (2)$$

where the CTi represents the Compiled Trust of node i computed by combining the various individual behavior trust values of node i. Since the ranges of CT and PT are between 0 and 1, the Overall Trust can be at best 1.0 by (2). This allows us to use a node's predictability as a factor in the Overall Trust computation. A system designer can set a certain threshold for Overall Trust that excludes a node from being used in the system if it has a trust value below the threshold. If PT is low enough, our mechanism can lower the Overall Trust so that it is below the threshold. Thus, PT has an effect on Overall Trust that is independent of the specific trust behaviors. Equations (1) and (2) are computed after every transaction in which a node is used, thus employing Behavior Based Redemption.

D) Redemption Speed Adjustment

Redemption Factor (RF) is also referred to as a forgetting factor as well as a fading factor. This helps to give a second chance to a node, which has been evaluated as malicious, by recovering a trust value based on time. The other way in which PT is used allows it to control the redemption speed by computing a RF of node i. We compute a RF using a current trust value with the current PT value, so a node can dynamically recover the trust value of neighboring node i depending on the current trustworthiness. RF is calculated by (3) and is used by (4) to allow trust to be redeemed at the calculated rate

$$RFi = \left(T_i^k \times PTi\right) \times \alpha + 1.0 \quad (3)$$

$$T_{i,after}^k = T_{i,before}^k \times RFi \quad (4)$$

where, the $T_{i,before}^k$ and $T_{i,after}^k$ represent the individual kth type of trust before and after the adjustment of node i. Since T_i^k is used for calculating RFi, each trust type can be recovered at a different speed. In (3), $0 < \alpha \leq 1$, and it represents a mechanism to allow a system designer to control the tolerance of a system. If a system needs to be strictly secured, α would be smaller than that in a more tolerant system. When PTi is low because a node has behaved unpredictably, the redemption will take more time than for a more predictable node. Equations (3) and (4) are computed periodically, thus employing time-based redemption.

V. ON-OFF ATTACK DETECTION

Using the equations described in Sections IV (C) and IV(D), we can reduce the effectiveness of On-off attacks. In particular, the technique in Section IV (C)reduces the Overall Trust of the On-off attackers, and Section IV (D) reduces the redemption speed of the trust values of the On-off attack nodes. However, if we only depend on the above two techniques, the On-off attackers will not be detected in a short time. Since the trust of the attacker is reduced, the attacker may have fewer opportunities to be used in the system and fewer observations will be made. It will take a long time to collect sufficient evidence to mark a node as malicious and the problem exists in many trust-based schemes. There are two ways to address this problem. The first way is to give more opportunities to low-trust nodes to act. This is why we allow trust redemption. The second way is to

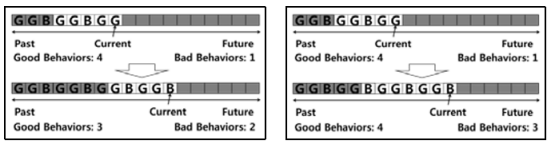
adjust the method of evidence collection based on the PT, using Sliding Windows.

A) Sliding Windows

The main purpose of a Sliding Window (SW) is to keep track of the past behaviors of a node. It would be best if we could observe the entire history of each node, but this is unattainable when a system has limited storage and processor speed, as in a WSN. For these reasons, we implemented a SW to allow a certain number of behaviors to be stored for calculating trust. A SW updates and stores the latest behavior history. When an event is observed and the SW is full, the SW removes the oldest behavior in its memory and stores the latest behavior. We use two types of SW in our trust computation: a fixed sliding window for good behaviors (GBW) and a dynamic sliding window for bad behaviors (BBW).

a) Good Behavior Window: The purpose of the GBW is to count the good behaviors among the most recent behaviors. It stores both good behaviors and bad behaviors, but counts only the good behaviors. This makes it possible to consider only the fresh good behaviors, while keeping in mind the overall pattern of behaviors in the recent past. A system designer is able to provide a specific number of opportunities for bad nodes by setting the size of GBW.

Fig. 1(a) shows how the GBW works. The top part of the figure shows the window status with a size of five with four good behaviors and one bad behavior. The bottom part of the figure shows the



(a) Good Behavior Window (b) Bad Behavior Window

Figure1. Sliding Windows

same window after four more behaviors. In this figure, the GBW is represented by the white boxes, so it forgets any behaviors in the shaded boxes. The boxes on the left of the "Current" arrow are the past behaviors that it remembers. The box that the arrow points out is the latest behavior, and the boxes on the right side of the arrow represent space for future behaviors that have not occurred yet. In this example, the number of good behaviors changed from four to three after four behaviors in 2G-1B attack.

b) Bad Behavior Window: We are more interested in the bad behaviors than the good behaviors because they are harmful to the system, and the primary purpose of PT is to isolate malicious nodes. However, to avoid erroneously labeling nodes as malicious, we need to be cautious in discriminating the malicious nodes. For these reasons, we developed a BBW that allows us to observe more previous bad behaviors depending on the current trust value. So as trust decreases the size of BBW increases. The BBW stores good behaviors and bad behaviors, but counts only the bad behaviors. The size of the window changes dynamically as the trust of the node changes, and a system designer is able to set a maximum window size for the BBW. We have performed a set of analytical tests that indicate the tolerance to attack for a given maximum window size and we discuss these tests in Section 4. Thus, it is up to the system designer to determine how much damage the system can tolerate, and set the BBW maximum size accordingly.

$$Size := \beta \times OT + \gamma \tag{5}$$

$$MaxSize := \beta \times MaxTrust + \gamma; \quad MinSize := \beta \times MinTrust + \gamma \tag{6}$$

While the maximum and minimum BBW size is set by the system designer, the dynamic size of the BBW is computed by (5) and (6), where 0.0≤Min Trust≤MaxTrust ≤ 1.0, and β and γ are computed using simultaneous equations. By adjusting the values of MaxTrust and MinTrust the designer can affect the range of size for the BBW. When a node observes a bad behavior by one of the other nodes, the current Overall Trust for that node gets lower. Thus, we want to consider more previous behaviors, so the BBW size is increased. In Fig. 1(b), we see on the top a window size of five with one bad behavior. Later, as represented in the window on the bottom of the figure, two more bad behaviors have occurred, so the BBW size has increased so that it keeps all three bad behaviors in the calculation of trust.

B) Predictability Trust Based Redemption (PTR)

PT counts the number of the behaviors that did and did not satisfy the designer's expectation. In these simulations, we considered a 100% forwarding transaction as a good behavior. The Overall Trust of a node is calculated based on the forwarding behaviors and is shown in (7).

$$GFB_i = GFB_i + NFP_i, BFB_i = BFB_i + NLP_i$$
$$FT_i = \frac{NFP_i + 1}{NFP_i + NLP_i + 2}$$
$$GPB_i = GPB_i + SB_i, BPB_i = BPB_i + DB_i \tag{7}$$
$$PT_i = \frac{GPB_i + 1}{GPB_i + BPB_i + 2}$$
$$OT_i = FT_i \times PT_i$$

and every six seconds, each node updates the trust of its neighbouring nodes by using (8).

$$RF_i = \left(\frac{GFB_i + 1}{GFB_i + BGB_i + 2}\right) \times PT_i + 1.0 \tag{8}$$
$$FT_i = FT_i \times RF_i$$
$$OT_i = FT_i \times PT_i$$

whereSBi represents satisfied behavior of node i and DBi represents disappointed behavior of node i. When the behavior satisfied the designer's expectation, SBi is set to 1 and DBi is set to 0. Otherwise, SBi is set to 0 and DBi is set to 1.

CONCLUSION

Predictability Trust is a concept that allows for accumulation of previous behaviors to compute trust of a node in a system. When a smart attacker knows that the collaborating entities in a network allow for trust redemption, the attacker can try to plan attacks in such a way to fool the system into regaining trust after time or after subsequent good behaviors. With the use of Predictability Trust, the previous behaviour of attacker will make a collaborator more wary of working with it again, and will redeem trust more slowly. If the accumulated behaviors prove to be bad enough, the collaborator will choose not to work with the attacker again. We have demonstrated how PT with sliding windows can allow for flexible design in which a system designer is able to decide the number of opportunities a node should be allowed before being eliminated from the system.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," Computer networks, vol. 38, no. 4, pp. 393-422, Elsevier, 2002.

[2] A. R. Dhakne and P. N. Chatur, "Distributed Trust based Intrusion Detection approach in wireless sensor network," 2015 Communication, Control and Intelligent Systems (CCIS), Mathura, pp. 96-101, IEEE, 2015.

[3] K. Paul and D. Westhoff, "Context aware detection of selfish nodes in dsr based ad-hoc networks," GLOBECOM'02. IEEE, vol. 1, pp. 178-182, 2002.

[4] S. Buchegger and J. Le Boudec, "A robust reputation system for mobile ad-hoc networks," Proceedings of P2PEcon, June, 2004.

[5] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security, pp. 107-121, 2002.

[6] S. Bansal and M. Baker, "Observation-based cooperation enforcement in ad hoc networks," Arxiv preprint cs/0307012, 2003.

[7] Q. He, D. Wu, and P. Khosla, "SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks," IEEE WCNC 2004, vol. 2, pp. 825-830, 2004.

[8] A. R. Dhakne and P. N. Chatur, "Detailed Survey on attacks in wireless sensor network," Proceedings of the International Conference on Data Engineering and Communication Technology: ICDECT 2016, vol. 2, pp.319-331, Springer,2017.

[9] X. Mao and J. McNair, "Effect of on/off misbehavior on overhearing based cooperation scheme for MANET," MILCOM 2010, pp. 1086-1091, IEEE, 2010.

[10] G. Maciá-Fernández, J. Díaz-Verdejo, P. García-Teodoro, and F. de Toro-Negro, "LoRDAS: A low-rate DoS attack against application servers," Critical Information Infrastructures Security, pp. 197-209, Springer, 2008.

[11] T. Qin, H. Yu, C. Leung, Z. Shen, and C. Miao, "Towards a trust aware cognitive radio architecture," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 13, no. 2, pp. 86-95, 2009, ACM, 2009.

[12] C.-L. Cheng, X.-L.Xu, and B.-Z. Gao, "METrust: A mutual evaluation-based trust model for P2P networks," International Journal of Automation and Computing, vol. 9, no. 1, pp. 63-71, Springer, 2012.

[13] D. M. Shila, Y. Cheng, and T. Anjali, "Mitigating selective forwarding attacks with a channel-aware approach in WMNs," Wireless Communications, IEEE Transactions on, vol. 9, no. 5, pp. 16611675, IEEE, 2010.

[14] J. Wang, Y. Liu, and Y. Jiao, "Building a trusted route in a mobile ad hoc network considering communication reliability and path length," Journal of Network and Computer Applications, vol. 34, no. 4, pp. 1138-1149, Elsevier, 2011.

[15] B.-J. Chang and S.-L.Kuo, "Markov chain trust model for trustvalue analysis and key management in distributed multicastMANETs," Vehicular Technology, IEEE Transactions on, vol. 58, no. 4, pp. 1846-1863, IEEE, 2009.

[16] Z. Liu, S. S. Yau, D. Peng, and Y. Yin, "A flexible trust model for distributed service infrastructures," IEEE ISORC 2008, pp. 108115, IEEE, 2008.

[17] A. Jøsang and R. Ismail, "The beta reputation system," Proceedings of the 15th Bled Electronic Commerce Conference, pp. 41-55, 2002

[18] L. C. DiPippo, Y. Sun, and K. Rahn.Jr., "Secure Adaptive Routing Protocol for Wireless Sensor Networks", University of Rhode Island Department of Computer Science Technical Report, TR10-329, 2010.

[19] J. Mundinger and J. Le Boudec, "Analysis of a reputation system for mobile ad-hoc networks with liars," Performance Evaluation, vol. 65, no. 3, pp. 212-226, Elsevier, 2008.

[20] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson, "One hop reputations for peer to peer file sharing workloads," NSDI, vol. 8, pp. 1-14, 2008

[21] H. Xia, Z. Jia, L. Ju, X. Li, and Y. Zhu, "A Subjective Trust Management Model with Multiple Decision Factors for MANET based on AHP and Fuzzy Logic Rules," IEEE/ACM GreenCom '11, pp. 124-130, IEEE, 2011.

★★★