Journal of Information Processing Vol.20 No.1 54–64 (Jan. 2012)

[DOI: 10.2197/ipsjjip.vol.20.54]

Invited Paper

Secured Geographic Forwarding in Wireless Multimedia Sensor Networks

Taye Mulugeta[1] Lei Shu[2,a)] Manfred Hauswirth[3] Zhangbing Zhou[4] Shojiro Nishio[2]

Received: July 8, 2011, Accepted: October 17, 2011

Abstract: Two Phase geographic Greedy Forwarding (TPGF) is a pure on-demand geographic greedy forwarding protocol for transmitting multimedia streams in wireless multimedia sensor networks (WMSNs), which has explicit route discovery, i.e., a node greedily forwards a routing packet to the neighbor that is the closest one to the destination to build a route. Like most geographic routing protocols, TPGF is vulnerable to some greedy forwarding attacks, e.g., spoofing or modifying control packets. As the first research effort that investigates the secure routing protocol in WMSNs, in this paper, we identify vulnerabilities in TPGF and propose corresponding countermeasures, e.g., secure neighbor discovery and route discovery, and propose the SecuTPGF, an extended version of TPGF, which exactly follows the original TPGF protocol's routing mechanism but with enhanced security and reliability. The effectiveness of SecuTPGF is proved by conducting security analysis and evaluation experiments.

Keywords: wireless multimedia sensor networks, security, geographic forwarding, two phase geographical greedy forwarding

1. Introduction

The emergence of wireless sensor networks (WSNs) and various multimedia devices provide the bridge between physical and virtual worlds, which brings people into information explosion era. Along with the fast development of WSNs applications, traditional sensory information provided by scalar sensor nodes can no longer satisfy the information needs as simple scalar sensory data cannot efficiently describe some complicated events in the WSNs fields. Multimedia sensor nodes are developed to provide more comprehensive information to enhance the capability of traditional WSNs for event description. Efficiently transmitting multimedia streams in wireless multimedia sensor networks (WMSNs) is a significant challenging issue, due to the limited transmission bandwidth and power resource of sensor nodes [1].

Two Phase geographical Greedy Forwarding (TPGF) [4] is one of the first designed routing protocols for WMSNs, which uses geographic greedy forwarding for exploring one or multiple node-disjoint optimized hole bypassing transmission paths in WMSNs. Like most network protocols, TPGF is not designed for non adversarial networks and is susceptible to outsider attacks. For example, an enemy, who is able to compromise an authentic network node, may easily launch more serious insider attacks, by extracting key and security information from the compromised node, and then act as an authentic network participant [6]. Thus, when TPGF is used in WMSNs for transmitting multimedia streaming data, it should be devised in a way that it is resilient to security attacks, since attacks at the networking layer (specifically those against the routing protocols) can disrupt the whole network operation [5].

Therefore, in this paper, the focus is providing efficient security for TPGF protocol: the SecuTPGF, a modified version of TPGF applying Identity-Based Non-Interactive Key Distribution Scheme (ID-NIKDS) [7], which provides both node authentication and symmetric key establishment. In SecuTPGF, we mainly secure the *neighbor discovery* and *route discovery*. Securing *neighbor discovery* prevents malicious nodes from joining the WSN and hence nodes establish a neighbor table free of malicious nodes. Securing *route discovery* authenticates the intermediate nodes involved in the routing path.

To the best of our knowledge, SecuTPGF is the first research effort for providing secured routing protocol in WMSNs, which clearly distinguish the novelty of SecuTPGF and its scientific impact in the WMSNs research community. As the more concrete scientific contributions of this research work, the SecuTPGF protocol provides the following functions:

(1) Preventing outside adversaries from joining the network;
(2) Limiting the impact of insider attacks in a localized area;
(3) Partially detecting insider attacks and avoided them in the network;
(4) Authenticating control messages exchanged between nodes.

This paper is carefully extended from our previous accepted paper in Globecom 2010 [2] with the following organization: Section 2 provides expanded information on basic operations of TPGF. Section 3 presents system assumptions and further dis-

1 Department of Electrical and Computer Engineering, Addis Ababa University, Addis Ababa, Ethiopia
2 Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, Suita, Osaka 565–0871, Japan
3 Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland
4 Institute TELECOM & Management SudParis, Evry, France
a) lei.shu@ist.osaka-u.ac.jp

cusses the vulnerabilities of TPGF in terms of routing attacks. Section 4 introduces the Identity-Based Non-Interactive Key Distribution Scheme and its advantages, which were not introduced in the Globecom 2010 paper. Section 5, as a new section presents the design goals of SecuTPGF. Section 6 presents the detailed design of SecuTPGF with more information on the routing path maintenance. Security analysis is carried out in Section 7. Simulation based performance evaluations and more direct comparisons on path exploration are given in Section 8 and Section 9. Section 10 concludes this paper.

2. TPGF Routing Protocol

2.1 TPGF Overview

Two-Phase geographic Greedy Forwarding (TPGF) explores one or multiple (near) shortest hole-bypassing paths in WMSNs. The TPGF routing algorithm includes two phases: 1) the first phase is responsible for exploring the possible routing path; 2) the second phase is responsible for optimizing the found routing path with the least number of hops. The TPGF routing algorithm finds one path per execution and can be executed repeatedly to find more node-disjoint routing paths. An example of executing TPGF to explore multiple paths in a duty cycle based WMSNs is given in **Fig. 1**. In TPGF there are *route control messages* forwarded to 1) discover a route, 2) optimize a found route, 3) avoid block node situation (*step back and mark* message), and *release control message* to free those nodes that do not receive a *route Acknowledgment message*.

Some of the features that make TPGF be different from most existing geographic routing algorithms, e.g, GPSR [8], are:

- TPGF is a pure geographic routing algorithm. It does not include the face routing concept [8] [*1].
- TPGF does not require the computation and preservation of the planar graph [*2], e.g., RNG [22] and GG [23], in WSNs. This point allows more links to be available for TPGF to explore more node-disjoint routing path.

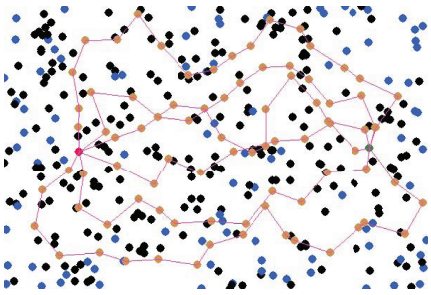


Fig. 1 An example of executing TPGF to explore multiple paths. The red color node is the multimedia source node. The black color nodes are sleeping nodes. The blue color nodes are awake nodes, which can be used to explore the node-disjoint transmission paths. The green color node is the sink node.

[*1] Face routing: A message is routed along the interior of the faces of the communication graph, with face changes at the edges crossing the Source-Destination-line. Greedy forwarding can lead into a dead end, where there is no neighbor closer to the destination. Then, face routing helps to find a path to another node, where greedy forwarding can be resumed.

[*2] In graph theory, a planar graph is a graph that can be embedded in the plane, i.e., it can be drawn on the plane in such a way that its edges intersect only at their endpoints.

- TPGF does not have the well-known Local Minimum Problem [4], which is defined as "a sensor node finds no next-hop node that is closer to the base station than itself."

2.2 Two Major Operations of TPGF

In general, the operation of TPGF relies on two activities: *Neighbor Discovery* and *Route Discovery*.

2.2.1 Neighbor Discovery

Neighbor discovery is the basic operation for building up neighbor table in every sensor node. When a node A wants to determine its neighbor nodes, it broadcasts a neighbor discovery request (HELLO) message which contains: its ID (ID_A), its geographic location (L_A), and then waits for each neighbor node to respond. Every node that receives this request responds with a neighbor discovery reply that contains its *ID* and *geographic location*. For each received reply, node A puts the *ID* and *geographic location* of the responding node in its neighbor table.

2.2.2 Route Discovery

TPGF route discovery is based on unicast *greedy forwarding* route finding and returning an *Acknowledgment*. In TPGF, a route *request message* contains: 1) the identifiers of the source node and the base station, 2) a record listing of identifiers of every chosen (intermediate) node that forwards this particular *request message*. Each *request message* also has a *path number* (request identifier), which, together with the identifier of the source node, uniquely identifies the request.

When a source node wants to explore one transmission path, it generates a route *request message*, which contains a new *path number* and an empty list of forwarding nodes and forwards it to its chosen neighbor based on the *greedy forwarding* rule: a forwarding node always chooses the next-hop node that is closest to the based station among all its neighbor nodes, the next-hop node can be further to the base station than itself. The chosen neighbor node appends its digressive *node number* together with its node ID to the list of identifiers in the *request message* and greedily forwards the request to next hop. If the chosen node finds that it has no next node available for transmission, it will *step back* (send a *block node* [*3] message) to its previous-hop node and mark itself as a *block node* [4] (marking the block node is to forbid the loop). The previous-hop node will attempt to find another available neighbor node as the next-hop node. This procedure is repeated until the request reaches the base station.

Whenever a routing path reaches the base station, an *Acknowledgement* is requested to send back to the source node. The base station generates an *Acknowledgment* by copying the recorded list of identifiers from the route *request* message into the *Acknowledgement* message. The *Acknowledgement* is then sent back to the source node. During the reverse travelling in the found routing path, *Label Based Optimization* [4] [*4] is performed in each

[*3] For any sensor node, during the exploration of a routing path, if it has no next-hop node that is available for transmission except its previous-hop node, this node is defined as a block node, and this kind of situation is defined as a block situation.

[*4] *Label Based Optimization*: Any node in a path only relays the acknowledgement to its one-hop neighbor node that has the same *path number* and the largest *node number*.

intermediate node to eliminate *path circles* [4] [*5]. The intermediate node by seeing the recorded list of identifiers, it only relays the *Acknowledgement* to its one-hop neighbor node that has the largest *node number* and then sends a *release message* to its previous-hop that does not get an *Acknowledgement*. This procedure is repeated until an *Acknowledgment* reaches the source node. When the source node receives the successful *Acknowledgment*, it starts to send out multimedia streaming data to the successful path with the pre-assigned *path number*.

3. Network Model, Vulnerabilities and Attacks

3.1 Network Model

In our considered WMSN, all nodes are stationary and all communication links are symmetrical. It is feasible for applying the public key cryptography to WSNs with care [9]. The base station is trustworthy and not resource-constrained, which is a common assumption in WSN security [6]. To determine geographic location, sensor nodes are equipped with their own Global Positioning System (GPS) or use localization algorithms [10]. We assume that each sensor node can sustain a certain time interval before it is compromised, which is also assumed by previous work [11]. Sensor nodes are not trusted unlike the base station, which is also a common assumption in WSNs [6], [12], because it is relatively easy for an adversary to capture and compromise sensor nodes. If a node is compromised, its keying and security primitives become to be available to the adversary, making it possible for an adversary to control the node in an arbitrary way [6]. Finally, we use Identity Base Cryptography (IBC) [13] scheme in the WMSNs.

3.2 Vulnerabilities

Since TPGF has no security mechanisms, malicious nodes can perform many attacks just by not behaving according to the TPGF rules. A malicious node M can carry out the following attacks (among many others) against TPGF:

(1) Impersonating a source (S) node by forging a route *request* with its address as the originator address.
(2) Using fake identity, it can send the *step back & mark* message to the previous node to create incorrect routing state.
(3) Selectively, not forwarding certain *request*, *acknowledgment*, or multimedia data messages. This kind of attack is especially hard to even detect since transmission errors can have the same effect.
(4) When forwarding an *acknowledgment* message generated by the base station as reply to *request* message, not performing *path optimization*, so it can increase the end-to-end delay of the found path.
(5) Spoofing a *release* command to create incorrect routing state.

3.3 Attacks

In this subsection, we review the routing attacks studied in Ref. [6] and discuss attacking mechanism in TPGF.

- By spoofing, altering, or replaying routing information, adversaries may be able to create attract or repel network traffic, extend or shorten transmission routes, generate false error messages, increase end-to-end transmission latency, etc. Since TPGF is an on-demand source routing protocol, without further protections, TPGF is vulnerable to these attacks, e.g., when TPGF forwards route *request* packets, the adversaries may insert malicious node identifiers in the path and make themselves participate in the transmission.
- Sybil attack [14]. A malicious node illegitimately takes on multiple identities by impersonating other nodes or simply by claiming false identities. In TPGF, nodes requires to exchange coordinate information to forward geographically addressed packets. By using Sybil attack, an adversary can forge location advertisement and advertise multiple bogus nodes.
- Node replication attack [15]. An adversary intentionally puts many replicas of a compromised node at many places.
- Wormhole attacks. Wormhole attacks are used to convince two possibly distant nodes that they are neighbors, so that the attacker can place himself on the route between them. Basically, the adversary tunnels messages from one part of the network to another through an out-of-bound channel available only to the attacker. Wormholes typically involve two colluding nodes. During neighbor discovery, wormholes create false neighbor relationship.
- HELLO flood [6]. HELLO flood can be thought of one-way broadcast wormhole. As with wormholes, HELLO flood is a threat to TPGF and prevented by using similar technique as that of wormhole attacks.
- Selective forwarding [6]. Attackers selectively forward packets instead of faithfully forwarding all received packets or completely drop all packets.
- Sinkhole attacks. By acting especially attractive to surrounding nodes with respect to the routing algorithm, a malicious node lures nearly all the traffic from a particular area and hence enables many other attacks. TPGF is a kind of greedy forwarding algorithm, which uses neighbor location information to build a path. The traffic is naturally routed towards the physical location of a base station, it is relatively difficult to attract it elsewhere to create a sinkhole.

3.4 Sybil Attack

In general, Sybil attack is the most common and powerful attack in geographic routing. In TPGF, an adversary may present multiple identities to other nodes with misrepresented location to increase its chance to be involved in routing path. Consider the hypothetical topology as an example in **Fig. 2**, an adversary A advertises multiple identities A_1, A_2, and A_3 with misrepresented location to increase chance of being selected as the intermediate relay node. When a source node S initiates route request to location $(7, 4)$, the virtual Sybil node A_1 is selected as the next-hop. As a result, the DATA will go through S-> A-> B-> D instead of S-> B-> D, which will increase the cost and the end-to-end delay of transmission. In situation when a source S needs multiple node-disjoint paths, it chooses Sybil nodes A_1 and A_2 as the relay nodes (shown in Fig. 2). The found paths will be S-> A_1

[*5] For any given routing path in a WSNs, if two or more than two sensor nodes in the path are neighbor nodes of another sensor node in the path, we consider that there is a *path circle* inside the routing path.

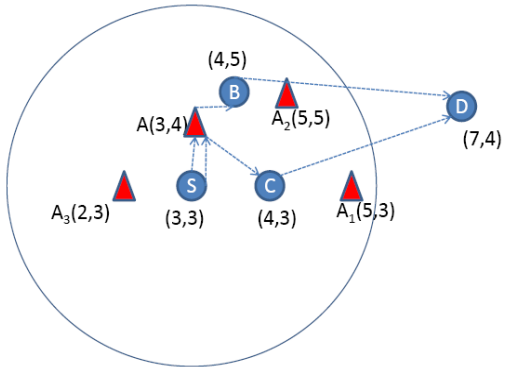


Fig. 2 Node A advertises Sybil IDs A_1, A_2 and A_3, and prevents source S from establishing multiple node-disjoint paths, in which nodes A_1, A_2 and A_3 are Sybil nodes.

-> B-> D and S -> A_2-> B-> D. However, these paths are actually not node-disjoint, since A_1 and A_2 are virtual Sybil identities of physical malicious node A. This gives an opportunity to the adversary A to raise many other attacks.

4. Identity-Based Non-Interactive Key Distribution Scheme

To establish a shared secret key between any two or more communicating nodes for subsequent cryptographic use is a fundamental problem of the security study in WSNs. Due to the constraints of WSNs, e.g., limit energy and computing capacity, it is believed that Public Key Cryptography (PKC) is too complex to be suitable for WSNs, which leads proposals based on pure symmetric key cryptography. However, the inherent limitations of symmetric-key cryptography render these proposals suffer from the lack of authentication, scalability and resilience to node compromise [18].

4.1 Overview of ID-NIKDS

Pairing-based cryptography [17], is an emerging technology that has drawn a great amount of research attention in the last a few years. In the field Pairing-based cryptography, Sakai, Ohgishi, and Kasahara proposed an identity-based non-interactive key distribution scheme (ID-NIKDS) [7] and that can be implemented using Tate Pairing [17]. In ID-NIKDS, for two nodes A and B that know each other's ID wish to decide on a secret key, first, the nodes need to have their own private key $[s]P_A$ and $[s]P_B$ placed on them by the the base station, where 's' is the master secret key of the base station. Then both nodes calculate public keys as, $P_A = H_1(ID_A)$ and $P_B = H_1(ID_B)$, where P_A and $P_B \epsilon$ G1, and H_1 is a mapping function that maps node's identity to a point in elliptic curve (H_1: 0, 1* ϵ G1). Finally, the symmetric key, k_{AB}, can be calculated by both nodes as

$$k_{AB} = \hat{e}([s]P_A, P_B) = \hat{e}(P_A, P_B)[s] = \hat{e}(P_A, [s]P_B) \qquad (1)$$

4.2 Advantages of IBC Based Key Agreement Protocol

With the advent of elliptic curves cryptography (ECC), identity base cryptography (IBC) based on pairing become more popular and is used for resource constrained networks, e.g., WSNs. As comparing to the conventional PKC, there are at least *three significant advantages* of IBC. First, IBC removes the need for *certificates* and hence the certificate distribution and verification. Considering the resource-constrained nature of WSNs, this often represents non-trivial savings in both communication and computation overheads, especially in large-scale WSNs. Second, IBC facilitates *non-interactive key agreement*. For any two parties, if both have an authentic public/private pair from the same trusted authority (TA) based on the IBC, have already shared a secret key without exchanging any message. This obviously can further reduce both communication and computation overheads. Finally, the fact that *any type of string can be a public key* in IBC provides many useful properties that do not exist with the conventional PKC. The drawback of IBC is pairing computation. However, the computation cost is very small as compared to transmission cost [*6]. In addition, pairing computation is done only once between any two parties and can be used for subsequent communication. Recently, the pairing computation cost is drastically reduced. As reported in TinyPBC [19], the computation time for Tate Pairing [17] on the MICA2 mote using the ATmega128L, 80-bit security level is improved from 10 s of seconds to 2.06 seconds.

In our proposal, we use the non-interactive key exchange protocol of Sakai et al. [7] to provide identity authentication and symmetric key establishment because it avoids the use of *certificate* for authentication [*7].

5. Design Goals of SecuTPGF

We aim at preventing unauthorized node from joining the network since multimedia data is costly and reducing the impact of insider attacks from paralyzing the network.

First, to prevent unauthorized node from joining the network, each node that interprets routing information must verify the origin and integrity of the data, which means it must authenticate the data. To achieve routing authentication, each control message should be authenticated by the originator, any intermediate node that adds information must resign the entire update message for protecting the mutable information in the data and the destination and any node that updates its state as a result of processing routing message must verify the authenticity of the control message. We need an authentication mechanism with low computation and communication overhead. Thus, for one-to-one and all-to-one authentication of a message, we use a message authentication code (MAC), e.g., HMAC [3] and a shared key between the two parties. To setup a shared key between any two parties, we need to use a key agreement protocol.

Our second target is to limit the impact of insider attacks. The cause of insider attacks is after compromising a legitimate node an adversary can initiate different kinds of malicious activities. For example, as shown in Section 3.4, Sybil attack can disturb the normal operation of TPGF. To prevent such attack the cryptographic mechanism should provide node ID authentication. How-

[*6] Wireless transmission of a bit can require over 1000 times more energy than a single 32-bit computation, as shown in Ref. [11].

[*7] In public-key schemes, to authenticate public keys, *certificate* is used that is ill-suited to WSNs.

ever, ID authentication is not enough to protect the sensor network. In node Replication attack, an adversary can introduce a malicious new node using compromised node ID. A solution to combat this attack is needed in authentication procedure.

6. SecuTPGF

The first problem that we address in SecuTPGF is achieving source authentication and protection of mutable information in routing messages. In our solution, we use message authentication code (MAC) to tackle these problems. The second problem addressed is authentication of node identity and calculation of symmetric key between nodes. In SecuTPGF, we use ID-NIKDS Scheme to mitigate these problems, which can avoid the using of certificates for public key authentication. And, no interaction is required to determine the symmetric key between nodes except their unique IDs. Finally, to limit the impact of insider attack, a bootstrapping time information [*8] is involved in authentication procedure.

6.1 Initialization and Key Setup

Setup: This stage is to be executed by the WSN manager (Base Station) acting as a trusted authority (TA), using its own facilities for processing in order to minimize the nodes power consumption.

To start up an ID-NIKDS scheme, the base station first needs to generate and distribute private keys and public parameters. This procedure can be accomplished as follows:

- The base station generates two groups G1 and G2 with prime order q satisfying the bilinear pairing e: G1 * G1 ϵ G2;
- Chooses a random generator point P ϵ G1;
- Generates a master secret key, [s] ϵ Zq* and set the base station's public key $Ppub = [s]P$;
- Computes node's public key by mapping each node's identity and bootstrapping time Ti to a point on the elliptic curve, via a hashing-and-mapping function H1; $P_X = H1(ID_X//Ti)$ for Node X;
- Calculates each node's private key, $S_X = [s]P_X$.

It next preloads each node X with values of the node's identity ID_X, the node's private key S_X, a preloaded individual symmetric key K_X shared with the base station, the bootstrapping time Ti (the time that the node X bootstraps itself to join the WSN) (ID_X, S_X, K_X, Ti) and also equipped with the function H1, so that it can easily compute public key of any node knowing the ID of the node. Once initialization stage is completed, all nodes are ready to be deployed into field. The neighbor discovery phase starts right after the network deployment.

6.2 Secure Neighbor Discovery

By securing neighbor discovery, outside adversaries are prevented from joining the WSN and only authentic nodes are allowed to join WSNs at the very beginning stage. Moreover, key establishment is also included to help the new node to establish shared keys with its neighbors so that it can perform secure communications with them. To authenticate nodes and establish symmetric key, the ID-NIKDS scheme is applied, which provides a pre-shared secret keys according to Eq. (1).

6.2.1 Neighbor Discovery

When it is deployed, node A bootstraps itself at a preset time Ti_A and tries to discover its neighbors. It broadcasts a HELLO message, which contains its ID (ID_A), its geographic location (L_A), bootstrapping time (Ti_A), and a random nonce (N_A), and then waits for each neighbor B to respond.

$$a \rightarrow * : HELLO(ID_A, L_A, Ti_A, N_A) \tag{2}$$

Node B first validates whether the bootstrapping time Ti_A is within a pre-specified threshold L with its current time t. If the check fails, node B simply discards the request. Otherwise, *B transmits to A a challenge message* that contains its ID (ID_B), geographic location (L_B), bootstrapping time (Ti_B), a random nonce (N_B), and an authenticator (V_B) calculated as $H(k_{BA}, L_B||L_A, Ti_B||Ti_A, N_B||N_A)$, where H is a hash function.

$$b \rightarrow a : (ID_B, L_B, Ti_B, N_B, V_B) \tag{3}$$

Upon receiving this *challenge*, node A proceeds to compute a verifier as

$$V'_B = H(\hat{e}([s]P_A, P_B), L_B||L_A, Ti_B||Ti_A, N_B||N_A) \tag{4}$$

By the bilinearity of the pairing ê in Eq. (1), the verification is successful if and only if both A and B have the authentic private keys corresponding to their claimed bootstrapping time. After verifying the equality of V'_B and V_B, node A computes a verifier as $V_A = H(k_{AB}, L_B, Ti_B, N_B)$ and sends valid response to node B. Node A also calculates symmetric key and add node B into its neighbor list.

$$a \rightarrow b : (ID_A, V_A) \tag{5}$$

Using a similar approach as node A, node B verifies that whether node A is an authentic neighbor and then establishes a secure link and adds it into its neighbor list.

6.2.2 Symmetric Key Establishment

After nodes A and B achieve mutual authentication, they calculate a symmetric key as

$$Z_{AB} = H(K_{AB}, N_A, N_B) = Z_{BA} = H(K_{BA}, N_A, N_B) \tag{6}$$

where K_{AB} is secret key, N_A is nonce of node A and, N_B is nonce of node B.

6.3 Secure Route Discovery

6.3.1 Route Request

In our SecuTPGF proposal, the source node initiates and forwards a *request message* to intermediate node that is the one hop neighbor nearest to the base station among all its neighbor nodes. The *request message* contains *message identifier* (*rreq*), *the ID of the source node* (*S*), *the geographic location of the base station* (*Dloc*), *a request path number* (*Pno*), *and a MAC field*. The MAC field is computed over all elements with a key shared by the Source (S) and the base station (D) ($MACk_{SD}(rreq, S, Dloc, Pno)$). The request path number is incremented each time when source node initiates a new route request.

[*8] A time that a node bootstraps itself to join a WSN.

The size of the generated MAC is 4 byte [*9].

When the intermediate node receives a *request message* for which it has no next-hop node to send, it sends *Block Node message* to its previous-hop node. The *Block Node message* is authenticated using a shared key between the intermediate node and the previous-hop node. Otherwise the intermediate node modifies the request by appending its ID in the path list of the *request message* and replacing the MAC field with a MAC computed on the entire *request message* using a key shared between the base station and the intermediate node. The intermediate node also checks if the path can be optimized. The path will be optimized, if the source or the farthest node listed in the path list (ID sequence) of the *request message* is a neighbor of the intermediate node. And, if the path is optimized, the intermediate node appends optimized neighbor ID in the path list before its ID when the request is modified. For example, an intermediate node 'e' receives a *request message* for which the path list contains "a->b->c->d" and nodes 'b' and 'c' are neighbors of node 'e'. The intermediate node 'e' checks whether the source node is a neighbor, if it is not, then searches the path list from the beginning node (node 'a') till it finds a neighbor node in the path list. The searching returns the farthest (the farthest in ID sequence, but not on geographic distance) neighbor node 'b', and then the path list in the *request message* for node 'e' will be modified as "a->b->c->d->b->e". Finally, the intermediate node records the address of the neighbor from which it received the request, and then the modified route request is forwarded. This process is repeated until the *request message* reaches the base station.

6.3.2 Route Acknowledgment

When the base station receives the *request message*, it verifies the MAC. If this verification is successful, the base station continues to search a duplicated node ID in the path list of the *request message* to get optimized path. If the base station finds a duplicate node ID, it assumes that the next node after the duplicated ID and the duplicated ID nodes are neighbors, so it removes the nodes' IDs in between the two neighbor nodes to get the optimized path. In previous example the path list of the *request message* is path "a->b->c->d->b->e," in which the path list node ID 'b' is duplicated. Therefore, the base station assumes that node 'b' and node 'e' are neighbors, so it removes the in-between nodes 'c' and 'd' to get the optimized path list as "a->b->e." After optimization, the base station constructs an *Acknowledgement message* containing *the ID of the source node, the geographic location of the base station, the request path number, the optimized path list, and the MAC field*, and sends it back to the source node via the reverse of the route obtained in the optimized path. The MAC field is computed over all elements with a key shared between the base station and the source node. When an intermediate node receives an *Acknowledgment message*, it checks whether the previous node ID that sends the *request message* is in the path list of the *Acknowledgment*; if not, it sends a *release command message* to this node. Finally, the source node verifies the *Acknowledgment message*. We describe SecuTPGF route discovery process in **Table 1** for a topology as shown in **Fig. 3**.

[*9] In Ref. [12], it claimed that 4 byte MAC is enough to protect the message authenticity and integrity in the context of flat WSNs.

Table 1 *Route Discovery* example in SecuTPGF. The initiator node A is attempting to discover a route to the base station (D).

A	:	$MAC_a = MAC_{aD}(rreq, a, D, Pno)$
a → b	:	$(rreq, a, D, Pno, [], [MAC_a])$
B	:	$MAC_b = MAC_{bD}(rreq, a, D, Pno, [b], [MAC_a])$
b → c	:	$(rreq, a, D, Pno, [b], [MAC_b])$
C	:	$MAC_c = MAC_{cD}(rreq, a, D, Pno, [b, c], [MAC_b])$
c → d	:	$(rreq, a, D, Pno, [b, c], [MAC_c])$
D	:	$MAC_d = MAC_{dD}(rreq, a, D, Pno, [b, c, a, d], [MAC_c])$
d → D	:	$(rreq, a, D, Pno, [b, c, a, d], [MAC_d])$
D	:	$MAC_a = MAC_{aD}(rreq, a, D, Pno, [a, d])$
D → d	:	$(rreq, a, D, Pno, [a, d], [MAC_a])$
d → a	:	$(rreq, a, D, Pno, [a, d], [MAC_a])$
d → c	:	$(rcom, a, D, Pno, [MAC_{dc}])$ a release command
c → b	:	$(rcom, a, D, Pno, [MAC_{cb}])$ a release command
b → a	:	$(rcom, a, D, Pno, [MAC_{ba}])$ a release command

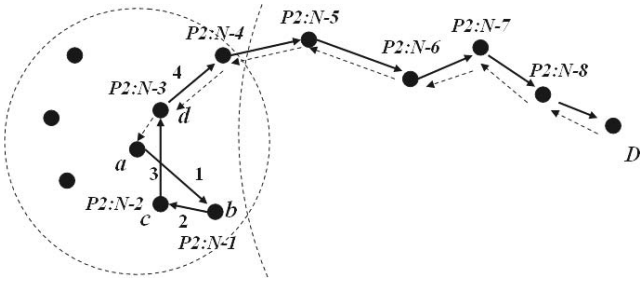


Fig. 3 The dash line shows the reverse traveling in the found path. Node b and c are not used for transmission, and will be released. The *path circle* [4] is eliminated, since node d directly sends the acknowledgement to node a.

6.4 Route Maintenance

Route maintenance mechanism detects malfunctioning, dead or subverted nodes along the routing path. In SecuTPGF, each node along the path forwards the data to the next hop node and then attempts to confirm that the data was received by the next hop node. If, after a limited number of local retransmissions of the data, a node in the route is unable to make this confirmation, it propagates a *route error message* (RERR) to the source node to inform that the link is broken. The initiator of *route error message* computed a MAC using a non interactive key. Upon receiving a *route error message*, the source authenticates the RERR and then may re-initiate the route discovery process for the destination. For example if node 'a' detects a link failure to its next hop node 'b', the generated RERR format is

$$k_{AS} = H([s]P_A, P_S) \tag{7}$$

$$RERR = MACk_{AS}(rerr, a, S, b, Ti_A) \tag{8}$$

where P_A and P_S are public key of the initiator node 'a' and the source node S, Ti_A is the bootstrapping time of node 'a'.

7. Security Analysis

In this section, we discuss attacks in which an adversary interferes the routing protocol from *outside and inside* and show how SecuTPGF prevents those attacks.

7.1 Outsider Adversary

An outsider adversary uses unauthorized nodes to attack the communication of some nodes, which is made easily by the usage of wireless channels.

- **Impersonation**: By securing neighbor discovery, adver-

saries cannot impersonate malicious nodes into WSNs. Because only legitimate nodes have TA-cleared private keys and are able to achieve mutual authentications. Also, adding the bootstrapping time in the public key can limit the period of a new node joining the WSN. Only if the node that has private key that corresponds to its ID and bootstrapping time can join the WSN. After that, it becomes an old node. Such mutual authentication also prevents the Sybil attacks, the identity replication attack and the wormhole attack. An adversary in fact could compromise existing nodes to introduce malicious new nodes. But the malicious new nodes do not have proper bootstrapping time and are not allowed to join the network. If an adversary compromising a new node during its bootstrapping phase, it has access to the secret keys of the new node and might introduce malicious nodes to launch attacks.

- **Fabrication and Modification**: In SecuTPGF, fabricated routing messages may include *route request, Acknowledgment* and *step back and mark* messages generated by malicious nodes. These messages cannot be injected into the network by unauthorized nodes, because SecuTPGF only receives each routing message from authenticated neighbors that are in its neighbor table. If the attacker also modifies the *request* or *Acknowledgment* message, such tampering will be detected since MAC checking will be failed.
- **Routing Loops and Location Spoofing**: In SecuTPGF, each participating node is authenticated therefore impersonation is not feasible. Location spoofing is also avoided because only legitimate nodes are allowed to join the network. It is possible for a compromised node to launch location spoofing attack, which affects only the localized part of the network.

7.2 Insider Adversary

Usually sensor nodes are not physically protected. They might be compromised by adversaries, unless nodes are capable of tamper proof hardware which triggers some type of self destruct mechanism upon attempted compromise. In this subsection, we focus on insider adversaries, in which a WMSN node is captured and compromised by the attacker. Insider attacks are more difficult to detect and prevent, thus our SecuTPGF proposal cannot avoid but limit the impact of these attacks from causing widespread damage in the whole network. Here, we discuss insider attacks specific to TPGF and the proposed solution in the following subsections.

- **Wormhole Attacks**: We mitigate wormhole attack by using a technique similar to Packet leashes [16]. During neighbor discovery phase, a node checks the maximum allowed distance which is approximately its transmission radius, before adding a neighbor into its neighbor table.
- **Sybil Attacks**: In SecuTPGF, an adversary cannot join the false IDs into the network as it does not have a TA cleared private key, so it fails to authenticate the false IDs. Thus in our proposal this attack is no longer feasible.
- **Node Replication Attack**: In SecuTPGF, we assume that a sensor node can sustain a certain time interval before it is compromised, which is also assumed by previous work [11]. With this assumption replication attack can be prevented because the replicated node bootstrapping time is out of range, thus it cannot join the WMSN network. However, if an adversary could compromise a sensor node within its bootstrapping time, it may introduce new nodes with the keys of a compromised node and deploy in different parts of the WSN. These new nodes can then be used to launch other attacks. We limit the impact of this attack by using consistency checking at the base station. Since in TPGF, all the found routing paths are node-disjoint routing paths, and if a node participates in more than one routing path simultaneously, definitely the node is a replicated node and thus its ID will be revoked from the WSN.
- **Selective Forwarding**: We use neighbor monitoring in the promiscuous mode to defend against selective forwarding attacks. Operating in promiscuous mode permits overhearing wireless transmissions of one-hop neighbors. Let's assume nodes A, B, and C be successive hops on a routing path. When a node A transmits a data packet to its next-hop neighbor B, node A will overhear the transmission from B to check whether node B has really transmitted the data message to B's next-hop neighbor which is C. Therefore, A can detect if B fails to forward or may forward the message, but not to the intended node C. By monitoring the behavior of the next-hop neighbor, if the legitimate previous hop node A decides that its next-hop neighbor B is a malicious node, it will send a routing failure back to the source node and blacklist node B's ID. The source node verifies routing failure message and then initiates another route discovery. In situation both B and C are malicious, B can forward the message correctly to C, and C drops the message. A cannot identify B is malicious, as one solution an *end-to-end Acknowledgment message* from the base station for every successful message received, but this may incur additional delay for streaming multimedia data. Our proposal cannot defend such kind of colluding attackers.

8. Simulation and Evaluation

Evaluation of SecuTPGF is analyzed in WSNs simulator NetTopo [20] [*10], in which the TPGF source code is available. We modified the simulator code to prevent malicious nodes without affecting the TPGF routing principles. In the simulation, the network size is fixed in 800 M×600 M and the sensor node transmission radius is 80 M. A source node is deployed at location (50, 50) and the base station is deployed at location (750, 550). The objective of this evaluation is to compare the routing performance of insecure TPGF protocol against our proposed SecuTPGF. We select *the end-to-end delay* (*routing path length*)*, and percentage of found path free of malicious node* as the indicators of routing performance. The comparative evaluation of the two routing protocols is done for various combinations of node density, and the presence of malicious nodes.

[*10] NetTopo is released as an open source sensor network simulator on the SourceForge: http://sourceforge.net/scm/?type=cvs&group_id=224160.

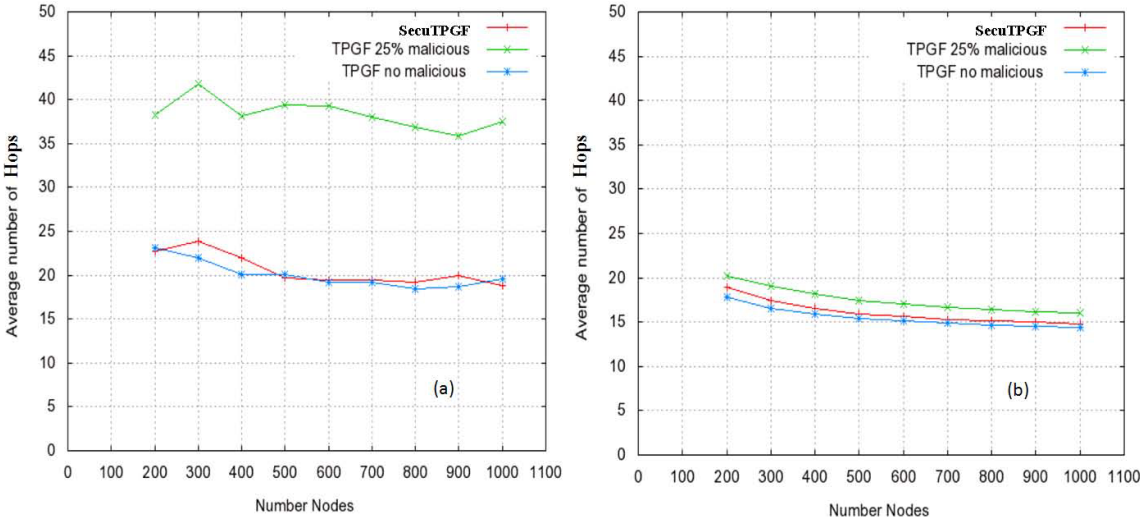


Fig. 4 Average number of hops with 25 percent malicious nodes: (a) Before optimization; (b) After optimization.

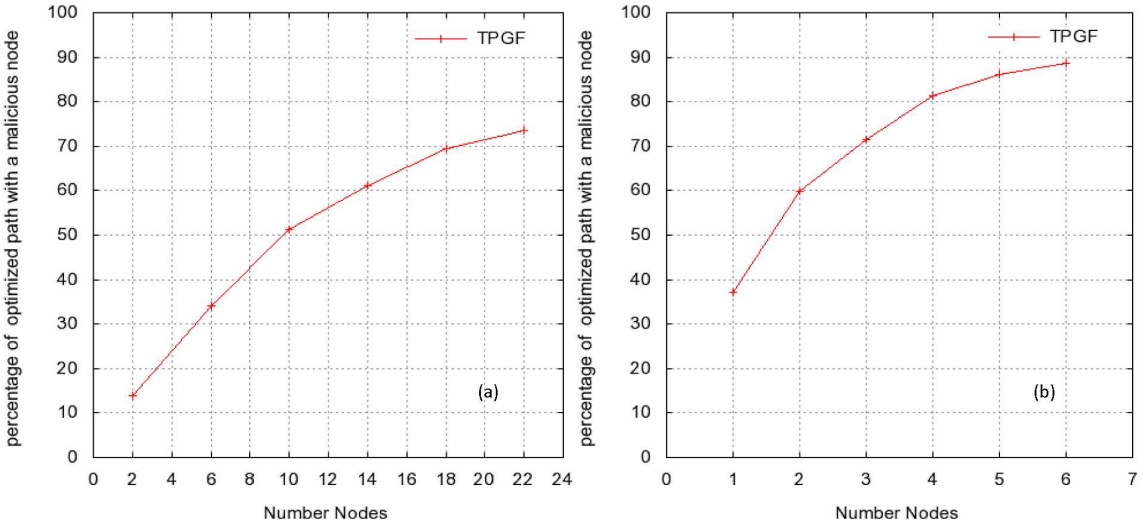


Fig. 5 Probability of a path with at least one malicious node: (a) Without virtual identities; (b) With each advertises 3 virtual identities.

8.1 Effects of Malicious Nodes In the Found Path Length During Route Discovery

In violation of TPGF routing, malicious nodes may increase the end-to-end delay of the message by randomly forwarding the *request message* and avoiding path optimization in the *Acknowledgment message*. In this attack, if a malicious node is the forwarding node of request message, it chooses the next-hop node *randomly* or to make it worse, it chooses the *farthest* node from the base station among all its neighbor nodes and forwards the *request message*. During the reverse travelling in the found routing path, the malicious node does not perform *label based optimization* that eliminates the path circles, it simply sends the *Acknowledgement* to its previous hop.

The effect of this attack on TPGF and SecuTPGF is studied in NetTopo simulation with 25% of malicious nodes on varying number of stationary sensor nodes. The sensor nodes number is changed from 200 to 1000 with 100 steps. Simulation results are collected by averaging the computed number of paths and path length from 100 runs using 100 different random seeds for network deployment.

Figure 4 shows the simulation results before and after applying optimizations on the average number hops of found paths. The average path length found by insecure TPGF routing grows as the malicious nodes force the insecure protocol to route in incorrect directions. The average number of hops for SecuTPGF routing with 25% malicious nodes is a little bit higher than that of TPGF in attack free environment, because SecuTPGF avoids malicious nodes for routing. The TPGF average path length is reduced after optimization, because there is a chance to remove malicious nodes when the honest nodes perform path optimization.

8.2 Percentage of Found Path With a Malicious Node

We evaluate the chance of an adversary to be selected in one of the paths generated by TPGF routing. The percentage is computed as the number of path that contains at least one malicious node to the total number of path generated. The simulation is performed with 500 stationary sensor nodes and varying number of malicious nodes. To increase the probability of an adversary to be involved the routing path, we deploy the malicious nodes ran-

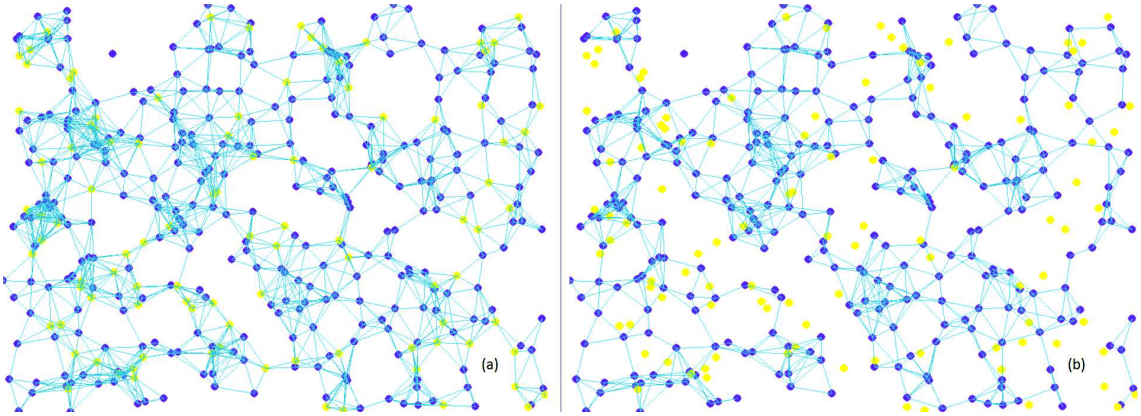


Fig. 6 The deployed WMSN network connectivity: (a) Without securing the *neighbor discovery* process; (b) With secured the *neighbor discovery* process.

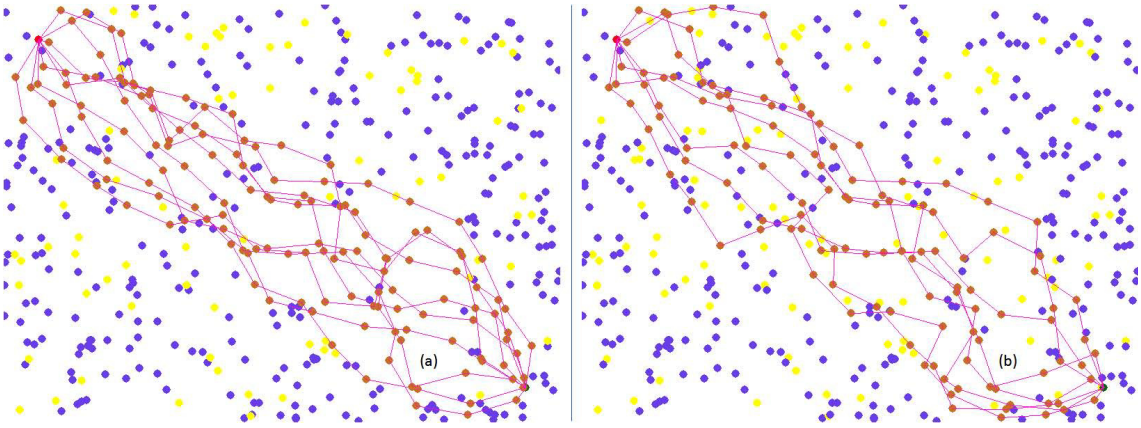


Fig. 7 (a) Ten transmission paths are found by executing TPGF; (b) Eight transmission paths are found by executing SecuTPGF.

domly near the direct line between the source node (50, 50) and base station (750, 550). To further increase the chance of an adversary to be in the path, each adversary creates 3 virtual (Sybil) identities randomly located around itself in a circle with a radius of the radio transmission range. **Figure 5** (a) is the simulation result for varying the number of malicious nodes and Fig. 5 (b) is the simulation result for varying the number of malicious nodes with each of them creates 3 virtual identities. In SecuTPGF, the adversaries and its virtual Sybil Identities fail to authenticate and cannot join the WSN, hence all the found paths are free of malicious node.

9. Demonstration and Comparison of TPGF and SecuTPGF

In this section, we show a number of figures, which are snapshots of the execution results of both TPGF and SecuTPGF in NetTopo. The blue color nodes are normal nodes. The yellow color nodes are malicious nodes.

9.1 Secured Neighbor Discovery

Figure 6 (a) shows the network connectivity, in which the malicious nodes are included in normal sensor nodes' neighbor lists. Figure 6 (b) shows the network connectivity after securing the *neighbor discovery* process.

9.2 Secured Route Discovery

Figure 7 (a) shows the execution of TPGF in NetTopo, in which malicious nodes are included in the transmission paths. Figure 7 (b) shows the execution of SecuTPGF in NetTopo, in which no malicious nodes are included in the transmission paths.

10. Conclusion and Future Work

Information explosion in ubiquitous computing [24] is caused by various new means for information gathering, e.g., wireless multimedia sensor networks. When people consume information provided by various resources and services, security should be guaranteed to ensure the correctness of information. Security in WMSNs is still a new and unexplored research field [5]. In this paper, the proposed SecuTPGF exactly followed the original TPGF protocol's routing mechanisms and applied ID-NIKDS scheme to provide both node authentication and symmetric key establishment, which allowed it to secure the neighbor discovery and route discovery. Current SecuTPGF is not a perfectly designed version yet, since some difficult attacks still cannot be handled. But, we believe that our effort for investigating the first secure routing protocol (SecuTPGF) in WMSNs had already brought a great contribution and impact to existing WSNs research community, in which new discussions and research ideas will appear soon from both the industry and the academic world.

In our future work, we are interested in bringing SecuTPGF

into a much more realistic network model, in which sensor nodes are duty-cycled to sleep for energy conservation [21]. There are also various attacks for sleep scheduling algorithms, which should be carefully secured, since time-varying network topologies produced by sleep scheduling algorithms in WSNs are the basic network graphs for SecuTPGF algorithm to explore the available multiple routing paths.

Acknowledgments The research work in this paper was supported: in part by Grant-in-Aid for Scientific Research (S) (21220002) of the Ministry of Education, Culture, Sports, Science and Technology, Japan, and in part by Lion project supported by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2), and in part by French ANR funded research project (PAIRSE). Lei Shu is the corresponding author.

Reference

[1] Shu, L., Zhang, Y., Zhou, Z., Hauswirth, M., Yu, Z. and Hynes, G.: Transmitting and Gathering Streaming Data in Wireless Multimedia Sensor Networks within Expected Network Lifetime, *ACM/Springer Mobile Networks and Applications* (*MONET*), Vol.13, No.3–4, pp.306–322 (2008).

[2] Mulugeta, T., Shu, L., Hauswirth, M., Chen, M., Hara, T. and Nishio, S.: Secure Two Phase Geographic Forwarding Routing Protocol in Wireless Multimedia Sensor Networks, *Proc. Intl. Conf. Global Communication* (*Globecom 2010*), Miami, Florida, USA (2010).

[3] Die, W. and Hellman, M.: New directions in cryptography, *IEEE Trans. Inf. Theory*, Vol.22, No.6, pp.644–654 (1976).

[4] Shu, L., Zhang, Y., Yang, L.T., Wang, Y., Hauswirth, M. and Xiong, N.X.: TPGF: Geographic Routing in Wireless Multimedia Sensor Networks, *Telecommunication Systems*, Vol.44, No.1–2, pp.79–95 (2010).

[5] Guerrero-Zapata, M., Zilan, R., Barcel-Ordinas, J., Bicakci, K. and Tavli, B.: The Future of Security in Wireless Multimedia Sensor Networks, *Telecommunication Systems*, Vol.45, No.1, pp.77–91 (2010).

[6] Karlof, C. and Wagner, D.: OhHelp: Secure routing in wireless sensor networks: Attacks and countermeasures, *Proc. Intl. Workshop on First Sensor Network Protocols and Applications in Conjunction with ICC 2003*, AK, USA, pp.113–127 (2009).

[7] Sakai, R., Ohgishi, K. and Kasahara, M.: Cryptosystems based on pairing, *Proc. Intl. Symposium on 2000 Cryptography and Information Security*, Okinawa, Japan, pp.26–28 (2000).

[8] Karp, B. and Kung, H.T.: GPSR: Greedy perimeter stateless routing for wireless networks, *Proc. Intl. Conf. on Mobile Computing and Networking*, Boston, USA, pp.243–254 (2000).

[9] Piotrowski, K., Langendoerfer, P. and Peter, S.: How public key cryptography influences wireless sensor node lifetime, *Proc. Intl. Workshop on Security in Ad Hoc and Sensor Networks* (*SASN 2006*) *in Conjunction with the CCS 2006*, pp.169–176, ACM, Alexandria, VA, USA (2006).

[10] Niculescu, D. and Nath, B.: Ad hoc positioning system (APS) using AOA, *Proc. Intl. Conf. 22nd Computer and Communications* (*INFOCOM 2003*), pp.1734–1743, IEEE, San Francisco, USA (2003).

[11] Zhang, Y., Liu, W., Lou, W. and Fang, Y.: Location based security mechanisms in wireless sensor networks, *IEEE Journal on Selected Areas in Communications*, Vol.24, No.2, pp.247–260 (2006).

[12] Karlof, C., Sastry, N. and Wagner, D.: TinySec: A Link Layer Security Architecture for Wireless Sensor Networks, *Proc. Intl. Conf. Embedded Networked Sensor Systems* (*SenSys 2004*), pp.62–72, ACM, Baltimore, Maryland, USA (2004).

[13] Shamir, A.: Identity-based cryptosystems and signature schemes, *Proc. Intl. Conf. Advances in Cryptology* (*CRYPTO 1984*), Santa Barbara, California, USA, pp.47–53 (1984).

[14] Newsome, J., Shi, E., Song, D. and Perrig, A.: The sybil attack in sensor networks: analysis & defenses, *Proc. Intl. Conf. 3rd Information Processing in Sensor Networks* (*IPSN 2004*), IEEE/ACM, pp.259–268, Berkeley, California, USA (2004).

[15] Parno, B., Perrig, A. and Gligor, V.: Distributed detection of node replication attacks in sensor networks, *Proc. Intl. Symp. Security and Privacy* (*S&P 2005*), pp.49–63, IEEE, Berkeley/Oakland, California, USA (2004).

[16] Hu, Y., Perrig, A. and Johnson, D.B.: Packet leashes: A defense against wormhole attacks in wireless networks, *Proc. Intl. Conf. 22nd Computer and Communications* (*INFOCOM 2003*), pp.1976–1986, IEEE, San Francisco, USA (2003).

[17] Joux, A.: The weil and tate pairings as building blocks for public key cryptosystems, *Proc. Intl. Symp. Security and Privacy* (*S&P 2005*), pp.49–63, IEEE, Berkeley/Oakland, California, USA (2004).

[18] Perrig, A., Stankovic, J. and Wagner, D.: Security in wireless sensor networks, *Comm. ACM*, Vol.47, No.6, pp.53–57 (2004).

[19] Oliveira, L.B., Aranha, D.F., Gouvêa, C.P.L., Scott, M., Câmara, D.F., López, J. and Dahab, R.: TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks, *Computer Communications*, Vol.34, No.3, pp.485–493 (2011).

[20] Shu, L., Hauswirth, M., Chao, H., Chen, M. and Zhang, Y.: NetTopo: A Framework of Simulation and Visualization for Wireless Sensor Networks, *Elservier, Ad Hoc Networks*, Vol.9, No.5, pp.799–820 (2011).

[21] Yuan, Z., Shu, L., Wang, L. Hara, T. and Qin, Z.: A Balanced Energy Consumption Sleep Scheduling Algorithm in Wireless Sensor Networks, *Proc. Intl. Conf. 7th Wireless Communications & Mobile Computing* (*IWCMC 2011*), Istanbul, Turkey (2011).

[22] Toussaint, G.T.: The relative neighborhood graph of a finite planar set, *Pattern Recogn.*, Vol.12, pp.261–268 (1980).

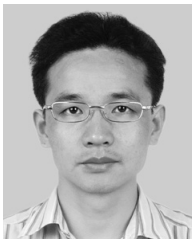
[23] Gabriel, K. and Sokal, R.: A new statistical approach to geographic variation analysis, *Systematic Zoology*, Vol.18, pp.259–278 (1969).

[24] Ngo, H., Nguyen, N., Le, X., Shu, L. and Lee, S.: A Survey on Middleware for Context Awareness in Ubiquitous Computing Environment, *Journal of Korean Information Processing Society Review*/ (*KIPS*), pp.97–121 (2003).

Taye Mulugeta received his B.Sc. degree in electrical engineering from Jimma University, Jimma, Ethiopia, in 2004. He received his M.S. degree in electrical and computer engineering from Addis Ababa University, Addis Ababa, Ethiopia, in 2010. His current research interests include wireless multimedia sensor network, security, digital signal processing and communication.

Lei Shu is currently a Specially Assigned Researcher in the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, Japan. He received his Ph.D. degree from Digital Enterprise Research Institute (DERI), the National University of Ireland, Galway (NUIG), in 2010. He has published over 100 papers. He had been awarded the Globecom 2010 Best Paper Award. His research interests include sensor network, multimedia communication, and security. He is a member of IEEE.

Manfred Hauswirth is Vice-Director of Digital Enterprise Research Institute (DERI), Galway, Ireland and professor at the National University of Ireland, Galway (NUIG). He holds a M.S. (1994) and a Ph.D. (1999) in computer science from the Technical University of Vienna. His main research interests are on semantic sensor networks, sensor networks middleware, peer-to-peer systems, Semantic Web services, and distributed systems security. He has published over 100 papers. He is a member of IEEE and ACM.



Zhangbing Zhou a Post-doctoral Research Fellow at TELECOM SudParis, France. He received his Ph.D. from Digital Enterprise Research Institute (DERI), the National University of Ireland, Galway (NUIG). His research interests include service-oriented computing, cloud computing, and sensor network middleware. He has published around 30 research papers.

Shojiro Nishio received his B.E., M.E., and Ph.D. degrees from Kyoto University, Kyoto, Japan. He is currently a Full Professor in the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, Japan. He served as a Trustee and Vice President of Osaka University from August 2007 to August 2011. His areas of expertise are in database systems, distributed systems, knowledge discovery, and multimedia systems. Dr. Nishio has authored or co-authored more than 560 refereed journal or conference papers, and served as the Program Committee Co-Chairs for several international conferences including DOOD 1989, VLDB 1995, and IEEE ICDE 2005. He has served and is currently serving as an editor of several international journals including IEEE Transaction on Knowledge and Data Engineering, the VLDB Journal, ACM Transaction on Internet Technology, and Data & Knowledge Engineering. Dr. Nishio is a fellow of IEEE, IPSJ, and IEICE.