A Survey on Security in Wireless Sensor Networks

Zhijun Li and Guang Gong
Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, Canada
leezj@engmail.uwaterloo.ca, ggong@calliope.uwaterloo.ca

Abstract

Advances in electronics and wireless communication technologies have enabled the development of large-scale wireless sensor networks (WSNs). There is numerous applications for wireless sensor networks, and security is vital for many of them. However, WSNs suffer from many constraints, including low computation capability, small memory, limited energy resources, susceptibility to physical capture, the lack of infrastructure, which impose unique security challenges and make innovative approaches desirable. In this paper we present a survey of security issues in WSNs, address the state of the art in research on sensor network security, and discuss some future directions for research.

1 Introduction

Wireless sensor networks (WSNs) are innovative large-scale wireless networks that consist of distributed, autonomous, low-power, low-cost, small-size devices using sensors to cooperatively collect information through infrastructureless ad-hoc wireless network. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control.

Security plays a fundamental role in many wireless sensor network applications. Because sensor networks pose unique challenges, security techniques used in conventional networks cannot be directly applied to WSNs because of its unique characteristics. First, sensor nodes are very sensitive of production cost since sensor networks consist of a large number of sensor nodes. Akyildiz *et al.* [2] argued that the cost of a sensor node should be much less than one dollar in order for sensor networks to be feasible. Therefore, most sensor nodes are resource restrained in terms of energy, memory, computation, and communication capabilities. Normally sensor nodes are powered by batteries, and recharging batteries are infeasible in many circumstances. Energy consumption becomes a key consideration for most sensor network protocols. Second, Sensor nodes may be deployed in public hostile locations, which make sensor nodes vulnerable to physical attacks by adversaries. Generally, adversaries are assumed to be able to undetectably take control of a sensor node and extract all secret data in the node. Furthermore, the scale of sensor networks is considerably large, and the network topology is dynamically adjusted, because some nodes may die out of running out of energy or failure, and new nodes may join the network to maintain desirable functionality. At last, sensor networks use insecure wireless communication channel and lack infrastructure. As a result, existing security mechanisms are inadequate, and new approaches are desired.

1.1 Security Goals

Wireless sensor networks are vulnerable to many attacks because of broadcast nature of transmission medium, resource limitation on sensor nodes and uncontrolled environments where they are left unattended. Similar to other communication systems, WSNs have the following general security goals:

- *Confidentiality*: protecting secret information from unauthorized entities
- *Integrity*: ensuring message has not been altered by malicious nodes
- *Data Origin Authentication*: authenticating the source of message;
- *Entity Authentication*: authenticating the user/node/base-station is indeed the entity whom it claims to be
- *Access control*: restricting access to resources to privileged entities
- *Availability*: ensuring desired service may be available whenever required

In addition, WSNs have following specific security objects:

- *Forward secrecy*: preventing a node from decrypting any future secret messages after it leaves the network
- *Backward secrecy*: preventing a joining node from decrypting any previously transmitted secret message
- *Survivability*: providing a certain level of service in the presence of failures and/or attacks
- *Freshness*: ensuring that the data is recent and no adversary can replay old messages
- *Scalability*: supporting a great number of nodes
- *Efficiency*: storage, processing and communication limitations on sensor nodes must be considered

1.2 Applications

There are extensive applications of wireless sensor networks [4, 34, 14], such as Great Duck (bird observation on Great Duck island), Cattle Herding, Bathymetry, ZebraNet, Glacier Monitoring, Ocean Water Monitoring, Cold Chain Management, Grape monitoring, Rescue of Avalanche Victims, Vital Sign Monitoring, Power monitoring, Parts Assembly, Tracking Military Vehicles, and Self-healing Mine Field and Sniper Localization. According to areas of deployment WSN applications can be categorized as the following fields: military, environmental, industrial, location oriented, public safety oriented, automotive, airport oriented, agricultural, emergency handling, medical and oceanic.

Military and medical solutions are two of the most security-oriented application fields of wireless sensor networks. Military sensing networks are designed to detect and gain as much information as possible about enemy movements, explosions, and other phenomena. Typically, wireless sensor nodes are integrated with military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting systems. Examples of military wireless sensor network applications are battlefield surveillance, guidance systems for intelligent missiles, detection of attacks by weapons of mass destruction such as nuclear, biological, or chemical, and other monitoring applications. Due to the nature of the military, it is apparent that those applications could not be mounted without appropriate security assurance.

Recently, many medical systems are equipped with a large number of tiny, non-invasive sensors, located on or close to the patient's body, for health monitoring purposes. Such systems are being designed to measure diverse physiological values including Blood Pressure, Electrocardiogram, Blood Oxygen level, activity recognition, etc., and are available in many different forms, including wrist wearable, ambulatory devices and as part of biomedical smart clothes. The term of body sensor network (BSN) [73] is conied to represent this kind of applications. A number of intelligent physiological sensors is integrated into a wearable wireless body sensor network, which can be used for computer assisted rehabilitation and even early detection of medical conditions. Such applications imply that outpatients can be monitored from their homes, freeing space in hospital beds. As physiological patient data is legally required to be kept private, the implemented network must invoke a strong security protocol.

1.3 Network Models

Typically, a wireless sensor network consists of a few base stations and hundreds and thousands of sensor nodes. Sensor nodes are battery powered, equipped with sensors, data processing units of limited computation capability, limited memory space, and short-range radio communications. Base stations are the gateways to other networks, with powerful data processing/storage centers, or access points for human interface. In general, base stations are many orders of magnitude more powerful than sensor nodes. As a rule, base stations are assumed to be trusted and to be temper resistant. Sensor nodes are usually deployed at random in targeted fields. Each of these scattered sensor nodes has the capabilities to collect data and route data back to base station via infrastructureless wireless architecture. Base stations issue task commands, collect sensor readings, perform costly operations on behalf of sensor nodes and manage the network. WSNs are dynamic in the sense that radio range and network connectivity changes by time. Sensor nodes dies and new sensor nodes may be added to the network.

There are different settings about WSN architectures.

- **Hierarchical Model vs. Distributed Model**

  In some scenarios, sensor nodes are organized as hierarchical structure. They are grouped into a number of clusters controlled by part of nodes playing a particular role denoted as cluster headers. Member nodes are associated with a cluster via a one-hop or multi-hop link and these member nodes performs sensing and forwarding. After gathering or aggregating localized sensing information from their cluster members nodes, the cluster heads send packets to the base station. In contrast, there is no concept of cluster or group in distributed mode. All nodes play the same roles in the network. Once nodes are deployed, they scan their radio coverage area to figure out neighbors and manage to form fully distributed networks. Nodes collaborate to collect, aggregate and forward information.

- **Homogeneous Model vs. Heterogeneous Model**

  In homogeneous system model, all nodes are similar in terms of communication, computation, and storage capabilities. By contrast, heterogeneous wireless nodes can be equipped with different transport medium with different range of coverage and different specifications including CPU, memory, and peripherals to meet specific needs.

The remainder of this paper is organized as follows. In Section 2 we discuss various key distribution schemes. Section 3 describes attacks and countermeasures, intrusion detection and intrusion tolerance. Section 4 presents a number of authentication protocols. In Section 5, we introduce secure data aggregation protocols. And we address some privacy-protection protocols in Section 6. Finally, we conclude this paper in Section 7.

2 Key Distribution and Management

Security of large scale densely deployed and infrastructure-less wireless networks of resource limited sensor nodes calls for efficient key distribution and management mechanisms. This is one of the most popular research fields in the secure sensor networks, and plenty of approaches are proposed.

2.1 Straightforward Approaches

The simplest method of key distribution is to preload a single network-wide key into all nodes before deployment. Only one single key is stored in the nodes' memory and once deployed in the network, there is no need for a node to perform key discovery or key exchange since all the nodes in communication range can transfer messages using the key which they already share. On the other hand, this scheme suffers a severe drawback that compromise of a single node would cause compromise of the entire network through the shared key. Thus it fails in providing the basic secure requirement of a sensor network by making it easy for an adversary trying to attack.

An alternative key distribution scheme is fully pairwise keys scheme, i.e., every node in the sensor network shares a distinct key with every other node in the network. The main problem with this pairwise key scheme is its poor scalability. The number of keys that must be stored in each node is proportional to the total number of nodes in the network. Since sensor nodes are resource-constrained, this brings significant overhead which limits the scheme's applicability except for it can only be effectively used in smaller networks.

The method of Kerberos-like key distribution is popular in a lot of networks environment. In sensor networks, we can use a trusted, secure base station as an arbiter to provide link keys to sensor nodes. The sensor nodes authenticate themselves to the base station, after which the base station generates a link key and sends it to both parties securely. An example of such a protocol is SNEP, a part of the SPINS security infrastructure [56]. However, this kind of schemes suffers high energy consumption, which makes it inapplicable in most of sensor network applications.

2.2 Schemes based on Initial Trust Model

In LEAP [79], Zhu, Setia, and Jajodia proposed a key distribution scheme based on initial trust mode. Every node shares a common master key K and a keyed one-way hash function H. Upon deployed, nodes begin to discover all neighbor nodes and establish pairwise key using K and H. For example, the pairwise key between node u and v can be $H_{H_K(u)}(v)$ or $H_K(u||v)$. After establishing every pairwise key, all nodes eliminate the master key. LEAP assumes that T (the time necessary for an adversary to compromise a sensor node) is larger than the maximum time for nodes to complete the key distribution. If this assumption holds, LEAP is secure. However, sensors may be deployed in different phases, and new sensors need to be added when previously deployed sensors fail or when the capability of the existing network is turned to be insufficient. A major disadvantage of LEAP is not supporting multi-phase deployment—new nodes cannot create pairwise keys with previous nodes.

Proposal in [22] partially solves this problem by that there are many distinct master keys, each of which is for one phase. Every node u in phase i stores the master key K_i and all other $H_{K_j}(u)$, where $j > i$. Every two adjacent nodes in same phase can establish pairwise key like LEAP. If node u in phase i and node v in phase j want to establish pairwise key, supposing $i < j$, they both can compute $H_{H_{K_j}(u)}(v)$ and get the pairwise key. Every node only eliminates its phase mater key, and keeps the rest. A drawback of this scheme steps up that an adversary who comprises one node can duplicate many nodes which can establish pairwise keys with later phase nodes. And the number of phases has to be determined prior to first deployment.

Another initial-trust-like scheme is addressed in [3], and is further enhanced in [35]. They assume that adversaries can monitor only small portion of sensor nodes, due to randomly deployment of sensor networks. Initially, each pair of neighbor nodes only broadcast their pairwise key in plaintext. Afterwards, they can utilize multihop and multipath indirect secure links to exchange other secret data, which results in higher security.

2.3 Basic Random Probabilistic Key Distribution Scheme

Eschenauer and Gligor [27] first proposed random key probabilistic distribution schemes (EG scheme) based on random graph theory [63]. A random graph is a graph that is generated by starting with a set of n vertices and adding edges between them at random. In Erdös-Rényi model, a random graph is denoted by $G(n, p)$, in which every possible edge occurs independently with probability p. Erdös and Rényi [26] showed that, to achieve almost one hundred percent graph connectivity, every two vertices only need to have relatively lower probability P' of existence of direct link. More than often, sensor node are randomly deployed, and the number of nodes in a sensor network is massive. We may think of a wireless sensor network as a graph, nodes as vertices, and links as edges. Using random graph theory, we can theoretically analyze the connectivity of sensor networks and design WSN-specific security protocols. Since EG scheme, the random probabilistic approaches are gaining many attentions in secure wireless sensor networks, and many interesting protocols are proposed. Pietro *et al.* [57] questioned the realistic assumption of random graph model in WSNs, and proposed another geometric random model for WSNs. Wu and Stinson [72] further discussed these models and validated the use of the random graph model in computing the connectivity of WSNs. Nevertheless, random-graph-based analyses are still prevailing in protocols of WSNs.

EG scheme works as follows.

(1) Key initialization stage. Let m denote the number of distinct cryptographic keys that can be stored on a sensor node. Before sensor nodes are deployed, an offline trusted key distribution server generates a *key pool* of S random keys out of a total possible key space. For each node, m keys are randomly selected from the key pool and loaded into the node's memory. This set of m keys is called the node's *key ring*. The number of keys in the key pool, S, is determined satisfying that two random subsets of size m in S will share at least one key with probability p such that the whole network can achieve almost fully connectivity probability P_c.

(2) Directly shared key discovery stage. After deployment, each node tries to discover its neighbors with which it shares common keys. There are many ways to determine whether two nodes share common keys or not. The simplest way is to make the nodes broadcast their key identity lists to other nodes. If a node finds out that it shares at least a common key with a neighborhood node, it can use the first common key for secure communication. Alternatively, the set of keys in the key ring of a node could be bound with the node's ID via a pseudorandom function. In this case, each node only needs to broadcast its ID to its neighbors.

(3) Path key establishment stage: A link exists between two nodes only if they share a key, but the path key establishment stage facilitates provision of the link between two nodes when they do not share a common key. Nodes can set up path keys with nodes in their vicinity that they did not happen to share keys with in their key rings. If the graph is connected, a path can be found from a source node to its neighbor. The source node can then generate a path key and send it securely via the path to the target node.

2.4 Enhancement of Random Key Distribution

Chan, Perrig, and Song [13] introduced two variations of EG Scheme, i.e., q-composite random key predistribution and multipath key reinforcement. The q-composite random key predistribution scheme requires that two nodes have at least q common keys to set up a link and use all common keys instead of first one to establish pairwise key. As the number of key overlap between two nodes increases, it becomes harder for an adversary to break their communication link. At the same time, to maintain the probability that two nodes establish a link with q common keys, it is necessary to reduce the size of the key pool, which poses a possible security breach in the network as the adversary now has to compromise only a few nodes to gain a large portion of key pool. Therefore the challenge of the q-composite scheme is to choose an optimal value for q while ensuring that security is not sacrificed. However, the optimal value for q is strictly related to the number of nodes that adversaries may capture, which is dynamic and cannot be precisely determined while network parameters are designated. Therefore, the benefit of q-composite ($q > 1$) mode might be trivial. The multipath reinforcement scheme is similar to [3], using multipath indirect secure link to exchange secret data to offer good security with additional communication overhead, suitable for occasions where security is more of a concern than bandwidth or power drain.

Liu, Ning and Li [49] proposed a key predistribution scheme which combines EG scheme with polynomial-based key predistribution protocol in [8]. During the key initialization stage, setup server generates a set of bivariate t-degree symmetric polynomials $f_l(x, y) = \sum_{i,j=0}^{t} a_{ij}x^i y^j$, where $a_{ij} = a_{ji}, l \in [0, S-1]$ over a finite field F_q. For each node u, a subset of these polynomials are then picked up by the server and all $f(u, y)$ are computed and placed in the node u. In the key discovery stage sensor node u finds every adjacent node, e.g. node v, with which it shares the same original bivariate polynomial and then both nodes can establish a common pairwise key, because $f(u, v) = f(v, u)$. Du *et al.* [20] independently discovered a technique which is equivalent to Liu-Ning-Li's scheme. Huang *et al.* [33] further analyzed the performance of polynomial-based key predistribution scheme. In general, the security performance of this kind of scheme overweighs that of original EG scheme. On the other hand, it has to be noticed that operations in those schemes are in finite field F_q, where q is necessary to be the minimal prime integer greater than the length of secret key k, typically 2^{128}. Sometimes the costly finite field operations may be not very suitable for some extremely resource-constrained sensor nodes.

In general, sensor nodes are randomly scattered into targeted area, thus it is difficult to obtain deployment knowledge of nodes. As a matter of fact, it is a general assumption for characteristic of sensor networks. However, some proposals argued that some information on deployment knowledge is achievable if the deployment of nodes follows some particular pattern. For example, if sensor nodes are scattered by an airplane, these nodes might be grouped or placed in a particular order before deployment and, based on this pattern, an approximate knowledge of node positions can be acquired. In these scenarios, combined with random key distribution scheme, several schemes were proposed. Those nodes which are more likely to become neighbors are allocated more same source material such that bigger size of key pool still suffice to maintain the same connectivity of global network, which strengthen resistance against node capture. The major issue in those schemes is how to develop suitable node deployment models. Deployment knowledge in [48] is modeled using non-uniform probability density functions (pdfs), which assumes the positions of sensor nodes to be at certain areas. Generally nodes are deployed in groups; therefore the pdfs of the final resident points of all the sensors in a group is highly likely to be the same as the group of sensors deployed in a single deployment point. Other models are addressed in [21, 16]. A reasonable doubt to those schemes is whether or how precisely their models reflect the true node deployment.

Traynor *et al.* [65] proposed a random key distribution scheme based on the heterogeneous sensor network model. Instead of a homogeneous composition of nodes, this kind of network now consists of a mix of nodes with different capabilities and missions. Level 1 (L1) nodes are assumed to be very limited in terms of memory and processing capability and perform the task of data collection. Level 2 (L2) nodes have more memory, processing ability, and additional radios (e.g., 802.11). These nodes are equipped with additional keys and take on the role of routers and gateways between networks. In addition to tamper-resistant casings, L2 nodes are assumed to be equipped with a fast encryption/deletion algorithm to protect their supplementary keys from compromise if they are captured. Under this assumption, a scheme for the unbalanced distribution of keys throughout a wireless sensor network builds upon the EG scheme. Intuitively, with more powerful nodes in a sensor network, it definitely can achieve better security

2.5 Key Distribution Using Combinatorial Design

Çamtepe and Yener [10] first proposed deterministic methods using combinatorial design in key distribution of wireless sensor networks. They showed how to map from two classes of combinatorial designs— balanced incomplete block designs and generalized quadrangles—to obtain deterministic key distribution schemes. Chakrabarti, Maitra, and Roy [11] presented a randomized block merging strategy for key pre-distribution in WSNs. Wei and Wu [69] provided two key predistribution schemes using difference families and all k-subsets of a set. Lee and Stinson [45] discussed how to employ two types of transversal designs, the set of all linear polynomials and the set of quadratic polynomials, to improve the performance of key predistribution schemes by carefully choosing a certain class of set systems as "key ring spaces".

2.6 Group Key Distribution

Wireless sensor networks are inherently collaborative environments in which sensor nodes often communicate in groups that typically are dynamic. Efficient group key management schemes are demanded for secure communications under this collaborative model. General speaking, many traditional binary-tree-based group key management schemes and broadcast approaches, such as logical key hierarchy, one-way function chain tree, and subset-cover broadcast encryption, can be adapted into wireless sensor networks. Currently many proposed group key management schemes in WSNs are based on exclusion basis systems (EBS), presented by Eltoweissy *et al.* [23], which is a combinatorial formulation of the group key management problem that produces optimal results with respect to the parameters n, k and m, where n is the size of the group, k is the number of keys stored by each member, and m is the exact number of re-key messages to exclude one member. It is defined as follows.

Definition 1 (Exclusion Basis System) *Let n, k and m be positive integers, where $1 < k, m < n$. An Exclusion Basis System of dimension (n, k, m), denoted by $EBS(n, k, m)$, is a collection Γ of subsets of $[1, n]$ such that for every integer $t \in [1, n]$, the following two properties hold:*

(a) t is in at most k subsets of Γ

(b) There are exactly m subsets, say $A_1, A_2, \ldots, A_m$, in Γ such that $\bigcup_{i=1}^{m} A_i$. is $[1, n] - \{t\}$. (That is, each element t can be excluded by a union of exactly m subsets in Γ)

In a collusion-free environment, using EBS for key management guarantees forward and backward secrecy. Eltoweissy *et al.* [23] proved that there exists a positive solution to the $EBS(n, k, m)$ problem, where $k + m$ is equal to the total number of keys, if and only if $C(k+m, k) \geq n$. Apparently, we can tradeoff between the number of re-key messages and the number of keys known to each user. Moreover, this suggests that, in general, for arbitrarily large numbers of users, n, there are systems satisfying the properties of $EBS(n, k, m)$ with k and m smaller than the corresponding values of k and m for a binary tree system. However binary-tree-based approaches ensure that collusion between users is not possible, whereas an arbitrary EBS needs an external technique to safeguard security through collusion attacks.

Eltoweissy *et al.* [25, 24] applied EBS to sensor networks using specific network models. In GKIP [25], all sensor nodes in the network are anonymous and are preloaded with identical state information. The proposed scheme leverages a location-based virtual network infrastructure, combined with EBS. GKIP implements group keys at the granularity of a set of nodes. The set granularity allows for an efficient peer monitoring mechanism within a particular set that enables detecting nodes that infiltrate the network or exhibit suspicious behavior. LOCK [24], localized combinatorial keying, is another dynamic key management scheme based on the EBS scheme. The assumed network model consists of a three level hierarchy, i.e., base station, cluster heads, and sensor nodes. LOCK does not use location information in the generation of keys. When the nodes are initially released into the environment, they create a set of backup keys. These sets of backup keys are only shared with the base station, not the local cluster leader nodes. If a node is captured, other nodes are rekeyed locally so that the compromised node is unable to communicate with them. If a cluster leader is compromised, the base station initiates a rekeying on at the cluster head level. Also, nodes within the group governed by the compromised cluster leader rekey with the base station. In LOCK, if an adversary compromises any node, it does not have any effect on the operations of other nodes in other clusters.

In order to reduces the potential of collusion among compromised sensor nodes in standard EBS system, Younis, Ghumman, and Eltoweissy [25] proposed SHELL scheme , using node location information to compute keys with the help of clusters and gateways. SHELL gathers node locations after employment and uses this information for assigning keys. Nodes that are located closer to each other share a higher number of keys than nodes that are located longer distance from each other. The clusters in this scheme track key assignments but not the keys themselves. The actual keys are stored in the gateways of other clusters. SHELL exploits the physical proximity of nodes so that a node would share most keys with reachable nodes, and thus very few additional keys would be revealed when colluding.

2.7 Public Key Feasibility

The common perception of public key cryptography is that it is complex, slow, power hungry, and not at all suitable for use in ultra-low power environments like wireless sensor networks. Gaubatz, Kaps and Sunar [30] first challenged the basic assertion of public key cryptography infeasibility in sensor networks, which are based on a traditional software based approach. They propose a custom hardware assisted approach for which they claim that it makes public key cryptography available in such environments, provided they use the right selection of algorithms and associated parameters, careful optimization, and low-power design techniques.

In the family of public key algorithms, Elliptic Curve Cryptosystem (ECC) and Hyper Elliptic Curve Cryptosystem (HECC) are widely thought of as the best balance in terms of speed, memory requirement and security level. Malan, Welsh, and Smith [51] presented the first implementation of elliptic curve cryptography over $GF(2^p)$ for sensor networks based on the 8-bit, 7.3828MHz MICA2 mote. Although public-key infrastructure have been thought impractical, they argue, through analysis of their implementation for TinyOS of multiplication of points on elliptic curves, that public-key infrastructure is, in fact, viable for sensor network key distribution, even on the MICA2. They demonstrate that public keys can be generated within 34 seconds and that shared secrets can be distributed among nodes in a sensor network within the same, using just over 1 kilobyte of SRAM and 34 kilobytes of ROM.

Bertoni, Breveglieri, and Venturi [7] proposed two coprocessor architectures suitable for sensor networks: a $12K$ gate processor able to perform one $k * P$ operation (i.e., the ECC primitive) over the finite field $GF(2^{163})$ in $17.05ms$, consuming $1.1mJ$ of energy, and a $18.5K$ gate coprocessor performing the same operation in $14.68ms$ but consuming only $0.66mJ$.

Doyle *et al.* [19] examined the practicality of using efficient elliptic curve algorithms and identity-based encryption to deploy a secure sensor network infrastructure. They evaluated the potential for realizing this on low-power, long-life devices by measuring power consumption of the operations needed for key management in a sensor network and provided further evidence for the feasibility of the approach. However, their platform based on ARM7TDMI processor is considerably more powerful than any of the devices that are used in WSNs at the moment.

The applicable implementation of public-key crytography in typical sensor nodes platform comes with TinyECC [46], NanoECC [64], and TinyPBC [53]. TinyECC is very useful since it is a configurable library for ECC operations in wireless sensor networks. TinyECC provides a number of optimization switches, which can turn specific optimizations on or off according to developers' needs. Different combinations of the optimizations cost different execution time and resource consumptions, giving developers great flexibility in integrating TinyECC into sensor network applications. Liu and Ning presented the design, implementation, and evaluation of TinyECC on several common sensor platforms, including MICAz, Tmote Sky, and Imote2 in [46]. In NanoECC [64], point multiplication in a curve takes 1.27s at 7.3828MHz on MICA2 mote. Pairing-based cryptography (PBC) is an emerging field related to ECC which has been attracting the interest of international cryptography community, since it enables the design of original cryptographic schemes (such as, identity-base encryption) and makes well-known cryptographic protocols more efficient. TinyPBC is able to compute pairings, the PBC primitive, in about 5.5s on an ATmega128L clocked at 7.3828-MHz. Although it appears not very good, it does show the applicability of pairing-based cryptography in WSNs.

2.8 Discussion

Random key distribution approaches are prevailing at present. However, few analyses about communication overload in these schemes have been conducted. Especially, finding a secure path in a random graph is a NP-complete problem. Most of those schemes just ignored this problem. Rekey and perfect backward secrecy are also serious issues for those random predistribution schemes. From the practical point of view, group key distribution and public key based might be the tendency. The progress in efficiently implementation of ECC and HECC and advances in sensor hardware will make public key cryptosystem practicable in a few years.

3 Attacks and Countermeasures

Like any wireless ad hoc network, WSNs are suffering many different attacks. In this section, we introduce the major attacks to WSNs and countermeasures.

3.1 Secure Routing

Routing is a basic functionality of any network, there are various attacks and corresponding countermeasures for WSNs. Sybil attack and wormhole attack are two major routing attacks specifically for WSNs.

Karlof and Wagner [41] first considered routing security in wireless sensor networks systematically. They addressed security goals for routing in sensor networks, showed how attacks against ad-hoc and peer-to-peer networks can be adapted into powerful attacks against sensor networks, introduced two classes of novel attacks against sensor networks — sinkholes and HELLO floods, and analyzed the security of all major sensor network routing protocols. Sink is an alias of base station in sensor networks. In a sinkhole attack, the adversary's goal is to lure nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the adversary at the center. Because nodes along or near the path that packets follow have many opportunities to tamper with application data, sinkhole attacks can enable many other attacks. They also described crippling attacks against all of them and suggest countermeasures and design considerations.

Sybil Attack:

Sybil attack is a harmful threat to sensor networks, in which a malicious node illegally forges an unbounded number of identities. The Sybil attack can disrupt normal functioning of the sensor network, such as the multipath routing, used to explore the multiple disjoint paths between source-destination pairs. Douceur [18] first presented the Sybil attack problem in the peer-to-peer distributed systems. He pointed out that it could defeat the redundancy mechanisms of the distributed storage systems. Newsome *et al.* [52] analyzed the threat posed by the Sybil attack to wireless sensor networks. They established a classification of different types of the Sybil attack, proposed several techniques to defend against the Sybil attack, and analyzed their effectiveness quantitatively.

Zhang *et al.* [77] proposed a light-weight identity certificate method using to thwart Sybil attack. This method uses a two-level Merkle hash tree to create certificates. Each sensor node is pre-assigned a unique secret key to derive one-way key chains. An identity certificate is also distributed to each node, which associates the node's identity with its one-way key chain. To securely demonstrate its identity, a node first presents its identity certificate, and then proves that it possesses or matches the associated unique information. An extension of this method exploits node deployment knowledge to reduce the computational overhead at each node. However, the scalability problem of this method adversely affects its use in a large scale sensor network.

Yin and Madria [75] proposed a light-weight Sybil attack detection method based on a hierarchical architecture in sensor networks. This method also uses a two-level Merkle hash tree to create certificates. A high-level certificate allows a node's identity to be proved to other nodes that it needs to communicate with. A node creating a false identity will not be able to easily forge an identity certificate because the result of a identity verification calculation must match commitment, which is publicly known, according to the properties of the Merkle hash tree. The low-level identity certificate makes this proof specific to a single receiving node.

Wormhole Attack:

Since sensors use a radio channel to send information, malicious nodes can eavesdrop the packets, tunnel them to another location in the network, and retransmit them. This generates a false scenario that the original sender is in the neighborhood of the remote location. The tunneling procedure forms a wormhole attack. In [68], Wang and Bhargava proposed a mechanism, MDS-VOW, to detect wormholes in a sensor network. MDS-VOW first reconstructs the layout of the sensors using multi-dimensional scaling. Then MDS-VOW detects the wormhole by visualizing the anomalies introduced by the attack. The anomalies, which are caused by the fake connections through the wormhole, bend the reconstructed surface to pull the sensors that are faraway to each other. Through detecting the bending feature, the wormhole is located and the fake connections are identified. Yun *et al.* [76] proposed another countermeasure named WODEM against the wormhole attack. In WODEM, a few detector nodes equipped with location-aware devices and longer-lasting batteries detect wormholes, and normal sensor nodes are only required to forward control packets from the detector nodes. Then a pair of detectors can detect the wormhole attack between them.

3.2 DoS Attack

Denial of service (DoS) attack is a pervasive threat to most networks. Due to the characteristics of energy-sensitiveness and resource-limitedness, sensor networks are very vulnerable for DoS attack. Wood and Stankovic [71] explored various DoS attacks that may happen in every network layers of sensor networks. In [43], Mihui, Inshil, and Kijoon proposed a DoS detection method via practical entropy estimation on hierarchical sensor networks reflecting resource constraints of sensors. In order to enhance the accuracy of detection even in the various deployments of attack agents, they deployed hierarchically entropy estimators according to network topology, and a main estimator synthesizes localized computation. This entropy estimator is simplified by only multiplication calculation instead of logarithm, in addition to providing higher estimation precision of entropy compared to the conventional entropy estimation.

3.3 Node Clone Attack

Sensor nodes deployed in hostile environments are vulnerable to capture and compromise. An adversary may extract secret information from these sensors, clone and intelligently deploy them in the network to launch a variety of insider attacks. Chan and Perrig [12] catalog a number of attacks that can be made using replicated nodes .

Parno, Perrig, and Gligor [54] provided two probabilistic algorithms to detect node clone. They assume that every node is aware of it's geographic coordinates location and broadcasts the information to specific witnesses. Randomized multicast distributes node location information to randomly-selected witnesses, exploiting the birthday paradox to detect replicated nodes, while line-selected multicast uses the topology of the network to detect replication, i.e., in addition to witness nodes, the nodes within the multicast path check the node replication. Apparently, both of them are very communication-consuming.

SET proposed by Choi, Zhu, and Porta [15] is to detect node replication by computing set operations (intersection and union) of exclusive subsets in the network. First, SET securely forms exclusive unit subsets among one-hop neighbors in the network in a distributed way. This secure subset formation also provides the authentication of nodes' subset membership. SET then employs a tree structure to compute non-overlapped set operations and integrates interleaved authentication to prevent unauthorized falsification of subset information during forwarding. Randomization is used to further make the exclusive subset and tree formation unpredictable to an adversary.

Brooks *et al.* [9] propose a clone detection protocol in the context of random key predistribution (Section 2.3). The basic idea is that keys that are present on the cloned nodes are detected by looking at how often they are used to authenticate nodes in the network. First each node makes a counting Bloom filter of the keys it uses to communicate with neighboring nodes and appends a nonce. Then Bloom filter and nonce are transferred to base station, which will count the number of times each key is used in the network. Keys used above a threshold value are considered cloned.

Bekara and Laurent-Maknavicius [5] describe a deterministic node clone detection protocol based on the initial trust assumption (Section 2.2). They also suppose that nodes are not mobile. Therefore cloned nodes of former generations can not request for key establishment.

3.4 General Intrusion Detection and Intrusion Tolerance

Agah *et al.* [1] proposed an intrusion detection framework of sensor networks using game theory. They applied three different schemes for defense. The main concern in all three schemes is finding the most vulnerable node in a sensor network and protecting it. In the first scheme they formulated attack-defense problem as a two-player, nonzero-sum, non-cooperative game between an attacker and a sensor network. This game achieves Nash equilibrium [1] and thus leading to a defense strategy for the network. In the second scheme they used Markov Decision Process to predict the most vulnerable senor node. In the third scheme they used an intuitive metric (node's traffic) and protected the node with the highest value of this metric.

Based on the DESERT tool, which has been proposed for component-based software architectures, Inverardi, Mostarda, and Navarra [36] derived a framework that permits to dynamically enforce a set of properties of the sensors behavior. This is accomplished by an IDS specification that is automatically translated into few lines of code installed in the sensors. This realizes a distributed system that locally detects violation of the sensors interactions policies and is able to minimize the information sent among sensors in order to discover attacks over the network.

Deng, Han, and Mishra [17] described an INtrusion-tolerant routing protocol for wireless SEnsor NetworkS (INSENS). INSENS constructs forwarding tables at each node to facilitate communication between sensor nodes and a base station. It decreases computation, communication, storage, and bandwidth requirements at the sensor nodes at the expense of increased computation, communication, storage, and bandwidth requirements at the base station. An important property of INSENS is that while a malicious node may be able to compromise a small number of nodes in its vicinity, it cannot cause widespread damage in the network.

3.5 Discussion

The attacks, detection, tolerance and countermeasures are highly application-specific. To effectively resist and detect those attacks, security consideration must be taken in account in various protocols in sensor networks from scratch. Many current security protocols have good performance to resist those attacks. For instance, in [78], the authors demonstrated the efficacy of their LBKs scheme in counteracting several notorious attacks against sensor networks such as the Sybil attack, the identity replication attack, and wormhole and sinkhole attacks.

Time synchronization and sensor location are very important in many sensor network applications. There are considerable schemes regarding secure time synchronization and secure localization in wireless sensor networks. Most of them, however, are not cryptographic approaches. Many schemes in these two topics are discussed in [58].

4 Authentication

Authentication is one of the most important security primitives. Simply speaking, authentication is a mechanism by which some means is provided to guarantee that entities are who they claim to be, or that information has not been manipulated by unauthorized parties. In fact, authentication is specific to the security objective which one is trying to achieve. Examples of specific objectives are message authentication (data origin authentication), entity authentication (identification), access control, data integrity, non-repudiation, and key authentication. In this section, we cover two principal categories of authentication in wireless sensor networks, broadcast message authentication and entity authentication.

4.1 Broadcast Message Authentication

A broadcast message authentication scheme permits any targeted node to verify the authenticity of the source of broadcasted messages. This can be achieved using digital signatures if public key cryptography is used, or if only symmetric cryptography is used, by appending verifiable authentication data, consisting of multiple shared-secrets based message authentication codes (MAC). Due to the properties of sensor networks, broadcast authentication is more pervasive than one-to-one message authentication, and there are a number of schemes to achieve broad authentication in sensor networks.

μTESLA, proposed by Perrig *et al.* [56], is the "micro" version of TESLA (Timed Efficient Stream Loss-tolerant Authentication) [55]. It emulates asymmetry through a delayed disclosure of symmetric keys and serves as the broadcast authentication service of SNEP[56]. μTESLA requires that the base station and the nodes are loosely time synchronized, and that all nodes know an upper bound on the maximum synchronization error. For an authenticated packet to be sent, the base station computes a MAC on the packet with the key that is secret at that point of time. When a node receives a packet, it can confirm that the base station has not yet disclosed the corresponding MAC key, according to its loosely synchronized clock, maximum synchronization error and the time at which the keys are to be disclosed. The node stores the packet in its buffer. When the keys are to be disclosed, the base station broadcasts the key to all receivers. Each MAC key is a member of a key chain, which has been generated by a one-way function F. In order to generate this chain, the base station chooses the last key K_n of the chain randomly, and applies F repeatedly to compute all other keys: $K_i = F(K_{i+1}), i \in [1, n-1]$. The nodes can then verify the correctness of the key and use it to authenticate the packet stored in the buffer.

Liu and Ning [47] presented a series of techniques to extend the capabilities of μTESLA. The basic idea is to predetermine and broadcast the initial parameters required by μTESLA instead of unicast-based message transmission. In the simplest form, this extension distributes the μTESLA parameters during the initialization of the sensor nodes. To provide more flexibility, especially to prolong the lifetime of μTESLA without requiring a very long key chain, they introduced a multi-level key chain scheme, in which the higher-level key chains are used to authenticate the commitments of lower-level ones. To further improve the survivability of the scheme against message loss and DoS attacks, they used redundant message transmissions and random selection strategies to deal with the messages that distribute key chain commitments. The resulting scheme, which is named multi-level μTESLA, removes the requirement of unicast-based initial communication between base station and sensor nodes while keeping the nice properties of μTESLA. μTESLA is also extended by Liu *et al.* [50] to support multiuser scenario but the scheme assumes that each sensor node only interacts with a very limited number of users.

Schemes based on delayed key disclosure, like μTESLA, can suffer from DoS attack. In the subsequent interval when the message is in the buffer and the receiver waits on the disclosure time, an attacker can flood the network with arbitrary messages, claiming that they belong to the current time interval. Only in the next time interval can the nodes determine that these messages are not authentic. The use of public key cryptography would eliminate the need for such complicated protocols, increasing the security of the system and only requiring the public key of the base station to be embedded into all of nodes. Ren, *et al.* [61, 60] presented several public-key-based schemes to achieve immediate broadcast authentication and thus avoid the security vulnerability intrinsic to μTESLA-like schemes. Those schemes are built upon the unique integration of several cryptographic techniques, including the Bloom filter, the partial message recovery signature scheme and the Merkle hash tree.

Kondratieva and Seo [44] studied the problem of optimizing the authentication tree structure for sensor network environments. The procedure for finding the tree structure is formalized, in which the number of nodes with the longest authentication path length is made minimal. An algorithm for hash tree generation is introduced and it is proven that the proposed tree structure is optimal. Optimization of the authentication procedure is achieved by proposing an indexing scheme, supported by the least path protocol.

μTESLA-family is interesting and can be applied to some WSN application area. However, as public-key primitives become more and more feasible in WSNs, most WSNs would employ asymmetric approaches. The performance of current public-key-based broadcast authentication is far from satisfaction, and more researches are demanded.

4.2 Entity Authentication

Access control is a classical problem in many existing computer systems and applications. To achieve access control in wireless sensor networks, it is essential to authenticate the identities of users.

Benenson *et al.* [6] first proposed an entity authentication scheme based on elliptic curve cryptography, utilizing redundancy to withstand node capture. They used the public key infrastructure approach to issue a certificate for every user of the sensor networks through a certification authority (CA)and the CA has its own private/public key pair. A legitimate User's certificate is the user's public key signed by the CA. Each sensor node in WSN has the public key of the CA preloaded with which the node can verify the user's certificate. Then the user and the requested sensor node can carry out a standard challenge-response scheme with digital signature. But their scheme is designed for only one sensor node to authenticate the user, not the whole wireless sensor networks. And their scheme requires more overheads for encryption and signature verification than decryption and signing which makes a heavy burden for the WSN.

Jiang and Xu [39] presented a distributed entity authentication scheme in wireless sensor networks. It is based on the self-certified keys cryptosystem, which is modified to use elliptic curve cryptography to establish pair-wise keys for use in the user authentication scheme.

Wong *et al.* [70] proposed a dynamic strong-password based entity authentication scheme for wireless sensor networks. It allows legitimate users to query sensor data at any of the sensor nodes in an ad hoc manner, and imposes low computational overload and requires only simple operations, such as one-way hash function and exclusive-OR operations. Tseng, Jan, and Wang [67] enhanced Wong *et al.*'s scheme to resist the replay and forgery attacks. It also allows legitimate users to choose and change their passwords freely.

Tripathy and Nandi [66] used cellular automata based components to achieve entity authentication. Cellular automata is a dynamic system consists of a grid of identical finite state machines, whose states are updated synchronously at discrete time steps according to a local update rule. The proposed security component is to achieve threshold authentication and group key establishment as a suitable alternate to countermeasure the node capture attacks. Information are distributed among several nodes and user can determine the correct answer only if at least some certain correct responses are obtained.

All of these schemes above are based on conventional cryptography, symmetric or public-key. In fact, if this kind of cryptographic primitives are allowed, there are plenty of general entity authentication schemes that can be applied to sensor networks. The main issue left is how to distribute and manage secret-keys, passwords, public-key certificate effectively and efficiently in the environment of sensor networks. Some ultra-light entity authentication may be very useful in some wireless sensor networks.

5 Secure Data Aggregation

In many applications of wireless sensor networks, the base station is more interested in aggregated data than exact individual values from all sensors. By aggregating data, it is also greatly helpful to reduce the amount of data to be transmitted for conserving valuable energy. Indeed, current in-network aggregation schemes are beneficial to communication energy consumption but they are designed without considering possible security issues. Furthermore, wireless sensor networks are often designed with neighbor nodes sharing keys or with decryption at aggregator nodes. In either situation the potential for aggregator nodes to be physically compromised means that data confidentiality is at high risk. Therefore secure data aggregation is desirable where data can be aggregated without the need for decryption at aggregator nodes. Aggregation becomes especially challenging if end-to-end confidentiality between a source and a destination is required.

Hu and Evans [32] proposed a secure hop-by-hop data aggregation scheme. In their scheme, individual packets are aggregated in some pattern so that the base station can detect non-authorized inputs. On the other hand, their solution introduces a considerable communication overhead per packet. Moreover, they assumed that only leaf nodes under a tree-like network topology sense data, whereas the intermediate nodes do not have their own data readings, which is a little unrealistic, or at least too restricted. Jadia and Muthuria [37] extended the Hu-Evans scheme. Instead of relying on keys shared between the base station and sensor nodes for authentication, Jadia-Muthuria scheme make use of one-hop as well as two-hop pairwise keys. It is intended to replace the data validation step of the Hu-Evans scheme with some other mechanism that does not require unnecessary key reception by all nodes. Those two schemes are resistent to only a single inside malicious node and outside intruder devices.

Yang, *et al.* [74] proposed SDAP, a secure hop-by-hop data aggregation protocol for sensor networks, using the principles of divide-and-conquer and commitand-attest. In SDAP, a novel probabilistic grouping technique is utilized to dynamically partition the nodes in a tree topology into subtrees. A commitment-based hop-by-hop aggregation is conducted in each subtree to generate a group aggregate. The base station identifies the suspicious subtrees based on the set of group aggregates. Finally, each subtree under suspect participates in an attestation procedure to prove the correctness of its group aggregate. Feng *et al.* [28] proposed a family of secret perturbation-based schemes that protect sensed information confidentiality without disrupting the data aggregation.

Several secure aggregation algorithms have been proposed under the scenario that there are a certain class of nodes called aggregators. Przydatek, Song, and Perrig [59] proposed secure information aggregation (SIA) to identify forged aggregation values from all sensor nodes in a network. In SIA scheme, aggregators compute an aggregation result over the raw data together with a commitment to the data based on a Merkle-hash tree and send data to a trustable remote user, who later challenges the aggregators to verify the aggregation. They assumed that the bandwidth between a remote user and aggregators is a bottleneck in this scenario. Therefore the SIA scheme is intended to reduce this communication overhead while providing a mechanism to detect with high probability if aggregators are compromised.

Homomorphic encryption [29] is semantically-secure encryption which, in addition to standard guarantees, has additional properties, e.g. the sum of any two encrypted values is equal to the encrypted sum of the values. There are several efficient homomorphic cryptosystems, such as Unpadded RSA, El-Gamal, Goldwasser-Micali, Benaloh and Paillier [29]. Using homomorphic encryption, Kifayat *et al.* [42] presented the extended structure and density independent group based key management protocol (SADI-GKM) with the additional feature of secure data aggregation to provide better data confidentiality to every single node in a large scale wireless sensor network. Ren, Kim, and Park [62] also proposed a secure data aggregation scheme which supports end-to-end encryption using homomorphic encryption as well as hop-by-hop verification using ECC based MAC.

Current proposed secure data aggregation schemes are rather elementary and more practical schemes are demanded. It is worthwhile to pay more attention how to apply homomorphic encryption to secure aggregation effectively. In the meantime, it is of great help to focus on specific popular aggregation protocols of WSNs to design realistic secure aggregation.

6 Privacy

One challenge threatening the successful deployment of sensor networks is privacy. Although many privacy-related issues can be addressed by security mechanisms, one sensor network privacy issue that cannot be adequately addressed by network security is source-location privacy. Adversaries may use RF localization techniques to perform hop-by-hop traceback to the source sensor's location.

Kamat *et al.* [40] provided a formal model for the source-location privacy problem in sensor networks and examined the privacy characteristics of different sensor routing protocols. They inspected two popular classes of routing protocols: flooding-based routing protocols, and the routing protocols involving only a single path from the source to the sink. While investigating the privacy performance of routing protocols, they considered the tradeoffs between location-privacy and energy consumption. They argued that most of the current protocols cannot provide efficient source-location privacy while maintaining desirable system performance. They devised new techniques to enhance source-location privacy that augment these routing protocols. One of strategies, a technique called phantom routing, has been proven flexible and capable of protecting the source's location, while not incurring a noticeable increase in energy overhead. Further, they examined the effect of source mobility on location privacy. It is shown that even with the natural privacy amplification resulting from source mobility, the phantom routing techniques yield improved source-location privacy relative to other routing methods.

The model of k-anonymity is one of major mechanisms in protecting privacy. Gedik and Ling [31] described a personalized k-anonymity model for protecting location privacy against various privacy threats through location information sharing. First, they provided a unified privacy personalization framework to support location k-anonymity for a wide range of users with context-sensitive personalized privacy requirements. This framework enables each mobile node to specify the minimum level of anonymity it desires as well as the maximum temporal and spatial resolutions it is willing to tolerate when requesting for k-anonymity preserving location-based services (LBSs). Second, they devised an message perturbation engine which runs by the location protection broker on a trusted server and performed location anonymity on mobile users' LBS request messages, such as identity removal and spatio-temporal cloaking of location information. They developed a suite of spatio-temporal cloaking algorithms, called Clique Cloak algorithms, to provide personalized location k-anonymity, intending to avoid or reduce known location privacy threats before forwarding requests to LBS providers.

Jian *et al.* [38] proposed a location privacy routing protocol (LPR) that provides path diversity and protects receiver-location privacy in WSNs. Combining with fake packet injection, LPR is able to minimize the traffic direction information that an adversary can retrieve from eavesdropping. By making the directions of both incoming and outgoing traffic at a sensor node uniformly distributed, this system makes it hard for an adversary to perform analysis on locally gathered information and infer the direction to which the receiver locates.

7 Conclusion

As WSNs grow in application area and are used more frequently, the need for security in them becomes inevitable and vital. However, the inherent characteristics of WSNs incur constraints to of sensor nodes, such as limited energy, processing capability, and storage capacity, etc. These constraints make WSNs very different from traditional wireless networks. Consequently, many innovative security protocols and techniques have been developed to meet this challenge. In this paper, we outline security and privacy issues in sensor networks, address the state of the art in sensor network security, and discuss some future directions for research.

References

[1] A. Agah, S. K. Das, K. Basu, and M. Asadi. Intrusion Detection in Sensor Networks: a Non-Cooperative Game Approach. In *Proceedings of Third IEEE International Symposium on the Network Computing and Applications (NCA'04)*, pages 343 – 346. IEEE Computer Society, 2004.

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

[3] R. Anderson, H. Chan, and A. Perrig. Key Infection : Smart Trust for Smart Dust. In *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP'04)*, pages 206–215, 2004.

[4] T. Arampatzis, J. Lygeros, and S. Manesis. A Survey of Applications of Wireless Sensors and Wireless Sensor Networks. In *Proceedings of the 13th Mediterranean Conference on Control and Automation*, pages 719–724, 2005.

[5] C. Bekara and M. Laurent-Maknavicius. A New Protocol for Securing Wireless Sensor Networks against Nodes Replication Attacks. In *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMOB 2007)*, pages 59–59, 2007.

[6] Z. Benenson, N. Gedicke, and O. Raivio. Realizing robust user authentication in sensor networks. In *Real-World Wireless Sensor Networks (REALWSN)*, 2005.

[7] G. Bertoni, L. Breveglieri, and M. Venturi. ECC Hardware Coprocessors for 8-bit Systems and Power Consumption Considerations. In *Third International Conference on Information Technology: New Generations (ITNG 2006)*, pages 573–574, 2006.

[8] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-Secure Key Distribution for Dynamic Conferences. *Information and Computation*, 164(1):1–23, 1998.

[9] R. Brooks, P. Y. Govindaraju, M. Pirretti, N. Vijaykrishnan, and M. T. Kandemir. On the Detection of Clones in Sensor Networks Using Random Key Predistribution. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1246–1258, 2007.

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

[11] D. Chakrabarti, S. Maitra, and B. Roy. A Key Pre-distribution Scheme for WirelessSensor Networks: Merging Blocks in Combinatorial Design. In *Information Security*, pages 89–103. LNCS 3650, 2005.

[12] H. Chan and A. Perrig. Security and privacy in sensor networks. *Computer*, 36(10):103–105, 2003.

[13] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, 2003 IEEE Symposium on Security And Privacy, pages 197–213, Berkeley, CA, United States, 2003. Institute of Electrical and Electronics Engineers Inc.

[14] W. Y. Chang. Wireless Sensor Networks and Applications. In *Network-Centric Service-Oriented Enterprise*, pages 157–209. 2008.

[15] H. Choi, S. Zhu, and T. F. La Porta. SET: Detecting node clones in sensor networks. In *Third International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm 2007)*, pages 341–350, 2007.

[16] J. Y. Chun, Y. H. Kim, J. Lim, and D. H. Lee. Location-aware Random Pair-wise Keys Scheme forWireless Sensor Networks. In *Third International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SECPerU 2007)*, pages 31–36, 2007.

[17] J. Deng, R. Han, and S. Mishra. INSENS: Intrusion-Tolerant Routing in Wireless Sensor Networks. Technical Report Technical Report CU-CS-939-02, University of Colorado, Department of Computer Science, 2003.

[18] J. R. Douceur. The Sybil Attack. In *First International Workshop on Peer-to-peer Systems (IPTPS' 02)*, pages 251–260. LNCS 2429, 2002.

[19] B. Doyle, S. Bell, A. F. Smeaton, K. McCusker, and N. E. O'Connor. Security Considerations and Key Negotiation Techniques for Power Constrained Sensor Networks. *The Computer Journal*, 49(4):443–453, 2006.

[20] W. Du, J. Deng, Y. S. Han, P. Varshney, J. Katz, and A. Khalili. A Pairwise Key Predistribution Scheme for Wireless Sensor Networks. *ACM Transactions on Information and System Security (TISSEC)*, 2005.

[21] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A key predistribution scheme for sensor networks using deployment knowledge. *IEEE Transactions on Dependable and Secure Computing*, 3(1):62–77, 2006.

[22] B. Dutertre, S. Cheung, and J. Levy. Lightweight Key Management in Wireless Sensor Networks by Leveraging Initial Trust. Technical Report SRI-SDL-04-02, SRI International, April 6 2004.

[23] M. Eltoweissy, M. H. Heydari, L. Morales, and I. H. Sudborough. Combinatorial Optimization of Group Key Management. *Journal of Network and Systems Management*, 12(1):33–50, 2004.

[24] M. Eltoweissy, M. Moharrum, and R. Mukkamala. Dynamic key management in sensor networks. *IEEE Communications*, 44(4):122–130, 2006.

[25] M. Eltoweissy, A. Wadaa, S. Olariu, and L. Wilson. Group key management scheme for large-scale sensor networks. *Ad Hoc Networks*, 3(5):668–688, 2005.

[26] P. Erdös and A. Rényi. On the evolution of random graphs. *Bulletin of the Institute of International Statistics*, 38:343–347, 1961.

[27] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and Communications Security*, pages 41–47, Washington, DC, USA, 2002.

[28] T. Feng, C. Wang, W. Zhang, and L. Ruan. Confidentiality Protection for Distributed Sensor Data Aggregation. In *IEEE The 27th Conference on Computer Communications (INFOCOM 2008)*, pages 56–60, 2008.

[29] C. Fontaine and F. Galand. A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security*, 2007(1):1–15, 2007.

[30] G. Gaubatz, J.-P. Kaps, and B. Sunar. Public key cryptography in sensor networks-revisited. In *1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*, 2004.

[31] B. Gedik and L. Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *Proceedings of 25th IEEE International Conference onDistributed Computing Systems. (ICDCS 2005)*, pages 620–629, 2005.

[32] L. Hu and D. Evans. Secure aggregation for wireless networks. In *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, pages 384 – 391, 2003.

[33] D. Huang, M. Mehta, A. v. d. Liefvoort, and D. Medhi. Modeling Pairwise Key Establishment for Random Key Predistribution in Large-Scale Sensor Networks. *IEEE/ACM Transactions on Networking*, 15(5):1204 – 1215, 2007.

[34] Y.-M. Huang, M.-Y. Hsieh, and F. E. Sandnes. Wireless Sensor Networks and Applications. In *Sensors, Advancements in Modeling, Design Issues, Fabrication and Practical Applications*, pages 199–219. 2008.

[35] J.-B. Hwang, Y.-S. Hwang, and J.-W. Han. Consideration of efficient nearest node discovering mechanisms for Key Infection. In *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, volume 3, pages 1686–1689, 2006.

[36] P. Inverardi, L. Mostarda, and A. Navarra. Distributed IDSs for enhancing Security in Mobile Wireless Sensor Networks. In *20th International Conference on Advanced Information Networking and Applications (AINA 2006)*, volume 2, pages 116–120, 2006.

[37] P. Jadia and A. Mathuria. Efficient Secure Aggregation in Sensor Networks. In *High Performance Computing (HiPC 2004)*, pages 40–49. LNCS 3296, 2004.

[38] Y. Jian, S. Chen, Z. Zhang, and L. Zhang. Protecting Receiver-Location Privacy in Wireless Sensor Networks. In *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, pages 1955–1963, 2007.

[39] C. Jiang, B. Li, and H. Xu. An Efficient Scheme for User Authentication in Wireless Sensor Networks. In *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW '07)*, pages 438–442, 2007.

[40] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing Source-Location Privacy in Sensor Network Routing. In *Proceedings of 25th IEEE International Conference onDistributed Computing Systems. (ICDCS 2005)*, pages 599–608, 2005.

[41] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: attacks and countermeasures. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127, 2003.

[42] K. Kifayat, M. Merabti, Q. Shi, and D. Llewellyn-Jones. Applying Secure Data Aggregation techniques for a Structure and Density Independent Group Based Key Management Protocol. In *Third International Symposium on Information Assurance and Security (IAS 2007)*, pages 44–49, 2007.

[43] M. Kim, I. Doh, and K. Chae. Denial-of-Service(DoS) Detection through Practical Entropy Estimation on Hierarchical Sensor Networks. In *The 8th International Conference Advanced Communication Technology (ICACT 2006)*, volume 3, pages 1562–1566, 2006.

[44] V. Kondratieva and S.-W. Seo. Optimized Hash Tree for Authentication in Sensor Networks. *Communications Letters, IEEE*, 11(2):149–151, 2007.

[45] J. Lee and D. R. Stinson. On the Construction of Practical Key Predistribution Schemes for Distributed Sensor Networks Using Combinatorial Designs. *ACM Transactions on Information and System Security (TISSEC)*, 11(2):1–35, 2008.

[46] A. Liu and P. Ning. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In *International Conference on Information Processing in Sensor Networks (IPSN '08)*, pages 245–256, 2008.

[47] D. Liu and P. NIing. Multi-Level TESLA: Broadcast Authentication for Distributed Sensor Networks. In *Proceedings of the 10th Annual Network and Distributed Systems Security Symposium*, pages 263–276, 2003.

[48] D. Liu and P. Ning. LocationBased Pairwise Key Establishments for Static Sensor Networks. In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, Conference on Computer and Communications Security, pages 72 – 82, Fairfax, Virginia, 2003.

[49] D. Liu, P. Ning, and R. Li. Establishing Pairwise Keys in Distributed Sensor Networks. *ACM Transactions on Information and System Security (TISSEC)*, 8(1):41 – 77, 2005.

[50] D. Liu, P. Ning, S. Zhu, and S. Jajodia. Practical broadcast authentication in sensor networks. In *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2005)*, pages 118–129, 2005.

[51] D. J. Malan, M. Welsh, and M. D. Smith. A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON 2004)*, pages 71–80, 2004.

[52] J. Newsome, E. Shi, D. Song, and A. Perrig. The Sybil attack in sensor networks: Analysis and defenses. In *Third International Symposium on Information Processing in Sensor Networks, IPSN 2004*, pages 259–268, Monterey, CA, United States, 2004.

[53] L. B. Oliveira, M. Scott, J. Lopez, and R. Dahab. TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. In *Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on*, pages 173–180, 2008.

[54] B. Parno, A. Perrig, and V. Gligor. Distributed Detection of Node Replication Attacks in Sensor Networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 49 – 63, 2005.

[55] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. Efficient authentication and signing of multicast streams over lossy channels. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 56–73, Berkeley, CA, USA, 2000.

[56] A. Perrig, R. Szewczyk, V. W. D. Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 189–199, Rome Italy, 2001. IEEE.

[57] R. D. Pietro, L. V. Mancini, A. Mei, A. Panconesi, and J. Radhakrishnan. Connectivity properties of secure wireless sensor networks. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, Washington DC, USA, 2004. ACM.

[58] R. Poovendran, C. Wang, and S. Roy. *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks* . Springer Verlag, 2007.

[59] B. Przydatek, D. Song, and A. Perrig. SIA: Secure Information Aggregation in Sensor Networks. In *Proceedings of the first international conference on Embedded networked sensor systems*, pages 255 – 265, Los Angeles, California, USA, 2003.

[60] K. Ren, W. Lou, K. Zeng, and P. J. Moran. On Broadcast Authentication in Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 6(11):4136–4144, 2007.

[61] K. Ren, W. Lou, and Y. Zhang. Multi-user Broadcast Authentication in Wireless Sensor Networks. In *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '07)*, pages 223–232, 2007.

[62] S. Q. Ren, D. S. Kim, and J. S. Park. A Secure Data Aggregation Scheme for Wireless Sensor Networks. In *Frontiers of High Performance Computing and Networking ISPA 2007 Workshops*, pages 32–40. LNCS 4743, 2007.

[63] J. Spencer. *The Strange Logic of Random Graphs*, volume 22 of *Algorithms and Combinatorics*. Springer-Verlag, 2001.

[64] P. Szczechowiak, L. Oliveira, M. Scott, M. Collier, and R. Dahab. NanoECC: Testing the Limits of Elliptic Curve Cryptography in Sensor Networks. In *Wireless sensor networks*, pages 305–320. LNCS 4913, 2008.

[65] P. Traynor, R. Kumar, H. Choi, G. Cao, S. Zhu, and T. La Porta. Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. *IEEE Transactions on Mobile Computing*, 6(6):663–677, 2007.

[66] S. Tripathy and S. Nandi. Defense against outside attacks in wireless sensor networks. *Computer Communications*, 31(4):818–826, 2008.

[67] H.-R. Tseng, R.-H. Jan, and W. Yang. An Improved Dynamic User Authentication Scheme for Wireless Sensor Networks. In *IEEE Global Telecommunications Conference (GLOBECOM '07)*, pages 986–990, 2007.

[68] W. Wang and B. Bhargava. Visualization of Wormholes in Sensor Networks. In *Proceedings of the 2004 ACM workshop on Wireless security*, pages 51 – 60, Philadelphia, PA, USA, 2004.

[69] R. Wei and J. Wu. Product Construction of Key Distribution Schemes for Sensor Networks. In *Selected Areas in Cryptography*, pages 280–293. LNCS 3357, 2005.

[70] K. H. Wong, Y. Zheng, J. Cao, and S. Wang. A Dynamic User Authentication Scheme for Wireless Sensor Networks. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, pages 244–251, 2006.

[71] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002.

[72] J. Wu and D. R. Stinson. Minimum node degree and k-connectivity for key predistribution schemes and distributed sensor networks. In *Proceedings of the First ACM Conference on Wireless Network Security (WiSec'08)*, Alexandria, Virginia, USA, 2008.

[73] G.-Z. Yang and M. Yacoub. *Body Sensor Networks*. Springer, 2006.

[74] Y. Yang, X. Wang, S. Zhu, and G. Cao. A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 356 – 367, 2006.

[75] J. Yin and S. K. Madria. Sybil attack detection in a hierarchical sensor network. In *Third International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm 2007)*, pages 494–503, 2007.

[76] J.-H. Yun, I.-H. Kim, J.-H. Lim, and S.-W. Seo. WODEM: Wormhole Attack Defense Mechanism in Wireless Sensor Networks. In *Ubiquitous Convergence Technology (ICUCT 2006)*, pages 200–209. LNCS 4412, 2007.

[77] Q. Zhang, P. Wang, D. S. Reeves, and P. Ning. Defending against Sybil attacks in sensor networks. In *25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS 2005 Workshops)*, pages 185–191, 2005.

[78] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Location-based compromise-tolerant security mechanisms for wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 24(2):247–260, 2006.

[79] S. Zhu, S. Setia, and S. Jajodia. LEAP : efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of the 10th ACM conference on Computer and Communication Security (CCS' 03)*, pages 62–72, Washington D.C., 2003. ACM.