Veteran Identity-Based Signature for Wireless Sensor Network

Ashish Savita
CSE Department
Rama University, Kanpur
kitashishcse@gmail.com

C.Rama Krishna
CSE Department
NITTTR, Chandigarh
rkc_97@yahoo.com

Abstract: **Wireless sensor networks continue to grow, so does the need for effective Safety net. Sensor networks can interact with sensitive because Data and / or hostile work environment unattended, it is necessary these safety concerns have been addressed by the introduction of the system Temple design. However, because of the inherent lack of resources and computing; Security in sensor networks poses different challenges than traditional network / computer security. Currently enormous research potential Wireless sensor network security. On this type, belonging many researchers will benefit research in this area rent. With this in mind, we survey the major topics in the wireless sensor network security, and current many of the requirements of security barriers and sensors classified finally, the current attacks, and their corresponding list of protective measures.**

Keywords- Wireless Sensor, Security; Authentication Protocols, Network*;*

.

I. INTRODUCTION

Sensor networks refer to a heterogeneous system combining tiny sensors and actuators with general-purpose computing elements. Target tracking, surveillance, environmental monitoring etc. Today's sensors can monitor temperature, pressure, humidity soil makeup, vehicular movement, noise levels, lighting conditions, the presence or absence of certain kinds of objects or substances, mechanical stress levels on attached objects, and other properties.

We present an online/offline identity-based signature scheme for the Wireless Sensor Network (WSN). The authors argue that due to significant reduction in costs of computation and storage, the scheme is particularly suitable for the WSN[1] environment with severely constrained resources. One of the interesting features of the scheme is that it provides multi-time usage of the offline storage, which allows the signer to re-use the offline Pre-computed information in polynomial time, in contrast to onetime usage in all previous online/offline signature schemes. However, since sensors usually have very constrained resources in terms of computing, communication, memory, and battery power, providing authenticity in WSN poses different challenges than in traditional network/computer security .This requires lightweight and power-saving cryptographic algorithms to support WSN security. Symmetric-key cryptographic algorithms have been regarded as suitable tools for providing WSN with security. Contrary to this common belief, it has recently been reported that public-key cryptographic algorithms are feasible to be realized in WSNs and in fact practical .If appropriate algorithms are chosen [2]. A significant benefit one can obtain from using public-key cryptographic algorithms for WSN security is that this simplifies essential security services including key distribution/management and hence reduces transmission power due to less protocol overhead **[**3**]**.

II. SECURITY CHALLENGES

A. Hardware

One major challenge in a WSN is to produce *low cost* and *tiny* sensor nodes. There are an increasing number of small companies producing WSN hardware and the commercial situation can be compared to home computing in the 1970s. Many of the nodes are still in the research and development stage, particularly their software. Also inherent to sensor network adoption is the use of very low power methods for data acquisition.

In many applications, a WSN communicates with over a local area network or wide area network through a gateway. The Gateway acts as a bridge between the WSN and the other network. This enables data to be stored and processed by device with more resources, for example, in a remotely located server.

B. Software

Energy is the scarcest resource of WSN nodes, and it determines the lifetime of WSNs. WSNs are meant to be deployed in large numbers in various environments, including remote and hostile regions, where ad hoc communications are a key component. For this reason, algorithms and protocols need to address the following issues:

- Lifetime maximization
- Robustness and fault tolerance
- Self-configuration

Lifetime maximization: Energy/Power Consumption of the sensing device should be minimized and sensor nodes should be energy efficient since their limited energy resource determines their lifetime. To conserve power the node should shut off the radio power supply when not in use.Some of the important topics in WSN (Wireless Sensor Networks) software research are:

- Operating systems
- Security
- Mobility

Operating systems for wireless sensor network nodes are typically less complex than general-purpose operating systems. They more strongly resemble embedded systems, for two reasons. First, wireless sensor networks are typically deployed with a particular application in mind, rather than as a general platform. Second, a need for low costs and low power leads most wireless sensor nodes to have low-power microcontrollers ensuring that mechanisms such as virtual memory are either unnecessary or too expensive to implement.It is therefore possible to use embedded operating systems such as eCos or uC/OS for sensor networks. However, such operating systems are often designed with real-time properties.

TinyOS is perhaps the first [4] operating system specifically designed for wireless sensor networks. TinyOS is based on an event-driven programming model instead of multithreading. TinyOS programs are composed of event handlers and tasks with run-to-completion semantics. When an external event occurs, such as an incoming data packet or a sensor reading, TinyOS signals the appropriate event handler to handle the event. Event handlers can post tasks that are scheduled by the TinyOS kernel some time later. LiteOS is a newly developed OS for wireless sensor networks, which provides UNIX-like abstraction and support for the C programming language.Contiki is an OS which uses a simpler programming style in C while providing advances such as 6LoWPAN and Protothreads.RIOT implements a microkernel architecture. It provides multithreading with standard API and allows for development in C/C++. RIOT supports common IoT protocols such as 6LoWPAN, IPv6, RPL, TCP, and UDP.[5]

III. ONLINE COLLABORATIVE SENSOR DATA MANAGEMENT PLATFORMS

Online collaborative sensor data management platforms are on-line database services that allow sensor owners to register and connect their devices to feed data into an online database for storage and also allow developers to connect to the database and build their own applications based on that data. Examples include Xively and the Wikisensing platform. Such platforms simplify online collaboration between users over diverse data sets ranging from energy and environment data to that collected from transport services. Other services include allowing developers to embed real-time graphs & widgets in websites; analyze and process historical data pulled from the data feeds; send real-time alerts from any DataStream to control scripts, devices and environments. The architecture of the Wikisensing system is described in [8] describes the key components of such systems to include APIs and interfaces for online collaborators, a middleware containing the business logic needed for the sensor data management and processing and a storage model suitable for the efficient storage and retrieval of large volumes of data.

IV. OBSTACLES OF SENSOR SECURITY

A wireless sensor network is a special network which has many constraints compared to a traditional computer network. Due to these constraints it is difficult to directly employ the existing security approaches to the area of wireless sensor networks. Therefore, to develop useful security mechanisms while borrowing the ideas from the current security techniques, it is necessary to know and understand these constraints first **[9].**

A. *Very Limited Resources***:**

All security approaches require a certain amount of resources for the implementation, including data memory, co de space, and energy to power the sensor. However, currently these resources are very limited in a tiny wireless sensor.

a) *Limited Memory and Storage Space:*

A sensor is a tiny device with only a small amount of memory and storage space for the co de. In Order to build an effective security mechanism, it is necessary to limit the co de size of the security algorithm.

For example, one common sensor type (TelosB) has an 16-bit, 8 MHz RISC CPU with only 10K RAM, 48K program memory, and 1024K flash storage **[10].**

With such a limitation, the software built for the sensor must also be quite small. The total co de space of TinyOS, the de-facto standard operating system for wireless sensors, is approximately 4K [13], and the core Scheduler occupies only 178 bytes. Therefore, the co de size for the all security related co de must also be small.

b) Power Limitation Energy:

It is the biggest constraint to wireless sensor capabilities. We assume that once sensor no des are deployed in a sensor network, they cannot be easily replaced (high operating cost) or recharged (high cost of sensors). Therefore, the battery charge taken with them to the field must be conserved to extend the life

of the individual sensor node and the entire sensor network. When implementing a cryptographic function or proto col within a sensor node, the energy impact of the added security co de must be considered. When adding security to a sensor no de, we are interested in the impact that security has on the lifespan of a sensor (i.e., its battery life). The extra power consumed by sensor nodes due to security is related to the Processing required for security functions (e.g., encryption, decryption, signing data, verifying signatures), the energy required to transmit the Security related data or overhead (e.g., initialization vectors needed for encryption/ decryption), and the energy required to store security parameters in a secure manner (e.g., cryptographic key storage).

B. Unreliable Communication

Certainly, unreliable communication is another threat to sensor security. The security of the network relies heavily on a defined protocol, which in turn depends on communication.

a) *Unreliable Transfer*

Normally the packet-based routing of the sensor network is connectionless and thus inherently unreliable. Packets may get damaged due to channel errors or dropped at highly congested nodes. The result is lost or missing packets. Furthermore, the unreliable wireless communication channel also results in damaged packets. Higher channel error rate also forces the software developer to devote resources to error handling. More importantly, if the proto col lacks the appropriate error handling it is possible to lose critical security packets. This may include, for example, a cryptographic key.

b) Conflicts

Even if the channel is reliable, the communication may still be unreliable. This is due to the broadcast nature of the wireless sensor network. If packets meet in the middle of transfer, conflicts will occur and the transfer itself will fail. In a crowded (high density) sensor network, this can be a major problem. More details about the effect of wireless communication can be found at [12].

c) Latency

The multi-hop routing, network congestion, and no de processing can lead to greater latency in the network, thus making it Midcult to achieve synchronization among sensor no des. The synchronization issues can be critical to sensor security where the security Mechanism relies on critical event rep orts and cryptographic key distribution. Interested readers please refer to **[11]** on real-time communications in wireless sensor networks.

C. Unattended Operation

Depending on the function of the particular sensor network, the sensor nodes may be left unattended for long periods of time. There are three main caveats to unattended sensor nodes:

a) *Exposure to Physical Attacks*

The sensor may be deployed in an environment open to adversaries, bad weather, and so on. The likelihood that a sensor suffers a physical attack in such an environment is therefore much higher than the typical PCs, which is located in a Secure place and mainly faces attacks from a network.

b) Managed Remotely

Remote management of a sensor network makes it virtually impossible to detect physical tampering (i.e., through tamper proof seals) and physical maintenance issues (e.g., battery replacement). Perhaps the most extreme example of this is a sensor node used for remote reconnaissance missions behind enemy lines. In such a case, the node may not have any physical contact with friendly forces once deployed.

c) No Central Management Point

A sensor network should be a distributed network without a central management point. This will Increase the vitality of the sensor network. However, if designed incorrectly, it will make the network organization difficult, inefficient, and fragile. Perhaps most importantly, the longer that a sensor is left unattended the More likely that an adversary has compromised the node.

V. SECURITY REQUIREMENTS

The goal of security services in WSNs is to protect the information and resources from attacks and misbehavior. The security requirements in WSNs include:

• *Availability*,
which ensures that the desired network services are available even in the presence of *denial of service attacks*
• *Authorization*,
which ensures that only authorized sensors can be involved in providing information to network services
• *Authentication*,
which ensures that the communication from one node to another node is genuine, that is, a malicious node cannot masquerade as a trusted network node
• *Confidentiality*,
which ensures that a given message can-not be understood by anyone other than the desired recipients
• *Integrity*
, which ensures that a message sent from one node to another is not modified by malicious intermediate nodes
• *No repudiation*
, which denotes that a node cannot deny sending a message it has previously sent
• *Freshness*

, which implies that the data is recent and ensures that no adversary can replay old messages Moreover, as new sensors are deployed and old sensors fail, we suggest that forward and backward secrecy should also be considered:

• *Forward secrecy*

a sensor should not be able to read any future messages after it leaves the network.

• *Backward secrecy:*

A joining sensor should not be able to read any previously transmitted message.

The security services in WSNs are usually centered on cryptography. However, due to the constraints in WSNs, many already existing secure algorithms are not practical for use.

VI. IDENTITY BASED CRYPTOGRAPHY

One important issue that should be resolved in order to fully utilize public-key cryptography in WSN is to build up a public-key infrastructure (PKI) for WSN, which is to establish a trusted identity. However, as pointed out in the PKI for WSNs is not trivial to construct. Especially, distributing signed public-key certificates of numerous sensors could be difficult in many situations. We argue in this paper that at least for providing data authentication services, identity-based signature schemes are useful due to the feature that a signer does not have to hold a signed public-key certificate for other entities to verify signatures that the signer generates. Identity-based (ID-based) cryptography, introduced by Shamir **[6]** eliminates the necessity for checking the validity of certificates. In an ID-based cryptography, public key of each user is easily computable from a string corresponding to this user's identity (e.g. an email address, a telephone number, etc.).

A private key generator (PKG) then computes the private keys from a master secret for the users. This property avoids the requirement of using certificates and associates an implicit public key (user identity) to each user within the system. In the case of ID-based signature (IBS), verification takes only the identity together with the message and signature pair as input and executes the algorithm directly. This is different from the traditional public-key cryptography, whereas an additional certificate verification algorithm is needed, which is equivalent to the process of two signatures verification.

Identity-based cryptography could particularly be suitable for WSN. The absence of certificate eliminates the costly certificate verification process. In addition, when there is a new node added to the network, other nodes do not need to have its certificate in order to communicate in a secure and authenticated way. This can greatly reduce communication overhead and computation cost, which is a significant factor in the design of WSN. Recently, Tan et al. proposed an identity based encryption scheme for body sensor network (BSN), a network of sensors deployed on a person's body to collect physiological information.

VII. ONLINE/OFFLINE SIGNATURE

In order to further reduce the computational cost of signature generation, online/offline signature is preferable in WSN. The notion of Online/offline signatures was introduced by Even, Goldreich and Micali **[7]** .It performs the signature generation procedure in two phases. The first phase is performed offline (prior to the knowledge of the message to be signed), and the second phase is performed online (after knowing the message to be signed).

In WSN, the offline phase can be executed at the base station, while the online phase is to be executed in the WSN node. The online phase is typically very fast and hence can be executed efficiently even on a weak processor, such as a node in WSN.

Even, Goldreich and Micali proposed a general method for converting any signature scheme into an online/offline signature scheme. However, the method is impractical since it increases the size of the signature by a quadratic factor. Later, Shamir and Tauman proposed a new paradigm called "hash-sign-switch" for designing more efficient online/off-line signature schemes. Both schemes are in generic setting, and thus not actually very efficient or practical to be used.

VIII. ID-BASED ONLINE/OFFLIINE SIGNATURE

An ID-based online/offline signature scheme consists of five algorithms as follows:

1. *System Setup (SS):* Given a security parameter 1k, outputs a master secret key SK_{PKG} and system parameters SP.
2. *Key Extraction (KE):* Given a user's identity ID_i and a master secret key SK_{PKG}, outputs a corresponding private key $D_{ID}i$, i.e., $D_{ID}i \leftarrow KE(ID_i, SK_{PKG})$.
3. *Offline Signing (OffSign):* Given a signing key $D_{ID}i$ † and system parameter SP, outputs an offline signature S, i.e., $S \leftarrow$ Off Sign ($D_{ID}i$' SP).
4. *Online Signing (OnSign):* Given a message m and an offline signature S, outputs an online signature σ, i.e., $\sigma \leftarrow$ OnSign (m, S).
5. *Signature Verification (Ver):* Given a message m, user's identity IDi, signature σ and system parameters SP, returns 1 if the signature is valid and 0 if not. Namely, $0/1 \leftarrow$ Ver (m,IDi, σ, SP).

IX. RELATED WORKS

In their scheme, the signer needs to execute the offline phase every time when he wants to produce a signature. We call it "one-time" meaning the offline signature part can be used only once, and hence, it cannot be re-used. If we apply this one-time scheme into WSN, it becomes impractical since, assuming the offline phase is done at the base station, non-reusability of the storage implies that sensors need to go back to the base station every time for obtaining the next offline signature part.
We do not expect a node can execute such a heavy operation, which makes the signature scheme not appropriable for node-to-node signatures in WSNs.

X. CONCLUSION

In this paper, we proposed an efficient identity-based online/offline signature scheme, which is very suitable for wireless sensor networks. As compared with the existing four pairing-based schemes, our scheme has the lowest computational cost. . More importantly, our offline signing algorithm does not require any secret key information. It can be pre-computed by a PKG. The offline information can also be re-used.

REFERENCES

[1] Akyildiz I.F., Su W., Sankarasubramaniam Y., Cayirci E.: A survey on sensor networks. IEEE Commun. Mag40 (8), 102–114 (2002).

[2] Baek J., Tan H., Zhou J., Wong J.: Realizing stateful public key encryption in wireless sensor network. In: Proc. IFIP-SEC '08, pp. 95–108. Springer, Boston (2008).

[3] Gaubatz G., Kaps J.P., Oztruk E., Sunar, B.: State of the art in ultra-low power public key cryptography for wireless sensor networks. In: Proc. PerSec '05, pp. 146–150. IEEE ,2005.

[4] Oliver Hahm, Emmanuel Bacilli, Mesut Günes, Matthias Wählisch, Thomas C. Schmidt, RIOT OS: Towards an OS for the Internet of Things, In: Proc. of the 32nd IEEE INFOCOM. Poster Session, Piscataway, NJ, USA: IEEE Press, 2013.

[5] Shamir A., Tauman Y.: Improved online/offline signature schemes. In: Proc. CRYPTO '01, vol. 2139. Lecture Notes in Computer Science, pp. 355–367, 2001.

[6] Even S., Goldreich O., Micali S.: On-line/off-line digital signatures. In: Proc. CRYPTO '89, vol. 2442. Lecture Notes in Computer Science, pp. 263–277, 1989.

[7] Silva, D.; Ghanem, M.; Guo, Y. (2012). "Wikisensing: An Online Collaborative Approach for Sensor Data Management". Sensors 12 (12): 13295. doi:10.3390/s121013295,2012.

[1] D. W. Carman, P. S. Kruse, and B. J. Matt. Constraints and approaches for distributed sensor network security. Technical Rep ort 00-010, NAI Labs, Network Associates, Inc., Glenwood o d, MD, 2000.

[2] http://www.xbow.com/wirelesshome.aspx, 2006.

[3] J. A. Stankovicet al. Real-time communication and coordination in embedded Sensor networks. Proceedings of the IEEE, 91(7):1002–1022, July 2003.

[4] I. F. Akyildiz, W. Su, Y. Ankara Subramanian, and E. Cayirci. A survey on sensor networks. IEEE Communications Magazine, 40(8):102–114, August 2002.

[5] Du, W., Deng, J., Han, Y. S., and Varshney, P. K., "A pairwise key pre-distribution scheme for wireless sensor networks", Proc. of the 10th ACM conference on Computer and communications security, 2003, pp.42-51.

[6] M.C. Gorantla and A. Saxena. An efficient Certificateless signature scheme 2005 International Conference on Computational Intelligence and Security, Xi'an, China, 2005; 110–116.