International Journal of Innovations in Computing
(ISSN : 2319-8257) Vol. 1 Issue 2

Security Protocols for Sensor Networks

Omprakash Deshmukh
Senior Software Engineer,
GalaxE Solutions
om052002@gmail.com

Abstract - ***As sensor networks edge closer towards wide-spread deployment, security issues become a central concern. So far, much research has focused on making sensor networks feasible and useful, and has not concentrated on security. This paper is a review of suite of security building blocks optimized for resource constrained environments and wireless communication. SPINS has two secure building blocks: SNEP and µTESLA. SNEP provides the following important baseline security primitives: Data confidentiality, two-party data authentication, and data freshness. µTESLA is a new protocol which provides authenticated broadcast for severely resource-constrained environments.***

Keywords- Security; Sensor Networks; SNEP ; µTESLA.

I. INTRODUCTION

We envision a future where thousands to millions of small sensors form self-organizing wireless networks. How can we provide security for these sensor networks? Security is not easy; compared with conventional desktop computers, severe challenges exist — these sensors will have limited processing power, storage, bandwidth, and energy. Despite the challenges, security is important for these devices.

As we describe below, we have studied about prototype wireless network sensors .These sensors measure environmental parameters and experiments have done with them to control air conditioning and lighting systems. Serious privacy questions arise if third parties can read or tamper with sensor data. In the future, we envision wireless sensor networks being used for emergency and life-critical systems – and here the questions of security are foremost. This paper presents a review on set of Security Protocols for Sensor Networks, SPINS. The chief contributions of this paper are:

- Exploring the challenges for security in sensor networks.
- Designing and developing µTESLA (the "micro" version of

The Timed, Efficient, Streaming, Loss-tolerant Authentication Protocol), providing authenticated streaming broadcast._ Designing and developing SNEP (Secure Network Encryption Protocol) providing data confidentiality, two-party data authentication, and data freshness ,with low overhead. Designing and developing an authenticated routing protocol using SPINS building blocks.

II. SECURITY OF SENSORS

These constraints make it impractical to use the majority of the current secure algorithms, which were designed for powerful workstations. For example, the working memory of a sensor node is insufficient to even hold the variables (of sufficient length to ensure security) that are required in asymmetric cryptographic algorithms (e.g. RSA[1] , Diffie-Hellman[2]), let alone perform operations with them.

A particular challenge is broadcasting authenticated data to the entire sensor network. Current proposals for authenticated broadcast are impractical for sensor networks. Most proposals rely on asymmetric digital signatures for the authentication, which are impractical for multiple reasons (e.g. long signatures with high communication overhead of 50-1000 bytes per packet, very high overhead to create and verify the signature).TESLA is efficient for the Internet with regular desktop workstations, but does not scale down to our resource-starved sensor nodes. In this paper, we extend and adapt TESLA such that it becomes practical for broadcast authentication for sensor networks. We call this new protocol µTESLA .The measurements show that adding security to a highly resource-constrained sensor network is feasible. The paper studies an authenticated routing protocol and a two-party key agreement protocol, and demonstrates that our security building blocks greatly facilitate the implementation of a complete security solution for a sensor network.

III. KEY FEATURES FOR SENSOR NETWORK SECURITY

In this section, we formalize the security properties required by sensor networks, and show how they are directly applicable in a typical sensor network.

A. Data Confidentiality

A sensor network should not leak sensor readings to neighboring networks. In many applications (e.g. key distribution) nodes communicate highly sensitive data.

The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that only intended receivers possess, hence achieving confidentiality. Given

The observed communication patterns, we set up secure channels between nodes and base stations and later bootstrap other secure channels as necessary.

B. Data Authentication

Message authentication is important for many applications in sensor networks. Within the building sensor network, authentication is necessary for many administrative tasks (e.g. network reprogramming or controlling sensor node duty cycle). At the same time, an Adversary can easily inject messages, so the receiver needs to make sure that the data used in any decision-making process originates from the correct source. Informally, data authentication allows a receiver to verify that the data really was sent by the claimed sender.

C. Data Integrity

In communication, data integrity ensures the receiver that the received data is not altered in transit by an adversary. In SPINS, we achieve data integrity through data authentication, which is a stronger property. Data Freshness Given that all sensor networks stream some forms of time varying measurements, it is not enough to guarantee confidentiality and authentication; we also must ensure each message is fresh.

IV. SPINS SECURITY BUILDING BLOCKS

To achieve the security requirements we established in Section 3 we have designed and implemented two security building blocks: SNEP and µTESLA. SNEP provides data confidentiality, two-party data authentication, integrity, and freshness. µTESLA provides authentication for data broadcast. We bootstrap the security for both mechanisms with a shared secret key between each node ad the base station.

A. SNEP

SNEP provides a number of unique advantages. First, it has low communication overhead since it only adds 8 bytes per message. Second, like many cryptographic protocols it uses a counter, but we avoid transmitting the counter value by keeping state at both end points. Third, SNEP achieves even semantic security, a strong security property which prevents eavesdroppers from inferring the message content from the encrypted message. Finally, the same simple and efficient protocol also gives us data authentication, replay protection, and weak message freshness. Data confidentiality is one of the most basic security primitives and it is used in almost every security protocol.. The basic technique to achieve this is randomization: Before encrypting the message with a chaining encryption function (i.e. DES-CBC), the sender precedes the message with a random bit string. This prevents the attacker from inferring the plaintext of encrypted messages if it knows plaintext-cipher text pairs encrypted with the same key. However, sending the randomized data over the RF channel requires more energy. So we construct another cryptographic mechanism that achieves semantic security with no additional transmission overhead. Instead, we rely on a shared counter between the sender and the receiver for the block cipher in counter mode (CTR). Since the communicating parties share the counter and increment it after each block, the counter does not need to be sent with the message. To achieve two-party authentication and data integrity, we use a message authentication code.(MAC). The combination of these mechanisms forms our Sensor Network Encryption Protocol SNEP. The encrypted data has the following format where D is the data, the encryption key is and the counter is C.

- Semantic security: Since the counter value is incremented after each message, the same message is encrypted differently each time. The counter value is long enough that it never repeats within the lifetime of the node.
- Data authentication: If the MAC verifies correctly, a receiver can be assured that the message originated from the claimed sender.

B. µTESLA

Current proposals for authenticated broadcast are impractical for sensor networks. First, most proposals rely on asymmetric digital signatures for the authentication, which are impractical for multiple reasons. They require long signatures with high communication overhead of 50-1000 bytes per packet, very high overhead to create and verify the signature. Even previously proposed one-time signature schemes that are based on symmetric cryptography (one-way functions without trapdoors) have a high overhead: Gennaro and Rohatgi's broadcast signature based on Lamport's one-time signature[3] requires over 1 Kbyte of authentication information per packet[4] , and Rohatgi's improved k-time signature scheme requires over 300 bytes per packet[5].

The recently proposed TESLA protocol provides efficient authenticated broadcast[6][7]. However, TESLA is not designed for such limited computing environments as we encounter in sensor networks for three reasons.

First, TESLA authenticates the initial packet with a digital signature. Clearly, digital signatures are

too expensive to compute on our sensor nodes, since even fitting the code into the memory is a major challenge. For the same reason as we mention above, onetime signatures are a challenge to use on our nodes.

Standard TESLA has an overhead of approximately 24 bytes per packet. For networks connecting workstations this is usually not significant. Sensor nodes, however, send very small messages that are around 30 bytes long. It is simply impractical to disclose the TESLA key for the previous intervals with every packet: with 64 2In case the MAC does not match, the receiver can try out a fixed, small number of counter increments to recover from message loss.In case the optimistic re-synchronization fails, the two parties engage in a counter exchange protocol, which uses the strong freshness protocol described below. bit keys and MACs, the TESLA-related part of the packet would be constitute over 50% of the packet.

Finally, the one-way key chain does not fit into the memory of our sensor node. So pure TESLA is not practical for a node to broadcast authenticated data.

V. APPLICATIONS

In this section we demonstrate how we can build secure protocols out of the SPINS secure building blocks. First, we build an authenticated routing application, and second, a two-party key agreement protocol.

Using the µTESLA protocol, we developed a lightweight, authenticated ad hoc routing protocol that builds an authenticated routing topology. Ad hoc routing has been an active area of research[15] .However, none of these solutions offer authenticated routing messages. Hence it is potentially easy for a malicious user to take over the network by injecting erroneous, replaying old, or advertise incorrect routing information. The authenticated routing scheme we developed mitigates these problems. The routing scheme within our prototype network assumes bidirectional communication channels, i.e. if node A hears node B, then node B hears node A.

The route discovery depends on periodic broadcast of beacons. Every node, upon reception of a beacon packet, checks whether it has already received a beacon (which is a normal packet with a globally unique sender ID and current time at base station, protected by a MAC to ensure integrity and that the data is authentic) in the current epoch 5. If a node hears the beacon within the epoch, it does not take any further action. Otherwise, the node accepts the sender of the beacon as its parent to route towards the base station. Additionally, the node would repeat the beacon with the sender ID changed to itself. This route discovery resembles a distributed, breadth first search algorithm. However, in the above algorithm, the route discovery depends only on the receipt of route packet, not on its contents. It is easy for any node to claim to be a valid base station. We note that the µTESLA key disclosure packets can easily function as routing beacon. We accept only the sources of authenticated beacons as valid parents. Reception of a µTESLA packet guarantees that that packet originated at the base station, and that it is fresh. For each time interval, we accept as the parent the first node that sends a packet that is later successfully authenticated. Combining µTESLA key disclosure with the distribution of routing beacons allows us to charge the costs of the transmission of the keys to network maintenance, rather than the encryption system. This scheme leads to a lightweight authenticated routing protocol. Since each node accepts only the first authenticated packet as the one to use in routing, it is impossible for an attacker to reroute arbitrary links within the sensor network. Furthermore, each node can easily verify whether the parent forwarded the message: by our assumption of bidirectional connectivity.

VI. CONCLUSION

We have successfully demonstrated the feasibility of security subsystem for an extremely limited sensor network platform. We have identified useful security protocols for sensor networks: authenticated and confidential communication, and authenticated broadcast. Many elements of design are universal and apply easily to other sensor networks. Since our primitives are solely based on fast symmetric cryptography, and use no asymmetric algorithms, our building blocks are applicable to a wide variety of device configurations.

The computation costs of symmetric cryptography are low. Since the data authentication, freshness, and confidentiality properties require transmitting a mere 8 bytes per unit, it is feasible to guarantee these properties on a per packet basis, even with small 30 byte packets. It is difficult to improve on this scheme, as transmitting a MAC is fundamental to guaranteeing data authentication. A more powerful device would also allow for more basic modes of authentication..

REFERENCE

[1] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2):120–126 2008.

[2] W. Diffie and M. E. Hellman. New directions in cryptography. IEEE Trans. Inform. Theory, IT-22:644–654, November 2006.

[3] L. Lamport. Constructing digital signatures from a one-way function. Technical Report CSL-98, SRI International, October 2006.
[4] R. Gennaro and P. Rohatgi. How to sign digital streams. In Burt Kaliski, editor, Advances in Cryptology - Crypto '07, pages 180–197, Berlin, 2007.
[5] Pankaj Rohatgi. A compact and fast hybrid signature scheme for multicast packet authentication. In 6th ACM Conference on Computer and Communications Security, November 2005.
[6] Adrian Perrig, Ran Canetti, J.D. Tygar, and Dawn Song. Efficient authentication and signing of multicast streams overlossy channels. In IEEE Symposium on Security and Privacy May 2004.
[7] Adrian Perrig, Ran Canetti, Dawn Song, and J. D. Tygar. Efficient and secure source authentication for multicast. In Network and Distributed System Security Symposium, NDSS '04, February .2004.
[8] Alfred J. Menezes, Paul van Oorschot, and Scott Vanstone.Handbook of Applied Cryptography. CRC Press, 2007.
[9] Bruce Schneier. Applied Cryptography (Second Edition). John Wiley & Sons, 2006.
[10] Joan Daemen and Vincent Rijmen. AES proposal: Rijndael, March 2006.
[11] U. S. National Institute of Standards and Technology (NIST). Data Encryption Standard (DES). Draft Federal Information Processing Standards Publication 46-3 (FIPS PUB 46-3), January 2007
[12] David Wheeler and Roger Needham. TEA, a tiny encryption algorithm. http://www.ftp.cl.cam.ac.uk/ftp/
[13] Gideon Yuval. Reinventing the Travois: Encryption/MAC in 30 ROM bytes. In Proc. 4th Workshop on Fast Software Encryption, 2007
[14] R. L. Rivest. The RC5 encryption algorithm. Proc. 1st Workshop on Fast Software Encryption, pages 86–96, 2005..
[15] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, November 2003.
[16] Johnson and D.A. Maltz and J. Broch. The dynamic source routing protocol for mobile ad hoc networks (internet-draft). In Mobile Ad-hoc Network (MANET) Working Group, IETF, October 2006.

\