2011 Eighth International Conference on Information Technology: New Generations

Security of Wireless Sensor Networks

Daniel E. Burgner
Department of Computer Science
Norfolk State University
700 Park Avenue
Norfolk, Virginia 23504
USA
d.e.burgner@spartans.nsu.edu

Luay A. Wahsheh
Department of Computer Science
Norfolk State University
700 Park Avenue
Norfolk, Virginia 23504
USA
law@nsu.edu

***Abstract*—Wireless sensor networks have been researched extensively over the past few years. They were first used by the military for surveillance purposes and have since expanded into industrial and civilian uses such as weather, pollution, traffic control, and healthcare. One aspect of wireless sensor networks on which research has been conducted is the security of wireless sensor networks. These networks are vulnerable to hackers who might go into the network with the intent of rendering it useless. An example of this would be an enemy commandeering a drone and getting it to attack friendly forces. In this paper, we review the security of wireless sensor networks. Areas that are covered include: architectures and routing protocols; security issues that include context and design as well as confidentiality, integrity, and authenticity; algorithms; and performance issues for wireless sensor network design. Performance of the Self-Originating Wireless Sensor Network (SOWSN), Practical Algorithm for Data Security (PADS), and mechanisms for in-network processing were investigated in further detail with SOWSN having the best performance as a result of it being based on realistic scenarios.**

***Keywords*-algorithms; architectures; performance; routing protocols.**

I. Introduction

The security of wireless sensor networks is an area that has been researched considerably over the past few years. Applications for these networks are varied and they all involve some level of monitoring, tracking, controlling, or a combination thereof. Wireless sensor networks have characteristics that are unique to them, such as the ability to withstand unfavorable environmental conditions, dynamic network topology, communication failures, large scale of deployment, scalable node capacity, node mobility, unattended operation as well as limited power, to name a few. They also have base stations, which have more resources, that act as a gateway between the sensor nodes and the end user.

This paper reviews the architectures and routing protocols associated with wireless sensors networks in Section II, security issues in Section III, algorithms designed for these networks in Section IV, and performance issues in Section V. Finally, we conclude the paper in Section VI.

II. Architectures and Routing Protocols

One important aspect associated with wireless sensor networks is the architectures and routing protocols. Architectures are the backbone of any network and the routing protocols are the means in which the network uses to communicate. One architecture and one routing protocol are presented as examples.

A. Architecture

One architecture used for wireless sensor networks is a Self Organizing Wireless Sensor Network (SOWSN) that utilizes a star-mesh topology. It consists of two types of wireless nodes: a base station and a sensor node. The sensor node is responsible for the collection of events resulting from malicious targets. This involves up to and including: a) gathering information about potentially malicious tasks including target nature and relative position; b) event generation in real-time regarding detected targets with event transmission to an event analysis center via a base station; and c) relaying generated events to the base station.

Base stations control the actions performed for efficiency sensing support. A base station computes the relative position of the event source and transmits it along with source position and timestamp to the analysis center. Should a base station receive an alert related to a specific target, an identity to this target should be assigned, which allows all target-related alerts receiving the appropriate treatment. All targets have the ability to be identified but their movement is unpredictable. Each sensor in the network has an initial amount of power with base stations being gateways that connect the sensors to the analysis center. Figure 1 illustrates an example of a SOWSN architecture [1].

B. Routing Protocols

Routing protocols are an important component of any wireless sensor network. For wireless sensor networks, design principles of secure routing protocols are poorly understood since no clear definition exists for secure routing in wireless sensor networks. Ács et al. [2] suggest a design for secure routing protocols based upon a formal security

978-0-7695-4367-3/11 $26.00 © 2011 IEEE
DOI 10.1109/ITNG.2011.62

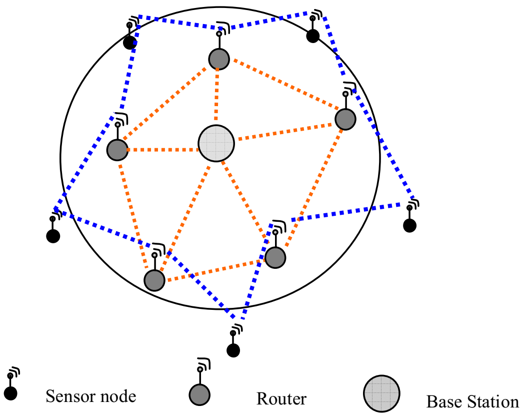


Figure 1. SOWSN architecture.

model. The formal security model would be the general definition of routing security and includes an adversary model with a description of the ideal-world and real-world models. The adversary model is represented by nodes in the network that are considered adversaries. They can relay messages between honest nodes, which have cryptographic secrets used in the wireless sensor network, who are unable to directly communicate with each other or they can hear the communications between the honest nodes. The network model is usually based on static nodes with a single base station as opposed to the adversary model that uses multiple base stations. The real-world model represents the operation of the protocol in the real world while the ideal-world model represents how the model should work ideally. When there is a real-world and an ideal-world model, one also has a real-world and an ideal-world adversary model. The difference between the real-world and ideal-world adversary model is that the real-world adversary model can inject extra messages or modify messages while the ideal-world model cannot as a result of its construction. Other than that, the ideal-world adversary can deliver the same effects as the real-world adversary model. Routing security is defined as a routing protocol that is statistically secure with a security objective. It also states for any configuration and real-world adversary, we have an ideal-world adversary such that the output from the real-world model is indistinguishable from the output of the ideal-world model. It also states that all of this is valid with a negligible function of security functions being forged. Figure 2 illustrates how messages are communicated with one and two adversarial nodes [2].

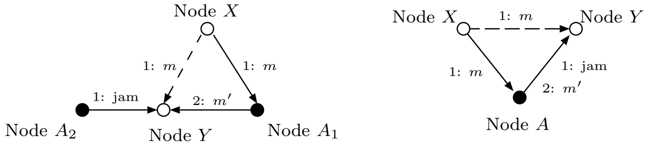


Figure 2. Communication with adversarial and honest nodes.

III. Security Issues

The heart of the matter when it comes to security of these networks are the issues themselves. Three issues that are presented in this paper are the context and design implications, middleware applications as well as confidentiality, integrity, and authenticity.

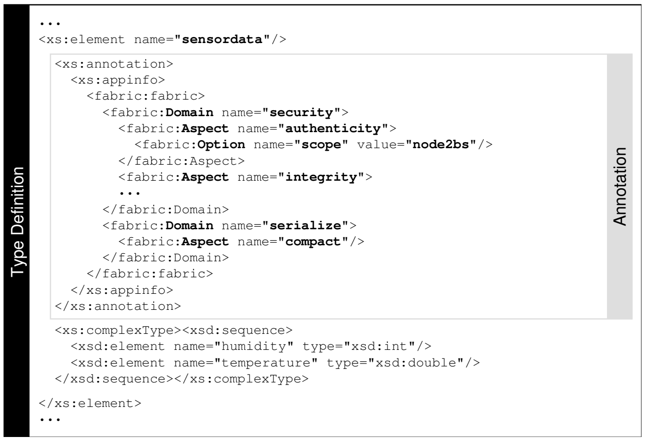
A. Context and Design Implications

Security designs for wireless sensor networks should be placed in a context that relates it to a set of factors confining wireless sensor networks to a region consistent with these networks. Two factors coming into play are attacker motivation as well as vulnerabilities and opportunities. Attacker motivation involves the gains one would make, such as mission interference and benefit gained from the data. Vulnerabilities and opportunities refer to aspects unique to wireless sensor networks, such as: physical access, wireless communication, attacks on coordination and self-configuration and network observability.

These factors affect designs of wireless sensor networks in that they are driven by cost, energy-efficiency, and application-level performance. Networks with high attacker motivation may have to trade off performance or cost for reduction of vulnerabilities to acceptable levels. This would include purchasing of hardware such as sensors, sometimes more expensive and secure or having multiple base stations as opposed to a single base station. When dealing with aspects such as software, protocols, and services, then one would have to contend with more specific tradeoffs between security and performance. Some protocols may expose the location of a destination in each packet, which could result in possible attacks on critical points of the infrastructure. These can be mitigated through the use of encryption at the cost of computational resources [3].

B. Middleware Applications

One middleware application that is implemented involves using a synthesis tool called FABRIC. Support for wireless sensor networks application development is added through the generation of custom-tailored instances for target platforms. Routing and sensor data structures are defined with them being attached to the data type definitions that are represented as domains. The concept is visualized with the Extensible Markup Language (XML) Schema with XML documents being incorporated into each data type definition. Based on each module's self-description, FABRIC selects the best modules for a given annotated type definition. This scheme is implemented as a plug-in for a framework where a user can use a graphical user interface to enter parameters. Security setups and residual risk tables, which display for each security setup for all types of risk attack paths, are generated and presented by the plug-in so that the application developer can select the optimal one for his situation. FABRIC is then invoked after the plug-in rejects values in the annotations for the individual data types. The application developer can implement his application through the use of security mechanisms and key material that are included in the generated middleware. The code size for a manual implementation is comparable in size to that generated by security mechanisms, meaning that no overhead is generated as a result of FABRIC's implementation. Figure 3 illustrates an example FABRIC configuration file [4].


Type Definition
Annotation

```
...
<xs:element name="sensordata"/>
  <xs:annotation>
    <xs:appinfo>
      <fabric:fabric>
        <fabric:Domain name="security">
          <fabric:Aspect name="authenticity">
            <fabric:Option name="scope" value="node2bs"/>
          </fabric:Aspect>
          <fabric:Aspect name="integrity">
          ...
        </fabric:Domain>
        <fabric:Domain name="serialize">
          <fabric:Aspect name="compact"/>
        </fabric:Domain>
      </fabric:fabric>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexType><xsd:sequence>
    <xsd:element name="humidity" type="xsd:int"/>
    <xsd:element name="temperature" type="xsd:double"/>
  </xsd:sequence></xs:complexType>
</xs:element>
...
```

Figure 3. Example excerpt from a FABRIC configuration file.

C. *Integrity, Authenticity, and Confidentiality*

Security of wireless sensor networks as well as all networks has three goals: integrity, authenticity, and confidentiality. Confidentiality refers to the disclosure of information to authorized parties. Integrity refers to checking if data was modified between source and destination. Authenticity refers to data coming from an authorized party along with data confirming the identity of the source. We look at three examples of these goals in action.

1) Practical Algorithm for Data Security (PADS): This algorithm is primarily used for one-time pads (OTP). The message's integrity and authenticity are based on the security of the message authentication code. A 4-byte Message Authentication Code (MAC) is used, meaning an attacker would have to go through 232 attempts, at most, to get a MAC that is a match. The security of the OTP is dependent on the key that is generated [5]. A MAC, also known as a cryptographic checksum, results from the public application of an input via a secret key. Usually of fixed length, it is attached to the input to validate the input's integrity and authenticity [6]. For confidentiality, a new key is generated at each transmission, with the security of the protocol involved dependent upon the Key Derivation Function described by IEEE's Standard Specifications for Public-Key Cryptography [7]. These factors reduce the ability of an attacker to create an OTP as a match. The time an attacker would need to create a match will be past the lifetime of a typical sensor network [5]. An example of such a method is SPINS, which is a three-part approach providing for an authentication routing protocol as well as a three-part approach providing authenticated streaming broadcasts as well as two-party data authentication, data confidentiality, and freshness [8].

2) SOWSN: For the SOWSN described in [1], security requirements would include: not allowing replay of transmitted alerts, not authorizing Denial-of-Service attacks performed by malicious nodes and based on false alerts being generated, not allowing impersonation attacks to succeed and guaranteeing integrity and confidentiality of alerts transmitted by sensor nodes. The base station's two main responsibilities in this regard are to verify the integrity of the messages received based on the signature delivered and to authenticate requests for route establishment before building of any path is authorized.

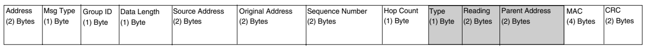
3) Application-Driven Perspective: Like PADS, the encryption of data will support confidentiality. A MAC can also be used via a keyed one-way hash function to support integrity and authenticity. Applications such as SPINS and TinySec can also support integrity, authenticity, and confidentiality. Habitat Monitoring, one application for wireless sensor networks, is not as focused on confidentiality since the expectation is that there is no observer effect. When it comes to integrity and authenticity, they have to be supported with a cost-effective approach through the use of SPINS and TinySec. Another application, Battlefield Monitoring, is more stringent in that confidentiality, integrity, and authenticity would have to be guaranteed. Adversaries could extract transmitted data and modify it to their wishes (injection of false data). The end result could lead to falsified location of enemy forces [3].

IV. Algorithms

Algorithms have been constructed for wireless sensor networks that are tailored to meet the needs of these networks. Examples at which we will look include PADS, SOWSN, RC5 algorithms, and an algorithm used for Indoor Location System (ILS).

A. *PADS*

Albath and Madria [5] use an algorithm that does basic embedding. It calculates a MAC using the static part of the packet. The MAC is added to the data and a time synced key is created based on a secret key shared between the sender and the receiver. Any attacker would have to be time synced with the network or he or she would be unable to break the encryption. They also use a basic detection algorithm to locate the embedded pad, remove it, and return it to its original value. The location and removal are done by the base station since it shares with the embedded sensor node the secret key. Figure 4 illustrates a multi-hop packet structure used by an OTP with certain fields protected by the MAC calculation [5].



Address (2) Bytes	Msg Type (1) Byte	Group ID (1) Byte	Data Length (1) Byte	Source Address (2) Bytes	Original Address (2) Bytes	Sequence Number (2) Bytes	Hop Count (1) Byte	Type (1) Byte	Reading (2) Bytes	Parent Address (2) Bytes	MAC (4) Bytes	CRC (2) Bytes

Figure 4. Multi-hop packet structure. The fields shaded gray are protected by the MAC calculation.

B. SOWSN

For the SOWSN, a Range-Based Algorithm using point-to-point distances/angles is considered. One use for the Range-Based Algorithm involves making detected sensors perform detection with high frequency that will allow alerts to be correlated with respect to target position and collection instants. It will use a multifactor dimensionality reduction (MDR) algorithm to allow nodes to route messages through intermediate node to the closest base station. Sensor nodes use it to send alerts to the base station [1].

C. RC5 Algorithms

Yang et al. [9] implemented a cryptographic primitive of a secure hash function which is based on a block cipher using RC5 algorithms. This allows for the derivation of location-binding keys as well as creating and verifying MACs. This was implemented on MICA2 motes, which are equipped with an 8-bit, 4 MHz microprocessor using a microthread operating system with a limited memory size, 128 KB for program memory and 4 KB for data memory.

RC5 was also used in cryptographic algorithms that generate hash codes and MACs on Berkeley motes. These motes are equipped with 4 MHz processor along with 128 KB flash memory, 4 KB RAM as well as an RFM monolithics TR 1000 radio operating at 19.2 Kbps. In this instance, MACs are generated via standard CBC mode. The sequence number chain is started by the base station, which chooses a random key and encrypts a well-known plaintext with this key. The cipher is used to generate other ciphers, with the process continuing until all the keys have been generated [10].

D. ILS

Paradells et al. [11] use an algorithm that takes into account time and signal strength to obtain accurate localization. This is used for an ILS, which relies on signal strength (in terms of Received Signal Strength Indication or RSSI) and Time of Arrival/Time Difference of Arrival (TOA/TDOA). Data collected on these measures is collected by the reference nodes which send it to a controller computer to determine the best estimate of a node's local position. Their implementation involves obtaining attenuation values present with the system using RSSI measures between reference nodes. The system performs an RSSI measure between known references with the difference between measured and analytically computed values being attenuation resulting from obstacles between reference nodes. The system then performs location measures which are translated into distance estimations plotted as continuous circles. This system is not perfect as it has a situation where RSSI measures between references that have different attenuations than reported between reference and mobile nodes.

V. Performance Issues

Now we look at the performance issues for the wireless sensor networks presented. These approaches include the SOWSN, PADS, and proposed mechanisms for in-network processing.

A. SOWSN

For the SOWSN, the parameters are chosen for input and output. Input parameters are used to represent real scenarios that can happen. They include: base stations, capacity of each base station, number of sensors deployed along with transmission range, maximum transmission power of each sensor, sensor speed, number of targets, and simulation period. Output parameters are used to represent the effects that will be studied. They include the average number of events, connections per event, paths per event, and handoffs.

From experiments performed by Boudriga et al. [1], for an increasing number of sensors, the average number of paths and connections goes up. This goes up as the transmission range goes up as well. As the sensor speed increases, the average number of handoffs increases as well. The probability of loss is dependent on a number of factors, which include transmission range, base station capacity, and the number of sensors. As the number of sensors increases, the probability of loss decreases and this is true as base station capacity increases as well as decreasing the transmission range. Boudriga et al. [1] noted that as the number of sensors is 600 or greater, the probability of loss becomes constant regardless of base station capacity and transmission range.

B. PADS

PADS was placed through simulations along with two other routing protocols, TinySec and AODV, a non-secure routing protocol. Two sets of simulations were performed on each of these protocols. One set involved a total message size of 23 bytes while the other involved a 2-byte payload, resulting in a message size of 18 bytes for AODV, 22 for PADS, and 23 for TinySec. Three areas were evaluated: latency (the average time a packet takes to reach the base station), throughput in bits per second, and average energy use per node.

When the first set of simulations was run, latencies for AODV and PADS were similar, but were worse on average over increasing number of nodes for TinySec. Throughput over increasing number of nodes for PADS and AODV were also comparable but PADS had a level of security. TinySec fared the worst in throughput. Energy usage for PADS and AODV was also comparable except when the number of nodes was 45. TinySec had the worst energy consumption as a result of high failure rates of messages received.

For the second set of simulations, latency has increased for PADS and AODV, but TinySec outperforms both. TinySec's performance regarding latency is irrelevant since it has a low success rate. Throughput for AODV and PADS is significantly better than with TinySec and this is a result of TinySec's performance as it received much fewer messages. As in the first set of simulations, TinySec's energy usage is much higher than that of AODV and PADS [5].

C. In-Network Processing

A prototype has been simulated along with implementation of cryptographic primitives consisting of one-way hash chain generators and MAC on Berkeley motes. Measurements taken from the prototype include network setup overhead, data aggregation performance, and aggregator storage requirements. Measurements taken from the implementation on motes include computation and memory requirements.

Measurements of network setup overhead involve constructing a multi-level hierarchical wireless sensor network with the base station at the center of the network. The network was divided into multiple sensor groups with each level of a sensor group divided into multiple lower-level sensor groups. The network setup involves multiple levels of message exchanges with measurements for a number of packets exchanged for setup in a multi-level hierarchical network. As the number of levels increases, so does the network level overhead and with it, the number of sensor groups.

Measurements of in-network processing performance were taken through an experiment for all sensor nodes report sensor data to their respective aggregators with each aggregator computing a single sensor value. This information is received from its group members, which is forwarded to its aggregator. The network setup is the same for this experiment. As the number of messages increased, the number of packets increased at a slow pace for aggregators at the three levels, but increased significantly when it had no aggregation. In addition, as the level of aggregation increased, fewer packets were exchanged as the number of messages increased.

Aggregators require memory to store cryptographic keys, hash chains, and topology information of their sensor group. The memory needed to store these items is small, but as the network increases in terms of topology, the memory requirements will increase as well. The memory requirements for the topology itself will outstrip those for the keys and subkeys. A Berkeley mote has 4 KB RAM and 512 KB EEPROM, meaning that the topology information can be stored on the EEPROM and if the network is a top-level aggregator, most of its shared subkeys can be stored in the EEPROM.

Aggregators can use two mechanisms to send commands to all nodes in their sensor group: μTESLA and ripple keys. Packets required to disseminate a command were measured using different sensor group sizes with the same density. Compared with using a unicast message, both μTESLA and ripple keys use less overhead. Ripple keys outperform μTESLA for small networks, where μTESLA outperforms ripple keys when larger networks are involved. It should be noted that ripple keys do not require time synchronization and are not hindered by delayed key release, increasing total time needed to disseminate commands [10].

VI. Conclusion

The research reviewed in this paper represents the tip of the iceberg when it comes to the security of wireless sensor networks. Architectures play a key purpose in wireless sensor networks as do unique security issues such as how security affects context and design matters as well as working with confidentiality, integrity, and authenticity. Algorithms also have a role in the process of constructing a wireless sensor network. Lastly, performance issues are addressed to determine if such a proposed design is feasible for use. Based on security analysis of all the wireless sensor networks, it is concluded that SOWSN has the best performance since it is based on real scenarios. More research in this area will continue as it is an emerging technology for years to come.

References

[1] N. A. Boudriga, M. Baghdadi, and M. S. Obaidat, "A New Scheme for Mobility, Sensing, and Security Management in Wireless Ad Hoc Sensor Networks", In *Proceedings of the 39th Annual Symposium on Simulation*, pp. 61–67, 2006.

[2] G. Ács, L. Buttyán, and I. Vajda, "Modelling Adversaries and Security Objectives for Routing Protocols in Wireless Sensor Networks", In *Proceedings of the 4th ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 49–58, 2006.

[3] E. Sabbah, A. Majeed, K. D. Kang, K. Liu, and N. Abu-Ghazaleh, "An Application-Driven Perspective on Wireless Sensor Network Security", In *Proceedings of the 2nd ACM International Workshop on Quality of Service & Security for Wireless and Mobile Networks*, pp. 1–8, 2006.

[4] S. Ransom, D. Pfisterer, and S. Fischer, "Comprehensible Security Synthesis for Wireless Sensor Networks", In *Proceedings of the 3rd International Workshop on Middleware for Sensor Networks*, pp. 19–24, 2008.

[5] J. Albath and S. Madria, "Practical Algorithm for Data Security (PADS) in Wireless Sensor Networks", In *Proceedings of the 6th ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pp. 9–16, 2007.

[6] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Fifth edition, Pearson Education, 2011.

[7] IEEE, IEEE Standard 1363-2000, Standard Specifications for Public Key Cryptography, 2000.

[8] A. Perrig, J. Stankovic, and D. Wagner, “Security in Wireless Sensor Networks”, *Communications of the ACM*, 47(6):53–57, June 2004.

[9] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, “Toward Resilient Security in Wireless Sensor Networks”, In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 34–45, 2005.

[10] J. Deng, R. Han, and S. Mishra, “Security Support for In-Network Processing in Wireless Sensor Networks”, In *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 83–93, 2003.

[11] J. Paradells, J. Vilaseca, and J. Casademont, “Improving Security Applications Using Indoor Location Systems on Wireless Sensor Networks”, In *Proceedings of the International Conference on Advances in Computing, Communication, and Control*, pp. 689–695, 2009.