Indian Journal of Communications Technology and Electronics (IJCTE)
Vol.2.No.1 2014pp 7-12.
available at: www.goniv.com
Paper Received :05-03-2014
Paper Published:28-03-2014
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

DISRUPTION MALLEABILITY IN MOBILE UNATTENDED WSN

1K.Kavitha, and 2J. Jothi
Department of Electronics and Communication Engineering,
Mailam Engineering college, Mailam.
Email: kavisajas03@gmail.com

ABSTRACT

Wireless sensor networks are susceptible to a wide range of attacks. Some wireless sensor networks preclude the constant presence of a centralized data collection point, that is, a sink. In such a disconnected or unattended setting, nodes must accumulate sensed data until it can be off loaded to an itinerant sink. Furthermore, if the operating environment is hostile, there is a very real danger of node and data compromise. The unattended nature of the network makes it an attractive target for attacks that aim to learn, erase, or modify potentially valuable data collected and held by sensors. We argue that adversarial models and defense techniques in prior WSN literature about security are unsuitable for the unattended WSN setting. We define a new adversarial model by taking into account special features of the UWSN environment. We show that in the presence of a powerful mobile adversary, we focus on intrusion resilience in mobile unattended wireless sensor networks, where sensors move according to some mobility models. By using IDR protocol, secure the information.

Keywords: WSN, sink, UWSN, mobility models, IDR

1. INTRODUCTION

Within the last decade, sensors and sensor networks have been extremely popular in the research community. In particular, security issues in wireless sensor networks (WSNs) have received a lot of attention. Because of the low cost of individual sensors and due to meager resources, security poses unique and formidable challenges [1]. One common assumption in prior WSN security research was that data collection is performed in (or near) real time: a trusted entity — usually called a sink — is assumed to always be present. Presence of an online sink enables nodes to submit measurements soon after sensing. As a consequence, an adversary capable of compromising nodes and corrupting data has relatively little time to attack. Although many WSNs operate in this mode, there are WSN scenarios and applications that do not fit into the real-time data collection model. We refer to such networks as unattended WSNs (UWSNs).

Consider the following examples:

• A tree-mounted WSN composed of noise sensors, installed in a protected area (e.g., in a national park) to monitor firearm discharge (to detect poaching) or sawing (to detect illegal tree logging). The size of the protected area, its inaccessibility, and/or the difficulty of hiding a sink, can motivate the requirement for an itinerant sink.

• A subterranean WSN deployed along an international border to record illegal crossings. The scale, in terms of both the number of sensors and the area they must cover, might make it too costly to install a multitude of stationary sinks, one per border segment. Instead, periodic visits by a mobile sink (e.g., mounted on a border patrol vehicle) might be more realistic.

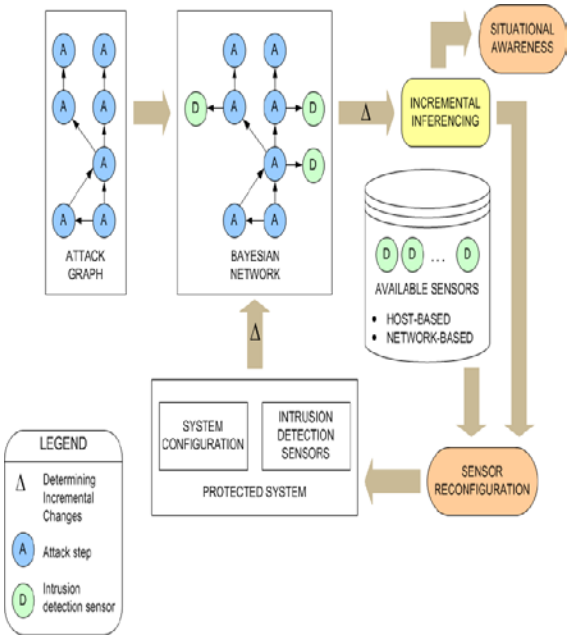

Fig 1.1 Detailed designs

One common feature in these examples is that constant physical access to the entire network is impossible, and sink visits are sporadic. Consequently, sensors cannot off load data in real time: they must accumulate data in situ and wait for an explicit upload signal. We further narrow the scope to UWSNs operating in hostile environments. Unattended sensors deployed in such environments represent an attractive and easy target for an adversary. The inability of the sensors to off load data in real time exposes them and their data to increased risk. Without external connectivity, sensors can be compromised (without detection), and collected data can be read, altered, or simply erased. Sensor compromise is a realistic threat because a typical sensor is a mass-produced commodity device with no specialized secure hardware or tamper-resistant components. In prior security research, often it was assumed that a number of sensors can be compromised during the entire operation of the network. (Thus, the main challenge is to detect such compromise.) This is a reasonable assumption because — given a constantly present sinks — attacks can be detected and isolated. The sink can take appropriate action immediately to prevent further compromise. In contrast, in a UWSN, the adversary can compromise a maximum number of sensors within a particular time interval.

2. ADVERSARY MODEL

The UWSN model considered in prior work assumes a mobile adversary that migrates among different subsets of compromised sensors. In our UWSN setting, sensors are mobile, while the adversary is static. This latter operating hypothesis, other than being worth investigating on its own, is also motivated by the fact that the adversary might not have enough "resources" to move or there might just be no incentive for it to be mobile, i.e., it might as well be stationary and wait for sensors to move to its controlled area. Previous work [12] has shown that the adversarial mobility model has no or very little impact on the network performance in terms of resiliency, when sensor is mobile. Hence, in this paper we focus on the impact on self-healing of a distributed, static adversary.

Further, 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.

3. ENVISAGED NETWORK ENVIRONMENT

We assume a UWSN consisting of a multitude of homogeneous low-cost sensor nodes distributed over a certain geographical area. The term "unattended" means, as discussed previously, that sensors cannot communicate with the sink at will and that the network is not under constant supervision. The unattended nature of the network might be caused by the design requirement to avoid a central/single point of failure. Alternatively, as illustrated in the two examples in the previous section, it might be caused by poor or sporadic accessibility of the deployment area. In any case, the sink is assumed to be mobile.

4. THE MOBILE ADVERSARY

We anticipate a powerful mobile adversary, hereafter referred to as a μADV. One important feature that separates it from other adversarial models is its *mobility*. We assume that a μADV can compromise a subset (up to a certain size) of sensors within a given time interval. The subset of compromised nodes might not be clustered or contiguous, that is, concurrently compromised nodes can be spread throughout the entire UWSN topology. Furthermore, in the next interval, a μadv can migrate and compromise a different subset of sensors [3]. The time it takes a μADV to compromise a set of nodes is much shorter than the time between two successive visits of the sink. Thus, given enough compromise intervals, a μADV can gradually subvert the entire network.

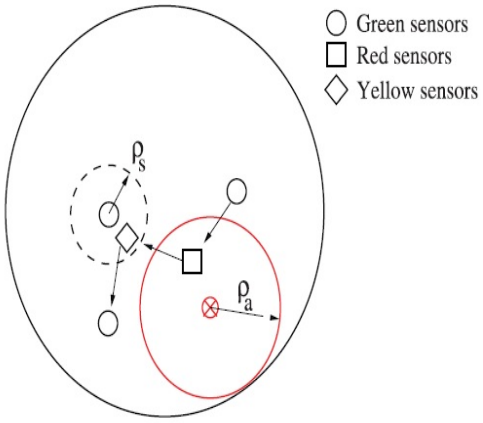

Fig 4.1 Scenario of sensors

While a μADV occupies a given node, it can read and possibly write to the storage, memory, and all of the communication interfaces of the node. At the very least, it can learn all node secrets, as well as eavesdrop on all relevant communication. Of course, nothing prevents a μADV from physically destroying or damaging sensors, in particular because the network is unattended most of the time. However, such crude behavior leaves evidence. Another one of our assumptions is that a μADV is subtle and prefers to operate in a stealthy manner [4]. Therefore, because it wants to "leave no trail," the movements of a μADV are unpredictable, and they also are untraceable. Specifically, it is impossible to detect if or when a μADV compromised a particular sensor.

We now consider several types of μADVs, each with slightly different goals:

4.1 Curious μADV — aims to learn as much sensed data as possible. It is not hard to read data from RAM and/or ROM of a commodity sensor [4]. With no countermeasures, a μADV can compromise nodes and read their data directly. Of course, a μADV might be especially interested in learning some specific measurements that represent critical or high-value data.

4.2Search-and-erase μADV — Aims to prevent certain target data from reaching the sink. Consider, for example, a sensor network that monitors nuclear emissions, where the sink raises an alarm if one of the nodes reports a value above a certain threshold. The goal of the μADV might be to find that value and erase it before it ever reaches the sink. If we assume that the sink tolerates some missing measurements (due to occasional errors or malfunctions), an ad remains undetected even if it succeeds.

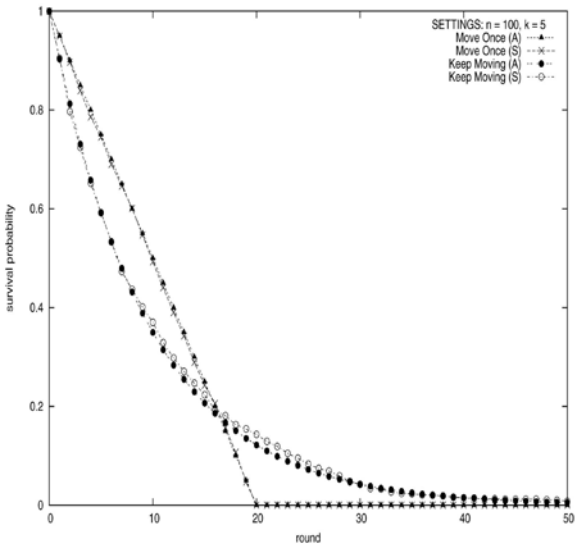

Fig 4.2Survival of different defense strategies against search- and -erase

4.3Search-and-replace μADV — if we assume that the sink has no tolerance for lost data, the corresponding adversary model changes from search-and-erase to search-and replace. In this version, a μADV also aims to prevent target data from reaching the sink. However, it wants to replace the target data with concocted value(s). It is not hard to envision other types of adversarial behavior relevant to UWSNs. For example, we can imagine a *polluter* μADV whose goal is to confuse or mislead the sink by introducing many fraudulent measurements into the UWSN; or, one that aims to indiscriminately erase as much data as possible, the main goal being denial-of-service. However, at least initially, we do not consider these types of behavior because
They violate our previously stated assumption that a μADV wishes to remain stealthy.

We further distinguish between a *proactive* and a *reactive* adversary. The latter is assumed to be dormant (inactive) until it receives a signal to respond to certain target data. As soon as this happens, the μADV *reacts* and starts compromising nodes in order to accomplish its goal. In contrast, a proactive μADV roams the network ahead of time, compromises subsets of nodes, and waits for a signal to respond to certain target data. A proactive μADV, as we discuss below, has certain advantages over its reactive counterpart.

5. GENERAL CONCEPTS

In this section, we introduce preliminary general concepts.

5.1 Forward and Backward Security— considering that the compromise of a given sensor has certain duration, we can partition sensor-collected data into the following three categories, based on the time of compromise:

1 Before compromise
2 During compromise
3 After compromise

The security of category 1 data is referred to as forward security. *Forward security* means that even if a μADV obtains the current secrets of a sensor, it cannot decrypt (or forge authentication tags for) data collected and encrypted (or authenticated) before compromise; whereas, the security of category 3 data is referred to as backward security. *Backward security* means that a μADV that obtains the current secrets of a sensor cannot decrypt (or forge authentication tags for) data in category 3. Of course, nothing can be done about the security of category 2 data because during that time, a μADV is in full control of the sensor.

Modern cryptography offers a number of tools and techniques to achieve either forward security only or both forward and backward security (sometimes referred to as *keyinsulation*). Key evolution is a common theme in all of these techniques. Time is divided into fixed intervals. If public cryptography is used, the public key remains fixed throughout the entire lifetime of the system; whereas, the private key is updated in each interval. After the private key of the next interval is computed, the current private key and other intermediate values are deleted. With symmetric cryptography, the pair wise (shared) key is evolved at the end of each interval.

Forward-secure techniques rely on key evolution alone, without resorting to any trusted third party. That is, usually, the private key is updated by its owner through a one-way function. In contrast, in key-insulated schemes, key evolution is performed collectively by the owner and an outside helper called a *base* — a separate secure entity, typically in the form of a remote trusted server or local tamper-resistant hardware.

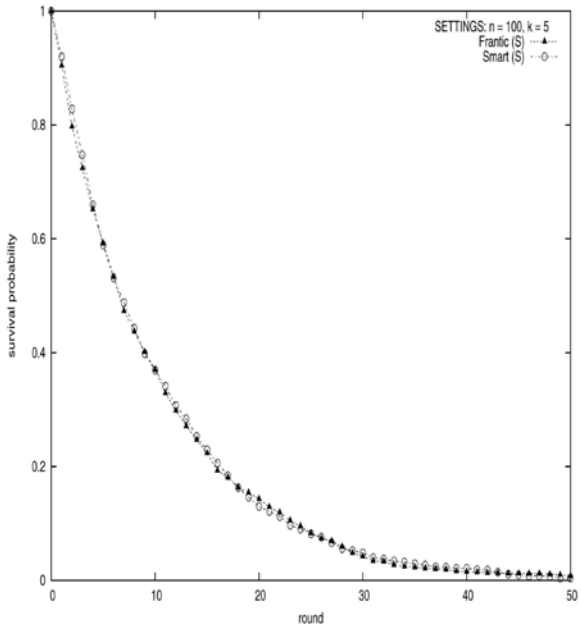

Fig 5.1 Comparison of survival probability and round

5.2 Random Number Generators— we distinguish between two types of random number generators: a true random number generator (TRNG) and a pseudo-random number generator (PRNG). The former extracts randomness from physical phenomena and generates information, theoretically independent values. That is, given an arbitrarily long sequence of consecutive TRNG-generated numbers, removing any one number from the sequence makes any guess of the missing number equally likely. In other words, learning a value in a given position of the TRNG generated sequence provides no information on the values in the previous or in the following positions. A PRNG is an algorithm that starts with an initial value — *seed* — and uses some function(s) to produce a sequence of values that appear random. Often, one-way functions (functions that are computationally infeasible to invert) are used as PRNGs. Thus, a typical PRNG provides forward security.

5.3 Randomized Encryption

Randomized encryption uses a nondeterministic, probabilistic algorithm to encrypt data. Informally, given two cipher texts encrypted under the same key, it is infeasible to determine whether the corresponding plaintexts are the same. A form of randomness is always involved in the randomized encryption process. In the context of symmetric encryption, different keys can be used to achieve randomized encryption. In the context of public-key encryption, because the public key used to

encrypt data usually remains constant, a random value is required as one of the inputs (along with plaintext) to the encryption function.

5.4 Potential Defense Strategies

Some potential defense strategies against the aforementioned μADV types. As can be expected, encryption and authentication, as well as data dissemination can mitigate attacks to some extent. There is an important difference between a curious μadv and the other two types (search-and-erase and search-and replace) that is related to data location. With a curious μADV, moving data around the network (to keep it away from compromised nodes) does not help. Because a μADV is mobile, it eventually *catches up* with all data, whether the data is moved around or not. Thus, to protect against a curious μADV, nodes might not be required to communicate among themselves, except perhaps for the purposes of encryption or authentication techniques that might require cooperation.

5.5Curious μADV

Because a curious μADV is interested only in learning actual sensed data, encryption seems like an effective defense strategy; without knowing a sensor's encryption key, the μADV cannot learn the data. Any secure encryption scheme can be used, whether symmetric or public key.

6. CONCLUSION

In this paper, we presented our vision of an emerging type of unattended WSN and a model of an adversary more powerful than that traditionally encountered in other network (and sensor- network) settings. We argue that the unattended nature of the environment lends itself to a nimble μADV that aims to gradually undermine the security of the entire network. Our initial analysis shows that current techniques do not offer sufficient protection against a μADV in UWSNs. The limited resources of individual sensors combined with the power of the μADV make the problem very challenging. By investigating the problem further, we identified two properties — forward security and backward security — crucial for mitigating μADV threats. However, current encryption and authentication techniques that provide these properties are unsuitable for the UWSN model due to sensor computational limitations, as well as lack of trusted hardware components and/or online trusted parties. One conclusion is that to defend against the envisaged adversary, sensors must collaborate to:

• Help each other recover from compromise (to regain a secure state)

• Hide the origin, contents, and locations of sensed data until

The next visit of the itinerant sink new ideas and techniques are clearly required to address the challenge posed by the UWSN μADV model. We hope that this exploratory article offers a new research direction and stimulates some discussion within the research community.

REFERENCES

[1] R. Ostrovsky and M. Yung, "How to Withstand Mobile Virus Attacks," *ACMPODC '91*, 1991.

[2] Y. Frankel *et al*., "Proactive RSA," *Crypto'97*, 1997.

[3] T. Rabin. "A Simplified Approach to Threshold and Proactive RSA," *Crypto'98*, 1998.

[4] J. Deng *et al*., "A Practical Study of Transitory Master Key Establishment forWireless Sensor Networks," *IEEE SecureComm '05*, 2005.

[5] M. Bellare and S. Miner, "A Forward Secure Digital Signature Scheme,"*Crypto '99*, Aug. 1999.

[6] M. Bellare and B. Yee, "Forward Integrity for Secure Audit Logs," UCSD CSEDept. tech. rep. 1997 no. 23, 1997.

[7] G. Itkis and L. Reyzin, "Forward-Secure Signatures with Optimal Signing andVerifying," *CRYPTO '01*, 2001.

[8] Y. Dodis *et al*., "A Generic Construction for Intrusion-Resilient Public KeyEncryption," *CT-RSA '04*, Feb. 2004.

[9] Y. Dodis *et al*., "Key-Insulated Public Key Cryptosystems," *Eurocrypt '02*,May 2002.

[10] Y. Dodis *et al*., "Strong Key-Insulated Signature Schemes," *PKC'03*, 2003.

[11] G. Itkis, "Intrusion-Resilient Signatures: Generic Constructions," *SCN '02*,2002.

[12] G. Itkis and L. Reyzin, "SiBIR: Signer-Base Intrusion-Resilient Signatures,"*CRYPTO '02*, 2002.

[13] V. Shoup, "OAEP Reconsidered," *Crypto '01*, 2001.

[14] R. Di Pietro *et al*., "POSH: Proactive Cooperative Self-Healing in UnattendedWireless Sensor Networks," *IEEE SRDS '08*, 2008.

[15] D. Ma and G. Tsudik, "DISH: Distributed Self-Healing in Unattended SensorNetworks," *SSS '08*, 2008.

[16] R. Di Pietro *et al*., "Catch Me (If You Can): Data Survival in UnattendedSensor Networks," *IEEE PERCOM '08*, Mar. 2008.

[17] R. Di Pietro *et al*., "Collaborative Authentication in Unattended Sensor Networks,"*ACM WiSec '09*, 2009.

[18] T. Hara and S. Madria, "Data Replication for Improving Data Accessibility in AdHoc Networks," *IEEE Trans. Mobile Comp*., vol. 5, 2006, pp. 1515–32.

[19] V. Gianuzzi, "Data Replication Effectiveness in Mobile Ad Hoc Networks,"*ACM PE-WASUN '04*, 2004.
[20] D. Chakrabarti *et al*., "Information Survival Threshold in Sensor and P2PNetworks," *IEEE ICC '07*, 2007.
[21] V. Berman and B. Mukherjee, "Data Security in MANETs Using MultipathRouting and Directional Transmission," *IEEE ICC '06*, 2006.
[22] P. Papadimitratos and Z. Haas, "Secure Data Communication in Mobile AdHoc Networks," *IEEE JSAC*, vol. 24 no. 2, 2006, pp. 343–56.
[23] Z. Benenson, P. Cholewinski, and F. Freiling, "Simple Evasive Data Storagein Sensor Networks," *PDCS'05*, 2005.
[24] S. Chessa and P. Maestrini, "Dependable and Secure Data Storage andRetrieval in Mobile, Wireless Networks," *DSN '03*, 2003.
[25] A. Kamra *et al*., "Growth Codes: Maximizing Sensor Network Data Persistence,"*ACM SIGCOMM '06*, 2006.
[26] N. Subramanian, C. Yang, and W. Zhang, "Securing Distributed Data Storageand Retrieval in Sensor Networks," *Pervasive Mobile Comp*., vol. 3, no.6, 2007.