A Proactive Data Security Framework for Mission-Critical Sensor Networks

Kui Ren[1], Wenjing Lou[1], and Patrick J. Moran[2]

[1]Department of ECE, Worcester Polytechnic Institute, Worcester MA 01609
{kren,wjlou}@wpi.edu

AirSprite Technologies, Inc., Marlborough, MA 01532
pmoran@airsprite.com

ABSTRACT

The resource-constrained sensors in mission-critical applications are subject to both random failures and intentional compromise, which poses severe security threats in wireless sensor networks (WSNs). The different types of security threats have been identified and addressed in an individual manner in the past. In this paper, we argue that cryptography alone is insufficient to fully address the insider attacks in the existence of both the compromised and faulty sensor nodes. We further propose a proactive data security framework (PDSF) to identify compromised and faulty nodes proactively and prohibit them from participating network activities in a dynamic manner. The rationale behind our approach is that a sensor's future behavior can be predicted (at least) probabilistically by its past behavior. PDSF is divided into two key modules, that is, misbehavior characterization & monitoring, and trust management. PDSF characterizes different types of misbehavior in WSNs and defines a set of monitoring criteria. PDSF also develops a trust management model, which adapts to the resource constrained nature of the WSNs.

I. INTRODUCTION

In mission-critical applications such as battlefield reconnaissance and homeland security monitoring, wireless sensor networks (WSNs) can be deployed in unattended and hostile geographic areas, where a large number of sensor nodes collaboratively collect, process, and report information of interest to the end-user(s). Data security in such WSNs is critical and demands inherent collaborations among sensor nodes due to the decentralized and infrastrutureless nature of the WSNs. However, the resource-constrained sensors may randomly fail or be intentionally compromised over network lifetime by the adversary. The compromised and faulty nodes can hence become malicious or unpredictable, and pose severe security threats in WSNs.

In the past few years, many cryptography-based security designs have been proposed for WSNs. Providing lightweight and decentralized cryptographic mechanisms is the primary focus of these works in order to meet the stringent resource constraints of the sensors. However, cryptography-based approaches alone are not sufficient to address the security threats posed by compromised and faulty sensors [1]. These approaches cannot adequately defend against the insider attacks and node random failures, although they are effective to the outsider attacks. This is because the compromised and faulty nodes are legitimate network members originally and do posses all the corresponding cryptographic keys. Hence, additional security mechanisms have to be developed (on top of cryptography-based approaches) to fully address the problem.

Some recent works have paid attention to the presence of compromised and faulty sensors [2]–[4], [14], [15]. These works focus on increasing security resilience, and use the scale and redundancy in the WSN to their advantage. These works usually introduce a threshold property to their designs to gain the resilience against up to a certain number of compromised and faulty nodes. However, the effectiveness of these passive approaches is suspicious in practice, where the predefined threshold parameter may deviate significantly from the practical situation. Furthermore, these works are limited in scope. They usually each deal with one individual type of insider attacks, and the corresponding solutions are highly specific and not applicable to other attacks. A number of different solutions are thus demanded to address different types of insider attacks and node random failures. This is extremely inefficient if not impossible in WSNs due to lack of resources, not to mention the compatibility issue and repetitive designs.

To systematically address the problem, this paper explores a unified proactive approach, which serves as the front line in defending against different types of security threats. The approach seeks to identify compromised and faulty nodes proactively and prohibit them from participating network activities in a dynamic manner. The rationale behind our approach is that a sensor's future behavior can be predicted (at least) probabilistically by its past

behavior. Following this logic, we propose a proactive data security framework (PDSF), which consists of two key parts: misbehavior characterization & monitoring and trust management. The first part characterizes different types of misbehavior in WSNs and defines a set of monitoring criteria. The second part develops a trust management model, which evaluates the detection results of the first part and establishes (maintains) a distributed reputation table at each sensor. This reputation table is consulted every time a sensor plans to interact with other sensors.

The remaining part of this paper is organized as follows. We start from the analysis of limitations of current security designs in WSNs. Then, we describe the network model and security model assumed by the proposed proactive data security framework. Next, we detail its design. And finally, we conclude the paper by pointing out some open problems and future directions.

II. Limitation analysis of current security designs

Through a concrete example, we show that cryptography-based approach alone is not sufficient to ensure data security in WSNs. In a battlefield scenario, a WSN is deployed to monitor the battlefield, where both network legitimate users and the adversaries present. The WSN is aimed at providing dynamic information, regarding the adversary's information and so on, to the legitimate network users, i.e., soldiers, on demand. In such a scenario, some of the sensor nodes may fail or be compromised, and the compromised sensors will be completely controlled by the adversary soldiers. We assume that all cryptography-based mechanisms are already in place, when the WSN is deployed. Now we state three different types of attacks that cannot be addressed by cryptography-based approach alone.

Attacks against data confidentiality: In WSNs, we require that, on the one hand, all the data collected, processed, and transmitted by the normal network sensors should be kept confidential against the adversary; on the other hand, compromised nodes should be prohibited from exposing the information (except for those directly available to themselves) to the adversary. However, in our scenario, a compromised node may initiate a query asking about the location of a legitimate soldier's position and send this information to an adversary soldier. Hence, the legitimate soldier's life can be in a great danger. Cryptographic mechanisms cannot prevent this attack because the query sent by the compromised node is valid for authentication.

Attacks against data authenticity: The requirements of data authenticity is also two-fold: only authenticated data that reflect the real status of the environment should be processed and relayed to the network user(s); bogus data from compromised and faulty nodes should be prohibited from being injected into network or at least filtered by network nodes. However, in our scenario, a compromised or faulty node may report false information in reply to the query of a legitimate soldier. Hence, the corresponding reactions will be intentionally misled. Again, cryptographic mechanisms cannot fully prevent this attack because the information is generated to be erroneous.

A few resilient approaches have been proposed to mitigate this attack by introducing a threshold property into their designs and thus gain the resilience against up to a certain number of compromised and faulty nodes. For instance, in order to prevent compromised nodes from reporting false alarm to the sink, SEF requires multiple (t) sensors, which sensed the event simultaneously, to collectively generate a report. Therefore, as long as there are no more than t compromised nodes, the event report will be secure. However, the effectiveness of these approaches is suspicious in practice, where the predefined threshold parameter may deviate significantly from the real situation.

Furthermore, these works are also limited in scope. That is, they usually each deal with one individual type of attack, and the resulted solutions are highly specific and not applicable to other threats in general. For example, The security design of SEF, which targets to mitigate false data injection attack, is not suitable in application of secure data aggregation. The problem of secure data aggregation has to be addressed separately by other schemes. To fully address different types of insider attacks and node random failures thus demands a number of different solutions. This is extremely inefficient if not impossible in WSNs due to lack of resources, not to mention the compatibility issue and repetitive designs.

Attacks against data availability: It is also the requirement that information of interest should be always available to the legitimate network user(s) on demand. Moreover, the adversary should be prohibited from obtaining the information from any normal sensor node without physically compromising it. However, in our scenario, a compromised node may drop an alarm report so that the legitimate soldier misses the alert. Hence, no preventive actions can be taken and the result could be vital. Cryptographic mechanisms are obviously not relevant to this attack.

Through above exemplary attacks, we have found that cryptography-based mechanisms are not sufficient to address security threats posed by compromised and faulty nodes. This motivates us to seek the proactive approaches. By monitoring the misbehavior of sensors, the compromised and faulty sensors can be proactively identified. Hence, we can dynamically prevent the compromised and faulty sensors from participating the network activities. In this

way, the impact of the compromised and faulty sensors can be greatly restricted. Also, the proactive approach is totally compatible with the current passive approaches. The two approaches hence can be combined together to provide even more promising effectiveness, if possible.

III. Network Model and Security Model

Network Model: In the proposed framework, we consider a large-scale uniformly distributed WSN that monitors a vast terrain via a large number of static homogenous sensor nodes. The WSN can be deployed through approaches such as aerial scattering. We make the following assumptions. 1) Once deployed, each node can obtain its authentic geographic location via a localization scheme. 2) The WSN is well connected and densely deployed to support fine-grained collaborative sensing and be robust against node loss and failure. 3) There could be multiple mobile network users that may quest the WSN for certain information on demand. 4) Sensor nodes are not tamper-resistant and have limited communication range. 5) Wireless links among sensors are symmetric. We assume that the signals generated by the targets or phenomena of interest form certain distributions that spread in space. 6) Multiple sensors are required to collaboratively resolve any single query.

Security Model: In the proposed framework, we assume that the known cryptography-based mechanisms are already in place when the network is deployed. That is, every sensor is assumed to be able to authenticate itself to its neighbors and the network users through its possessed cryptographic keys. We assume that there is a short bootstrapping phase right after network deployment during which no sensor nodes are compromised.

We then assume that, during network operation time, the adversary could physically compromise a small portion of network sensors and gain full control over them. We also assume the communication between the compromised nodes and the adversary cannot be detected by the sensors. Thus, all the *insider* attacks are possible in the proposed framework. We assume that the adversary is sophisticated and its goal includes: 1) obtain the information from the WSN; 2) inject false information into network; 3) prevent the legitimate users from obtaining the information. At the same time, the adversary still wants to remain undetected by other network sensors and hence, the behavior of the compromised nodes is strategic. That is, no simple deterministic assumption can be made on their behavior. Hence, brute force attacks like consistent message flooding attack, jamming attack, and message dropping attack are not the concern of this work because they can be easily detected.

In the proposed framework, we also assume that sensors may fail in terms of radio failure, sensing function error, and even system crash. Since the malfunctioning of the faulty sensors could also result in the generation of bogus data, they bring equally detrimental effects to the functioning of the network. Also, their failure patterns could be persistent, transient or probabilistic. Therefore, no simple deterministic assumption can be made on their behavior.

Attack Types		Cryptography -based solution*
Data forwarding related	Message delay attack	No
	Selective forwarding attack	No
	Message alteration attack	Yes
	Message replay attack	Yes
	Sinkhole attack	No
	Message collision attack	No
Data generation related	Bogus data attack	No
	Bogus query attack	No
	Report disruption attack	Yes
Routing related	Hello attack	Yes
	Whomhole attack	Yes
	Bogus routing info. attack	Yes
	Sybil attack	Yes
Physical related	Byzantine attack	No
	Node replication attack	Yes
	Node relocation attack	Yes
*The judgements are obtained in the context of static and location-aware WSNs with cryptographic mechanisms in place.		

TABLE I

Summary of different types of insider attacks in WSNs

IV. Misbehavior Characterization and Monitoring

A. Analysis of Insider Attacks and Sensor Failures

In this section, we attempt to characterize the misbehavior of different types of insider attacks and node random failures. To this end, we first broadly categorize different insider attacks into four different types according to their different natures: 1) Data forwarding related, 2) Data generation related, 3) Routing related, and 4) Miscellaneous. In Table 1, we classify a variety of known insider attacks into these four types. Furthermore, we have the following observations:

Regarding the attacks that are addressable by cryptography-based approaches, whatever approaches the compromised nodes take always result in authentication failures when processed by the receiver node(s). This, in turn, results in packet dropping as observed by other neighbors, if the receiver node(s) are not the final destination of the packet.

Regarding the attacks that are data forwarding related, the strategic approach for a compromised node is as follows: it may drop only up to one packet per message[1] to maximize

[1]Here we assume one message contains multiple packets.

its attack efficiency and minimize its risk of exposure; moreover, the compromised nodes may collude and drop packets multihops away to circumvent localized monitoring. However, if every message is required to be explicitly end-to-end acknowledged, such attacks can always be detected.

One strategic approach regarding attacks that are data generation related can be as follows: a compromise node occasionally (or upon request from the adversary) 1) comes up with an erroneous sensing result to cheat its neighbors during the process of collaborative sensing/aggregation, and 2) replies a received query with an erroneous report. Here, by an erroneous sensing result or report, we mean that this result is significantly different from the normal result. In reality, however, the sensing results within a neighborhood should not vary significantly. For example, the acoustic signal generated by an enemy tank attenuates spatially according to a certain distribution (e.g., normal distribution). The sensing results from the neighboring sensors thus could be assumed to form a sample of a normal distribution. Hence, the nodes with significantly different sensing results are the signs of such attacks.

Another strategic approach regarding attacks that are data generation related can be as follows: a compromise node occasionally (or upon request from the adversary) initialize a bogus query. However, message generation/relay action of a sensor can be expected by its neighbors in general. For example, a node initiates a corresponding query only when it received such a request from a legitimate user. Due to the broadcast nature of wireless links, there will be multiple nodes in a neighborhood receiving the same request at the same time. Therefore, if a neighbor node receives a query originated from its neighbor but obtains no corresponding request from the user, that neighbor can be suspicious. Moreover, the MAC protocols in WSNs is usually collision avoidance. Hence, a packet relay action can also be expected.

In general, it is very hard to detect the node launching message collision attacks. However, a sensor does have the ability to determine whether itself is jammed or not by exploring its packet delivery ratio and the neighbors' location information.

A Byzantine attack compromises the software platform of a sensor node and runs malicious code provided by the adversary [13]. Such attacks can be detected by the sensor node itself through using code attestation techniques.

At the same time, we also consider the following four types of sensor random failures: 1) message random alteration; 2) random message broadcast; 3) sensing function error; 4) random packet dropping. Obviously, these failures pose the same security threats as some of the insider attacks do. In the view of neighbor nodes, the resulted misbehavior are also the same as those from insider attacks. Thus, both insider attacks and node random failures can be characterized in the same manner.

B. Misbehavior Characterization

Based on above analysis, the following attack evaluation rules are developed accordingly in PDSF.

R1–Message acknowledgement rule: Any unacknowledged message will be treated as an evidence of an attack or a failure with respect to the responsible neighbor sensor, although the message packet may be dropped by sensors multihops away. PDSF distinguishes between the two. Hence, there are three different cases: 1) the message is successfully acknowledged; 2) the message is unacknowledged or timeout happens, but the corresponding neighbor node did relay the packet; 3) the message is unacknowledged / timeout happens, and the corresponding neighbor node did not relay the packet.

R2–Authentication failure rule: An authentication failure raised by the cryptography module is an evidence of an attack or node failure. PDSF distinguishes between end-to-end and hop-wise authentication failures. Only hop-wise authentication failure will counted as the evidence of attacks (failures) from a neighbor node.

R3–Data validity rule: PDSF assumes that the sensing results from the neighboring sensors could be modelled through a normal distribution. If a node reports significantly different sensing results, it is may be the sign of such attacks or node failures. We note that result consistency check is usually application specific. The data abnormality may only be obvious given the application context.

R4–Traffic awareness rule: PDSF assumes packet generation/relay actions of a sensor can be expected by the neighbor nodes assuming the underlying MAC protocols is collision avoidance. Any unexpected packet generation/relay action hence is the sign of such attacks or node failures.

The above four rules are used by every node to evaluate its neighbor nodes and detects misbehaved hosts. Note that all these rules are somewhat subjective. Therefore, attack detection is not always and false judgements are possible due to the inherent locality constraint of the knowledge acquired by each node. The following rules are used by every node to evaluate itself.

R5–Packet delivery rule: PDSF defines packet delivery ratio as the ratio of the number of packets that are successfully delivered to a destination compared to the number of packets that have been sent out by the sender. If a sensor finds that its packet delivery ratio is below a threshold value, it treats this as a sign of message collision attack and/or a possible type of node failures.

R6–Memory consistency rule: The memory status of a sensor node should be consistent and keeps integrity. Any code change is a sign of a compromising attack and/or a hardware failure.

R7–In-situ rule: As we assume a static WSN, every node should keep in-situ after deployment. Therefore, any location change is a sign of a compromising attack and/or a hardware failure; such change can be reflected in sensor's accelerator, if applicable.

C. Behavior Monitoring

PDSF requires each sensor operates at promiscuous mode, monitoring all its neighbor nodes and itself. Specifically, PDSF monitors the following aspects:

- Packet Forwarding Behavior: PDSF overhears the channel and compares ongoing data traffic with the recorded routing/MAC messages. In addition, timer and explicit acknowledgement mechanisms are used to detect packet drop and duplication.
- Time-Space Data Consistency: PDSF obtains data validity information from the application modules regarding their neighbor nodes. Note that the definition of validity is application specific and thus should be individually designed according to different application natures.
- Traffic-related Behavior: PDSF assumes neighbor nodes' traffic-related behavior, such as message generation, relay, and duty cycle (sleep schedule) to be expectable and clear from the context.
- Cryptographic Failures: PDSF captures every failure raised by the cryptography module. Such failures are inevitably the sign of attacks and/or failures.
- Self-Status: PDSF keeps tracking its own packet delivery ratio, attesting its own memory status at real time, and checking its physical location information (e.g., through the accelerator or GPS).

So far, we have developed a set of behavior monitoring criteria for WSNs. Based on our in-depth analysis, these summarized criteria can be used to effectively detect the insider attacks and node failures. However, the detection is not always and false detections are possible. The design logic behind these criteria is to keep the individual monitoring action of each sensor independent. Each sensor monitors its neighborhood and makes decisions by itself. We intentionally avoid using collaborative monitoring mechanism, although it has better detection efficiency and accuracy. This is because collaborative monitoring mechanism is usually very complicated and energy inefficient in WSNs. PDSF follows a much simpler decision fusion approach as described in next section to enhance the detection efficiency and accuracy.

V. Trust Management Framework

Besides misbehavior characterization, the other building block of the proactive data security framework is trust management, which takes the output of behavior monitoring as its input. Generally, the main functionalities of trust management in PDSF consist of two parts. One is to dynamically adjust the trust level, or say reputation, that is assigned to one sensor according to its behavior. The other is to generate the alert messages to expel malicious or selfish sensors from the network.

A. Principles of Designing The Trust Management Model

There has been extensive research on trust and reputation systems in terms of peer-to-peer networks [5]–[7] and mobile ad hoc networks [8], [9]. Here, we present the principles employed to design the trust management model in PDSF and point out the differences between our model and previous work, wherever they exist. The proposed trust management model follows the below principles.

Principle I: *At the initialization stage, each sensor can fully trust all its immediate neighbor nodes in the first place unless they are proved guilty.*

It is different from reputation systems in peer-to-peer networks and mobile ad hoc networks. The reason behind is that, in those networks, the major responsibility of the trust party or the group manager (if any) is to control the access to the network. She may simply assign an identity to a new node without any verification. Or in cases that the traceability property is desirable, she may verify and bind the identity of a new node with the entity holding it. However, neither the trust party nor the group manager can judge whether the newly-joined node is good or bad. In contrast, in sensor networks, all the sensors are provided and scattered by the same trusted entity, and thus they are supposed to be good at least at the beginning.

Principle II: *The trust system should be built upon previous both good and bad experiences (i.e. positive and negative feedbacks), which, at the same time, should be direct knowledge.*

The trust management models based on only one type of feedback are insufficient. The trust models based on previous positive feedbacks only can be cheated in a way that, colluded sensors send good reports for each other. On the other hand, counting only negative feedbacks is insufficient as well, because sensors executing a few malicious operations would be rated the same or better than those sensors which successfully transmit a large amount of data but have a few failures (e.g. due to unreliable wireless communication), which is obviously unfair. In PDSF, since the monitoring module provides both positive and negative feedbacks among the neighbor nodes, a trust model based

on both feedbacks can hence be implemented. Moreover, to reduce the complexity, each sensor node maintains the reputation information only for its immediate neighbors.

Principle III: *Reputation fading is turned off, when the trust value of a sensor drops below a predetermined threshold.*

As soon as the trust value of a given sensor drops below a predetermined threshold, the detecting sensor(s) will refuse any further interaction with this sensor by itself. And an alert message will also be generated to call for expelling the sensor nodes from network. If a sensor is expelled, it will not be able to participate in any operation or enjoy any service, e.g. requesting other sensors to forward its packet from then on. Many reputation systems in peer-to-peer networks and mobile ad hoc networks [7], [8], [11] allow the reputation fading. The purpose of reputation fading to allow for the redemption of nodes that are no longer misbehaving. In sensor networks, sensor random failures pose the similar security impacts as certain types of insider attacks do. However, we view sensor random failures as events which are less frequent, unless the hardware is broken, and thus it is safe to set a threshold such that, if the trust level of a sensor is lower than this threshold, we assert that this sensor is a malicious one, although node random failures might owe partially to its low trust value. Moreover, we argue that, once it happens, this sensor either is compromised or has hardware problems. Given the fact that there is no sensor recovery mechanism in place, in PDSF, the reputation fading mechanism should be turned off for those sensors, whose trust values are below the threshold.

Principle IV: *To expel a sensor node in question, a voting procedure among its neighbor nodes is required.*

To prevent potential slanders against innocent sensors, before determining whether to expel a given sensor node, a voting procedure is required among the neighbors of that sensor. That is, consensus has to be reached among the neighbor nodes.

B. The Trust Management Scheme for PDSF

Most trust and reputation systems proposed so far can be extended under the principles described in Section V-A. Therefore, instead of giving a detailed trust metrics for computing the trust value/reputation of a sensor, in this paper, we present the procedures of building the trust management scheme for PDSF in a more generic way.

1) Calculating Trust Values: In PDSF, the trust management scheme bootstraps itself as soon as the WSN is deployed. Based on Principle II, each sensor builds up a table recording the trust values and the counters for positive and negative operations of all its neighbors, which it is able to interact with and monitor on. And all the trust values in the list are set to the maximum value, according to Principle I. For example, given that the trust value is normalized to $[0,\ 1]$, the initial trust values for all the neighbors are 1. The initial values for both of the counters are 0.

The fundamental functions of WSNs are data sensing, processing, and reporting. Besides that, the data needs to be forwarded towards the sink, and in order to lower down the communication cost the data might be aggregated along the route to the sink. Hence, in PDSF, each sensor is allowed to accumulate its trust by correctly participate in these activities. For instance, each successful completion of these activities increases the positive counter by 1. In some applications, the value or significance of the activity is context-based. In such cases, the increase of the positive counter is activity-dependent. In peer-to-peer networks and mobile ad hoc networks, certain incentive or say credit can be given to nodes answering the queries on the trust value. However, in sensor networks, considering the fact that all the sensors are from the same party, responding to the reputation queries is not voluntary but compulsory. Consequently, sensors in PDSF do not increase the positive counters for others because of their cooperation in the trust scheme.

On the other hand, the decrease of negative counters is based on the detections of the violations of rules defined in Section IV-B. For each violation of R1 to R4, the sensor detecting the violation will increase the negative counter for the one breaking the rule. Note that, refusing or failing to respond to a query on the trust value, detected by the behavior monitoring mechanism described in Section IV-C, is viewed as a violation of R1.

Based on the values of both the positive and negative counters, the trust value of a sensor can be calculated and updated accordingly. Different trust metrics, e.g. the beta distribution [12], can be used at this stage. To prevent from advanced attacks, e.g. strategic dynamic personality attacks where malicious peers can build a reputation and then start cheating or oscillating between building and milking the reputation, more refined trust metrics like TrustGuard [7] can be employed. Certainly, there is a trade-off between the computation and communication cost and the accuracy of evaluating the trust values of sensors.

In certain cases, it is possible that the trust value of a sensor is set directly to a value, instead of being calculated from the trust metrics. For example, when R6 or R7 is violated, every node in the neighborhood of the rule-breaker denoted as R will set the trust value of R to zero and prohibit it from participating any network activities forever.

In case of violations of R5, an alert message will be sent to all the neighbors and the sink. If applicable, certain

higher level intrusion detection action can be taken by the sink to identify the source of the attack sensors. This is because it is usually impossible for the sensor nodes to detect which of its neighbors is the attacker node(s). This is especially true when the message collision attack (jamming attacks) happens below the network layer (i.e., at MAC or physical layer).

2) Expelling Malicious Sensors: To increase the efficiency of the detection and prevent potential slanders against innocent sensors, PDSF adopts a neighbor voting approach as follows.

When a sensor detects a neighbor denoted as R with a reputation below a predetermined threshold[2], it automatically excludes the sensor from any of activities it involves in. In addition, an alert message is sent out to call for a vote on R. If there is no less than m out of all the n nodes in the corresponding neighborhood with similar opinions on R, it will be excluded by all the nodes in its neighborhood. Note that, PDSF does not require sensor nodes to perform collaborate monitoring, since it is too complex and cheat-prone for WSNs. Rather, PDSF uses the much simpler decision fusion approach to improve detection efficiency and accuracy, but keeps monitoring activity independent to reduce protocol complexity and overheads.

3) On-demand Querying Trust Values of Remote Sensors: When a sensor node interacts with a remote node (e.g., receiving a query from a remote node), it indirectly judges the trustworthiness of that node on-the-fly and decides whether or not to answer the query. Since each sensor only has reputation knowledge of its neighbors, the reputation information of a remote node has to be evaluated using a different approach. PDSF follows a distance-aware minimal trustworthy route approach. Specifically, this approach first estimates the number of hops between the remote node and the local node based on their location information. Then the local node tries to find at least one path of the same hops to reach the remote node. All the consecutive pair of nodes along the route have to have a minimal trust value no less than 0.5 mutually. If the route is successfully found, then the remote node is assumed to be trustworthy.

VI. Concluding remarks

In this paper, we argued that cryptography alone is insufficient to fully address the insider attacks in the presence of both the compromised and faulty sensor nodes. We explored a unified solution, which serves as the front line in defending against all types of security threats. Specifically, we further proposed a proactive data security framework (PDSF) to identify compromised and faulty nodes proactively and prohibit them from participating network activities in a dynamic manner. PDSF consists of two key modules, misbehavior characterization & monitoring, and trust management. The former characterizes different types of misbehavior in WSNs and defines a set of monitoring criteria. And the latter develops a trust management model, which adapts itself to the resource constrained and application specific nature of the WSNs.

[2]The threshold is chosen in such a way that, the probability that the trust value of a sensor neither compromised nor physically broken is below the threshold is negligible.

Acknowledgement

This work was supported in part by a research grant from AirSprite Technologies, Inc., Marlborough, MA, USA.

References

[1] D. Wagner, "Security for Sensor Networks: Cryptography and Beyond," http://www.cs.berkeley.edu/ daw/talks/SASN03.ppt.

[2] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An Interleaved Hop-by-hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks", IEEE Symposium on Security & Privacy, Oakland, CA, May 2004.

[3] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statiscal Enroute Filtering of Injected False Data in Sensor Networks", IEEE Infocom, HongKong, China, Mar. 2004.

[4] K. Ren, W. Lou and Y. Zhang, "LEDS: Providing Location-aware End-to-end Data Security in Wireless Sensor Networks," IEEE INFOCOM, Barcelona, Spain, 2006.

[5] E. Damiani, S. Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks", ACM CCS, 2002, pp. 207-216.

[6] S. Kamvar, M. Schlosser, and H. Garcia- Molina, "The eigentrust algorithm for reputation management in P2P networks," WWW, 2003, pp. 640-651.

[7] M. Srivatsa, L. Xiong, and L. Liu, "TrustGuard: Countering vulnerabilities in reputation management for decentralized overlay networks", WWW 2005, pp. 422-431.

[8] S. Buchegger and J. Boudec, "Performance analysis of the CONFIDANT protocol (cooperation of nodes: Fairness in dynamic ad-hoc NeTworks)", MobiHOC, June 2002.

[9] Pietro Michiardi and Refik Molva, "CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks", IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security, 2002, pp. 107-121.

[10] P. Michiardi and R. Molva, "Analysis of coalition formation and cooperation strategies in mobile ad hoc networks", Ad Hoc Networks, vol. 3, pp. 193-219, 2005.

[11] S. Buchegger, "Coping with Misbehavior in Mobile Ad-Hoc Networks", Ph.D. thesis, EPFL, Feb. 2004.

[12] A. Jøang and R. Ismail, "The beta reputation system", The 15th Bled Electronic Commerce Conference, 2002.

[13] E. Shi, L. Doorn, and A. Perrig, "BIND: A Time-of-use Attestation Service for Secure Distributed Systems," IEEE Symposium on Security and Privacy 2005.

[14] A. Perrig, B. Przydatek, and D. Song "SIA: Secure Information Aggregation in Sensor Networks," ACM SenSys 2003.

[15] D. Wagner, "Resilient Aggregation in Sensor Networks," ACM SASN '04, October 25, 2004.