ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Intrution Resilience Using Self-Healing Mechanism In Mobile Unattended Wsns

Roshil K Das[1], J.Arun[2]

Department of Information Technology, Maharaja Engineering College, Avinashi, Tamilnadu, India [1]

Asst. prof, Department of Information Technology, Maharaja Engineering College, Avinashi, Tamilnadu, India[2]

Abstract: Wireless Sensor Networks (WSNs), by their distributed nature, limited sensor resources, and lack of tamper resistance the networks are affected to a large area of attacks. The security measures become ineffective, if a sensor is corrupted and thereby the adversary learns all secrets. Trusted third party or access to a source of high-quality cryptographic randomness is required to recover secrecy, once a sensor is compromised with an adversary. But in the case of an Unattended Wireless Sensor Networks (UWSNs) the operation is not similar to Wireless Networks; here the sink visits the network periodically. The sensors in this network move according to some mobility models, that is why it is named as intrusion resilience in Mobile Unattended Wireless Sensor Networks. The advantage of this network is, it is independent from security (e.g., sensors move to improve area coverage). By leveraging sensor mobility it also introduces a cooperative protocol and it allows compromised sensors to recover secure state after compromise.

Keywords: Network security, mobility models, cooperative protocol.

I. INTRODUCTION

A wireless sensor network (WSN) of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The adversary can compute secrets for round r0 > r, but it cannot compute any secrets used in prior rounds. Note that this is possible even if the adversary is no longer in control of the sensor in round r0. Backward secrecy is much more challenging, because knowledge of Kr allows the adversary to compute secrets for future rounds. It would be trivial to obtain backward secrecy if each sensor had a True Random Number Generator (TRNG). Because a TRNG yields information- theoretically independent values, even if the adversary learns many (but not all) TRNG outputs, it cannot compute the missing values, whether they correspond to the past or to the future. In other words, when the adversary compromises the sensor, it cannot learn past secrets; once the adversary leaves the sensor it will not be able to compute future sensor secrets. Unfortunately, TRNGs are not found on commodity sensors and not expected to be available for the near future. An alternative to per-sensor TRNGs is the presence of a trusted third party (TTP); this is assumed in key-insulated schemes. In such schemes, forward and backward secrecy is achieved by having end-devices (e.g., the sensors) evolve their secrets in cooperation with a TTP, called a base. Unless both the end-device and the base are compromised at the same time, per-round keys are insulated. Key-insulated schemes are well matched for WSNs with a constantly present sink, where the latter acts as a base. However, there are settings where the constant presence of a sink is not viable.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

II. RELATED WORK

Di Ma and Gene Tsudik: Suggested, Unattended wireless sensor networks (UWSNs) operating in hostile environments face the risk of compromise. Unable to off-load collected data to a sink or some other trusted external entity, sensors must protect themselves by attempting to mitigate potential compromise and safeguarding their data. It refers on techniques that allow unattended sensors to recover from intrusions by soliciting help from peer sensors. Hence define a realistic adversarial model and show how certain simple defense methods can result in sensors re-gaining secrecy and authenticity of collected data, despite adversary's efforts to the contrary. Here presents an extensive analysis and a set of simulation results that support observation and demonstrate the effectiveness of proposed techniques.

Mauro Conti, Roberto Di Pietro and Luigi V. Mancini: Suggested, the nature of mobile ad hoc networks (MANETs), often unattended, makes this type of networks subject to some unique security issues. In particular, one of the most vexing problems for MANETs security is the node capture attack: An adversary can capture a node from the network eventually acquiring all the cryptographic material stored in it. Further, the captured node can be reprogrammed by the adversary and re-deployed in the network in order to perform malicious activities. In particular, one of the most vexing problems for MANETs security is the node capture attack: The nature of mobile ad hoc networks (MANETs), often unattended, makes this type of networks subject to some unique security issues. In particular, one of the most vexing problems for MANETs security is the node capture attack: An adversary can capture a node from the network eventually acquiring all the cryptographic material stored in it. It also addresses the node capture attack in MANETs. Start from the intuition that mobility, in conjunction with a reduced amount of local cooperation, helps computing effectively and with limited resource usage network global security properties. Then, developed this intuition and use it to design a mechanism that to detect the node capture attack. It support our proposal with a wide set of experiments showing that mobile networks can leverage mobility to compute global security properties, like node capture detection, with a small overhead.

Jokhio, I.A and Kemp, A.H: Suggested, one of the most vexing problems in wireless sensor network security is the node capture attack. An adversary can capture a node from the network as the first step for further different types of attacks. For example, the adversary can collect all the cryptographic material stored in the node. Also, the node can be reprogrammed and re-deployed in the network in order to perform malicious activities. To the best of the knowledge no distributed solution has been proposed to detect a node capture in a mobile wireless sensor network. And hence propose an efficient and distributed solution to this problem leveraging emergent properties of mobile wireless sensor networks. In particular, introduce two solutions: SDD, that does not require explicit information exchange between the nodes during the local detection, and CCD, a more sophisticated protocol that uses local node cooperation in addition to mobility to greatly improve performance. Here also introduce a benchmark to compare these solutions with. Experimental results demonstrate the feasibility of our proposal. For instance, while the benchmark requires about 9,000 seconds to detect node captures,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

CDD requires less than 2,000 seconds. These results support our intuition that node mobility, in conjunction with a limited amount of local cooperation, can be used to detect emergent global properties.

Mauro Conti, RobertoDi Pietro, AndreaGabrielli, Luigi Vincenzo Mancini, and AlessandroMei: Suggested, Mobile Ad Hoc networks are subject to some unique security issues that could delay their diffusion. Several solutions have already been proposed to enforce specific security properties. Hover, mobility pattern nodes obey to can, on one hand, severely affect the quality of the security solutions that have been tested over"synthesized" mobility pattern. On the other hand, specific mobility patterns could be leveraged to design specific protocols that could outperform existing solutions. Hence, there investigate the influence of a realistic mobility scenario over a benchmark mobility model (Random Waypoint Mobility Model), using as underlying protocol a recent solution introduced for the detection of compromised nodes. Extensive simulations show the quality of the underlying protocol. However, the main contribution is to show the relevance of the mobility model over the achieved performances, stressing out that in mobile ad-hoc networks the quality of the solution provided is satisfactory only when it can be adapted to the nodes underlying mobility model.

III. THE NETWORK SIMULATOR 2.33 (NS2)

Network Simulator (NS2) is a discrete event driven simulator developed at UC Berkeley. It is part of the VINT project. The goal of NS2 is to support networking research and education. It is suitable for designing new protocols, comparing different protocols and traffic evaluations NS2. A large amount of institutes and people in development and research use, maintain and develop NS2. This increases the confidence in it. Versions are available for FreeBSD, Linux, Solaris, Windows and Mac OS X.

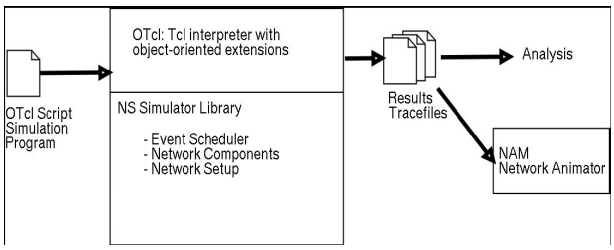

Figure 1.Simplified User's View of Ns

NS2 interprets the simulation scripts written in OTcl. A user has to set the different components (e.g. event scheduler objects, network components libraries and setup module libraries) up in the simulation environment. The user writes his simulation as a OTcl script, plumbs the network components together to the complete simulation. If he needs new network components, he is free to implement them and to set them up in his simulation as ill. The event scheduler as the other major component besides network components triggers the events of the simulation (e.g. sends packets, starts and stops tracing). Some parts of NS2 are written in C++ for efficiency reasons. The data path (written in C++) is separated from the control path (written in OTcl). Data path object are compiled and then made available to the OTcl interpreter through an OTcl linkage (tclcl) which maps methods and member variables of the C++ object to methods and variables of the linked OTcl object. The C++ objects are controlled by OTcl objects. It is possible to add methods and member variables to a C++ linked OTcl object.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

IV. PROTOCOLS

4.1.1 Table Driven (Or Proactive)

The nodes maintain a table of routes to every destination in the network, for this reason they periodically exchange messages. At all times the routes to all destinations are ready to use and as a consequence initial delays before sending data are small. Keeping routes to all destinations up-to-date, even if they are not used, is a disadvantage with regard to the usage of bandwidth and of network resources. These protocols ire designed to overcome the wasted effort in maintaining unused routes. Routing information is acquired only when there is a need for it. The needed routes are calculated on demand. This saves the overhead of maintaining unused routes at each node, but on the other hand the latency for sending data packets will considerably increase.

4.1.2. On-Demand (Or Reactive)

These protocols are designed to overcome the wasted effort in maintaining unused routes. Routing information is acquired only when there is a need for it. The needed routes are calculated on demand. This saves the overhead of maintaining unused routes at each node, but on the other hand the latency for sending data packets will considerably increase. The nodes maintain a table of routes to every destination in the network, for this reason they periodically exchange messages.

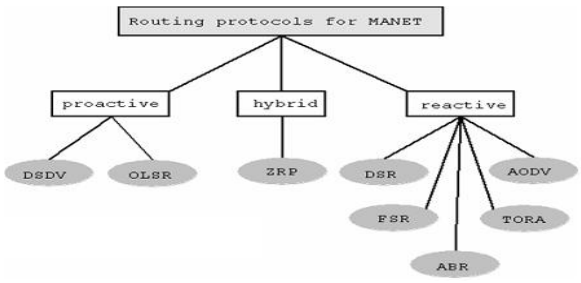

Figure2. Routing protocol structure

4.2 DYNAMIC SOURCE ROUTING PROTOCOL

DSR is a reactive routing protocol which is able to manage a MANET without using periodic table-update messages like table-driven routing protocols do. DSR was specifically designed for use in multi-hop wireless ad hoc networks. Ad-hoc protocol allows the network to be completely self-organizing and self-configuring which means that there is no need for an existing network infrastructure or administration.

4.2.1 Path-finding-process: Route Request & Route Reply

If node A has in his Route Cache a route to the destination E, this route is immediately used. If not, the Route Discovery protocol is started:

1. Node A (initiator) sends a RouteRequest packet by flooding the network

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

2. If node B has recently seen another RouteRequest from the same target or if the address of node B is already listed in the Route Record, Then node B discards the request!
3. If node B is the target of the Route Discovery, it returns a RouteReply to the initiator. The RouteReply contains a list of the "best" path from the initiator to the target. When the initiator receives this RouteReply, it caches this route in its Route Cache for use in sending subsequent packets to this destination.
4. Otherwise node B isn't the target and it forwards the Route Request to his neighbors (except to the initiator).

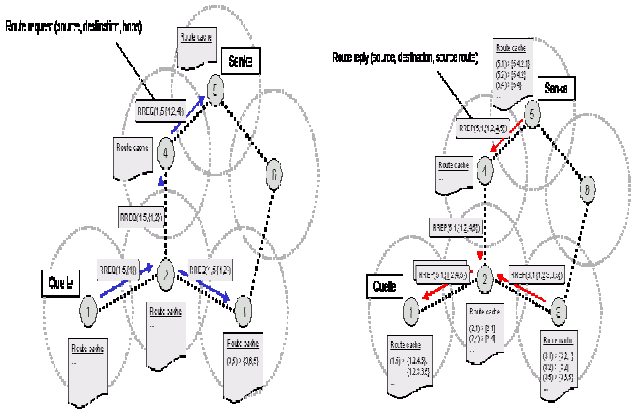

Figure 3.Path-finding-process: Route Request & Route Reply

V. ANALYSIS

5.1. Topology Formation

Constructing Project design in NS2 should takes place. Each node should send hello packets to its neighbor node which are in its communication range to update their topology.

5.2. Adversarial Compromising

In UW SN setting, sensors are mobile, while the adversary is static. The envisioned adversary differs from other adversarial models considered in most prior WSN security literature. The latter is static in terms of the set of sensors it corrupts, i.e., it compromises k out of n sensor throughout the network lifetime. Our adversary (ADV) is stationary with respect to the portion of the deployment area it controls; but, the set of compromised sensors changes as nodes move in and out of the adversary-controlled area.

5.3 Collaborative Intrusion Resilience

A number of techniques discover the sensor compromise when the adversary modifies the sensor code. Hence, if the adversary is limited to "read-only" attacks and keeps the sensor code unchanged, there is no way to tell whether that sensor has ever been compromised. This allows ADV to stay undetected and benefit from repeated attacks to the network. For small neighborhood sizes (i.e., B ' 2), the network exhibits a self-healing property that, with high probability, allows sensors to regain secret state as soon as they move away from the adversary-controlled regions.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

ADVANTAGES:
Node mobility helps to solve network connectivity problems caused by sensor failures and allows sensors to adapt their sampling power to respond to precise events.

1. a green sensor remains green until it moves at distance less from ADV ;

2. a red sensor cannot become green without becoming yellow first as ADV eavesdrops on red sensors;

3. a yellow sensor can become green only if it receivesat least one contribution from a green sensor.

Main motive of this paper is Minimize the red sensors, maximizing the green sensors using self healing mechanism.

VI. CONCLUSION

In this paper, it provides several contributions to the UWSN field. First, a new adversary model that spreads over different areas of the deployment field is introduced. Second, when assessing self-healing protocols in autonomous distributed systems there exist two novel metrics that, other than being interesting on their own, are of general help. Third, it is studied, for a large range of system parameters, it understood that how the degree distribution of the adversary affects the self-healing protocol. Especially the latter shows a great capability to recover from compromising for several deployment settings while incurring a negligible overhead—only local communications are required. Finally, repeated analysis and extensive simulation do support these findings.

REFERENCES

[1]. Ma D and Tsudik g., (2008). "Dish: Distributed Self-Healing," Proc. 10th Int'l Symp. Stabilization, Safety, and Security of Distributed Systems (SSS '08), pp. 47-62.
[2]. Dutta R., Wu Y.D and Mukhopadhyay S., (2007) "Constant Storage Self-Healing Key Distribution with Revocation in Wireless Sensor Network," Proc. IEEE Int'l Conf. Comm. (ICC '07), pp. 1323-1328.
[3]. Naik V., Arora A., Bapat, and Gouda M.G., (2003)"Whisper: Local Secret Maintenance in Sensor Networks," IEEE Distributed Systems Online, vol. 4, no. 9,.
[4]. Dantu K., RahimiM.H., Shah H., Babel S., Dhariwal A., and Sukhatme G.S., (2005) "Robomote: Enabling Mobility in Sensor Networks," Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN '05), pp. 404-409,.
[5]. Cortes J., Martınez S., Karatas T., and Bullo F., (2002) "Coverage Control for Mobile Sensing Networks," Proc. IEEE Int'l Conf. Robotics and Automation (ICRA '02), pp,.
[6]. Wang G., Cao G., La Porta T.F.,, and Zhang.W., (2005) "Sensor Relocation in Mobile Sensor Networks," Proc. IEEE INFOCOM, pp. 302-312,.
[7]. Rahimi M H., Shah H., Sukhatme G S., Heidemann J.S., and Estrin D.,(2003) "Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network," Proc. IEEE Int'l Conf. Robotics and Automation (ICRA '03), pp. 19-24,.
[8]. Camp T., Boleng J, and Davies V., (2002) "A Survey of Mobility Models for Ad Hoc Network Research," Wireless Comm. and Mobile Computing (WCMC), Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, pp. 483-502,.
[9]. Di Pietro R., Oligeri.G., Soriente.C., and Tsudik.G,,(2010) "Intrusion-Resilience in Mobile Unattended WSNs," Proc. IEEE INFOCOM, pp. 2303-2311,.
[10]. Francillon A., and Castelluccia C., (2008) "Code Injection Attacks on Harvard-Architecture Devices," Proc. 15th ACM Conf. Computer and Comm. Security (CCS '08), pp. 15-26,.