A Survey of Wireless Sensor Networks in Key Management Scheme

Raju M
PG Scholar
Department of CSE
CMS College of Engineering
Namakkal, Tamilnadu, India

Selvan M
Assistant Professor
Department of CSE
CMS College of Engineering
Namakkal, Tamilnadu, India

Lanitha B
Assistant Professor
Department of CSE
KGiSL Institute of Technology
Coimbatore, Tamilnadu, India

ABSTRACT
This paper presents an entire study of Key Management Schemes for wireless sensor networks. Key management schemes (KMS) suggested in the literature, it is difficult for a warning network designer to learn exactly that KMS best connects in a certain WSN application. This review, that the KMS plays a important propose in determining the security performance of a WSN network with given application requirements and also develop a technique that changes the network designers to choose probably the almost suitable KMS for a particular WSN network setting. Broad reviews on the application requirements and the properties of altered key management schemes resolve each other and deal the problems on the present clearly-of-the-art research on the KMS for homogeneous networks to provide solutions for demonstrating link-layer keys in a variety of WSN applications and scenarios.

Keywords
Sensor Networks, Key Management, Network Design, Security.

1. INTRODUCTION
Small-sized and power- constrained pervasive devices, called sensor nodes, physical information (temperature, humidity, and forth) accessible easily from any other application. Wireless sensor networks (WSN) help to bridge the gap among the real world computer systems and the Internet. Security is major challenges when implementing WSNs. Challenges in managing these distributed and pervasive sensor networks, particularly when numerous of these wireless-enabled and self-configurable devices that may definitely not fit in with the same network are deployed within the same supervision area. Characteristics of a WSN, it is difficult to setup a safe link-layer channel (based on pairwise key) among neighboring nodes. The internal design of different strategies varies substantially typically, allowing for different features or properties. Protocol for WSN applications are resorting to a revolutionary search that is really a very time consuming process. This paper conducts analysis of different KMS properties, highlighting the relationship among these properties and certain requirements of WSN applications. This analysis and a method (the sensekey tool), may highlight the significance of selecting the almost suitable KMS protocols for limited or critical contexts. Properties may be mapped to certain requirements of WSN applications. Network designer have choice of the very appropriate protocol for different WSN application scenarios for development. Paper expands the concepts delivered in a prior version [1], providing an overview of the existing KMS properties and protocols and addition uncovers open issues by examining the suitability and applicability of the specific clearly. Analysis may focus only on homogeneous sensor networks, may not consider those KMS protocols that try to take advantage of the existence of a powerful device. KMS protocol to declare a particular key management scheme for WSN application. Different KMS protocols and highlight the properties that define their overall behaviors. Present clearly-of-the-art research in KMS may provide a viable solution to the link-layer key management problem, and determine future research effort must be focused to design substantially better KMS protocols for real-world WSN applications.

2. SECURITY AND KEY MANAGEMENT
Wireless channels attacks by using various mechanisms such as for instance secure communication channels, secure protocols (Routing, aggregation, and time synchronization), context awareness and trust measures, secure location mechanisms [3]. Secure protocols may be the security primitives, such as for instance symmetric key cryptography (SKC) and public key cryptography (PKC). Secure communication channel at the link layer among several devices, providing confidentiality, integrity, and authentication. The fundamental of the wireless channels and the limited capabilities of the sensor nodes, it may be not as difficult for a knowledgeable antagonist to monitor as well as assume control of the behavior of an unprotected WSN [2]. Protection mechanisms is instance time-stamping. Instance time-stamping is possible to avoid external attacks like instance message injection, eavesdropping, and packet relaying. Each device that requires opening a secure channel, it is neighbours must share among them some security credentials and secret keys. KMS solve the subject of creating, distributing, and maintaining those secret keys. The present constraints (memory, computational capabilities, etc.) of sensor nodes discourage the usage of resource. Instance network size, nodes connectivity, energy spent in key setup processes that also effect a change in the style of a KMS. Every time a limited set of nodes have to communicate with another in a secure manner, a mechanism to produce and maintain group keys is necessary. KMS for sensor networks is unwise to count on centralized entities because of the distributive and self-configurable nature of the WSN networks. Secret key opening process is secure end-to-end channel among two nodes. Communication security using link-layer keys is the foundations for security assurance of

sensor networks, the left of the paper may concentrate on the schemes that establish the link-layer keys.

3. KMS FRAMEWORKS

One of the sensor network link-layer standards is IEEE 802.15.4.

Secret keys should be exchanged, it is important to make use of a KMS protocol.

A homogeneous sensor network, protocol is classified into four major frameworks:

1. Key pool framework,
2. Mathematical framework,
3. Negotiation framework, and
4. Public key framework.

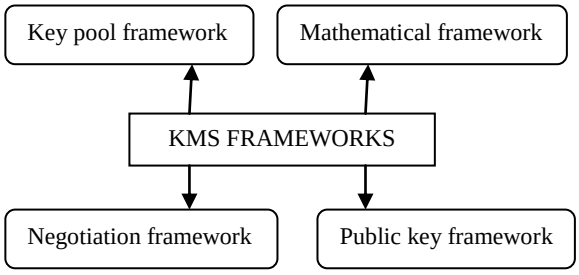

Fig 1: Key Management Scheme Framework

The key pool framework makes for a role in the pool framework, one of the KMS frameworks that ever been proposed far and quite easy [4].

- First, the network designer makes a key pool, a big set of pre-calculated secret keys.
- Second, earlier the network deployment each node is allotted a distinctive key chain, a tiny subset of the keys from the important thing pool (key pre-distribution).
- Third, succeeding the network deployment, the nodes exchange their identification (ID) variety of the keys from their key chains, searching for a typical shared secret key (shared-key discovery).
- Finally, two nodes do not share the same key; they try to look for a key path among them to negotiate a pairwise key (path-key establishment).

Framework is to make a restricted secure connectivity among the nodes, never mind the size of the network. KMS protocols proposed using mathematical algorithms (linear algebra, combinatory, and algebraic geometry) for calculating the pairwise keys of the nodes. KMS protocols cantered on linear algebra, the almost scheme may be the Bloom scheme [5]. KMS protocols developed cantered on algebraic geometry, probably the almost scheme is on the basis of the bi-variate polynomials [7]. Using a bi-variate polynomial f, every node A in a network has the capacity to obtain a pairwise key with another node y by solving f (A, y) and creation of pairwise keys among the nodes with no communication overhead. Mathematical framework protocols may be modified either to enhance certain properties (instance network resilience [13]) or to provide some properties (instance extensibility [14]). Drawbacks of mathematical framework protocols are solved partially by using the key pool paradigm [7]. Designs are frequently difficult to utilize, and it is challenging to make them scalable. Nodes negotiate their keys with their close neighbors right following the deployment of a network and this requires several steps in the pre-distribution. Protocols that generate their keys through mutual agreement belong to the negotiation framework. Other mechanisms and protocols (Guy Fawkes protocol [9]) ensure the authenticity of the peers in just about any stage of the network deployment. Framework is may organize a network into dynamic or static clusters [10]. Symmetric key agreement schemes of being WSN standards for instance ZIGBEE PRO [11], wirelessHART and ISA100.11a are fall within this family. Public key cryptography is used to securely bootstrap the pairwise key of two nodes throughout a public communication channel. Public key framework, two nodes need to exchange their public keys and several other information to effectively create their pairwise secret keys. Resource-constrained sensor nodes are able to use Public key cryptography through Elliptic curve cryptography (ECC) the total amount of memory required implementing the algorithm and the time/energy needed to complete the negotiation, substantially higher than other KMS frameworks. Advantage of the frameworks is deployment knowledge (knowledge of the ultimate node locations in the WSN field). This knowledge is building of each framework (keys inside the important thing pool, polynomials stored inside a node) be optimized. Communication overhead is another element in the development of any sensor network protocol. Optimization focused on reducing the number and/or size of the messages discovering or negotiating shared secret keys one of the peers [12]. Public key cryptography primitives are used only for re-keying where two nodes do not have a typical secret key.

4. MAPPING APPLICATION REQURIMENT TO KMS PROPERTIES

Some solutions available that may be used to bootstrap the link-layer keys of a particular WSN deployment and it is a hard task to learn that solution fit is best with a particular WSN. Security is the resilience of that design and communication channel be completely vulnerable against any attacks once an antagonist captures a really small group of nodes.

Table 1. KMS Properties

Property name		**Abbr**	**Description**
Memory footprint		Mm	ROM and RAM used for the protocol
Communication overhead		Cm	Number of messages exchanged between peers
Processing speed		Sp	Computational cost of the protocol
Network bootstrapping		Sec	Confidentiality of the bootstrap process
Network resilience		Rs	Resistance against stolen credentials
Connectivity	Global Conn.	GC	Existence of a key path between any node
	Local Conn.	LC	Existence of a shared secret between neighbour nodes
	Node Conn.	NC	Existence of a shared secret between any nodes
Scalability		Sc	Support for big networks
Extensibility		Ex	Capability of adding new nodes
Energy		En	Optimization of the energy usage

Avoid such situations, time-consuming trial-and-error experiments and exhaustive protocol analyses are the real solutions. Various sets of security and operational requirements, such as for instance their ability to assistance large networks (scalability), a secure shared-key discovery (confidentiality), and some protocols may appear to be better than the others. Existing KMS protocols are work to distribute secret keys in a wireless sensor network. Symmetric design approach [6] grants the creation of a pairwise key among any set of nodes that is requirement for applications with mobile entities. Constraints of a particular WSN scenario are possible to pick a KMS protocol the properties of that may meet requirements of the application form to be deployed. WSN protocols as instance routing they have to bind to some application-specific requirements. Routing protocols may need to use geographical information to discover the nodes and send packets. So information of the physical locations of the nodes is important. These properties that are shown in Table 1 were obtained the analysis of existing surveys. Properties are defined based on the security and operational requirements of sensor networks.

4.1 Memory Footprint (Mm):

Total amount of data memory in a KMS requires for storing the security credentials. Data memory in a KMS is useful for bootstrapping the whole infrastructure. Development viewpoint, need enough free memory for implementing different WSN applications and for storing temporary data. A warning node is usually constrained with involves to memory (< 4 kb of data memory and < 48 kb of instruction memory). Amount of instruction memory that a KMS is must implement the entire protocol. At more space should be produced designed for storing security credentials.

4.2 Communication Overhead (Cm):

Lots of WSN assumptions where the communication overhead should to be reduced to a small level. KMS protocols, the nodes must exchange information with their matches through communication channels to establish their pairwise keys. Protocols expect the exchange of a little bit of information and some another protocols need to undergo complex negotiation processes among matches.

4.3 Processing Speed (Sp):

WSN applications require setting up a secure channel among two previously unknown nodes. Communication overhead property, plenty time consumed in duration in sending and receiving messages through the wireless channels. Processing times for different key management schemes are associated with the communication overhead required by the KMS. Wireless sensor nodes usually are severely constrained with regards to their computing power. Fortunately there are many KMS protocols that are not very computationally intensive. Fast establishment of the communication link is essential. Reduction in the overhead may help shorten the hyperlink establishment time.

4.4 Network Bootstrapping (Sec):

Some protocols are not requiring switching sensitive information (ids of the nodes).The deployment environment is secure enough and confidentiality. When deployment area is open to public the information managed by the sensor nodes is important. Some protocols do think that the network is less clever to be at risk and it inclining to switch some secret information without the protection. The entire means of the keys distribution must be secure by it itself. The confidentiality of the key distribution process taking invest the early stage of the life-time should to be assured. Information exchange generally in protocols provide information about the key and antagonists may derive the key it itself from that information.

4.5 Network Resilience (Rs):

Network resilience is, the lower the chance is for a malicious attacker to manage an important area of the network. Network resilience indicates the capability to with stolen credentials. KMS protocols where nodes share pairwise keys only with their direct neighborhood. Network resilience is capturing some nodes is extremely low, and then the network resilience is not a critical element. Requirement for network resilience increases with the chance of a node being subverted by an adversary. Network is deployed is heavily protected.

4.6 Connectivity:

Connectivity is relates to the ability for two sensor nodes to share the exact same security credentials. Three main connectivity properties, as listed below.

- Global connectivity (GC): If the GC is 100%, this means that there's always a key path, a safe routing path, among any two nodes in the network.

Global connectivity is important property. Global connectivity property in many WSN scenarios all nodes are equally very important to providing the network services.

- Local connectivity (LC): If LC is 100%, then any node may securely keep in touch with any one of it is neighbors without negotiation.

Locations of the nodes inside the network are unknown. It is usually important to a high local connectivity; to assure nodes may have a way to setup a pairwise key using their neighborhood, producing the least overhead possible.

- Node connectivity (NC): If NC is 100%, then any node in the network may open a pairwise secure channel with any node.

Node connectivity is essential in a few WSN application scenarios where nodes are mobile, requiring a safe channel to be opened with any node in their neighborhood.

4.7 Scalability (Sc) and Extensibility (Ex):

All protocols provide scalability and extensibility. Scalability is not an issue for the WSN applications that require small several sensor nodes. When the network increases, the scalability becomes more important. Some protocols are not designed to control a network with a sizable amount of nodes as a result of memory constraints. The extensibility property is very important where there be hostile external entities and those applications that have to offer something for a comparatively long period of time. A KMS protocol provides scalability and it may support a WSN network with a sizable amount of nodes. KMS protocol is extensible and it allows the inclusion of new nodes following it is initial deployment.

4.8 Energy (En):

Establishment of pairwise security credentials among nodes may be an energy-consuming task and how much energy is consumed during the operation of a KMS protocol. An indicator node usually depends on batteries for powering it itself. The nodes are getting into unlimited energy sources, instance solar energy as well as normal power lines. Energy saving is not really a critical element. A WSN with a relatively short lifetime not need to take into account energy saving.

4.9 Identifying properties:

Let SK represent the properties of sensekey (Cm, GC, LC, NC, En, Ex, Mm, Rs, Sc, Sec, Sp) and
Let SR denote the identified requirements (Criticality, Remote Location, Isolation, Maintenance, Growth, and Performance).
x = Criticality ⟶ y = Sec & Rs
x = Connectivity ⟶ y = GC & LC & NC
x = Remote Location ⟶ y = Sec & Rs
x = Isolation ⟶ y = Sec & Rs
x = Maintenance ⟶ y = Ext
x = Growth ⟶ y = Sc
x = Performance ⟶ y = Cm

Table 2. List of different WSN application

Scenario Type		**Examples**	**Properties**
One-hop networks		Management of industrial machinery	LC
Simple networks	Simple	Office monitoring	GC, LC
	Medium	wine production industry	Sc, GC, LC
	Large	Wildfire detection	Sc, GC, LC, Cm
Mobile base station	Small	Vehicle tracking	GC, LC
Mobile and static nodes	Small	Monitoring of assisted-living residents	NC, GC, Cm, LC
	Medium	Hazards in safety-critical structures	NC, Sc, GC, Cm, LC
Short-lifetime networks	Small	Measuring noise pollution	LC, Cm, En, Ex

4.10 Main and Secondary properties:

The properties that may be used in sensekey are the next:
Main properties = Rs, GC, LC, NC, Ext, Sc
Secondary properties = Sec, Cm and deployment knowledge.

4.11 SenseKey and Results:

Determine the shape with the main and secondary properties. The consequences suggest that seven possible protocols be fitted to this particular scenario. Sensekey provides some other protocols that not have several the disadvantages of the conventional. The protocols are the standards developed for these forms of environments and ZIGBEE Smart Energy 2.0.

5. CONCULATION

Present protocols satisfy some of the needs of existing sensor network applications. Properties characterizing a KMS protocol and made an attempt to function those properties to the requirements of WSN applications. Proposed KMS selection method that may be used by network designers to decide that protocols and protecting their networks this scheme is used along with other methods about offer network designers with an improved overview on the suitability of specific protocols. This work has focused on key management schemes that establish link-layer secret keys among neighbors in sensor networks with homogeneous nodes. Other protocols have other specific properties like instance self-healing and currently employed in this direction. Key management in WSNs has been thoroughly studied and there are yet some problems that require more research.

6. REFERENCES

[1] R. Roman, J. Lopez, C. Alcaraz, H. Chen, SenseKey - Simplifying the Selection of Key Management Schemes for Sensor Networks, in: 5th International Symposium on Security and Multimodality in Pervasive Environments (SMPE 2011), Singapore, 789–794, 2011.

[2] Y. Zhou, Y. Fang, Y. Zhang, Securing Wireless Sensor Networks: a Survey, IEEE Communications Surveys & Tutorials 10 (3) (2008) 6–28.

[3] W. Zhu, Y. Xiang, J. Zhou, R. Deng, F. Bao, Secure Localization with Attack Detection in Wireless Sensor Networks, International Journal of Information Security 10 (2011) 155–171.

[4] L. Eschenauer, V. Gligor, A Key-Management Scheme for Distributed Sensor Networks, in: 9th ACM Conference on Computer and communications Security (CCS), Washington, DC, USA, 41–47, 2002.

[5] W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, A. Khalili, A Pairwise Key Predistribution Scheme for Wireless Sensor Networks, ACM Transactions on Information and System Security (TISSEC) 8 (2) (2005) 228–258.

[6] S. Camtepe, B. Yener, Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks, IEEE/ACM Transactions on Networking 15 (2) (2007) 346–358.

[7] D. Liu, P. Ning, Establishing Pairwise Keys in Distributed Sensor Networks, in: 10th ACM Conference on Computer and Communications Security (CCS), Washington D.C., USA, 52–61, 2003.

[8] R. Anderson, C. Haowen, A. Perrig, Key Infection: Smart Trust for Smart Dust, 20 in: 12th IEEE International Conference on Network Protocols (ICNP), Berlin, Germany, 206–215, 2004.

[9] A. Seshadri, M. Luk, A. Perrig, SAKE: Software Attestation for Key Establishment in Sensor Networks, in: Distributed Computing in Sensor Systems, vol. 5067 of LCNS, Springer Berlin / Heidelberg, 372–385, 2008.

[10] B. Panja, S.Madria, B. Bhargava, Energy and Communication Efficient Group Key Management Protocol for Hierarchical Sensor Networks, in: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, vol. 1,Taichung, Taiwan, 2006.

[11] ZigBee Alliance, ZigBee-08006r03: ZigBee-2007 Layer PICS and Stack Profiles (ZigBee-PRO), Revision 3, http://www. zigbee.org/, Retrieved on December 2011, 2008.

[12] D. S. H. Chan, A. Perrig, Random Key Predistribution Schemes for Sensor Networks, in: 2003 IEEE Symposium on Security and Privacy, IEEE, Oakland, USA, 197–213, 2003.

[13] W. Zhang, M. Tran, S. Zhu, G. Cao, A Random Perturbation-based Scheme for Pairwise Key Establishment in Sensor Networks, in: 8th ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc), Montreal, Canada, 90–99, 2007.

[14] M. Wen, Y. Zheng, W. Ye, K. Chen, W. Qiu, A Key Management Protocol with Robust Continuity for Sensor Networks, Computer Standards and Interfaces 31 (4) (2009) 642–647.