DEFEND AGAINST TOPOLOGICAL ATTACKS IN SENSOR NETWORKS

Yun Zhou and Yuguang Fang
Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611
Tel: (352)392-8576; Fax: (352)392-0044
Email: {yzufl@, fang@ece.}ufl.edu

ABSTRACT

Sensor networks are vulnerable to many active attacks due to the defects of the network protocols that are not designed carefully to involve security defenses at the beginning. Most of the attacks try to cause topological distortion by spoofing or replaying routing information. This paper proposes to use a location-based naming (LBN) mechanism for sensor nodes, in which location information is embedded into node identifier and acts as an inherent node characteristic to provide authentication service in local access control. When LBN is enforced, the impacts of many attacks to sensor network topology can be limited in a small area. A link layer authentication (LLA) scheme is also proposed to further decrease the impacts of those attacks. Our LBN and LLA can be combined and act as an efficient solution against a wide range of topological attacks in sensor networks.

INTRODUCTION

Sensor networks are vulnerable to malicious attacks in unattended and hostile environments such as battlefield surveillance and homeland security monitoring. Adversaries can easily eavesdrop messages transmitted over the air between nodes, or disable the entire network by launching physical attacks to sensor nodes or logical attacks to communication protocols [1], [2]. By using encryption, we may prevent eavesdropping attacks. However, an intelligent adversary may launch many active attacks by utilizing the defects in the network protocols which are not designed carefully to involve security defenses at the beginning. Karlof and Wagner [2] classified a series of attacks to sensor networks, most of which try to spoof or replay routing information to make two distant sensor nodes believe they are neighbors to each other. These attacks will cause serious topological distortion, which lead to the rapid deterioration of network performance.

The reason that those topological attacks can make effect is that they break the underlying assumption of routing protocols that all sensor nodes are cooperative and all routing information received are trustful. However, the assumption is not true in hostile environments. For example, a malicious node may impersonate other normal nodes in the network by changing its node ID, but the network routing protocol can not detect this spoofed node ID and may launch a new round of route discovering algorithm. The effect is that a route through the malicious node may be used as an "optimal" one in stead of the previous route, which leads to the failure of the routing protocol.

Thus, to guarantee the proper operation of routing protocols, some trustiness should be set up between sensor nodes such that routing protocols can run on this trustworthy infrastructure. In this paper, we propose a *location-based naming* (LBN) mechanism for sensor networks. Specifically, some location information is embedded into node IDs and acts like an inherent node characteristic in stationary sensor networks, thus it can be used to provide authentication services in local access control [3]. We further propose a link layer authentication scheme LLA, which incorporates LBN, to provide a neighborhood authentication service in sensor networks. Due to the utilization of location information, our scheme can limit the impacts of those topological attacks in a small area. For example, when LBN and LLA are employed, a malicious node can not take on IDs of nodes far away from itself, because the malicious node may be detected if its ID does not belong to its vicinity area.

We make the following contributions in this paper:

1) We propose a location-based naming mechanism LBN and explore its security value for sensor networks;
2) We propose a link layer authentication scheme LLA, which incorporates LBN, to provide a neighborhood authentication service;
3) We show that our LBN mechanism and LLA scheme can be combined to provide an efficient defense against many notorious topological attacks in sensor networks.

The rest of this paper is organized as follows. We illustrate how topological attacks cause severe impacts in sensor networks in Section II. Then we describe our location-based naming mechanism LBN in Section III and link layer authentication scheme LLA in Section IV. We will discuss how our LBN mechanism and LLA scheme can be combined to defend against many attacks in Section V. Some discussions are given in Section VI, and conclusion and future work are given in Section VII.

This work was supported in part by the US Office of Naval Research under grant N000140210464 (Young Investigator Award) and under grant N000140210554, and by the US National Science Foundation under grant ANI-0093241 (CAREER Award) and under grant ANI-0220287.

TOPOLOGICAL ATTACKS

Due to the resource constraints of sensor nodes, most of routing protocols in sensor networks are quite simple, and thus are sometimes more susceptible to attacks than routing protocols in other networks. Many routing attacks have been classified in [2]. They try to raise havoc by spoofing or replaying routing information to skew network topology, thus leading to the failures of routing protocols.

A famous topological attack is the *Sybil attack* [4], in which a malicious node illegitimately takes on multiple identities, which may be fabricated IDs or impersonated IDs. The Sybil attack may pose a serious threat to routing protocols, especially multipath routing and geographic routing, in sensor networks [2]. Besides, it may also cause negative impacts to other applications such as data aggregation, voting, fair resource allocation, misbehavior detection, etc [4]. A similar attack is the *identity replication attack* [4], in which an adversary may put many replicas of a captured node at many places in the network to incur inconsistency.

In the *Wormhole attack* [5], two malicious nodes collude to tunnel packets from one place to another distant place in the network. This attack may distort the network topology by making two distant nodes believe they are neighbors, thus become a serious attack to routing protocols.

In the *sinkhole attack* [2], a malicious node tries to lure nearly all the traffic from a particular area, creating a metaphorical sinkhole with the malicious node at the center. This kind of attack typically works by making the malicious node look especially attractive to surrounding nodes by claiming a lower routing cost to the base station in the sensor network. In the *HELLO flood* attack [2], a malicious node may broadcast HELLO packets with large enough transmission power to convince most nodes in the network that the malicious node is their neighbor, thus lead the network into the state of confusion. In the *acknowledgement spoofing attack* [2], a malicious node may spoof link layer acknowledgments for the packets destined to a neighboring node which is dead or the packets lost due to the bad channel reliability, thus make the source node form a wrong routing decision based on the belief that the dead destination node is alive or the channel is reliable.

LOCATION-BASED NAMING MECHANISM

In this section, we propose a *Location-based Naming* (LBN) mechanism for sensor networks. The idea is to embed some location information into node identifier (ID) and use the location information to defend against topological attacks in sensor networks. The details are given as follows.

A. Location Determination

To utilize location information, it is the first requirement to acquire location information. The *location determination* is not a trivial task in stationary sensor networks. It is infeasible to install every node with a GPS system due to the desire for low price sensor nodes. So, some facilitating methods, such as mobile robots with GPS capability [6] or other coarse-gained location estimation algorithms based on the *Received Signal Strength Indicator* [7] or ultrasound measurements [3], are needed in conventional location determination schemes. These post-deployment methods rely on the intensive cooperation between sensor nodes, thus lead to a large amount of communication overhead.

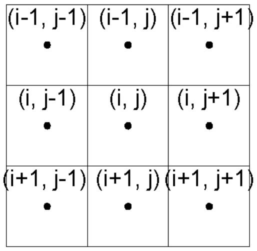

Figure 1. A square cell deployment model.

However, when a sensor network is deployed in an area, some location information is known *a priori*. Hence, if we deploy a group of nodes into an area, we may preload the location information of the area into the nodes' memory. Due to deployment errors, the a-priori location information is less precise than that of posterior measurements, however, it obviates the need to use expensive positioning devices and complex distributed location determination algorithms, thus it is pretty suitable for some applications in resource constrained sensor networks. In this paper, we uses the course-grained a-priori location information to develop a security scheme to defend against many attacks to network topology.

Before deploying a group of sensor nodes, we should decide which place the group should reside. Thus the entire deployment area is divided into many adjacent non-overlapping cells. Every cell is centered with a *deployment point*. Based on specific deployment models, the contour of cell may be square [8]–[10] or hexagon [11]. For simplicity, square cell (Fig. 1) is used as an instance in this paper. However, other shapes are still applicable with a few modifications.

Each group of nodes is intended to be deployed in a predefined cell. Due to deployment errors, every node will be deployed around the deployment point of its cell according to some probability distribution function(PDF), such as *Gaussian distribution* or *Uniform distribution*.

B. Location-based Name

When the deployment model is defined, the location of each deployment point is known. By associating each group of nodes with a specific cell, we may know in which cell of the network each node will reside. In a large scale sensor network, the coordinates of deployment points usually have length of several bytes. However, in current link layer protocols for sensor networks, the node ID field length is usually less than 4 bytes. For example, in TinyOS packet format, the node ID field length is only 16 bits [12]. It is impossible to include the location coordinates of deployment points directly into node ID field in large scale sensor networks. However, our scheme does not rely on precise location information, we only count on the relative location information between sensor nodes. Hence, we tend to use indices.

In our deployment model, each cell is marked with a *cell index*, which is a pair of integers (i, j), where i is the row index

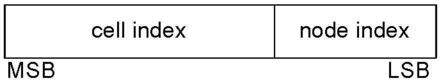

Figure 2. Location-based name.

and j is the column index. Thus we can identify each cell and its associated group of nodes by cell index. The indices are not absolute location coordinates, so they could be very small integers. With this benefit, we may allocate several bits from the node ID field for cell index, and the rest bits from the node ID field as *node index* in the associated cell. Thus each node is identified by a pair *(cell index, node index)*. For example, we may allocate 10 bits from a 16 bits ID field for cell index, and the rest of 6 bits for node index (Fig. 2). Then the maximum affordable network may consist of cells of 32 rows and 32 columns, where each cell contains 64 nodes, and the total number of nodes is 65536.

Only index can not provide more information other than cell identification. What we care about is how the indices describe the relationship between nodes. Thus in our deployment model all cells are indexed according a fixed order from top to right and from left to right such that each cell index (i, j) acts like a coordinate in a two dimensional plane (Fig. 1). In another word, cell indices are normalized coordinates of cells. Hence, the indices reflect the spatial relationship between nodes. By checking node ID fields in received packets, a node may tell whether the sources of packets come from its own cell or neighboring cells or other distant cells. If we treat each node as a kind of resource, and the packets reception by the node as a kind of resource access, then the orderly naming mechanism may provide an authentication service for the access control at link layer in that every node should only accept the packets from the nodes in its cell or neighboring cells, and deny the packets from other distant cells. Obviously, our LBN mechanism has its significance for securing sensor networks. An example is that most ID-spoofing attacks may be defeated because of inherent location information in node IDs. We will show in Section V that our LBN mechanism may defend against a wide range of attacks in sensor networks.

Link Layer Security

In the overall network security infrastructure, link layer security is the basic tile, because all communications are established on the neighbor-to-neighbor communication mode. A node should only accept the packets from authenticated neighboring nodes. To establish trustiness between neighboring nodes, authentication services at link layer are required. To prevent eavesdropping attacks, two neighboring nodes need to negotiate a shared key used for encryptions at link layer. Some proposals [8]–[11] used location information in key management in sensor networks. However, they have not addressed the authentication problem. Motivated by their work, we propose a *Link Layer Authentication* (LLA) scheme in this section, which incorporates the LBN mechanism to provide a neighborhood authentication service.

Our LLA scheme consists two phases. The first one is the *bootstrapping phase* (B-Phase), which is the initial time period after network deployment. The second is the normal *communication phase* (C-Phase) during which nodes communicate normal packets to fulfill kinds of applications. In each phase a two-step authentication is enforced. The first step is the *ID-based authentication*, in which every node decides to accept or reject a packet by checking the packet ID field according to LBN. The second step is the *key-based authentication*, in which the two communicating nodes verify the IDs of each other by the shared key between them.

A. Polynomial Distributing

We use t-degree bivariate polynomials to establish a secure infrastructure between neighboring nodes. A t-degree bivariate polynomial is defined as

$$f(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{t} a_{ij} x^i y^j \tag{1}$$

over a finite field $\mathbb{F}_q$, where q is a prime that is large enough to accommodate a cryptographic key. By choosing $a_{ij} = a_{ji}$, we can have $f(x, y) = f(y, x)$. Assume that every sensor node has a unique, integer-valued, non-zero ID in our LBN mechanism. For a pair of nodes u and v where u and v are node IDs, we can assign a *polynomial share* $f(u, y)$ to u and another share $f(v, y)$ to v. By assigning polynomial shares, we mean the coefficients of univariate polynomials $f(u, y)$ and $f(v, y)$ are loaded into node u's and v's memory, respectively. To establish a shared key, both nodes broadcast their IDs. Subsequently, node u can compute $f(u, v)$ by evaluating $f(u, y)$ at $y = v$, and node v can as well compute $f(v, u)$ by evaluating $f(v, y)$ at $y = u$. Due to the polynomial symmetry, a shared key between nodes u and v has been established as $K_{uv} = f(u, v) = f(v, u)$. This shared key may be used as the link layer key for authentication or encryption.

We use the method proposed in [8] to distribute polynomials such that two nodes in the same cell and neighboring cells hold shares of the same set of polynomial(s) [1]. Each cell is associated with a unique t-degree bivariate polynomial, and the nodes destined to the cell are preloaded with shares of the corresponding polynomial. Besides, the polynomial is also assigned to the horizontal and the vertical neighboring cells. For example, in Fig. 1, the polynomial of cell (i, j) is also assigned to cells $(i, j - 1)$, $(i, j + 1)$, $(i - 1, j)$, and $(i + 1, j)$. Thus a node in cell (i, j) may establish shared keys with nodes in it cells and all neighboring cells because they have shares from the same polynomials set $\mathcal{P}$. We refer readers to [8] for more technical details.

B. B-Phase Authentication

After deployment, the network is in the bootstrapping phase. In this phase, a trustiness should be set up between nodes so that other high layer protocols may begin to work on this trustworthy infrastructure. This is achieved by the B-phase authentication protocol described in Table I.

At very begin, every node broadcasts its node ID (step (1) in Table I) to inform its neighbors its existence. When node u hears

---

[1] We have developed a more efficient scheme in [11] using hexagon cells. It can also be used in LBN design if we choose to use hexagon cells in place of square cells.

TABLE I. B-Phase Authentication Protocol

(1)	$v \rightarrow * :$	$< v >$
(2)	$u \rightarrow v :$	$< u, v, p_f, \{n_u\}_{K_{uv}} >$
(3)	$v \rightarrow u :$	$< v, u, n_u, \{n_v\}_{K_{uv}} >$
(4)	$u \rightarrow v :$	$< u, v, n_v >$

node v, it first checks the cell index field in v's node ID. In LBN mechanism, the cell index should be the same as that of u or the one of the neighboring cell indices which may be easily verified because all cell indices are orderly sorted. If it is not the case, the received ID v may be a spoofed value from a malicious node, and node u just ignores node v's packets.

If the received ID v is acceptable, node u knows immediately the shared polynomials set $\mathcal{P}$ with node v. Because node u and node v have shares derived from the polynomials in $\mathcal{P}$, node u may further verify node v through a challenge-response method. Node u randomly selects a polynomial $f(x, y)$, which has a unique index p_f [2], from $\mathcal{P}$ and uses the corresponding share $f(u, y)$ to calculate a shared key $K_{uv} = f(u, v)$ with node v. Then node u picks a nonce n_u, which is a random number, and sends to node v a challenge packet including the ID u, index of the polynomial $f(x, y)$, and encrypted n_u by $f(u, v)$ (step (2) in Table I).

If node v does have the ID it claims, it sure has the shared polynomials set $\mathcal{P}$ with node u. Then node v may use the polynomial index in the received packet to find the shared key K_{uv} and be able to decrypt the nonce n_u. Next, node v also picks a nonce n_v, returns to node u a response packet including the node ID v, nonce n_u, and the encrypted n_v by $f(u, v)$ (step (3) in Table I).

After get the response from v, node u may check the returned value of n_u. If it is the same as that it has sent to node v, then node v is an authenticated node, otherwise not.

To authenticate itself, node u also decrypts n_v and returns it to node v at step (4) in Table I.

Following the handshake authentication procedure, every node may set up trustiness with its neighbors during the bootstrapping phase. Besides, a shared key is established between neighboring nodes. This shared key may act as the master key and be used to derive other keys for different purposes, such as encryption, authentication, etc. Thus, the future communications between neighboring nodes are secured by the shared key.

C. C-Phase Authentication

After the bootstrapping phase, normal communications may run between neighboring nodes to fulfill kinds of applications. During this phase, an adversary may inject, modify, or spoof packets to raise havoc among the network. To guarantee normal operation of the network, every packet should be authenticated so that the sink node knows it is talking with the authenticated source node.

A normal way to achieve packet authentication and integrity is to use *Message Authentication Code* (MAC), which is a digest calculated by a one-way and collision-resistant hash function with messages and some secrets as inputs. Every node may check whether a received packet is tampered by recalculating the MAC and comparing it with that in the packet.

When a node v needs to send a packet to node u, it constructs the packet like,

$$v \rightarrow u : \quad < v, u, n_v, m, H(v \parallel u \parallel n_v \parallel m \parallel K_{uv}) > ,$$

where n_v is a nonce, m is the message, $H()$ is a hash function, "$\|$" is the concatenation operator, and K_{uv} is a shared key between u and v. To protect the master key established in the bootstrapping phase, it is better to use a derived authentication key here. For example, we may calculate an authentication key as $H(K_{uv} \| 1)$ and an encryption key as $H(K_{uv} \| 0)$. Here the message m may be in plaintext if only authentication is needed or be encrypted if both authentication and encryption are desired.

When node u receives the packet from node v, it first checks the cell index field in v's ID according to LBN. If the ID v is not acceptable, node u simply drops the packet, thus it does not need to check the MAC field. Moreover, node u may check the cell index field just after extracting node v's ID from the packet and stop receiving the remaining part of the packet to save energy if node v's ID is not acceptable, because packet transmission and reception are the most energy-costly radio operations in sensor nodes. Only if the ID v is acceptable, node u proceeds to verify the MAC field in the packet and authenticate the packet.

TinySec [12] defined link layer packet formats including *Auth* packet format, in which only authentication is provided, and *AE* packet format, in which both authentication and encryption are provided. It is similar to our scheme, however, it does not address how to establish authentication and encryption keys. It is obvious that we can combine TinySec with our scheme to provide a complete solution for link layer security in sensor networks.

SECURE SENSOR NETWORKS

In this section, we will show that LBN and LLA are combined to defeat many topological attacks.

A. Topological Attacks

To defend against the Sybil attack, several potential methods are proposed in [4], including radio resource testing, verification of key sets for random key predistribution, registration, position verification and code attestation. The methods to defend against the identity replication attack include centralized computing based on location or number of simultaneous connections [4]. However, those methods rely on either strict physical assumptions or cooperations between a bunch of nodes, thus leading to a large communication overhead. In our scheme, every node ID should appear only in a small area of the network due to the LBN mechanism. The IDs not belonging to a cell may be easily found out by the nodes in the cell. Thus the impact of the attacks is limited in a small area where malicious nodes reside. Moreover, the spoofed IDs can be detected because the malicious node can not have the corresponding polynomial shares belonging to the node whose ID is claimed by the malicious node, and the convergence of the replicas of the same node ID in a small area may be easily detected by surrounding normal nodes. So the Sybil

---

[2] Polynomial indices may be preloaded into nodes memory, or may be calculated by a hash function with cell indices as inputs.

attack and the identity replication attack can not get success in our scheme.

To defend against the Wormhole attack, Hu *et al.* proposed to use *packet leashes* [5] to limit the maximum range over which packets can be tunneled by the two colluding nodes. *Directional antennas* [13] are also used to defend against the Wormhole attack. However, these defenses are targeted to the Wormhole attack in ad hoc networks, and require expensive hardware devices, which are infeasible for most resource constrained sensor networks. Wang and Bhargava [14] proposed to use centralized computing to defend against the Wormhole attack in sensor networks, in which a controller collects all nodes' location information to reconstruct the network topology such that any topological distortion may be visualized. However, this approach causes much intensive communication overhead and is not realistic if malicious nodes move around in the entire network because each location change will trigger a new round of execution of the topology reconstruction algorithm. By using LBN, a node may check the cell index fields in the received packets and simply drop those packets coming from a distant place. So the impact of the Wormhole attack is limited in neighboring cells automatically. Though the two colluding nodes may tunnel packets in a small area, in this case they can not cause severe network scale topological distortions and may even be helpful to facilitate local communications. So, the Wormhole attack may be defeated in our scheme.

It is hard to defend against the sinkhole attack [2], because different metrics may be used in routing protocols. If geographical routing protocols are used, every route is found based on geographical information, which can be extracted from node IDs. In this case, the malicious node can not cheat other nodes because other nodes may easily find whether the malicious node is on the route to the base station based on the ID of the malicious node. If different routing criteria such as reliability are used, it is rather difficult to detect the sinkhole attack. However, the node ID may still provide some information about the location of the malicious node, thus if the source node finds the location of the malicious node is far away from the direction of the base station, it means a potential threat and some methods may be used to verify the routing information.

The HELLO flood attack [2] may be defeated because it is easy to check whether a HELLO packet is acceptable from its ID field in our scheme. The *acknowledgement spoofing attack* [2] can be detected by LLA because the malicious node does not have corresponding link layer keys.

B. The Node-compromise Attack

In our link layer authentication scheme, predistributed polynomials are used to establish shared keys between nodes. It is under the threat of the *node-compromise attack*, in which a small number of compromised nodes may expose a large amount of secrets in the network. It has been proved in [15] that a t-degree bivariate polynomial is *t-collusion resistant*, meaning that the collusion of no more than t nodes can not expose the polynomial. However if one t-degree bivariate polynomial is used by more than t nodes, an adversary may compromise more than t nodes holding shares of a same polynomial to reconstruct it, and then use the reconstructed polynomial to derive shared keys between non-compromised nodes that hold shares of the same polynomial. We have proposed an efficient scheme [11] in which every t-degree bivariate polynomial is reused no more than t times, thus we may achieve the perfect resilience to the node-compromise attack. For the lack of space, we do not investigate this topic here, and refer readers to [11] for technical details.

C. The Memory Exhaustion Attack

The B-phase authentication in our scheme is not stateless, because every node needs to keep the nonce in its memory so that it can verify the returned nonce value from its neighbor. For each authentication request, a nonce should be generated. A malicious node may launch the *memory exhaustion attack* by sending authentication requests at very high frequency to neighbors, thus cause its neighbors unusable by exhausting memory resources of the neighbors. However, it is also easy to detect frequent authentication requests from a malicious node. To defend against this kind of attack, normal nodes just need to drop those authentication requests if the frequency of request is too high. Some countermeasures can also be triggered to punish the malicious node.

DISCUSSION

To the best of our knowledge, there has been no research on the additional value of node identifier in sensor networks, where every node's identifier is taken from an one dimension name space that has no meaning but the identification function. Though many schemes [8]–[11] use node identifiers in key establishment, they simply use the identification function. Our scheme is the first investigation that tries to dig out more application values of node identifier. We have shown that by embedding location information into node identifiers our LBN has intrinsic immunity from many attacks against network topology. Besides security value, we believe our LBN can still be used in other applications in sensor networks.

Our LLA scheme incorporates LBN as the first step authentication method, and uses shared key to further verify node identity. In LLA, predistributed polynomials are used to achieve key agreement to provide authentication service. However, other shared-key-based authentication schemes can also work well with LBN in the second authentication step, as long as they guarantee neighboring nodes can establish a unique shared key. Similar schemes are *SPINS* [16], *LEAP* [17]. The building block *SNEP* in SPINS [16] can provide neighbor authentication by a shared key. However, two neighboring nodes rely on the base station to negotiate a shared key, which is not efficient in terms of communication overhead. In *LEAP* [17], a global key is used to derive shared keys to achieve neighbor authentication, where the underlying assumption is that adversaries can not compromise any node during network bootstrap phase, thus the global key can be safe. However, our scheme does not rely on this assumption and is resilient to node compromise attacks.

Zhang *et al.* [18] proposed to use location-based keys to secure sensor networks. Their scheme is based on public key cryptography, while our scheme is based on symmetric key cryptography.

Besides, in their scheme each location-based key is tight to a precise location in the network and the location information should be obtained by mobile robots. When a node moves, its location-based key associated with its previous location is invalid. Hence, their scheme is only applicable in stationary sensor networks, where sensor nodes do not move after deployment. Our scheme only uses course-gained a-priori deployment knowledge and does not need any positioning devices. Though our scheme is targeted for stationary sensor networks, low mobility can also be supported as long as nodes only move in their vicinity.

CONCLUSION AND FUTURE WORK

In this paper, we have introduced the naming problem for sensor networks in the literature for the first time. We believe that more benefits can be achieved by endowing node ID more meaningful information. A location-based naming mechanism LBN has been proposed to fulfill our idea. Our LBN obviates the need to use expensive positioning devices and complex distributed location determination algorithms in conventional location-based schemes. By using LBN, the impacts of many attacks to topology in sensor networks can be limited in a small area. We also proposed a link layer authentication scheme LLA, which incorporates LBN, to provide a neighborhood authentication service. It has been shown that our LBN and LLA can be an efficient defense against a wide range of topological attacks in sensor networks.

We have investigated the security value of our location-based naming mechanism. However we believe it may also find other applications in sensor networks, such as geographic routing, target tracking, environment surveillance, etc, especially those applications in which security is desired. We will develop more efficient solutions in those applications based on our new idea in our future work.

REFERENCES

[1] A. Wood and J. Stankovic, "Denial of service in sensor networks," in *IEEE Computer*, pp. 54-62, October 2002.

[2] Chris Karlof, David Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[3] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *ACM Wise'03*, San Diego, California, September 19, 2003.

[4] J. Newsome, E. Shi, D. Song, A. Perrig, "The sybil attack in sensor networks: Analysis & defenses," in *the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, Berkeley, California, USA, April 26-27, 2004.

[5] Y. Hu, A. Perrig, D. B. Johnson, "Pachet leashes: a defense against wormhole attacks in wireless networks," in *IEEE INFOCOM'03*, 2003.

[6] P. Corke, R. Peterson, and D. Rus, "Networked robots: flying robot navigation using a sensor net," in *Proc. of 11th Internatonal Symp. of Robotics Research (ISRR)*, October, 2003.

[7] C. Savarese, J. Rabaey, and J. Beutel, "Locationing in distributed ad-hoc wireless sensor networks," in *Proceedings of the ICASSP*, May 2001.

[8] D. Liu, and P. Ning, "Location-based pairwise key establishments for relatively static sensor networks," in *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03)*, October 2003.

[9] W. Du, J. Deng, Y. S. Han, S. Chen and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in the *IEEE INFOCOM 2004*, Hong Kong, March 2004.

[10] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wireless sensor networks," in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks(SASN'04)*, October 25, 2004, Washington, DC, USA.

[11] Y. Zhou, Y. Zhang, and Y. Fang, "LLK: A link-layer key establishment scheme in wireless sensor networks," in *IEEE WCNC'05*, New Orleans, LA, March 2005.

[12] C. Karlof, N. Sastry, and D. Wagner, "TinySec: A link layer security architecture for wireless sensor networks," in the *Second ACM Conference on Embedded Networked Sensor Systems (SensSys'04)*, Baltimore, Marylaand, November, 2004.

[13] L. Hu, and D. Evans, "Using directional antennas to prevent wormhole attacks," in the *11th Annual Network and Distributed System Security Symposium(NDSS'04)*, San Diego, California, February 5-6, 2004.

[14] W. Wang, and B. Bhargava, "Visualization of wormholes in sensor networks," in *Proceedings of the 2004 ACM workshop on wireless security(Wise'04)*, Philadelphia, PA, USA, October 1, 2004.

[15] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro,and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Advances in Cryptology C CRYPTO 92, LNCS 740*, pages 471-486, 1993.

[16] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: Security Protocols for Sensor Networks," in *Wireless Networks* 8, 521-534, 2002.

[17] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanism for large-scale distributed sensor networks," in *ACM CCS'03*, Washington, DC, October 27-31, 2003.

[18] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing sensor networks with location-based keys," in *IEEE WCNC'05*, New Orleans, LA, March 2005.