DESIGNING SECURE SENSOR NETWORKS

ELAINE SHI AND ADRIAN PERRIG, CARNEGIE MELLON UNIVERSITY

Sensor networks are a promising approach for a variety of applications, such as monitoring safety and security of buildings and spaces, measuring traffic flows, tracking environmental pollutants, etc. Sensor networks will play an essential role in the upcoming age of pervasive computing.

ABSTRACT

Sensor networks are expected to play an essential role in the upcoming age of pervasive computing. Due to their constraints in computation, memory, and power resources, their susceptibility to physical capture, and use of wireless communications, security is a challenge in these networks. The scale of deployments of wireless sensor networks require careful decisions and trade-offs among various security measures. The authors discuss these issues and consider mechanisms to achieve secure communication in these networks.

INTRODUCTION

Sensor networks are a promising approach for a variety of applications, such as monitoring safety and security of buildings and spaces, measuring traffic flows, and tracking environmental pollutants. Sensor networks will play an essential role in the upcoming age of pervasive computing, as our personal mobile devices will interact with sensor networks in our environment.

Many sensor networks have mission-critical tasks, so it is clear that security needs to be taken into account at design time. Security will be important for most applications for the following reasons. Most sensor networks actively monitor their surroundings, and it is often easy to deduce information other than the data monitored. Such unwanted information leakage often results in privacy breaches of the people in the environment. Moreover, the wireless communication employed by sensor networks facilitates eavesdropping and packet injection by an adversary. The combination of these factors demands security for sensor networks to ensure operation safety, secrecy of sensitive data, and privacy for people in sensor environments.

Security in sensor networks is complicated by the constrained capabilities of sensor node hardware and the properties of the deployment:

- Since sensor nodes usually have severely constrained computation, memory, and energy resources, asymmetric cryptography is often too expensive for many applications. Thus, a promising approach is to use more efficient symmetric cryptographic alternatives. In contrast to asymmetric cryptography (e.g., the RSA signature algorithm or the Diffie-Hellman key agreement protocol), symmetric cryptography (e.g., the AES block cipher or the HMAC-SHA-1 message authentication code) is three to four orders of magnitude faster to compute. However, symmetric cryptography is not as versatile as public key cryptographic techniques, which complicates the design of secure applications.
- Sensor nodes are susceptible to physical capture, but because of their targeted low cost, tamper-resistant hardware is unlikely to prevail. Therefore, when designing a secure sensor network we must assume that nodes may be compromised by an attacker. Compromised nodes may exhibit arbitrary behavior and may collude with other compromised nodes.
- Sensor nodes use wireless communication, which is particularly easy to eavesdrop on. Similarly, an attacker can easily inject malicious messages into the wireless network.
- Security also needs to scale to large-scale deployments. Most current standard security protocols were designed for two-party settings and do not scale to a large number of participants. We expect future sensor networks with thousands of sensor nodes, so it is clear that scalability is a prerequisite for any viable approach.

In this article we discuss security from a networking perspective and consider mechanisms to achieve secure communication. We will first discuss the threat and trust model for sensor networks. We will then discuss security requirements and propose specific countermeasures against attacks. Finally, we describe promising research directions and conclude.

THREAT AND TRUST MODEL

In this section we discuss the threat and trust models we expect to encounter in current sensor network applications. We consider insider and outsider attacks, and discuss a base-station-based trust model.

OUTSIDER ATTACKS

In an outsider attack, the attacker node is not an authorized participant of the sensor network. As the sensor network communicates over a wireless channel, a passive attacker can easily eavesdrop on the network's radio frequency range, in an attempt to steal private or sensitive information. For instance, in a commercial inventory application, it is clear that a competitor should not have access to inventory levels communicated across a wireless network. The adversary can also alter or spoof packets, to infringe on the authenticity of communication or inject interfering wireless signals to jam the network.

1536-1284/04/$20.00 © 2004 IEEE

Another form of outsider attack is to disable sensor nodes. To this end, an attacker can inject useless packets to drain the receiver's battery; he or she may capture and physically destroy nodes (e.g., with a hammer or explosives). Furthermore, benign node failures may result from nonadversarial factors such as battery depletion and catastrophic climate events. A failed node is indistinguishable from a disabled node. Therefore, although benign node failure is not really an attack, addressing benign node failures is inseparable from addressing disabled nodes, and is part of our security considerations.

Insider Attacks/Node Compromise

Node compromise is the central problem that uniquely characterizes the sensor network's threat model. With node compromise, an adversary can perform an insider attack. In contrast to disabled nodes, compromised nodes actively seek to disrupt or paralyze the network.

A compromised node may exist in the form of a subverted sensor node (i.e., a captured sensor node that has been reprogrammed by the attacker); or it can be a more powerful device such as a laptop, with more computational and memory resources and a more powerful radio. A compromised node has the following properties:

1. The device is running some malicious code that is different from the code running on a legitimate node and seeks to steal secrets from the sensor network or disrupt its normal functioning.
2. The device has a radio compatible with the sensor nodes such that it can communicate with the sensor network.
3. The device is an authorized participant in the sensor network. Assuming that communication is encrypted and authenticated through cryptographic primitives, the device must be in possession of the secret keys of a legitimate node such that it can participate in the secret and authenticated communications of the network.

In the worst case, a compromised node can exhibit arbitrary behavior, which is well known as the Byzantine model [1].

The Base Station as a Point of Trust

Sensor networks are usually deployed with one or more base stations. A base station is a much more powerful node with rich computational, memory, and radio resources. A base station usually exists in the form a PC or server. It serves as the data sink/processor, and as the interface between the sensor network and the external world. It is reasonable to assume that a base station is physically protected or has tamper-robust hardware, so we can conveniently rule out base station compromise. Thus, the base station can act as a central trusted authority in protocol design. Given the numerous security breaches of recent "secure" systems, we need to be very careful with such assumptions, and do our best to retain a maximum level of security in case even the base station is compromised.

However, scalability becomes a major concern if we make use of a central trusted authority in attack defense mechanisms. For instance, a simple way to establish pairwise keys between sensor nodes is to have the base station act as an intermediary: each node is configured with a secret key that it shares with the base station. We call the secret key node A shares with the base station K_A, and similarly K_B is the shared key between node B and the base station. If nodes A and B wish to establish a shared secret key K_{AB}, the base station can act as a trusted intermediary to establish that key, for example, by sending a random K_{AB} encrypted with K_A to node A and encrypted with K_B to node B. However, nonces or other mechanisms need to be used to ensure key freshness [2]. If each pair of neighboring sensor nodes wants to set up a shared secret key, the base station would become a scalability bottleneck as it would need to help set up $d \cdot n/2$ keys, assuming that each sensor node has d neighbors in a network with n nodes. Moreover, the nodes neighboring the base station suffer from higher communication overhead as they need to relay the key setup messages, and may thus run out of battery energy sooner.

In summary, while the base station may serve as a central trusted authority in a sensor network, we must use it with care and keep scalability concerns in mind.

A simple way to establish pair-wise keys between sensor nodes is to have the base station act as an intermediary: each node is configured with a secret key that it shares with the base station.

Security Requirements

In this section we discuss the security properties and requirements we would like to achieve in sensor networks. Before we present the standard security requirements, we discuss the desired properties of a secure sensor network protocol.

Desired Properties

For any secure sensor network protocol, we would like to achieve robustness against outsider attacks, and graceful degradation of security in case of insider attacks.

Robustness against Outsider Attacks — Most applications require security against outsider attacks. For well-known outsider attacks such as eavesdropping or packet injection, we may leverage standard security techniques; for example, we can use cryptographic primitives to guarantee the authenticity and secrecy of communication between legitimate nodes. In addition, it is necessary to design mechanisms that are robust to node failures. One way to achieve this is to deploy nodes in large quantities and leverage redundancy such that a few failed nodes will not cause network partitions. Also, network protocols need to be able to identify failed neighbors in real time and adjust according to the updated topology.

Resilience to Insider Attacks, Graceful Degradation with Respect to Node Compromise — Security-critical sensor networks require mechanisms to deal with compromised nodes. Ideally, we would like to be able to detect any compromised node and revoke its cryptographic keys. However, in practice this is not always possible.

An alternative design approach is to design mechanisms that are resilient to node compromise, such that performance *gracefully degrades*

Various attacks can compromise the availability of the sensor network. When considering availability in sensor networks, it is important to achieve graceful degradation in the presence of node compromise or benign node failures.

when a small fraction of nodes are compromised.

Realistic Levels of Security — While we discuss security requirements in general, the security concerns of a sensor network and the level of security desired may differ according to application-specific needs. For instance, in a health monitoring application where we use universally deployed sensor nodes to monitor people's locations and health conditions, we are concerned about protecting people's privacy. Yet we would hardly bother to protect the privacy of fish in a ocean monitoring application.

We now discuss specific security requirements in more detail.

Authentication

Since sensor networks use a shared wireless communication medium, authentication is necessary to enable sensor nodes to detect maliciously injected or spoofed packets. Authentication enables a node to verify the origin of a packet (source authentication) and ensure data integrity, that is, ensure that data is unchanged (data authentication). Almost all applications require data authentication. On one hand, for military and safety-critical applications, the adversary has clear incentives to inject false data reports or malicious routing information; on the other hand, even for civilian applications such as office/home applications where we expect a relatively nonadversarial environment, it is still risk-prone to go without authentication, for then people only moderately skilled would be able to meddle with the sensor network protocols solely out of mischief.

Although authentication prevents outsiders from injecting or spoofing packets, it does not solve the problem of compromised nodes. Since a compromised node has the secret keys of a legitimate node, it can authenticate itself to the network. However, we may be able to use intrusion detection techniques to find the compromised nodes and revoke their cryptographic keys network-wide.

Secrecy

Ensuring the secrecy of sensed data is important for protecting data from eavesdroppers. We can use standard encryption functions (e.g., the AES block cipher) and a shared secret key between the communicating parties to achieve secrecy. However, encryption itself is not sufficient for protecting the privacy of data, as an eavesdropper can perform traffic analysis on the overheard ciphertext, and this can release sensitive information about the data. In addition to encryption, privacy of sensed data also needs to be enforced through access control policies at the base station to prevent misuse of information. Consider, for example, a person locator application. Sensors are implanted in an office building to sense the location of people, and the information is sent to a Web server to answer requests to locate a person. Generally, people would like to limit the right to access their current location to a small group of people. Therefore, access control has to be enforced at the Web server to prevent misuse of information by unintended parties.

Node compromise complicates the problem of secrecy, for sensitive data may be released when a compromised node is one endpoint of the communication; or if a globally or group shared key is used, the compromised node can successfully eavesdrop and decrypt the communication between other sensor nodes within its radio frequency (RF) range.

Availability

Providing availability requires that the sensor network be functional throughout its lifetime. Denial-of-service (DoS) attacks often result in a loss of availability. In practice, loss of availability may have serious impacts. In a manufacturing monitoring application, loss of availability may cause failure to detect a potential accident and result in financial loss; in a battlefield surveillance application, loss of availability may open a back door for enemy invasion.

Various attacks can compromise the availability of the sensor network. When considering availability in sensor networks, it is important to achieve graceful degradation in the presence of node compromise or benign node failures.

Service Integrity

Above the networking layer, the sensor network usually implements several application-level services. Data aggregation is one of the most important sensor network services. In data aggregation, a sensor node collects readings from neighboring nodes, aggregates them, and sends them to the base station or another data processing node. The goal of secure data aggregation is to obtain a relatively accurate estimate of the real-world quantity being measured, and to be able to detect and reject a reported value that is significantly distorted by corrupted nodes. Another example is the time synchronization service. Current time synchronization protocols designed for sensor networks assume a trusted environment [3]. An open research problem is how to develop a time synchronization protocol that achieves graceful degradation in the presence of compromised nodes.

Attacks and Countermeasures

In this section we discuss countermeasures to the attacks we presented in the previous section.

On Secrecy and Authentication

Standard cryptographic techniques can protect the secrecy and authenticity of communication links from outsider attacks such as eavesdropping, packet replay attacks, and modification or spoofing of packets.

Key Establishment and Management — For two sensor nodes to set up a secret and authenticated link, they need to establish a shared secret key. The key establishment problem studies how to set up secret keys between a pair of nodes in the network. A naive idea is to use a global key stored on each sensor node prior to deployment, yet this is particularly vulnerable to node compromise, for the adversary only has to compromise one node and all communication links will be compromised. Public key cryptography is

a popular method for key establishment, but the computational cost may be too high for many applications, even if key establishment only needs to be performed when the sensors are initially installed. A drawback of public key cryptography is that it may open up the network to DoS attacks, as an attacker can send a bogus message to a sensor node, enticing it to perform seconds of signature verification only to notice that the message is fake. Recently, researchers proposed a class of random key predistribution techniques that address the problem of key establishment [4–7] However, further research is necessary to improve these algorithms in terms of scalability, resilience to node compromise, memory requirements, and communication overhead.

Broadcast/Multicast Authentication — Broadcast and multicast are indispensable for many sensor network protocols. In broadcast and multicast, source authentication poses a new research challenge. One possible approach is to use a digital signature, where the source signs each message with a private key and all the receivers verify the message using the public key. Unfortunately, public key cryptography is too costly for sensor networks. To address this problem, Perrig *et al.* proposed the μTesla protocol that provides secure broadcast authentication assuming loose time synchronization between sensor nodes [2]. The basic idea behind μTesla is to introduce asymmetry into symmetric key cryptography through delayed key disclosure and one-way function key chains.

On Availability

The class of attacks against network availability is often referred to as DoS attacks [8]. DoS attacks can be targeted at different layers of the networking stack.

Jamming and Packet Injection — Jamming can be targeted at different layers. At the physical layer, the attacker can send out interfering RF signals to impede communication. The jamming attacker can also aim at draining the nodes' battery by injecting irrelevant data or wasting battery energy on the receiving node for radio reception. The standard defense to physical jamming is frequency hopping and spread spectrum communication [9], requiring the attacker to expend significantly more energy to successfully jam communications.

Link-layer jamming exploits properties of the medium access control protocol employed. For instance, the attack can induce malicious collisions or attempt to get an unfair share of the radio resource. In defense, we need to design secure medium access control protocols. Wood and Stankovic studied link jamming systematically and proposed using error correcting codes to cope with the collision attack, rate limitation to deal with the exhaustion attack, and small frames to deal with an unfairness attack [8].

At the networking layer, the attacker can inject malicious packets. We can use authentication to enable the receiver to detect malicious packets, and message freshness through nonces to detect replayed packets.

The Sybil Attack — The Sybil attack is where a malicious node illegitimately claims multiple identities [10, 11]. The Sybil attack can be exploited at different layers to cause service disruption. At the MAC layer, by presenting multiple identities the malicious node can claim a dominating fraction of the shared radio resource, so legitimate nodes are left with little chance to transmit. At the routing layer, the Sybil attacker can lure network traffic to go through the same physical malicious entity. Imagine a simple routing protocol where a node chooses an upstream neighbor as the next hop with equal probability. By claiming to be a large number of identities, with high probability a Sybil identity will be selected as the next hop. Therefore, a "sinkhole" is created and the attacker can hence do selective forwarding [12].

We proposed several Sybil defense mechanisms suited for sensor networks [11]. One promising approach is to leverage the key predistribution process. The basic idea is to associate each node's identity with the keys assigned to it, so a node attempting to spoof identity A can succeed only when it has the corresponding keys of A; otherwise, it either fails to establish a communication link with the network or fails to survive validation.

Miscellaneous Attacks against Routing — At the networking layers, the adversary can mount miscellaneous attacks to disrupt routing availability. Routing availability is sacrificed if an intended recipient is denied the message. With compromised nodes, a simple attack is to drop packets or perform selective forwarding [12]. Multipath routing is a possible defense against this type of attack [13, 14]. The basic idea is to use multiple disjoint paths to route a message such that it is unlikely that every path is controlled by compromised nodes.

More sophisticated attacks include spreading bogus routing information, creating sinkholes or wormholes, and Hello flooding. Karlof and Wagner systematically study how different routing protocols are vulnerable to these attacks [12].

Stealthy Attacks against Service Integrity

In a stealthy attack, the attacker's goal is to make the network accept a false data value. In a data aggregation scenario, the false data value is a false aggregation result. The attacker has several options to achieve this goal. For instance, a corrupted sensor/aggregator can report significantly biased or fictitious values. A compromised node can also perform a Sybil attack, and all the imaginary identities can collude in reporting false data. The Sybil attack allows one compromised node to have greater impact on the aggregated result. The attacker can also perform DoS attacks so that legitimate nodes cannot report their sensor readings to the base station. Przydatek *et al.* [15] studied the stealthy attack in the data aggregation context and proposed SIA, a secure information aggregation protocol robust to the stealthy attack.

Consider time synchronization: a stealthy attacker's goal is to disseminate false timing information to desynchronize nodes. The attacker can intercept and delay synchronization mes-

At the networking layer, the attacker can inject malicious packets. We can use authentication to enable the receiver to detect malicious packets, and message freshness through nonces to detect replayed packets.

Sensor networks are usually immobile, and traffic patterns of a sensor network differ from that of an ad-hoc network, i.e., sensor network routing is often data-centric. Therefore, we need to design a secure routing protocol well-suited for sensor networks.

sages, or spread false synchronization messages. Similar to the data aggregation case, he or she can also exploit the Sybil and DoS attacks to disrupt the time synchronization protocol. So far, time synchronization protocols in sensor networks assume a trusted environment, making them particularly susceptible to various forms of stealthy attack.

Promising Research Directions

Code Attestation

Coping with compromised nodes is the most difficult challenge of sensor network security. To address this problem, a promising direction is to use code attestation to validate the code running on each sensor node. Because the code running on a malicious node must be different from that on a legitimate node, we can detect compromised nodes by verifying their memory content.

Code attestation may be achieved through either hardware or software. On the hardware side, the vision of a new trusted computing age sheds new light on future computing devices: they will be equipped with trusted hardware such as those being developed by the Trusted Computing Group (TCG) [16] or the Next-Generation Secure Computing Base (NGSCB) [17]. We can build attestation mechanisms exploiting the trusted hardware such that a remote party can verify the code running on a device. To enable the use of trusted hardware on sensor nodes, it will be essential to reduce cost, enhance efficiency, and minimize energy consumption. We may also strive toward code attestation through pure software means. So far little research has been done in this aspect, and we believe it is a promising research direction.

Secure Misbehavior Detection and Node Revocation

Since compromised nodes are particularly harmful to the sensor network, it is desirable to detect and revoke compromised nodes in a timely fashion. Chan *et al.* proposed to use a distributed voting system to tackle the problem (i.e., if node A discovers that node B is misbehaving, it may cast a vote against node B). If a sufficient number of votes against node B have been observed, all other nodes refuse to communicate with B [4]. A potential problem here is that malicious nodes can slander legitimate nodes (i.e., cast votes against legitimate nodes). Also, a malicious node can pretend to be a victim to make a legitimate node look bad. For instance, it can report a lost message and attribute the blame to its upstream node. Even worse, a malicious node may be able to make a legitimate node look bad to other legitimate nodes so that they will engage in revocation against each other. One way to start addressing these problems is to limit each node to m potential votes, such that when an attacker captures a node, it gets m votes against other innocent nodes. To achieve this, we could store the votes in each node's key ring prior to deployment in deactivated mode. On key setup, each node pair exchanges the activation value to allow its neighbor to vote against it [4].

Secure Routing

A secure routing protocol should enable communication despite adversarial activities. So far routing protocols for sensor networks, such as directed diffusion [18] and geographic routing [19], assume a trusted environment. Meanwhile, secure routing protocols have been proposed for ad hoc networks (e.g., Ariadne [20]). Ariadne prevents compromised nodes from tampering with uncompromised routes consisting of uncompromised nodes, and also prevents a large number of types of DoS attacks. It utilizes efficient symmetric key primitives, but would still be too heavyweight for sensor networks due to its communication, memory, and per-packet processing overhead. In addition, sensor networks are usually immobile, and traffic patterns of a sensor network differ from that of an adhoc network (i.e., sensor network routing is often data-centric). Therefore, we need to design a secure routing protocol well suited to sensor networks.

Secure Localization

Securing localization is an important primitive in sensor networks. This problem has two aspects: a sensor node can accurately determine its geographic coordinates in an adversarial environment, and a malicious sensor node cannot claim a false position to the infrastructure. Capkun and Hubaux studied the former problem and make use of secure distance bounding and distance estimation techniques [21]. Sastry, Shankar, and Wagner [22] and Capkun and Hubaux [23] studied the latter problem and propose mechanisms that enable an infrastructure to securely verify location claims.

Securing location determination is a prerequisite for secure geographic routing. It may also help us to solve problems such as the wormhole attack and the Sybil attack. For the wormhole attack, if a route consists of two consecutive nodes that are distant in geographic location, we may cast suspicion on the integrity of this route [24]. For the Sybil attack, a concentration of nodes in a small geographic area is suspicious. Thus, secure location determination is an important building block to secure sensor networks.

Efficient Cryptographic Primitives

Because sensor nodes are constrained in computational and storage resources, traditional security solutions for other types of networks such as the Internet are often too expensive for sensor networks. Perrig *et al.* designed the SPINS protocol suite, leveraging efficient block ciphers to perform a variety of cryptographic operations [2]. Karlof, Sastry, and Wagner designed TinySec [25], trading off efficiency and security. More research in this domain is necessary, especially in exploring the use of efficient asymmetric cryptographic mechanisms for key establishment and digital signatures.

Conclusion

Widespread deployment of sensor networks is on the horizon. Given their versatility, sensor networks will soon play an important role in

critical military applications as well as pervade our daily life. However, security concerns constitute a potential stumbling block to the impending wide deployment of sensor networks. Current research on sensor networks is mostly built on a trusted environment. Several exciting research challenges remain before we can trust sensor networks to take over important missions.

REFERENCES

[1] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Trans. Programming Languages and Systems*, vol. 4, no. 3, July 1982, pp. 382–401.

[2] A. Perrig *et al.*, "SPINS: Security Protocols for Sensor Networks," *Wireless Networks J.*, vol. 8, no. 5, Sept. 2002, pp. 521–34.

[3] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization Using Reference Broadcasts," *Proc. 5th Symp. Op. Sys. Design and Implementation*, Dec. 2002.

[4] H. Chan, A. Perrig, and D. Song, "Random Key Pre-distribution Schemes for Sensor Networks," *IEEE Symp. Security and Privacy*, May 2003.

[5] W. Du *et al.*, "A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks," *Proc. 10th ACM Conf. Comp. and Commun. Security*, Oct. 2003, pp. 42–51.

[6] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *Proc. 9th ACM Conf. Comp. and Commun. Security*, Nov. 2002, pp. 41–47.

[7] D. Liu and P. Ning. Establishing Pairwise Keys in Distributed Sensor Networks," *Proc. 10th ACM Conf. Comp. and Commun. Securit*, Oct. 2003, pp. 52–61.

[8] A. Wood and J. Stankovic, "Denial of Service in Sensor Networks," *IEEE Comp.*, Oct. 2002, pp. 54–62.

[9] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of Spread Spectrum Communications: A Tutorial," *IEEE Trans. Commun.*, vol. 30, no. 5, May 1982, pp. 855–84.

[10] J. R. Douceur, "The Sybil Attack," *1st Int'l. Wksp. Peer-to-Peer Systems*, Mar. 2002.

[11] J. Newsome *et al.*, "The Sybil Attack in Sensor Networks: Analysis and Defenses," *Proc. IEEE Int'l., Conf. Info. Processing in Sensor Networks*, Apr. 2004.

[12] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Proc. 1st IEEE Int'l., Wksp. Sensor Network Protocols and Applications*, May 2003.

[13] B. Deb, S. Bhatnagar, and B. Nath, "ReInForM: Reliable Information Forwarding Using Multiple Paths in Sensor Networks," *Proc. 28th Annual IEEE Int'l. Conf. Local Computer Networks (LCN 2003)*, Oct. 2003.

[14] D. Ganesan *et al.*, "Highly Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks," Mobile Comp. and Commun. Review, 5(4):10-24, 2002.

[15] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure Information Aggregation in Sensor Networks," *Proc. 1st ACM Int'l. Conf. Embedded Networked Sensor Sys.*, Nov. 2003, pp. 255–65.

[16] Trusted Computing Group, https://www.trustedcomputinggroup.org/, 2003.

[17] Next-Generation Secure Computing Base (NGSCB), http://www.microsoft.com/resources/ngscb/default.msp, 2003.

[18] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," 2000.

[19] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. MobiCom 2000*, 2000, pp. 243–54.

[20] Y.-Chun Hu, A. Perrig, and D. B. Johnson, "Ariadne: A Secure on-demand Routing Protocol for Ad Hoc Networks," *Proc. Mobicom 2002*, Sept. 2002.

[21] S. Capkun and J.-P. Hubaux, "Secure Positioning in Sensor Networks," Tech. rep. EPFLIICI200444, Swiss Fed. Inst. Tech., Lausanne, 2004.

[22] N. Sastry, U. Shankar, and D. Wagner, "Secure Verification of Location Claims," *Proc. ACM Wksp. Wireless Security*, Sept. 2003.

23] S. Capkun and J.-P. Hubaux, "Secure Positioning of Wireless Devices with Appplication to Sensor Networks," to appear, IEEE INFOCOM 2005.

[24] Y.-C. Hu, Adrian Perrig, and David B. Johnson, "Packet Leashes: A Defense Against Wormhole Attacks in Wireless Networks," *Proc. IEEE INFOCOM 2003*, Apr. 2003.

[25] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: Link Layer Security for Tiny Devices," http://www.cs.berkeley.edu/Thks/tinyseC/, 2003.

BIOGRAPHIES

ADRIAN PERRIG (perrig@cmu.edu) is an assistant professor with appointments in the Departments of Electrical and Computer Engineering, Engineering and Public Policy, and Computer Science at Carnegie Mellon University.

ELAINE SHI (rshi@cmu.edu) is a doctoral student in the Department of Computer Science at Carnegie Mellon University.

Current research on sensor networks is mostly built on a trusted environment. Several exciting research challenges remain before we can trust sensor networks to take over important missions.