WIRELESS SENSOR NETWORKS USING HIGH SECURITY FOR NODES

J.VIJAYARAJ, R.RAJASEKAR,MA.MOHAMMED ANSAR ALI
ASSISTANT PROFESSOR DEPT OF INFORMATION TECHNOLOGY
SKP ENGINEERING COLLEGE

ABSTRACT

Rapid technological advances in the area of micro electro-mechanical systems (MEMS) have spurred the development of small inexpensive sensors capable of intelligent sensing. A significant amount of research has been done in the area of connecting large numbers of these sensors to create robust and scalable Wireless Sensor Networks (WSNs). Proposed applications for WSNs include habitat monitoring, battlefield surveillance, and security systems. Although individual sensor nodes have limited capabilities, WSNs aim to be energy efficient, self-organizing, scalable, and robust. A substantial amount of research has centered on meeting these challenges, but relatively little work has been done on security issues related to sensor networks. The resource scarcity, ad-hoc deployment, and immense scale of WSNs make secure communication a particularly challenging problem. Since the primary consideration for sensor networks is energy efficiency, security schemes must balance their security features against the communication and computational overhead required to implement them. This paper will describe the fundamental challenges in the emergent field of sensor network security and the initial approaches to solving them.

Keywords: Security, sensor networks

1. INTRODUCTION

Rapid technological advances in the areas of micro electro-mechanical systems and miniaturization have spurred the development of a new kind of network. This network is composed of small, inexpensive sensors capable of intelligent sensing. Much research has been done with the aim of connecting large numbers of these sensors to create robust and scalable Wireless Sensor Networks (WSNs) on the order of hundreds of thousands of devices. Proposed applications for WSNs include habitat monitoring, battlefield surveillance, and security systems. Sensor devices, also called motes or nodes, typically consist of a sensing unit, a transceiver unit, a processing unit, and a power source unit. Depending on the application, the sensing unit may monitor various types of data including acoustic, seismic, visual, and temperature data. The transceiver unit is a low-power radio capable of short range communication (tens of meters). The processing unit contains memory and a processor with severely limited size and speed. Wireless sensor motes are powered by a battery energy source which is not intended to be recharged. Designers hope to mass produce nodes for a very low cost per device (less than a dollar) and deploy them liberally as disposable devices. Communication usually consists of source nodes which sense the data and return it to sink nodes over multiple hops. Sink nodes may be ordinary sensor nodes or specialized base stations with greater resources. [1]

2. SENSOR SECURITY CHALLENGES

The nature of large, ad-hoc, wireless sensor networks presents significant challenges in designing security schemes. Five of the most pronounced challenges are described below. [2][9][10]

Wireless Medium The pervasive applications proposed for sensor networks necessitate wireless communication links. Furthermore, the ad-hoc deployment of sensor motes makes wired communication completely inappropriate. The wireless medium is inherently less secure because its broadcast nature makes eavesdropping simple. Any transmission can easily be intercepted, altered, or replayed by an adversary. The wireless medium allows an attacker to easily intercept valid packets and easily inject malicious ones. Although this problem is not unique to sensor networks, traditional solutions must be adapted to efficiently execute on sensor networks.

Ad-Hoc Deployment The ad-hoc nature of sensor networks means no structure can be statically defined beforehand. The network topology is always subject to changes due to node failure, addition, or mobility. Nodes nodes may fail or be replaced the network must support self-configuration. The ever-changing nature of sensor networks requires more robust designs for security techniques to cope with such dynamics.

Hostile Environment A third challenging factor is the hostile environment in which sensor nodes function. Motes face the possibility of destruction or (perhaps worse) capture by attackers. Since nodes may be in a hostile environment, attackers can easily gain physical access to the devices. The highly hostile environment represents a serious challenge for security researchers.

Resource Scarcity the extreme resource limitations of sensor devices pose considerable challenges to resource-hungry security mechanisms. Security mechanisms must give special effort to be communication efficient in order to be energy efficient.

Immense Scale Finally, the proposed scale of sensor networks poses a significant challenge for security mechanisms. Simply networking tens to hundreds of thousands of nodes has proven to be a substantial task. Providing security over such a network is equally challenging. Security mechanisms must be scalable to very large networks while maintaining high computation and communication efficiency.

3. ATTACKS & DEFENSES

Security goals for sensor networks include the same four primary objectives as conventional networks: availability, confidentiality, integrity, and authentication. Although sensor network security is characterized by the same properties as traditional network security, WSNs are vulnerable to new methods of exploitation. Karlof and Wagner identify two major classes of attackers: mote-class and laptop-class. Mote-class attackers are constrained to the CPU, power, bandwidth, and range limitations of the mote platform. Laptop-class attackers, however, may possess more powerful hardware such as a faster CPU, a larger battery, a high-power radio transmitter, or a sensitive antenna. This hardware allows a broader range of attacks which are more difficult to stop. This section examines the security attacks and corresponding defenses at each level of the network. [3][8][9]

Physical Layer

Attacks at the physical level include radio signal jamming and tampering with physical devices.

Jamming It is a well-known attack on wireless communication is simply interference with the radio frequencies used by a device's transceiver. It represents an attack on the availability of a network. Jamming is only different from normal radio propagation in that it is unwanted and disruptive, thus creating a denial-of-service condition. The degree of the jamming is determined by physical properties such as the available power, antenna design, obstacles, and height above. The standard defense against jamming involves the use of spread-spectrum. Spread-spectrum communication uses a wider band for radio transmission. Although this class of countermeasures has been extensively studied, [6] the inherent complexity involved in spread-spectrum systems is particularly costly for sensor motes. Frequency hopping requires greater power and financial cost, two scarce resources in sensor networks.

Prevention of denial of service attacks is a difficult task. Since most sensor networks currently use single frequency communication, Wood, Stankovic, and Son have proposed a Jammed Area Mapping (JAM) service which emphasizes detection and adaptation in response to jamming. Nodes in the affected area switch to low power mode. If spread spectrum techniques cannot be incorporated into motes, then detection algorithms such as JAM may be important in defending against jamming attacks.

Tampering A second problematic issue at the physical layer is the relative ease and potential harm of device tampering. This problem is exacerbated by the large-scale, ad-hoc, pervasive nature of sensor networks. Access to thousands of nodes spread over several kilometers cannot be completely controlled. Attackers may very well have greater physical access to nodes than the network administrator. Nodes may be captured, interrogated, and compromised without difficulty. While node destruction is undesirable, node compromise may be even more dangerous because of the cryptographic material compromised. The preferred solution is algorithmic: algorithms that reduce the effect a single key compromise has on the security of the entire network. For example, if each node shares a key with its immediate neighbors, much less is compromised than when all the nodes in the network share a common key. Although this software approach may be cheaper to implement, it does not provide exhaustive protection. [4]

Link Layer The link and media access control (MAC) layer handles neighbor-to-neighbor communication and channel arbitration. Like the physical layer, the link layer is particularly susceptible to denial of service attacks.

Collision If an adversary can generate a collision of even part of a transmission, he can disrupt the entire packet. A single bit error will cause a CRC mismatch and possibly require retransmission. In some MAC protocols, a corrupted ACK may cause exponential back-off and unnecessarily increase latency. The advantage, to the adversary, of this MAC level jamming over physical layer jamming is that much less energy is required to achieve the same effect: preventing devices from successfully transmitting packets.

Exhaustion Another malicious goal is the exhaustion of a network's battery power [Perrig, Stankovic, and Wagner 2004]. In addition to the previous types of attacks, exhaustion may also be induced by an interrogation attack. In the IEEE 802.11-based protocols, for example, Request To Send (RTS) and Clear To Send (CTS) packets are used to reserve bandwidth before data transmission. A compromised node could repeatedly send RTS packets in order to elicit CTS packets from a targeted neighbor, eventually consuming the battery power of both nodes [Perrig and Wagner 2004].

Unfairness A more subtle goal of the previously described attacks may be unfairness in the MAC layer. A compromised node can be altered to intermittently attack the network in such a way that induces unfairness in the priorities for granting medium access. This weak form of denial of service might, for example, increase latency so

that real-time protocols miss their deadlines. Another form of this attack could target one particular flow of data in order to suppress detection of some event. The use of small frames which prevent a node from capturing the channel for a long period of time has been proposed as a defense against this sort of attack.

Network Layer The network layer is responsible for routing packets across multiple nodes. Due to the ad-hoc nature of sensor networks, every node must assume routing responsibilities. WSNs are particularly vulnerable to routing attacks because every node is essentially a router. Karlof and Wagner have identified a variety of routing attacks and have shown them to be effective against every major sensor network routing protocol. Their classifications of attacks are summarized below and are followed by a general discussion of secure routing techniques. [11]

False Routing Information The most direct attack on routing is to spoof, alter, or replay routing information. This false information may allow adversaries to create routing loops, attract or repel traffic, shorten or extend route lengths, increase latency, and even partition the network. Not surprisingly, authentication is an important element in the security systems proposed for sensor networks.

Selective Forwarding Selective forwarding is a more subtle attack in which some packets are correctly forwarded but others are silently and intentionally dropped. A compromised node could be configured to drop all packets, creating a so-called black hole. If an attacker can get in the path of a desired data flow, he can selectively drop packets from that flow.

Sinkhole Attack In the sinkhole attack, a node spuriously advertises a very good route to a sink node (base station) in order to lure all nearby traffic to itself. Thus all traffic within some sphere of influence is drawn into the sinkhole centered at the compromised node. This attack enables the selective forwarding attack along with other attacks. An adversary mounting a laptop-class attack may actually provide the fastest route to a sink by using its greater range to reach the sink in a single hop. Whether the route is real or imagined, the attacker has successfully attracted a large amount of network traffic to pass through himself.

Sybil Attack The Sybil attack occurs when a single node claims to be other nodes in the network. Karlof and Wagner claim that this attack significantly reduces the effectiveness of "fault-tolerant schemes" such as distributed storage, multipath routing, and topology maintenance. Geographic routing protocols are particularly vulnerable to the Sybil attack since they are designed with the assumption that no node can be in two places at once. If a node lies about it location, it can significantly disrupt routing performance in geographic routing protocols.

Wormhole Attack The wormhole attack is used to convince two possibly distant nodes that they are neighbors so that the attacker can place himself on the route between them. Basically, the adversary tunnels messages from one part of the network to another through an out-of-bound channel available only to the attacker. Wormholes typically involve two colluding nodes. This sort of attack is likely to be used in combination with selective forwarding or eavesdropping.

Acknowledgement Spoofing The last routing attack Karlof and Wagner identify is the acknowledgement spoofing attack. Several routing protocols rely on link layer acknowledgements for determining next-hop reliability. If an adversary can respond for weak or dead nodes, he can deceive the sender about the strength of the link and effectively mount a selective forwarding attack. The artificial reinforcement allows the attacker to manipulate the routing through the weak or dead node.

There have been several approaches to defend against network layer attacks. Authentication and encryption are a first step, but more proactive techniques such as monitoring, probing, and transmitting redundant packets have also been suggested. Secure routing methods protect against some of previous attacks. Proposed techniques are described below.

Authentication & Encryption Link layer authentication and encryption protect against most outsider attacks on a sensor network routing protocol. Even a simple scheme which uses a globally shared key will prevent unauthorized nodes from joining the topology of the network. In addition to preventing selective forwarding and sinkhole attacks, authentication and encryption also make the Sybil attack impossible because nodes will not accept even one identity from the malicious node. SPINS and TinySec are two proposed solutions for link level encryption and authentication. They are discussed in greater detail in the next section. [7]

Monitoring A more active strategy for secure routing is for nodes to monitor their neighbors and watch for suspicious behavior. In this approach, nodes act as "watchdogs" to monitor the next hop transmission of the packet. In the event that misbehavior is detected, nodes will update routing information to avoid the compromised node.

Redundancy Redundancy is another strategy for secure routing. An inelegant approach, redundancy simply transmits a packet multiple times over different routes. Hopefully, at least one route is uncompromised and will correctly deliver the message to the destination. Despite its inefficiency, this method does increase the difficulty for an attacker to stop a data flow.

4. PROPOSED SOLUTIONS

While the majority of the research in sensor networks has focused on making them feasible and useful, a few researchers have proposed solutions to the security issues

discussed previously. Sensor network security mechanisms can be divided into two categories: communication protocols and key management architectures [5]. Communication protocols deal with the cryptographic algorithms used to achieve availability, confidentiality, integrity, and authentication

Communication Protocols Currently there have been two major secure communication protocols proposed for sensor networks: SPINS and TinySec. Both protocols work at the link level to provide message confidentiality, authentication, and integrity using symmetric cryptography. The limited memory and CPU speeds of sensor nodes almost completely exclude the use of asymmetric cryptography sensor networks. Perrig et al. claim that a sensor node is unable to even store the variables for 1024 bit RSA encryption, much less perform the expensive exponentiation operations on them.

SPINS SPINS (Security Protocols for Sensor Networks) is comprised of two link layer protocols: SNEP and µTELSA. SNEP (Secure Network Encryption Protocol) provides data confidentiality, two-party authentication, and data freshness. Now identify three patterns of communication in sensor networks: node to base station, base station to node, and base station to all nodes. SNEP handles the first two types, and µTELSA handles the last. In order to minimize computation and memory requirements, SNEP bases all symmetric cryptographic primitives (encryption, message authentication code, hash, and random number generator) on the same block cipher, RC5. Another design goal is to minimize communication overhead.

The MAC is recalculated upon reception and compared to the value in the transmission. To implement replay protection, SNEP requires a synchronized counter value at each node. The MAC is calculated using a secret key and the counter. As a result, out-of-sync packets will not be accepted. SPINS includes a counter exchange protocol for synchronizing counter values between two hosts. Although maintaining a synchronized counter adds significant overhead, it allows semantic security, a strong security property which assures that identical messages are encrypted differently each time they are encrypted.

µTELSA, the second part of SPINS, provides authenticated broadcast for sensor networks. The goal of µTELSA is to allow base stations to transmit authenticated broadcasts to all of the nodes while preventing a compromised node from forging messages from the sender. µTELSA uses symmetric mechanisms to create an asymmetric system using a loosely synchronized clock. Receivers buffer broadcast packets until they receive the decryption key which is disclosed once in a specified time interval (epoch). The keys are calculated using a one-way hash function (F) and are disclosed in the reverse order that they are generated. Once a node receives a key, it can apply the same hash function to calculate the keys for previous epochs and decrypt buffered packets. **Figure 1** illustrates this process.

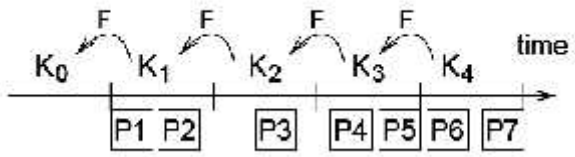

Figure 1: µTELSA key disclosure and computation. Each hash mark denotes an epoch. P1, P2,…P7 represent packets.

SPINS performs reasonably well according to its authors. Although key setup is expensive (4 ms), encrypting a 16 byte message and calculating its MAC only takes 2.5 ms. The limited bandwidth of the test platform, 10 kbps, allows time to perform key setup, encryption, and MAC calculation for every packet. The performance of µTELSA is bounded by the amount of buffer space available. Consequently, key disclosures must happen relatively frequently and must be reliably received.

The stated limitations of SPINS are that it does not completely deal with compromised nodes and it does not deal with denial-of-service attacks. The extremely limited storage space characteristic of sensors devices makes buffering particularly unattractive.

TinySec TinySec is a more recent solution to the sensor link layer security problem. The TinySec protocol provides access control, message integrity, and message confidentiality. TinySec explicitly omits replay protection, recommending it be performed at the application layer. The designers of the protocol emphasized usability and transparency in hopes of increasing TinySec's adoption. To this end, TinySec has been incorporated into the official release of TinyOS, the small, event-driven operating system designed for sensor motes. Unlike SPINS, TinySec has been fully implemented and exhibits promising performance. Encryption and authentication can be performed in software with only 10% energy overhead and 8% increased latency.

TinySec operates in two modes: authenticated encryption (TinySec-AE) and authentication only (TinySec-Auth). Like SPINS, TinySec implements authentication and integrity by the use of message authentication codes (MACs).

The performance of TinySec has proven that sensor network security can be efficiently done in software. TinySec requires 728 bytes of RAM and 7146 bytes of program space. The energy overhead imposed by TinySec is 3% for TinySec-Auth and 10% for TinySec-AE. The extra computation increases the time to transmit a packet 1.6% for TinySec-Auth and 7.9% for TinySec-AE. The energy, bandwidth, and latency of TinySec are all less than 10% and due almost entirely to the increased packet length. Not surprisingly, TinySec is being used by several other research projects throughout the country. With its impressive performance and ease of use, TinySec is the best sensor network security communication protocol to date.

Key Management Architectures Despite TinySec's merits as a communication protocol, it does not even attempt to solve the issue of key management. Key management handles the generation and secure distribution of cryptographic keys as well as techniques to protect the network from lost keys. This problem is also referred to as the bootstrapping problem since keys must be bootstrapped to devices in order to initiate a secure infrastructure. A variety of strategies exist for accomplishing this task. Some of the major approaches are summarized below.

LEAP The efficiency and speed of symmetric algorithms are well suited to sensor nodes and have been the default choice for sensor network designers. Most symmetric schemes require keys be loaded onto devices before deployment. Using a different key for every link provides the best security against compromised nodes but is incompatible with the basic nature of sensor networks. Sensor networks rely on data aggregation and in-network processing to increase network efficiency. Nodes along the path consolidate data to reduce the overall number of messages in the network. This cannot take place if messages are encrypted. In an effort to balance these two extremes, LEAP utilizes four types of keys for different security levels. LEAP supports an individual key shared only with the base station, a pair wise key shared with another sensor node, a cluster key shared with multiple neighboring nodes, and a group key shared by all the nodes in the network. The advantage of LEAP is that it supports in-network processing while minimizing the security impact of a compromised node to the node's immediate neighbors. LEAP provides a key for every need. This property offers convenience at the cost of storage space and complexity, neither of which are abundantly available to sensor nodes.

LKHW Another approach to key management is to use a hierarchy to store keys. Pietro et al. propose a scheme based on Logical Key Hierarchy (LKH) built on top of directed diffusion. Directed diffusion is a data-centric routing protocol that uses exploratory flooding to find the best path to send events of interest. The extension of LKH over directed diffusion comprises the LKH Wireless (LKHW) protocol. LKHW is a secure multicast scheme that enforces backward and forward secrecy. New nodes cannot decrypt old traffic, and evicted nodes cannot decrypt future traffic. LKHW uses a tree structure to store keys. The root of the tree serves as the key distribution center (KDC), and each leaf represents a user. Each leaf stores the set of keys belonging to its direct ancestors up to the KDC. The reason for using a tree structure is to increase the efficiency of re-keying. Re-keying occurs whenever a node joins or leaves the group. The energy required for re-keying is shown to be approximately logarithmic to the group size[12].

Random Key Predistribution Another novel approach to key management is random key predistribution [Chan, Perrig, and Song 2003]. In this strategy, a random pool of keys from the key space is preloaded into each node. Two nodes must find a common key in their sets in order to communicate. A challenge-response protocol is used to verify that two nodes have a key in common. Chan, Perrig, and Song extend this basic idea to a multipath-reinforcement scheme that strengthens the security between two nodes by exploiting the security of other links. Their work culminates in a random-pairwise key scheme which enables node-to-node authentication and quorum-based revocation. The strongest aspect of this strategy is that it provides complete resilience against node capture – a captured node reveals no information about the rest of the network.

TinyPK Despite the fact that asymmetric cryptography has been almost universally considered to be too resource-intensive for use in sensor networks, there have been some efforts to adapt public cryptography techniques to sensor devices. To minimize calculations by the sensor motes, e=3 is used as the public exponent. Encryption simply requires cubing a 1024-bit number and taking its residue modulo a large prime number. Implementing a public-key system requires a modest amount of infrastructure including a Certificate Authority (CA). The CA's public key is preloaded onto each node and is used to verify messages from the CA. Despite the adaptations, TinyPK still performs slowly by current standards. Table 1 summarizes the operation times for RSA encryption at various key sizes. [12]

RSA Key Size	Time (sec)
512	3.8
768	8.0
1024	14.5

`Table 1: RSA encryption (exponentiation) times

It has been suggested using TinyPK as a method of authenticating external parties to the sensor network and moving the computationally expensive operations to the external device when possible. Although public key cryptography possesses many advantages in handling key management, it is currently infeasible for node-to-node communication in sensor networks. Perhaps asymmetric techniques will be viable on more powerful hardware of the future. Most researchers predict, however, that devices will ride Moore's Law down the price curve instead of increasing in speed. If this is the case, then algorithmic optimizations will be required for public-key systems.

5. CONCLUSION

Sensor networks hold the potential to significantly transform the way computing affects life. In order to reach this potential, however, secure communication must be achieved. The wireless, ad-hoc, resource-limited nature of sensor networks creates substantial challenges for researchers. At the physical layer, probable attacks include frequency jamming and device tampering, two techniques with known solutions but entailing greater financial cost. The link layer of sensor networks is also susceptible to denial of service attacks in the form of maliciously induced collisions and exhaustion attacks.

The network layer is particularly vulnerable since every node in a sensor network is a router. Although link layer encryption and authentication serve as a first layer of defense, maximum security can only be achieved by designing routing algorithms with security in mind. SPINS and TinySec satisfactorily address the issue of link layer encryption, authentication, and integrity but require key management architectures to be practical. Current key management solutions are not sufficiently adapted to the unique requirements of sensor networks. If sensor networks are to reach their potential, secure communication must exist. For sensor networking protocols to accomplish this task, security must be a chief design consideration not an afterthought.

6. REFERENCES

[1]. Chi-Chou Kao, Chia-Nan Yeh, and Yen-Tai Lai *"**low-energy cluster head selection for clustering communication protocols in wireless sensor network"*** , DOI:10.2316/Journal.202.2011.1.202-2801. Jan. 2011.

[2]. Albert Levi, Sinan Emre Taşçı, Young Jae Lee, Yong Jae Lee, and Murat Ergun, ***"Simple, extensible and flexible random key redistribution schemes for wireless sensor networks using reusable key pools "***, Journal of Intelligent Manufacturing, Volume 21, Number 5, 635-645, DOI: 10.1007/s10845-009-0256-z.

[3]. Hill, Jason, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. ***"System architecture directions for networked sensors."*** In Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS IX) (November 2000).

[4]. Bing Wu, Jianmin Chen, Jie Wu and Mihaela Cardei, ***"A Survey of Attacks and Countermeasures in Mobile Ad Hoc Networks "*** Wireless Network Security Signals and Communication Technology, 2007, Part II, 103-135, DOI: 10.1007/978-0-387-33112-6_5.

[5]. Panayiotis Kotzanikolaou1,*, Emmanouil Magkos2, Dimitrios Vergados1, Michalis Stefanidakis2, ***"Secure and practical key establishment for distributed sensor networks"*** 17 FEB 2009, DOI: 10.1002/sec.102.

[6]. A. K. Das, I Sengupta, Indian Inst. of Technol., Kharagpur. ***"An effective group-based key establishment scheme for large-scale wireless sensor networks using bivariate polynomials"***, Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 6-10 Jan. 2008, On page(s): 9 - 16, ISBN: 978-1-4244-1796-4.

[7]. Yun Zhou; Yuguang Fang; Yanchao Zhang, Univ. of Florida, Gainesville, FL **"Securing wireless sensor networks: a survey"** Communications Surveys & Tutorials, IEEE , Third Quarter 2008, Volume: 10, Issue:3, page(s): 6 – 28, ISSN: 1553-877X, 16 September 2008.

[8]. Karlof, Chris, Naveen Sastry, and David Wagner. *"**TinySec: A Link Layer Security Architecture for Wireless Sensor Network."*** Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys'04) (November 3 - 5, 2004).

[9]. A Habib, Islamabad ***, "Sensor network security issues at network layer"*** Advances in Space Technologies, 2008. ICAST 2008. 2nd International Conference on , 29-30 Nov. 2008.

[10]. Kashif Kifayat, Madjid Merabti, Qi Shi and David Llewellyn-Jones ***"Security in Wireless Sensor Networks "*** Handbook of Information and Communication Security, 2010, Part E, 513-552, DOI: 10.1007/978-3-642-04117-4_26.

[11]. Junqi Zhang, Vijay Varadharajan, Macquarie University, Australia***, "Wireless sensor network key management survey and taxonomy"*** Journal of Network and Computer Applications Volume 33, Issue 2, March 2010, Pages 63-75.

[12]. Roberto, Roberto Di, Pietro Luigi, V. Mancini ***"LKHW: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks"*** Nov (2008), IST-2001-34734, doi-10.1.1.58.3114.