Ad Hoc Networks 7 (2009) 1434–1447

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Rethinking security properties, threat models, and the design space in sensor networks: A case study in SCADA systems

Alvaro A. Cardenas *, Tanya Roosta, Shankar Sastry

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, United States

ARTICLE INFO

Article history:
Available online 3 May 2009

Keywords:
Sensor networks
Survey
Taxonomy
Threat models
Security requirements

ABSTRACT

In recent years we have witnessed the emergence and establishment of research in sensor network security. The majority of the literature has focused on discovering numerous vulnerabilities and attacks against sensor networks, along with suggestions for corresponding countermeasures. However, there has been little guidance for understanding the holistic nature of sensor network security for practical deployments. In this paper, we discuss these concerns and propose a taxonomy composed of the security properties of the sensor network, the threat model, and the security design space. In particular, we try to understand the application-layer goals of a sensor network, and provide a guide to research challenges that need to be addressed in order to prioritize our defenses against threats to application-layer goals.

1. Introduction

A quick look at the research literature on sensor networks does not offer a hopeful view about their security. There appears to be innumerable threats to sensor networks, such as, replication (cloning) attack, Sybil attack, communication replay, wormhole attack, time synchronization attack, localization attack, routing attack, jamming, rushing of messages, aggregation attack, false sensor data injection, reputation attack, and many others.

Contributing to this grim outlook, sensor networks are generally presented as systems with very limited resources. Typical arguments include: (1) the hardware and energy constraints of sensor nodes severely limit their ability to implement traditional security solutions, (2) sensor nodes are left unattended and are therefore easily compromised, (3) there is no trusted infrastructure; therefore, distributed protocols must be resilient to Byzantine attackers, and (4) without an online trusted third party, it is difficult to bootstrap security associations.

As a result, if we implement a security countermeasure for each of the proposed attacks, the security overhead will overwhelm the (already scarce) available resources of the sensor network. In short, attempting to create a *secure* sensor network appears to be an impossible task.

This is not a problem unique to sensor networks, since obtaining perfect security is impossible. The problem, however, is that deployments of sensor networks have been used chiefly for either: (1) scientific purposes, where an adversary has little incentive to attack the sensors, or (2) military deployments, where very little public data is available: as a result, most of the academic research for the security of sensor networks has been done in abstract scenarios, where any assumption is valid; such as, the type of threats the sensor network is exposed, and the architecture and resource constraints of the sensor network.

However, recently sensor networks have found their way into real commercial applications. This offers us the opportunity to use concrete practical scenarios and avoid making assumptions about abstract deployments.

In this paper we begin to address these problems and we identify some key research challenges:

- Providing the background setting for the security of sensor networks in Supervisory Control and Data

* Corresponding author. Tel.: +1 510 642 8290.
E-mail addresses: cardenas@eecs.berkeley.edu (A.A. Cardenas), roosta@eecs.berkeley.edu (T. Roosta), sastry@eecs.berkeley.edu (S. Sastry).

1570-8705/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2009.04.012

Acquisition (SCADA) systems. Identifying (1) the common architecture and resource constraints of the sensor networks, and (2) the incentives and methods an attacker can follow.
- Providing a *holistic* view of the *security requirements* and *threat models* of the sensor networks. We express our holistic view with two considerations: (1) we focus on *high-level* security goals (we argue that previous research has focused on *low-level* security goals), and (2) we introduce a class of physical attacks. (previous research has focused mostly on cyber-attacks).
- Providing a ranking of threats and security mechanisms. While our rankings may not be general enough, we believe our taxonomy is an important first step to better understand the threats against a sensor network and to understand our priorities for protecting them.
- Defining the *high-level* security goals of a sensor network. While terms like *availability* and *integrity* tend to be understood informally, we provide a new interpretation of these properties in sensor networks.
- Identifying different ways that sensor measurements are reported back to the base station: Event-based sensor measurements can compromise confidentiality of the network even when we use standard encryption algorithms.

The rest of the paper is organized as follows: In Section 2 we discuss the use of sensor networks in SCADA systems and emphasize the importance of securing sensor networks. Section 3 outlines the *security properties* of the sensor network as seen from the point of view of a network user. Our goal is to analyze global requirements, such as, *confidentiality*, *availability*, *integrity*, and *privacy* of the network, instead of focusing only on the requirements for secure middleware (e.g., secure routing) as previous research has done. Section 4 describes the *threat model*. The goal is to provide a general framework to analyze the threat models against the global security requirements by determining the conditions necessary for an attack to succeed and its estimated consequences. This framework gives us a way to identify and evaluate the things that can go wrong in the network. In Section 5 we study the *security design space* to identify best practices for the design and configuration of secure sensor network. Our aim is to help a system designer decide how to best defend the deployed sensor network. Finally, Section 7 concludes the paper and describes challenges and future work.

2. A motivating example: Supervisory Control and Data Acquisition Systems

One of our main motivations is to understand the practical impact of security as sensor networks start transitioning from idealized concepts to concrete practical applications. In this section we present one example of a commercial application of sensor networks.

Supervisory Control and Data Acquisition Systems (SCADA) refers to large scale, distributed measurement (and control) networks. They are used to monitor or to control chemical or transport processes, municipal water supply systems, electric power generation, transmission and distribution, gas and oil pipelines, and other distributed processes.

A major drawback of typical SCADA systems is the cost of wiring devices to a network. Wireless sensor networking is a promising technology that can improve considerably the sensing capability of the SCADA system and significantly reduce the wiring costs. Motivated by these incentives, a number of companies have teamed up to bring sensor networks in the field of process control systems, and currently, there are two working groups to standardize their communications [1,2]. Fig. 1a and b show possible integrations of wireless sensor networks with SCADA systems.

While the deployment of sensors is beneficial for the operation of an industrial control system, deploying wireless devices without security considerations can be dangerous. For example, and adversary may be able to send (from a neighboring area within wireless range) spoofed packets to a controller, causing it to perform undesired effects.

The most well-known computer-based targeted attack to SCADA systems is the attack on Maroochy Shire Council's sewage control system in Queensland, Australia [3]. On January 2000, almost immediately after the wireless system for the sewage plant was installed by a contractor company, the plant experienced a series of problems. These problems continued for the next four months: pumps were not running when needed, alarms were not being reported, and there was a loss of communications between the control center and the pumping stations. These problems caused the flooding of the grounds of a nearby hotel, a park, and a river with a million liters of sewage. One of the insights in analyzing this attack, is that cyberattacks may be unusually hard to detect (compared to physical attacks). The response to this attack was very slow and the attacker managed to launch 46 reported attacks until he was caught. At the beginning, the sewage system operators thought there was a leak in the pipes. Then they observed that valves were opening without being commanded to do so, but they did not think it was an attack. It was only after months of logging that they discovered that spoofed controllers were activating the valves, and it took even more time to find the culprit: a disgruntled ex-employee of the contractor company that had installed the control system originally and who was trying to convince the water treatment company to hire him to solve the problem.

Because many SCADA systems perform vital functions in national critical infrastructures, such as electric power distribution, oil and gas refining, and water treatment and distribution, the disruption of control systems could have a significant impact on public health, safety, and lead to large economic losses. Securing control systems in critical infrastructures is thus a national priority [4,5].

When we asked several industry professionals about the security goals that SCADA networks should achieve, the majority of responders said that the main security requirements for the SCADA systems are, in the order of importance, availability, integrity and confidentiality. We explore this interpretation in the next section.

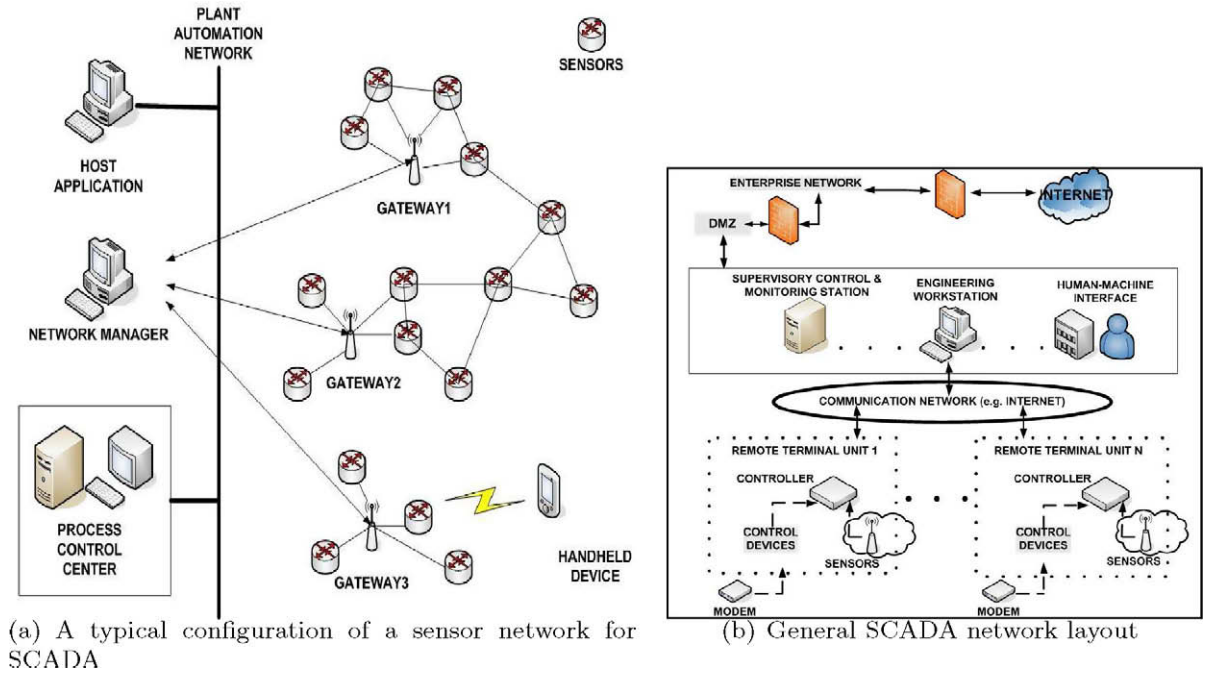

Fig. 1. (a) A typical configuration of a sensor network for SCADA; (b) general SCADA network layout.

The security of SCADA systems will be used as our baseline motivation. Although our framework can be used for the security of general sensor networks, we use the SCADA system as an example to justify some of our assumptions. This, in turn, facilitates the comparison of our assumptions and traditional assumptions made in the sensor network literature.

Remark 1. The sensor networks used in SCADA systems have a number of characteristics that are different than the generic characteristics of sensor networks:

- There is an online trusted third party (monitoring station).
- There is no aggregation: the controller collects all the data coming in from the sensors.
- The battery life of the sensors is expected to last several years; therefore, the energy efficiency of the protocols is not as critical as in other applications of sensor networks.
- Although there are some implementations of multi-hop routing, the majority of current deployments use a single hop between the sensors and the gateway.
- Sensors must be accessible and configurable by handheld devices used by network operators.

3. Security requirements

We classify the goals of a sensor network into two classes: (1) gathering information from a set of sensors in different locations, and (2) preventing the use of the resources of a sensor network by an unauthorized party.

Availability and *integrity* represent the goals of using a sensor network: availability refers to the ability to collect data and integrity refers to our confidence that the data collected is correct.

Confidentiality and *privacy* represent the protection against the possible side-effects of using a sensor network: the fact that an adversary can use the data collected by the network to obtain information that should be secret. An example of this breach of confidentiality due to side information is traffic analysis.

Motivated by RFC 2828 [6], we now define the security goals in a sensor network more precisely.

3.1. Service integrity

Definition 1. We define *service integrity* as the trustworthiness of the information provided by the sensor network. The quality of the received information allows the sensor network to perform its intended function, which is the collection of accurate data from the sensors (this is the main *application-layer* service). A violation of service integrity results in *deception*: a circumstance where an authorized entity receives false information about the phenomenon being monitored, and it believes it to be true.

Service integrity depends on the following two definitions:

Definition 2. *Message integrity:* prevents unauthorized modification of the data sent from the sensor node (message *tampering*). Message integrity tries to prevent digital tampering of the messages sent by the node. Besides preventing tampering, message integrity should also provide: (1) *Data liveness:* to prevent *replay* attacks by

giving the recipient of the data an indication of *when* the sensor reading was measured, and (2) *Source authentication:* to prevent *spoofing* attacks by providing the recipient of the data with the evidence of the identity of the message's source.

Remark 2. Data liveness is usually divided into *weak liveness* and *strong liveness*. Weak liveness means that the receiver knows the *time ordering* of the sensor readings, but not the exact time of the measurement. Weak liveness is usually achieved by adding a *message counter* that is incremented when each message is transmitted between a sender and a receiver. Strong liveness on the other hand, provides an approximation of the time when a sensor measurement was recorded. Strong liveness can be achieved by polling a sensor with a *nonce*, or by adding a *time stamp* (although care must be taken against time synchronization attacks [7,8]).

Remark 3. In the computer security literature the term *integrity* is regularly used to refer to *message integrity*. This limited notion, however, is not sufficient to analyze the integrity of the operational goal of the sensor network. The interactions of the sensors with the physical world and the fact that the information sent by the sensors depends on their location motivates us to define a new notion of integrity.

Definition 3. *Measurement integrity:* prevents the modification of the sensor measurements. An attack against measurement integrity succeeds when sensor nodes report data that is not representative of their intended environment.

Remark 4. False readings can be sent by: (1) a *malicious insider*: a compromised node sending incorrect data, (2) an *environment attack*: an attack affecting the environment around the sensor by the adversary—e.g., placing a magnet on top of a magnetometer—, or (3) a *false position attack*: changing the location of the sensor node by the attacker, and the sensor node is unable to detect this change and report it.

3.2. Network availability

Definition 4. *Network availability*: the information collected by the sensors is accessible and useable upon demand by a legitimate user. A violation of network availability results in *denial of service*: the prevention of authorized access to the sensor measurements.

To understand why industry practitioners consider that the availability of a sensor network is more important than its integrity, we consider a new definition.

Definition 5. *Service availability* is the union of *network availability* and *service integrity*—that is, *service availability* ensures that the measurements are: (1) received and (2) correct.

While *service availability* should be the top priority of a sensor network, we believe that *service integrity* is more important than *network availability*. In our definition, network availability just assumes we receive data, but the data may be incorrect and lead us to take incorrect actions. On the other hand, if there is no network availability we can notice the attack and take corrective actions.

Remark 5. Most of the literature in sensor networks uses the term *availability* to refer to *network availability*.

3.3. Confidentiality and privacy

Definition 6. *Confidentiality*: the information collected by sensor networks is only accessible to legitimate users. It is intended to prevent unauthorized users from learning the information collected by the sensors. A violation of confidentiality results in *disclosure*.

A typical way to ensure confidentiality is via encryption. However, due to the low-entropy of certain sensor measurement reports, confidentiality may be compromised by simple traffic analysis.

Definition 7. *Privacy* is the prevention of unauthorized users from learning sensitive *personal* information by using the sensor network. Privacy can be considered a special case of confidentiality when the data collected is personal (e.g., via surveillance camera networks, or sensors monitoring the vital signs of patients).

3.4. Access control

Definition 8. *Access Control*: the prevention of unauthorized access to the network. It prevents outsiders (unauthenticated principals) from gaining access to the network, while imposing and enforcing proper restrictions on what insiders (authenticated principals) can do. A security violation of access controls results in *usurpation*: the use of system services or functions by an unauthorized entity.

Confidentiality, privacy, and integrity depend to a large extent on enforcing *access control*. Access control is, however, more general: its goal is to protect *all* the resources of the network, including the misuse of the communication infrastructure. Consider for example the deployment of two sensor networks *A* and *B* within the same wireless range. Access control in network *A* would prevent the user of network *B* from using network *A* to route its packets. In this sense, access control will prevent *free riders* (a notion not captured by our definitions of service integrity, network availability, and confidentiality).

Although we consider *privacy* and *access control* implicitly, the main focus of the paper is to study service integrity, network availability and confidentiality.

3.5. High-level versus low-level security goals

All the security properties we have defined represent the general security requirements of the end-user of a network. We consider these requirements as *high-level* security goals: the goals of using a sensor network in the first place, without looking at the supporting services.

Other security goals, such as, *secure routing*, *secure key distribution*, *secure time-synchronization* and *secure neighbor discovery*, can be considered as *low-level* security goals. We argue that most of the research for the security of sensor networks has focused on the design of low-level *mechanisms*, such as, secure routing protocols, to achieve these low-level security goals. However, there has been very little research effort in trying to understand how these low-level mechanisms relate to the high-level security goals.

In this paper, we point out the need to understand this relationship by considering the ranking of the most essential security mechanisms for achieving the high-level security goals.

For example, instead of designing a new secure routing algorithm (a mechanism intended to provide network availability), we should ask how much do we gain in *network availability* by using a given routing protocol. Notice, however, that an attacker with the ability to jam the network will achieve a denial-of-service attack no matter which routing protocol is used. Therefore, to answer the question of how to select the appropriate security mechanisms requires a *threat model*.

4. Threat model

It is impossible to achieve perfect security. Not only will an all powerful adversary defeat any security mechanism, but defending against, and responding to every possible attack vector is prohibitively costly. Therefore, equally important to defining security (defining the security requirements) is *defining what we are secure against* (defining the *threat model*).

The goal of defining a threat model is to formalize our perceived *risk*. Risk is defined as the estimation of two quantities: (1) the likelihood of an attack, and (2) the consequences of the attack. The threat model should describe the capabilities of an adversary and identify the threats and attacks against the intended security requirements.

The types of threats in sensor networks may be different than the threats against traditional computer networks. For example, the Internet infrastructure is relatively well protected. Key computers (e.g., DNS roots) are kept in physically secure areas, and there is a level of redundancy and diversity that allows the infrastructure to survive several attacks. In contrast, in sensor networks it is generally assumed that the *infrastructure* is composed by the sensor nodes themselves, and they are assumed to be less protected than traditional infrastructure servers.

This perceived vulnerability of sensor nodes has been explored extensively in the research literature, leading to the suggestion of a wide range of attacks. We argue that researchers need to study the question of *which are the most likely attacks that an adversary will follow to compromise high-level security properties*.

4.1. Threat taxonomy

We categorize the attacks into three main types:

- *Outsider attacks*: this type of attack does not require any knowledge of secret keys being used by the network.
- *Key-compromise attacks*: these attacks help the attacker change type, i.e., go from an outside attacker to an inside attacker, by compromising the secret keys used in the network.
- *Insider attacks*: during these attacks the adversary can act as some legitimate nodes in the network. The legitimate identities the adversary can assume will depend on the secret communication keys it has captured.

In addition, each attack can be put into one of the two classes: (1) *final* attack, and (2) *intermediate* attack. Final attacks are the ones in which the attacker accomplishes its final goal: compromising one or more of our *high-level* security requirements. Intermediate attacks, on the other hand, are attacks which facilitate carrying out other attacks. They are used by the adversary as a stepping stone to accomplish its final goal and increase its capabilities.

In order to assess the damage caused by each attack and identify the path of least resistance for the attacker, we categorize each known attack in sensor networks into one of the three types. Then, we use a threat ranking scheme to score each attack based on the difficulty of being performed. Finally, we rank the impact of each attack on the main security requirements, i.e. integrity, availability, and confidentiality.

4.2. Outsider attacks

Some of the typical attacks considered in the literature include the following:

Spoofing attack: In this attack, a system entity illegitimately assumes the identity of an authorized system entity. If sensor nodes are not authenticated properly, this attack is very easy to launch. The lack of proper device authentication was the reason the attack on the sewage system at Maroochy Shire [3] was successful. *Attack class*: final; if an attacker can spoof a legitimate node in the network, then it can send arbitrary values on its behalf and compromise our system integrity.

Jamming attack: Jamming is the interference with the Radio Frequency (RF) used by the nodes in a network. It makes use of the broadcast nature of the communication medium. *Attack class*: final; it can affect the availability of some parts of the network.

Replay attack: In a reply attack, a transmitted packet is maliciously or fraudulently repeated or delayed by the adversary. *Attack class*: intermediate.

Wormhole attack: In this attack the adversary tunnels network messages to another part of the network through a low latency link. The attacker can use laptops or other wireless devices to send the packets on a low latency channel. Because we are assuming that the attacker has no secret keys, its impact depends on the final attack that is carried out by using wormhole. However, this attack could potentially have a higher impact if the adversary is able to infer/distinguish the types of the packets in transmission. By knowing the type of the packets, such as, data, acknowledgement, time update, or advertisement, the adversary can tunnel the 'control' packets and cause more damage to the underlying protocols. It may be possible for an attacker to infer the type of packet in transmission by gath-

ering information as a consequence of generation, transmission, and routing of data messages within the network via traffic analysis. An attacker can use the message generation rate, message size, and other peripheral information available to him through the broadcast medium to make the inference. *Attack class*: Intermediate.

Destroying a node: If the sensor network lacks physical access security, it is relatively easy for an attacker to walk up to a node and destroy it. This is an effective attack against availability. The main drawback of this attack from the adversary's point of view is the risk of apprehension. *Attack class*: final; the sensor node will not be available.

Environment tampering: The adversary in principle can compromise the integrity of the sensor readings by tampering with the deployment area. For example, he can place a magnet on top of a magnetometer, or temper with the temperature of the environment around temperature sensors. This is an effective attack against service integrity. The main drawback of this attack is the high risk of apprehension if the network is under some kind of surveillance. *Attack class*: final; we would receive incorrect data.

Node displacement: The attacker can change the location of the sensor nodes. By placing the sensor in an incorrect location, the measurements it is going to report to the base station will be erroneous. Therefore, this is an effective attack against service integrity. *Attack class*: final; if the node displacement is not detected we might interpret incorrectly the received sensor measurements.

Install new sensors: Again, if the area where the sensor nodes are deployed is left unattended, the adversary may be able to install its own sensors and monitor the physical event that we monitor. This is an effective attack against confidentiality. *Attack class:* final; notice that the goal of this attack is not to interact with the network, but just to monitor the physical system by other means.

4.3. Key-compromise attacks

In this section, we discuss the attacks which enable an adversary go from an outside attacker to an inside attacker. These attacks are intermediate attacks for availability and service integrity, and final attacks for confidentiality.

Cryptanalysis: This attack refers to transforming encrypted data into plaintext without having prior knowledge of the encryption parameters or processes. In the worst case, the attacker will obtain the secret keys of a set of devices, and would be able to impersonate them.

Exploit: An exploit takes advantage of a software vulnerability to compromise a system.

Remark 6. It can be argued that the number of vulnerabilities in sensor networks should be smaller than computers typically connected to the Internet because sensors provide less services. With less functionality and less complex code there will be less software bugs. Additionally, because of the resource constraints in sensor networks, programmers have to spend more time per line of code in sensor network applications, than in applications for regular computer networks.

Physical tampering: If an attacker has the necessary technical skills and equipment, he could physically compromise the sensor nodes and obtain the data and other keying material. Additionally, an attacker can succeed in performing a side-channel attack to analyze the physical activities of the system to extract the cryptographic keys.

4.4. Insider attacks

In this section we describe the insider attacks, which require having access to a subset of the secret keys used in the network.

Sybil: Sybil attack refers to the scenario where a malicious node pretends to have multiple identities. For example, the malicious node can claim false identities (*fabricated identities*), or impersonate other legitimate nodes in the network (*stolen identities*) [9,10]. *Attack class*: intermediate.

Replication: In this attack, the adversary attempts to add one or more nodes to the network that use the same ID as another node in the network [11]. *Attack class*: intermediate.

Denial of service at the link layer (or MAC layer): Examples of attack on the link-layer protocol are: 1) causing collision with packets in transmission, 2) exhaustion of the node's battery due to repeated retransmission, 3) unfairness in using the wireless channel among neighboring nodes [12]. *Attack class*: final: availability.

Routing attacks: In these type of attacks, an attacker tries to create routing loops or advertise false routes. The final objective is to degrade the availability of the system, or to receive more traffic for cryptanalysis [13]. *Attack class*: final: availability.

Time-Synchronization attack: Time-synchronization protocols provide a mechanism for synchronizing the local clocks of the nodes in a sensor network. As a result, when there is an attack on these protocols, a fraction of the nodes in the entire network will be out-of-sync with each other [8]. This in turn affects the sensor network applications that rely on tight synchronization to perform correctly, such as, TDMA-based protocols or object tracking [7]. *Attack class*: final; while time-synchronization problems may cause some DoS attacks, the message integrity of the nodes could be compromised if we use timestamps for an indication of liveness.

Slander attack: This attack is only possible if a distributed detection system is implemented, and the sensor nodes can accuse each other of misbehavior. Slander attacks are very dangerous to distributed node revocation techniques [14]. *Attack class*: depends on the system. It will not affect our high-level security properties unless we use a reputation scheme.

Wormhole: There are two types of wormholes that an insider can do: (1) an in-band wormhole attack (using the sensor network to tunnel packets), and (2) an out-of-band wormhole attack (using a low latency external communication link). An insider can use a wormhole more effectively than an outsider since it can certainly identify different types of packets. A distributed mechanism for detecting wormhole attack in sensor networks is given in [11]. *Attack class*: intermediate.

4.5. Security metrics

Once we have a threat model and a ranking, it is important to develop a function that captures the overall effect (or cost) of various attacks on our high-level security requirements, i.e., service integrity, network availability, and confidentiality.

There can be many definitions of costs for each attack. In this section we provide an example of some possible metrics: consider the following functions,

$$\begin{aligned} a_1(x,x') &= \alpha \times \sum_i \int \|x_i(t) - x'_i(t)\|^2 dt, \\ a_2(n_2,t) &= \beta \times n_2 \times T, \\ a_3(n_3,s) &= \gamma \times n_3 \times s \end{aligned} \tag{1}$$

where

- $a_1(.)$ measures the amount of compromise in integrity. $x_i(t)$ is the true value of the physical process we are monitoring by node i at time t, and $x'_i(t)$ is the value the attacker manages to send to the base station.
- $a_2(.)$ measures the amount of compromise in availability, which is a function of how many packets per second are intercepted and how long the attack continues T.
- $a_3(.)$ measures the amount of compromise in confidentiality. The arguments to the function are the number of nodes compromised (n_3) and the sensitivity of the data (s) which is application-dependent.

Furthermore (α, β, γ) is a vector composed of weights of the security objectives. For example, in our SCADA system scenario, we are more interested on providing integrity and availability.

For example, jamming is an attack on availability; therefore, for this attack we can measure $\beta \times n_2 \times T$. The number of nodes affected by the jamming attack are proportional to the area covered by the jamming device, which depends on how powerful the radio of the device is. We assume that the jamming radius is R, and the sensor deployment density is $\rho = \frac{ofnodes}{deploymentarea}$. Using these values, the number of nodes affected by the jamming device are $n_2 = \rho \times \pi R^2$. If the jamming is continued for t seconds, and the average number of packets in transmission in one unit of time are p, the total packets lost are $p \times t$. Therefore, $a_2 = \beta\rho\pi R^2 pt$.

Another example is when the attacker tries to compromise the measurement integrity by sending in accurate sensor measurements. Assume that the attacker has compromised one node and that measurements are taken periodically at times $t = k\Delta t$ for $k = 0, 1, \ldots$, where Δt is a time interval. In this scenario, $a_1(x,x') = \alpha \times \sum_k \|x(k\Delta t) - x'(k\Delta t)\|^2$. In our SCADA system example, the sensor reading x might correspond to the fluid level in a tank. In this case, the attacker can modify the readings to be $x' = x + \xi$ where ξ can be a positive or a negative value. There is no restriction on how the attacker chooses ξ. However, if he wants to affect on the decision making process of the system, he has to choose ξ intelligently so that the resulting x' is a valid fluid level, although it is not the *correct* fluid level.

4.6. Ranking

In this section, we present a ranking of various attacks in terms of their effect on availability and measurement integrity. The ranking is based on the difficulty of performing the attack versus the effect it has on the corresponding security objective. The effect of each attack is captured by the attacker's objective function defined in Eq. (1).

These rankings are based on the assumption that there are no security mechanisms in place. Therefore spoofing for example, is a very devastating attack, because an attacker can impersonate any node and send arbitrary data. We also assume there is no physical security, so launching one of our outside-physical attacks is assumed to be very easy for an attacker.

Having no security mechanisms in place for this analysis will give us a base line for attack rankings which can be used to decide what security mechanisms are most effective in preventing each attack.

In order to better quantify the feasibility of each attack, we define a threat ranking which gives an overall score of the *difficulty of accomplishing the attack*.

We assume that outsider attacks are easier to perform than insider attacks. For general insider attacks we only assume that the attacker has control of one node (but does not spoof any other node). For Sybil attacks and Wormhole attacks we assume the adversary has at least two identities (to launch a successful attack). We assume that having at least two identities is more difficult to obtain than a single identity.

We furthermore consider the following:

- Cost of extra hardware
  - [–] Attack requires more than commodity hardware.
  - [–] No extra hardware required, i.e. the attack can be accomplished using PC, laptops and sensor motes.
- Physical access: refers to physical proximity to the sensors and being able to touch them.
- Required technical skill
  - [–] Brute-force attacks: (e.g., destroying a node, or jamming a network).
  - [–] Logical attacks: these attacks require the knowledge of the specific protocol being used by the network.
    - [*] Automated attacks: the attack scripts can be acquired easily, such as, through the Internet.
    - [*] Non-automated attacks: The attacker needs to invest more amount of time and resources for these attacks.

Figs. 2 and 3, show a possible ranking of the service-integrity attacks. We have created a partial order because we believe some attacks are not comparable. An environment tampering attack may be very easy to perform in some cases, but if a sensor is installed in a pipe, or a water tank, the attack may be more difficult to launch than just compromising a node (key-compromise attack) to send fake data with it.

To have a better intuition of which attacks should be our first priorities, we have used an arbitrary representation of the partial order into a total order (the axes) in Fig. 4.

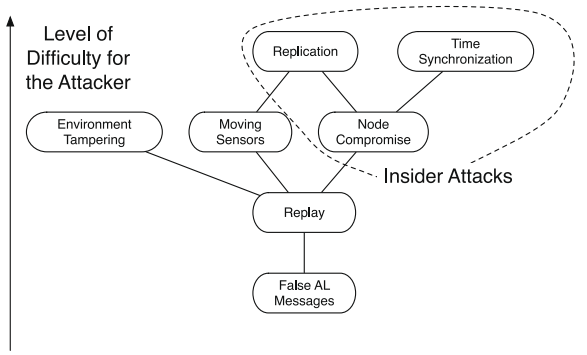

Fig. 2. This figure shows a partial order for the difficulty of performing each attack. These attacks are concerned with service integrity. False AL refers to false application-layer packets, and assumes a successful message insertion, message tampering, or spoofing attack.

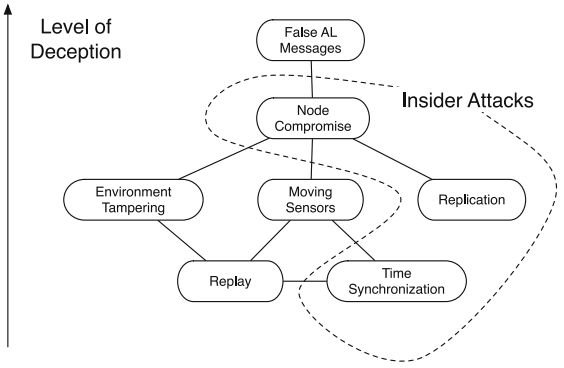

Fig. 3. Level of deception on service integrity, when each attack in this figure is carried out.

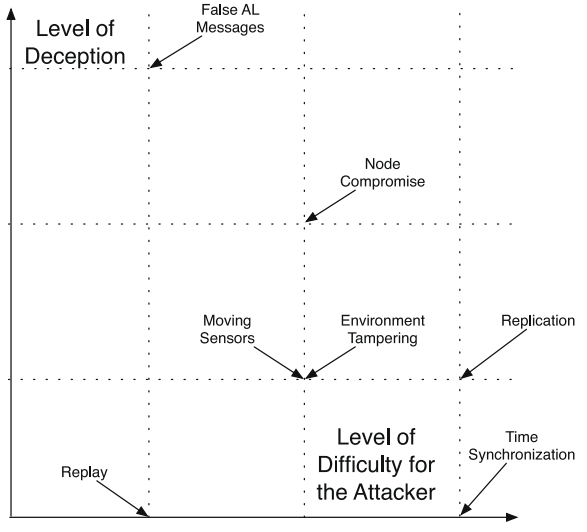

Fig. 4. A 2-D graph of the difficulty of each attack versus its consequence on the data integrity. We can see that the false-application later message attack is relatively easy to carry out and has a high impact on the integrity.

A similar analysis can be performed for availability in Figs. 5–7.

Some basic remarks from our analysis are:

Remark 7. We believe that the most basic security mechanisms that should be in place are mechanisms for application-layer *message integrity* (to avoid the most basic integrity attacks) and link-layer *message integrity* (to avoid the most basic DoS attacks).

Remark 8. Most of the literature on the security of sensor networks has focused on jamming, node displacement,

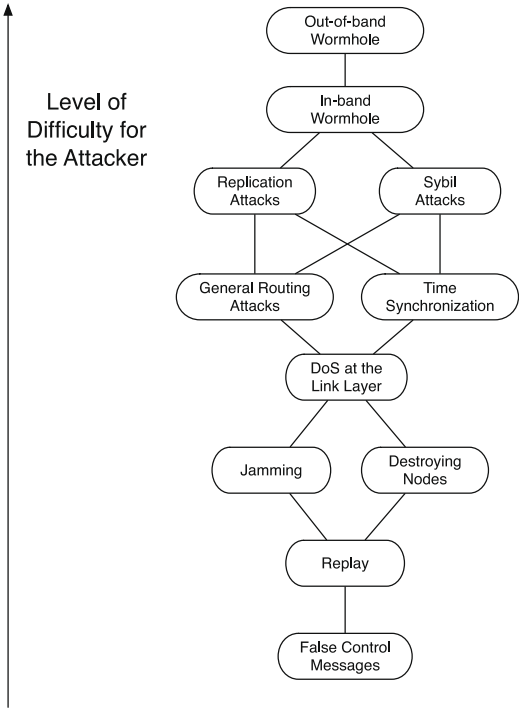

Fig. 5. Partial order for the difficulty of performing DoS attacks. DoS attacks impact the availability of the network. False control messages refer to attacks where the adversary can fake, spoof or tamper.

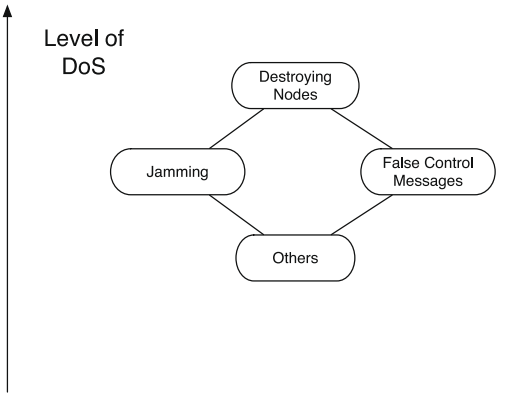

Fig. 6. The consequences of carrying out DoS attacks on the network availability.

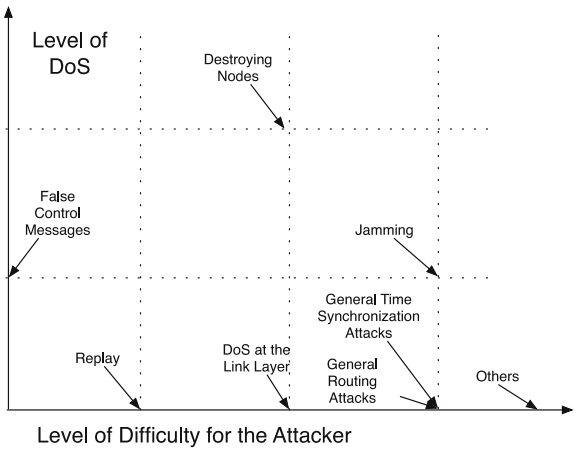

Fig. 7. A 2-D graph that illustrates the difficulty of carrying out each attack and its corresponding impact on the availability of the network. The figure indicates that if there is no physical security, there is no point in adding, for example, a secure routing protocol, because the adversary will destroy the nodes.

spoofing, replay, and wormholes. There has been very little analysis of physical attacks; such as destroying a node, environment tampering, and installing new sensors. While there is very few algorithmic solutions to these physical attacks, they need to be considered in a holistic security analysis. Without protection against these physical attacks, there is no point in trying to design secure systems against more complex attacks–such as wormhole attacks.

We are aware that our analysis may not be a definitive solution to the problem of threat modeling in sensor networks. Our goal with this analysis was to point out some future research directions and some possible ways on how to start addressing in a more systematic way the most likely attacks that an adversary may perform against a sensor networks, and how to allocate our resources effectively to combat these attacks.

5. Assumptions and design space

The majority of research in the security of sensor networks has focused on implementing security mechanisms for devices with severe resource constraints and no online trusted third party. While this scenario covers a large class of practical sensor networks, it is important to realize that these are not the only sensor networks available. Sensor networks have been used for a wide variety of applications and systems with vastly varying requirements and characteristics. In a recent study [15], the authors show the diversity of sensor networks in terms of deployment, cost, size, resources, energy, heterogeneity of the sensor nodes, infrastructure, size, lifetime, and other considerations.

We favor considering a more diverse design space because, (1) there is no global definition of what a sensor network is: each sensor network deployment will have its own goals, requirements, and constraints. (2) Our traditional assumptions about sensor networks might change radically as sensor networks start being deployed in practical commercial applications. (3) The security advantages that we obtain by investing in a more expensive infrastructure might outweigh its costs. (4) Attempting to add too many security countermeasures to a resource constraint sensor network will overwhelm the network.

We now discuss the advantages and disadvantages of some security mechanisms. The appropriate security mechanisms have to be chosen based on the importance of each security requirement and the ease of launching various attacks to compromise each security objective. This in turn means that we would like to secure the path of least resistance in our network first. This path can be found by considering the figures in Section 4.6. Each figure illustrates the ease/difficulty of launching each attack versus the impact it has on the network functionality. Therefore, to secure a sensor network, we should first start by using mechanisms that prevent attacks with less difficulty but more impact, such as, replay attack or spoofing.

The basic security mechanisms can be divided into: *prevention, detection and survivability*. *Prevention* mechanisms typically rely on cryptographic algorithms implementing authentication, access control. These mechanisms prevent the attacker from participating in the communication and compromising integrity. *Detection* relies on placing additional sensors for detecting unusual activities. We believe detection is most useful if the sensor node that is compromised is able to report this action. If the sensor node cannot report the compromise, distributed detection mechanisms have been proposed, such as, watchdog monitoring, or distributed revocation. These distributed protocols open a new class of problems, such as, *slander* attacks where compromised nodes can accuse good nodes of malicious behavior. *Survivability* is the ability of the network to remain in operation despite attacks on the system.

Before delving into details of these mechanisms, we discuss key management, which is the essential building block for most solutions.

5.1. Key management

There are three types of key agreement schemes: schemes with an online trusted server, public-key schemes, and key pre-distribution schemes. We now list the advantages and disadvantages of each scheme,

Key pre-distribution schemes [16–19]:

- Advantages: Key pre-distribution is based solely on symmetric cryptography. Furthermore, it does not rely on an online trusted third party (the trusted party is only used for the offline pre-distribution of keys).
- Disadvantages: It is difficult to guarantee end-to-end security among nodes. Furthermore, since there are repeated keys among several nodes, an adversary that compromises a few nodes can compromise the confidentiality and message integrity of a larger part of the network, and not only of the nodes it has just compromised (in its simplest case, key predistribution will deploy a single key for the whole network, and the compromise of the key will affect communications between all nodes.)

Online trusted server schemes [20]:

- Advantages: Online trusted server schemes rely solely on symmetric cryptography: each node shares a unique secret key with an online key distribution center. It is more resilient to node compromise than key pre-distribution, since the compromise of n nodes compromises the security of only these n nodes.
- Disadvantages: The network relies on the availability and integrity of the trusted server. If the trusted server is made unavailable, the sensor network would not be able to operate securely. Furthermore, if an adversary compromises the trusted server, the security of the sensor network is compromised (notice however that the same problem would occur if the offline trusted party for key pre-distribution schemes or the certificate authority for public-key schemes is compromised).

Public-key schemes [21]:

- Advantages: Public-key schemes do not rely on an *online* trusted third party: the trusted third party can function as an *offline* certificate authority. This schemes are more resilient to node compromise when compared to key pre-distribution schemes. Furthermore, the public key of the device binds its identity to the network. This is useful for authenticated one-to-many communications, such as, broadcast. Even with some compromised nodes in the network, a one-to-many message signed by the source of the message can authenticate its origin. Protocols like μTESLA [20] can achieve this property with symmetric-key primitives, but they incur in authentication delays and require tight time-synchronization.
- Disadvantages: Public-key schemes are based on asymmetric-key algorithms, and although asymmetric cryptography is now assumed to be feasible in most sensor network infrastructures [22,23] their use would still deplete the battery of sensor nodes faster than symmetric-key operations. Asymmetric cryptography, however, is typically used only to establish symmetric session keys, thus the influence to the lifetime of the network might not be significant to the new generation of sensor nodes [24]. In particular, the use of elliptic curve cryptography (ECC) and hardware support can help improve the efficiency of asymmetric algorithms.

By having a key-management architecture in place, the nodes in the sensor network can obtain the following,

1. *Network* key shared by all authorized entities,
2. *End-to-end* keys shared only between principals communicating at the application layer of the network, and
3. *Pairwise* link-layer keys shared between neighboring nodes at the link-layer of the network.

Although key pre-distribution schemes have been studied extensively in the research literature (motivated in large part by the seminal paper of Eschenauer and Gligor [16]), the availability of a trusted server in many practical scenarios has motivated a number of standard associations for sensor networks, such as, the ZigBee alliance [25], ISA SP100 [2] and WirelessHart [1] (technologies useful for sensor networks in SCADA), to propose the use of online trusted network managers for secure networks. The use of public-key cryptography is also being supported by the standards associations and several other companies, such as, NTRU's Aerolink [26].

5.2. Confidentiality mechanisms

Encryption is the primary way of preserving the confidentiality of the packets that contain sensor measurements. However, care must be taken in the use of the encryption technology. Algorithms resilient to chosen-plaintext attacks (semantic security) are a good solution since it is infeasible for a computationally-bounded adversary to derive any significant information about a message when he is only given its ciphertext. A few points to consider when deciding on what encryption mechanism to use are,

- A single network key is not resilient to insider attacks.
- A countermeasure for insider attacks on confidentiality is to use pairwise link-layer keys. If the adversary compromises the key of one node, it can only eavesdrop on communications passing through this node.
- A stronger guarantee is to use end-to-end encryption. If end-to-end encryption is the only encryption method used, any information below the application layer is disclosed (routing information, link-layer addresses etc.). Therefore, it might be necessary to use end-to-end encryption with pairwise link-layer keys, or a single network key used at the link layer. End-to-end encryption might limit the use of distributed protocols, such as aggregation schemes [27,28]. This essentially becomes a question of the tradeoff between confidentiality and energy efficiency.

Remark 9. Decryption of the message is not the only way in which an outsider can infer the contents of the message. Monitoring of the environment can be *polled-based, periodic or event driven*. A typical event-driven monitoring application uses *alarms*. A sensor will only send an alarm report if an event is detected. Therefore, an eavesdropper can identify that an event has happened if it observes the sensor node sending a packet. *Tracking*, for example, uses a similar event-driven monitoring application.

Sensor nodes, however, typically send health reports back to the base station (reporting battery status, and other network managing information). Thus, a possible countermeasure to mitigate these attacks is to randomize the time in which sensor nodes contact the base station (provided that the eavesdropper cannot distinguish between encrypted health reports and encrypted alarms).

Finally, an adversary might be able to place its own sensors for monitoring the environment. This is not a technical attack, but it shows the importance of protecting the physical deployment area of the sensor network. A possible mechanism to detect and deter this attack is to use surveillance cameras.

5.3. Service integrity mechanisms

A typical way to provide end-to-end data integrity, data liveness, and data origin authentication is to include a message integrity code in the packets sent by each party. The integrity code should include,

- The identities of the communicating parties at the application layer for data origin authentication
- The sensor reading for data integrity
- A counter for *weak* liveness, a time-stamp for *strong* liveness, or a nonce for *strong* liveness if the data is polled and the requesting party sends the nonce in the request. For polled messages, the nonce would provide the stronger guarantee of data liveness, since the time-stamp depends on accurate time-synchronization.

Similar to confidentiality, a network key for message integrity code is easily defeated as soon as the adversary compromises a single key. Therefore, using end-to-end key guarantees that the message cannot be tampered with, even when the adversary has compromised the keys of other nodes in the network. Again, this comes at the cost of limiting the use of distributed aggregation algorithms.

The use of pairwise link-layer keys will also limit the effect of insider attacks when attacking the time-synchronization protocol. This will increase our confidence that the time-stamps can be used for data liveness.

Measurement integrity can be protected to some extent by tamper-resistant or tamper-detection hardware. This increases the effort the adversary needs to put in to compromise the sensor node. The node might also include external sensors to detect when it is being moved to another location. Finally, a way to prevent or detect 'environment attacks' (the example of placing a magnet on top of a magnetometer sensing node) is to attempt to protect the physical area of the sensor node (again, surveillance cameras can be used).

Another attempt for *detecting* and *surviving* a measurement integrity attack is to use robust statistics [29], and of particular importance to SCADA systems, the use of robust control [30]. By identifying outliers and anomalies in the messages received, the measurement-integrity attack can be limited. Robust statistics and robust control come at a cost: even if there is no attack, they might discard true anomalous information.

5.4. Network availability mechanisms

5.4.1. Jamming

The design space for jamming and its countermeasures is highly situation dependent. Military sensor networks might have a very large design space for countermeasures, such as: (1) prevention: implementation of advanced waveforms using spread spectrum techniques for low probability of detection, and low probability of intercept. (2) Survivability: dynamic frequency reallocation, and raising the transmit power. However, commercial sensor networks cannot have the same flexibility in their design space because they have to conform to several norms – for example, in the US, commercial wireless systems have to be approved by the federal communications commission (FCC). The use of frequency hopping spread spectrum might make jamming more difficult to the adversary, but a dedicated adversary can always jam these signals as well. If the adversary has physical access to the sensor network it can also destroy the nodes. One way to discourage an adversary from performing jamming is to increase the physical security defenses to the sensor network deployment field.

5.4.2. Control packets

The packets in the network can be divided into *control packets* and *application packets*. Application packets are packets whose payload contains data sent by the application layer of the network, such as sensor readings. *So far our focus has been on protecting the integrity and confidentiality of application packets.*

On the other hand, there are packets whose payload contains data used to maintain the network services. These include: routing discovery packets, routing maintenance packets, time-synchronization packets, etc. *All of these control packets are of fundamental importance for availability.*

Contrary to confidentiality and integrity, *we cannot assume end-to-end security for control packets* because the network is by definition a distributed protocol. Therefore, *we can only use a network key or a pairwise link-layer key*. The same principles of service integrity and service confidentiality apply to control packets. To provide a secure communication infrastructure, control packets need to have authentication and replay protection. Therefore, these packets need to use message integrity codes. Also, to prevent an outsider from identifying the type of control packets being transmitted, we should have these packets encrypted. Without any key, an outsider can attempt to perform jamming or physical destruction of the sensing nodes. Other outsider attacks, such as replay, spoofing, and even a wormhole will have much limited effect if the the adversary cannot identify control packets and their payloads.

The availability of the network can be compromised if the attacker is able to gain access to the keying materials. To prevent an adversary from compromising the operation of all the network by capturing a single key, we require pairwise link-layer keys among neighboring nodes. Under these circumstances an adversary becomes a Byzantine attacker who can try to disrupt the network operation by 'confusing' peer devices. Therefore, the resiliency of the network management protocols will rely on redundancy and over-provisioning of resources.

Disrupting the functionality of the routing protocols compromises the availability of the network and services running on the network. Attacks against routing protocols include attempts to create routing loops or black holes (when attacker claims to be a short distance to all destinations and then selectively forwards payload traffic). We can identify two possible countermeasures: (1) measuring the link quality based on the number of dropped packets, and (2) using a routing protocol that builds path diversity: if a message sent along one path is not acknowledged by the recipient, then the protocol should use alternative path via a different neighbor. This could also mean using multi-

path routing protocols that send the data along different paths and take advantage of the redundancy in the received data.

6. Understanding the consequences of attacks against SCADA systems

While we believe that our models can be useful to model general sensor network deployments, in this final section we show an example of the role of sensor networks in SCADA systems.

Parallel to this work, we have been studying the consequences of attacks against control systems [31]. A proper threat assessment of control systems, and in particular, the role that sensor networks play in achieving the operational goals of the control system can help us integrate the ideas we introduced in this paper with a practical application.

The industrial control system we consider is a chemical reactor plant described by Ricker [32]. The chemical plant has four chemical elements (named *A*, *B*, *C* and *D* for simplicity). The goal of the control system is to produce a single irreversible chemical reaction $A + C \rightarrow D$ (B is an inert product) at a specified given rate while maintaining the pressure inside the tank below 3000 kPa.

The chemical plant has three actuators. The first actuator—controlled by $u_1(t)$—operates a valve that controls a feed F_1 containing products $A + B + C$. The second actuator—controlled by $u_2(t)$—is a valve that controls a feed F_2 containing product A. The final actuator—controlled by $u_3(t)$—is a valve that purges the gas created by the chemical reactions. Each control signal $u(t)_i$ has a range between 0 and 100 (the percent that the valve is to be open).

The control algorithm uses three sensors (y_4, y_5, and y_7) monitoring the product flow (D), the pressure inside the tank, and the amount of product A in the purge (respectively). u_1 is a function of y_5 and y_4, u_2 is a function of y_7, and u_3 is a function of y_5.

The primary safety goal is to keep the pressure inside the tank below 3000 kPa, and the primary operational goal is to keep the operational cost low. (The operational cost is proportional to the amount of products *A* and *C* lost in the purge, and inversely proportional to the amount of product *D*.)

Network: The network is representative of many industrial control configurations where there is only a single hop between each sensor and the base station (which forwards the data to the process control room). Under these conditions it is fairly easy to maintain an online trusted server scheme for key management.

Confidentiality: Because the sensors transmit data at a fixed sample rate, an attacker cannot use network traffic information to infer the state of the system, so there is no reason to implement a randomized transmission algorithm. Encrypting the transmissions of the sensors provides enough confidentiality. One important thing to realize, however, is that each sensor transmits different information, and therefore, an attacker may be more interested in obtaining information from one sensor than from others. In our example, we believe that sensor y_4 may provide a rival company valuable information about the production rate of the data, so protecting this sensor may be

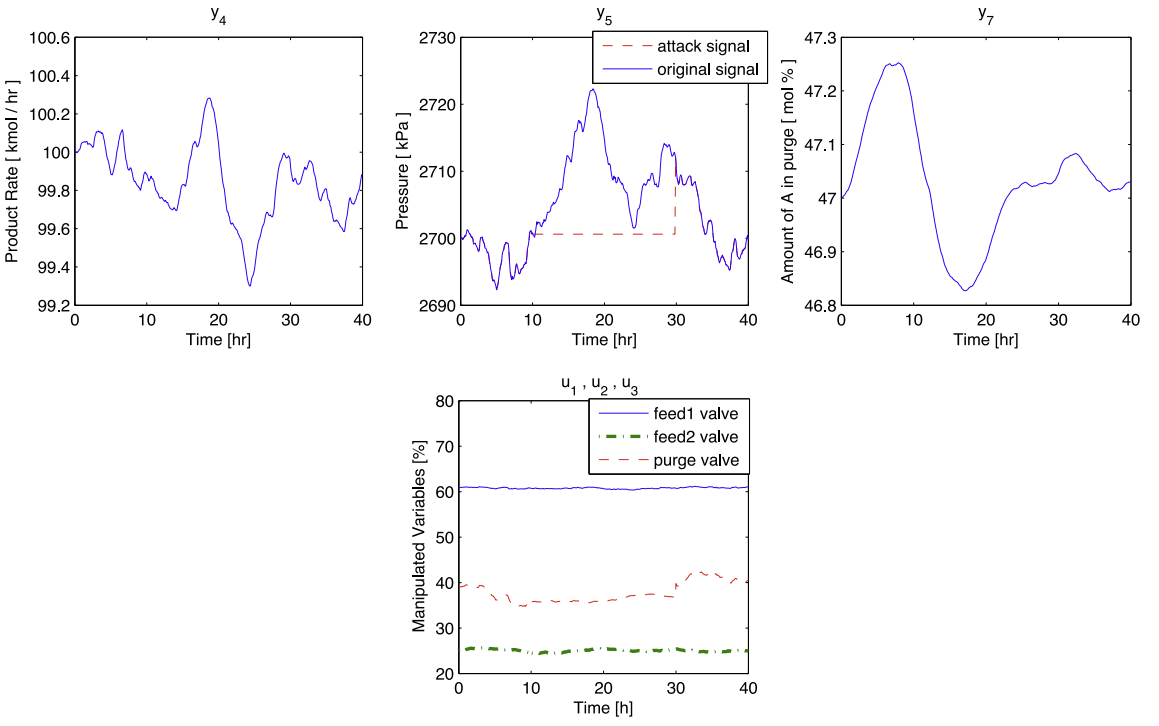

Fig. 8. Twenty hour DoS attack to sensor y_5.

a priority with respect to confidentiality (e.g., the company may decide to invest in a tamper-resilient sensor for y_4).

Availability: After performing several tests [31] we realized that DoS attacks on the sensors are not a major problem to the system. Under the assumption that a controller that does not receive information from a sensor, will continue using the last available data, we were able to maintain the plant in normal conditions (the pressure of the tank did not pass 3000 kPa and the operational cost did not change. For example, Fig. 8 shows how a DoS attack that lasts twenty hours does not affect the pressure of the tank beyond safety levels. Similar results were obtained for other tests [31].

If sensor measurement availability is not a priority to the system, we can decide that preventing complex time synchronization attacks, or even preventing jamming attacks may not be a cost-effective solution. In particular, given that the chemical reactor has very slow dynamics, it is easy to implement a detection and response practice by which the plant operator needs to respond to any loss of signal in a time frame of a couple of hours.

Integrity: By implementing basic message integrity codes we can prevent most outsider attacks, such as false application layer messages (i.e., injecting false sensor readings). Because sensors in industrial environments are tightly coupled within the pipe or tank, we believe that moving the sensors or performing an *environment tampering* attack is also very difficult.

If an attacker is an insider (i.e., it has comprised a sensor) then the situation changes. The question, however, is which sensor measurement is more valuable? We found out that to maintain the safety of the system, the attacker needs to compromise the pressure sensor and send a fake low pressure sensor reading. By compromising y_4 and y_7 an attacker cannot increase the pressure of the tank to unsafe levels [31]. Similarly, we found out that if an attacker wants to increase the operational cost of the plant, they need to compromise sensor y_4.

Assuming that safety is the first priority of the plant, we decide that the most important sensor to protect is the pressure sensor. Therefore, if we have enough resources to invest in one tamper resilient device, we should invest in a tamper resilient sensor to monitor the pressure. If we have resources to invest on two tamper resilient devices, we should protect the pressure and the product rate.

7. Conclusions

In this paper, we presented a taxonomy with the aim to provide a holistic view of the security of sensor networks. We believe this research direction will provide a better understanding of the security issues and will help the network designer decide on the most effective security mechanisms under resource constraints. However, there are many research challenges that need to be addressed first, such as, developing a systematic analysis of the threat model and its relation to the security countermeasures, the precise definitions of *security metrics*, and the detailed study of real world deployment scenarios.

Acknowledgements

We would like to thank Zong-Syun Lin, Saurabh Amin, Hsin-Yi Tsai, and Yu-Lun Huang for their work on the chemical reactor plant. We would also like to thank Kristofer Pister for discussions on the practical applications of sensor networks. This work was supported in part by TRUST (Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: AFOSR (#FA9550-06-1-0244) Cisco, British Telecom, ESCHER, HP, IBM, iCAST, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia, and United Technologies.

References

[1] Hart, <http://www.hartcomm2.org/frontpage/wirelesshart.html>, WirelessHart whitepaper.
[2] ISA, <http://www.isa.org/isasp100>, Wireless Systems for Automation [cited June, 2007]. http://www.isa.org/isasp100.
[3] J. Slay, M. Miller, Lessons learned from the Maroochy water breach, Critical Infrastructure Protection, vol. 253/2007, Springer, Boston, 2007, pp. 73–82.
[4] U.S.G.A. Office, Critical infrastructure protection, Multiple efforts to secure control systems are under way, but challenges remain, Techical Report GAO-07-1036, Report to Congressional Requesters, 2007.
[5] J. Eisenhauer, P. Donnelly, M. Ellis, M. O'Brien, Roadmap to secure control systems in the energy sector, energetics incorporated, Sponsored by the US Department of Energy and the US Department of Homeland Security, 2006.
[6] N.W. Group, Internet security glossary, <http://rfc.net/rfc2828.html>, May 2000.
[7] M. Manzo, T. Roosta, S. Sastry, Time synchronization attacks in sensor networks, in: SASN'05: Proceedings of the Third ACM Workshop on Security of Ad hoc and Sensor Networks, 2005.
[8] T. Roosta, W.-C. Liao, W.-C. Teng, S. Sastry, Testbed implementation of a secure flooding time synchronization protocol, in: IEEE Wireless Communication and Networking Conference, 2008.
[9] J. Newsome, E. Shi, D. Song, A. Perrig, The Sybil attack in sensor networks: analysis and defenses, in: IPSN'04: Proceedings of the Third International Symposium on Information Processing in Sensor Networks, 2004.
[10] J.R. Douceur, The Sybil attack, in: IPTPS'01: Revised Papers from the First International Workshop on Peer-to-Peer Systems, 2002.
[11] B. Parno, A. Perrig, V. Gligor, Distributed detection of node replication attacks in sensor networks, in: IEEE Symposium on Security and Privacy, May 2005.
[12] A.A. Cárdenas, S. Radosavac, J.S. Baras, Performance comparison of detection schemes for mac layer misbehavior, in: INFOCOM, 2007.
[13] Y.-C. Hu, A. Perrig, D.B. Johnson, Ariadne: a secure on-demand routing protocol for ad hoc networks, Wireless Networks 11 (2005) 21–38.
[14] H. Chan, V.D. Gligor, A. Perrig, G. Muralidharan, On the distribution and revocation of cryptographic keys in sensor networks, IEEE Transactions on Dependable and Secure Computing 2 (3) (2005) 233–247.
[15] K. Römer, F. Mattern, The design space of wireless sensor networks, IEEE Wireless Communications 11 (6) (2004) 54–61.
[16] L. Eschenauer, V. Gligor, A key-management scheme for distributed sensor networks, in: Proceedings of the Ninth ACM Conference on Computer and Communications Security, 2002, pp. 41–47.
[17] D. Liu, P. Ning, Establishing pairwise keys in distributed sensor networks, in: Computer Communication Society, October 2003.
[18] W. Du, J. Deng, Y. Han, P. Varshney, A pairwise key pre-distribution scheme for wireless sensor networks, in: Tenth ACM Conference on Computer and Communications Security (CCS03), October 2003.
[19] H. Chan, A. Perrig, D. Song, Random key predistribution schemes for sensor networks, in: IEEE Symposium Research in Security and Privacy, 2003.
[20] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, D.E. Culler, Spins: security protocols for sensor networks, Wireless Networks 8 (5) (2002) 521–534.

[21] A. Liu, P. Ning, Tinyecc: a configurable library for elliptic curve cryptography in wireless sensor networks, in: Technical Report TR-2007-36, North Carolina State University, Department of Computer Science, November 2007.
[22] V. Gupta, M. Wurn, Y. Zhu, M. Millard, S. Fung, N. Gura, H. Eberle, S.C. Shantz, Sizzle: a standards-based end-to-end security architecture for the embedded internet, Pervasive and Mobile Computing 1 (4) (2005) 425–445.
[23] D. Malan, M. Welsh, M. Smigh, A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography, in: First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Cummunications and Networks SECON, 2004, pp. 71–80.
[24] K. Piotrowski, P. Langendoerfer, S. Peter, How public key cryptography influences wireless sensor node lifetime, in: Proceedings of the Fourth ACM Workshop on Security of Ad hoc and Sensor Networks, 2006, pp. 169–176.
[25] Z. Alliance, http://www.zigbee.org, ZigBee Specification. <http://www.zigbee.org>.
[26] N. Corporation, <www.ntru.com/about/ntru_corp.pdf>, 2003.
[27] S. Oh, S. Russell, S. Sastry, Markov chain Monte Carlo data association for general multiple-target tracking problems, in: IEEE International Conference on Decision and Control, December 2004.
[28] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (Mobicom'00), August 2000.
[29] D. Wagner, Resilient aggregation in sensor networks, in: ACM Workshop on Security of Ad Hoc and Sensor Networks, October 2004.
[30] R. Berber, Methods of Model Based Process Control, Springer, 1995 (chapter 2).
[31] Z.-S. Lin, A.A. Cárdenas, S. Amin, H.-Y. Tsai, Y.-L. Huang, S. Sastry, Understanding the physical and economic consequences of attacks against control systems, International Journal of Critical Infrastructure Protection.
[32] N. Ricker, Model predictive control of a continuous, nonlinear, two-phase reactor, Journal of Process Control 3 (1993) 109.

Alvaro A. Cardenas received a B.S. with a major in electrical engineering and a minor in mathematics from the Universidad de los Andes, Bogota, Colombia, in 2002, and an M.S. and a Ph.D. in electrical and computer engineering from the University of Maryland, College Park, in 2002 and 2006, respectively. He is currently a postdoctoral scholar at the University of California, Berkeley. His research interests include information security, statistics, and machine learning. He received a two-year graduate school fellowship from the University of Maryland and a two-year distinguished research assistantship from the Institute of Systems Research.

Tanya Roosta received her B.S., M.S. and Ph.D. in electrical and computer sciences from the University of California at Berkeley. She also holds an M.A. from the University of California at Berkeley in statistics. She received the 3-year National Science Foundation fellowship for her graduate studies. Her research interests include sensor network security, fault detection, reputation systems, privacy issues associated with the application of sensors at home and health care, and sensor networks used in critical infrastructures. Her additional research interests include: robust statistics, outlier detection, statistical modeling, and the application of game theory to sensor network design.

Shankar Sastry received a B.Tech. from the Indian Institute of Technology, Bombay, in 1977, and an M.S. in EECS, an M.A. in mathematics, and a Ph.D. in EECS from the University of California at Berkeley, in 1979, 1980, and 1981, respectively. Dr. Sastry is currently the dean of the College of Engineering. He was formerly the director of CITRIS (Center for Information Technology Research in the Interest of Society) and the Banatao Institute. He served as the chair of the EECS Department, as the director of the Information Technology Office at DARPA, and as the director of the Electronics Research Laboratory at Berkeley, an organized research unit on the Berkeley campus conducting research in computer sciences and all aspects of electrical engineering. He is the NEC Distinguished Professor of Electrical Engineering and Computer Sciences and holds faculty appointments in the Departments of Bioengineering, EECS, and Mechanical Engineering.