Network and Complex Systems
ISSN 2224-610X (Paper) ISSN 2225-0603 (Online)
Vol.5, No.1, 2015


# Secure Leader Election for Intrusion Detection in MANET

G Murali[1] R Sivaram Prasad[2] K V Bhaskar Rao[3]
1.Asst Professor, Department of Computer Science Engineering, JNTUACE Pulivendula, AP, India
Muralig521@gmail.com
2.Research Director, Department of Computer Science Engineering, Acharya Nagarjuna University, AP, India
vaminenisivaram@yahoo.co.in
3.M-Tech,Department of Computer Science and Engineering, JNTUACE Pulivendula, AP, India
Bhaskarrao05@gmail.com

**Abstract**
This paper shows the leader election in presence of selfish nodes. To balance the resource consumption among the nodes and prolong the life time of manet. Nodes with highest resource should be elected as a leaders. But there is obstacles in doing so. First, node may lie about its available resources. Second, electing multiple leaders may leads to additional overhead. Considering first, solution is based on Mechanism design. It provides incentives to the nodes to honestly participate in the election process. The amount of incentives provide to the nodes is based on Vickrey, Clarke, and Groves (VCG) model. Considering second, series of algorithms are there to address optimal leader election.

## 1 Introduction

Unlike traditional networks, the Mobile Ad hoc Networks (MANETs) have no fixed chokepoints/bottlenecks where Intrusion Detection Systems (IDSs) [1], [2]can be deployed. Hence, a node may need to run its own IDS and cooperate with others to ensure security. This is very inefficient in terms of resource consumption since mobile nodes are energy-limited. To overcome this problem, a common approach is to divide the MANET into a set of 1-hop clusters where each node belongs to at least one cluster. The nodes in each cluster elect a leader node (cluster head) to serve as the IDS for the entire cluster. The leader-IDS election process can be either random [3] or based on the connectivity [4]. Both approaches aim to reduce the overall resource consumption of IDSs in the network. Unfortunately, with the random model, each node is equally likely to be elected regardless of its remaining resources. The connectivity index-based approach elects a node with a high degree of connectivity even though the node may have little resources left. With both election schemes, some nodes will die faster than others, leading to a loss in connectivity and potentially the partition of network. Next, we motivate further discussions through a concrete example.

## Example

The below figure 1 illustrates the MANET consists of 10 nodes from n1 to n10. These 10 nodes are organized into 5 one hop clusters. Here N5 and N9 belongs to more than one cluster. Each node differs in energy levels which is considered as secret information. At this point it is not desirable to have N5 and N9 as a leader. Using random model N5 and N9 are equally likely probable with others. N5 and N9 will be elected as a leader under connectivity index model. More over general approach for electing leader under resources is also failed because node may lie about energy levels. Finally nodes N5 and N9 are elected as leader and they refuse to run IDS. The consequences of such a refusal will lead normal nodes to launch their IDS, and thus die faster.

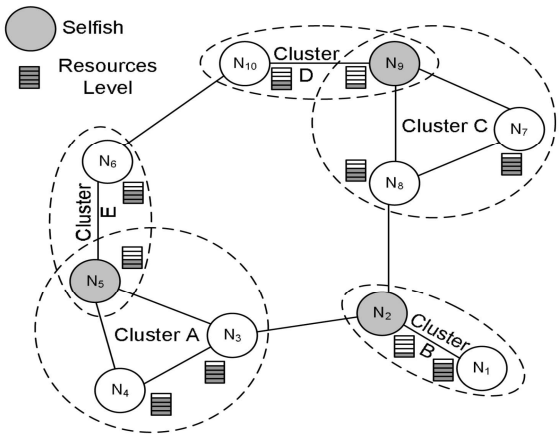


Fig 1 : Scenario of leader election in manet

 

## 2 Problem Statement

The aim is to achieve most efficient node as a leader. The following are the challenges arises. First, resource level should be considered as private information. As a result nodes may expose fake information. Second, node may behave properly during election procedure but after electing as leader it may not provide ids service.

In our model, let's consider MANET as undirected graph G = (N,L).where N denotes number of nodes in MANET and L denotes set of links in MANET. The Cost of analyzing the nodes is given by C = {c1,c2,…cn).The election procedure is denoted by the function vtk(C,i). If the node i votes for node k then vtk(C,i)=1, otherwise it is 0. Consider each leader allocates same budget to all nodes that elects it. If budget=15 and it gets 4 votes , then total sampling budget is 60. Goal of reducing the global cost of analysis function is expressed by the SCF and is given below

SCF=S(C) = min $\sum ck\cdot(\sum vtk(C,i)\cdot B)$

## 3 Mechanism Design

Mechanism design is a subfield of microeconomics and game theory [5]. Mechanism design uses game theory [6] tools to achieve the desired goals. The main difference between game theory and mechanism design is that the former can be used to study what could happen when independent players act selfishly. On the other hand, mechanism design allows a game designer to define rules in terms of the SCF such that players will play according to these rules. The balance of IDS resource consumption problem can be modeled using mechanism design theory with an objective function that depends on the private information of the players. In our case, the private information of the player is the cost of analysis which depends on the player's energy level. Here, the rational players select to deliver the untruthful or incomplete information about their preferences if that leads to individually better outcomes[10]. The main goal of using mechanism design[9] is to address this problem by: 1) designing incentives for players (nodes) to provide truthful information about their preferences over different outcomes and 2) computing the optimal system-wide solution, which is defined according to (1).

### 3.1 Mechanism model

We treat the IDS resource consumption problem as a
game where the N mobile nodes are the agents/players.
Each node plays by revealing its own private information (cost of analysis) which is based on the node's type. The type is drawn from each player's available type set {Normal; Selfish} . Each player selects his own strategy/type according to how much the node benefits from type. If the player's strategy is normal, then the node reveals the true cost of analysis. If it is selfish node, it reveals fake cost of analysis. Based on cost of analysis mechanism design calculates outputs and provides payments/punishments.
In the following sections, we will formulate the following components:

1. Cost of analysis function: It is needed by the nodes to compute the valuation function.
2. Reputation system: It is needed to show how:
   a. Incentives are used once they are granted.
   b. Misbehaving nodes are catched and punished.
3. Payment design: It is needed to design the amount of incentives that will be given to the nodes based on VCG.

#### 3.1.1 Cost of analysis

During the design of the cost of analysis function, the following two problems arise: First, the energy level is considered as private and sensitive information and should not be disclosed publicly. Such a disclosure of information can be used maliciously for attacking the node with the least resources level. Second, if the cost of analysis function is designed only in terms of nodes' energy level, then the nodes with the low energy level will not be able to contribute and increase their reputation values. To solve the above problems, we design the cost of analysis function with the following two properties: Fairness and Privacy. The former is to allow nodes with initially less resources to contribute and serve as leaders in order to increase their reputation. On the other hand, the latter is needed to avoid the malicious use of the resources level, which is considered as the most sensitive information. To avoid such attacks and provide fairness, the cost of analysis is designed based on the reputation value, the expected number of time slots that a node wants to stay alive in a cluster, and energy level. Note that the expected number of slots and energy level are considered as the nodes' private information.
The lifetime of a node can be divided into time slots. Each node i is associated with an energy level, denoted by Ei, and the number of expected alive slots is denoted by nTi. Based on these requirements, each node i has a power factor PFi = Ei/nTi.
The reputation of node i is denoted by Ri. Every node
has a sampling budget based on its reputation. This is
indicated by the percentage of sampling

$$PS_i = \frac{Ri}{\sum_{i=1}^{N} Ri} \quad (1)$$

 

The $c_i$ notation represents the cost of analysis for a single packet and Eids is used to express the energy needed to run the IDS for one time slot. The cost of analysis of each node can be calculated based on energy level. However, we considered energy level, expected lifetime and the present PS of node to calculate the cost of analysis. We can extend the cost of analysis function to more realistic settings by considering the computational level and cost of collecting and analyzing traffic. Our cost-of-analysis function is formulated as follows:

$$C_i = \begin{cases} \infty, \\ \frac{PSi}{PFi} \end{cases} = \frac{\frac{Ri}{\sum_{i=1}^{N} Ri} \times nTi}{Ei}, \quad \text{Otherwise} \quad (2)$$

According to the above formulation, the nodes have an infinite cost of analysis if its remaining energy is less than the energy required to run the IDS for one time slot. This means that its remaining energy is too low to run the IDS for an entire time slot. Otherwise, the cost of analysis is calculated through dividing the percentage of sampling by the power factor. The cost of analysis c is proportional to the percentage of sampling and is inversely proportional to the power factor. The rationale behind the definition of the function is the following. If the nodes have enough PS, they are not willing to loose their energy for running the IDS. On the other hand, if PF is larger, then the cost-of-analysis becomes smaller since the nodes have higher energy levels.

**3.1.2 Reputation System Model**

Before we design the payment, we need to show how the payment in the form of reputation can be used to: 1) motivate nodes to behave normally and 2) punish the misbehaving nodes. Moreover, it can be used to determine whom to trust.

To motivate the nodes in behaving normally in every election round, we relate the cluster's services to nodes' reputation. This will create a competition environment that motivates the nodes to behave normally by saying the truth. To enforce our mechanism, a punishment system is needed to prevent nodes from behaving selfishly after the election. Misbehaving nodes are punished by decreasing their reputation, and consequently, are excluded from the cluster services if the reputation is less than a predefined threshold. As an extension to our model, we can extend our reputation system to include different sources of information such as routing and key distribution with different assigned weights. Fig. 2 shows the abstract model of our reputation system where each node has the following components:

- Monitor or watchdog: It is used to monitor the behavior of the elected leader. To reduce the overall resource consumption, we randomly elect a set of nodes, known as checkers, to perform the monitoring process. The selected checkers mirror a small portion of the computation done by the leader, so the checkers can tell whether the leader is actually carrying out its duty. We assume that the checkers are cooperative because the amount of computation they conduct for monitoring the leader only amounts to a marginal resource consumption, which is dominated by the benefit of receiving intrusion detection service from the leader [7].
- Information exchange: It includes two types of information sharing:
    1. The exchange of reputation with other nodes in other clusters (i.e., for services purposes).
    2. To reduce the false positive rate, the checkers will exchange information about the behavior of the leader to make decision about the leader's behavior.
- Reputation system: It is defined in the form of a table that contains the ID of other nodes and their respective reputation R. The node that has the highest reputation can be considered as the most trusted node and is given priority in the cluster's services. Therefore, the rational nodes are motivated to increase their reputation value by participating in the leader election.
- Threshold check: It has two main purposes:
    1. To verify whether nodes' reputation is greater than a predefined threshold. If the result is true then nodes' services are offered according to nodes' reputation.
    2. To verify whether a leader's behavior exceeds a predefined misbehaving threshold. According to the result, the punishment system is called.
- Service system: To motivate the nodes to participate in every election round, the amount of detection service provided to each node is based on the node's reputation. Each elected leader has a budget for sampling, and thus, only limited services can be offered. This budget is distributed among the nodes according to their reputation. Besides, this reputation can also be used for packet forwarding. Packets of highly
  reputed nodes should always be forwarded. On the other hand, if the source node has an unacceptably low reputation, then its packet will have less priority. Hence, in every round, nodes will try to increase their reputation by becoming the leader in order to increase their services.
- Punishment system: To improve the performance and reduce the false positive rate of checkers in catching and punishing a misbehaving leader, we have formulated in [7] a cooperative game-theoretical

 

model to efficiently catch and punish misbehaving leaders with low false positive rate. Our catch-and-punish model was made up of k detection levels, representing different levels of selfish behaviors of the leader-IDS. This enables us to better respond to the misbehaving leader-IDS depending on which detection level it belongs to. Hence, the percentage of checkers varies with respect to the detection level. Once the detection exceeds a predefined threshold, the leader will be punished by decreasing its reputation value.

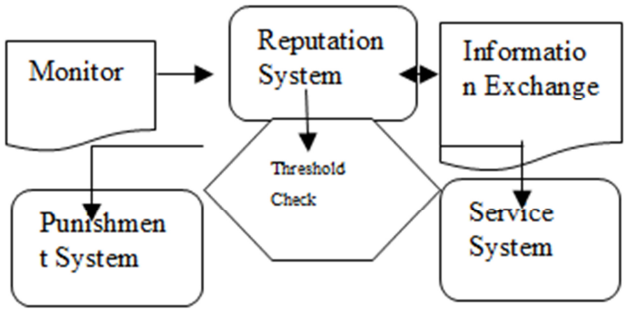


Fig 2 : Reputation System Model

### 3.1.3 CILE Payment Design

In CILE, each node must be monitored by a leader node that will analyze the packets for other ordinary nodes. Based on the cost of analysis vector C, nodes will cooperate to elect a set of leader nodes that will be able to analyze the traffic across the whole network and handle the monitoring process. This increases the efficiency and balances the resource consumption of an IDS in the network. Our mechanism provides payments to the elected leaders for serving others (i.e., offering the detection service). The payment is based on a per-packet price that depends on the number of votes the elected nodes get. The nodes that do not get any vote from others will not receive any payment. The payment is in the form of reputations, which are then used to allocate the leader's sampling budget for each node. Hence, any node will strive to increase its reputation in order to receive more IDS services from its corresponding leader.

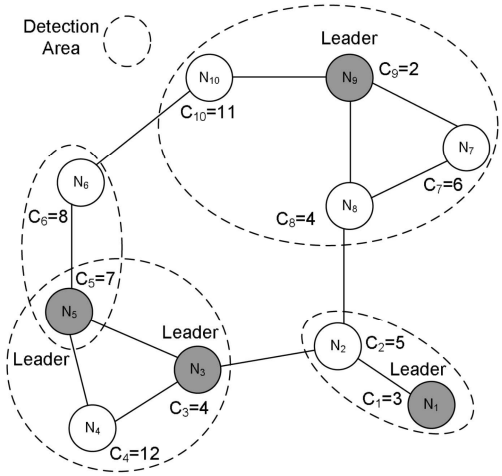


Fig 3: Example of leader election

Using following design of payment, truth telling is the dominant strategy.

$$P_k = \sum_{i \epsilon N} \upsilon tk(C,i) B \rho k, \quad (3)$$

Where

$$\rho_k = c_k + \frac{1}{\sum_{i \epsilon N} \upsilon tk(C,i)} \times \quad (4)$$
$$\left[\sum_{j \epsilon N} cj \sum_{i \epsilon N} \upsilon tj(C|ck = \infty, i) - \sum_{j \epsilon N} cj \sum_{i \epsilon N} \upsilon tj(C,i)\right]$$

**Example 1 :**

To show how the payment is calculated and used, we consider an MANET with 10 nodes, as shown in Fig. 3.

 

Since our model is repeatable, we present the election process at the 10th round. The reputation at the ninth round is given in the first row of Table 1. To elect a new leader in the 11th round, the nodes will first compute their cost of analysis using the cost of analysis function. The corresponding revealed cost is presented in the second row of Table 1. Given the nodes' cost and network topology, node 9 will be the leader among its neighbor since it has the lowest cost of analysis. Equation (3) is used to calculate the payment of node 9, which is in the form of reputation. The payment per packet is ρ9 = 2 + ¼( 8*1 + 4*3 – 2*4) = 5. This is because if node 9's cost is α, then node 10 would have voted for nodes 6 and nodes 7,8 and 9 would have voted for node 8. Hence, the total cost would have been 20 instead of 8. Therefore, the given payment of node 9 is $P_9 = \sum \upsilon_9 B\rho_9 = 4 * 5 * 5 = 100$., where B = 5 packets/sec is the sampling budget. After election, leader N9 distributes the IDS sampling budget over the protected nodes N7, N8, N9, and N10, according to their reputation, as follows: S = { $S_7$ = (90 * 20)/470, $S_8$ = (160*20)/470, $S_9$ = (120*20)/470, $S_{10}$ = (110*20)/470 }.

Table 1 : Leader-Ids Election Example

| Nodes | $N_1$ | N2 | N3 | N4 | N5 | N6 | N7 | N8 | N9 | N10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Reputation 10th | 100 | 140 | 120 | 80 | 120 | 60 | 90 | 160 | 20 | 110 |
| Cost of analysis | 3 | 5 | 4 | 12 | 7 | 8 | 6 | 4 | 2 | 11 |
| Reputation 11th | 140 | 140 | 215 | 80 | 160 | 60 | 90 | 160 | 120 | 110 |

### 3.1.4 CDLE Payment Design

In CDLE, the whole network is divided into a set of clusters where a set of 1-hop neighbor nodes forms a cluster. Here, we use the scheme of [8] to cluster the nodes into 1-hop clusters. Each cluster then independently elects a leader among all the nodes to handle the monitoring process based on nodes' analysis cost. Our objective is to find the most cost-efficient set of leaders that handle the detection process for the whole network. Hence, our social choice function is still as in (1).

To achieve the desired goal, payments are computed using the VCG mechanism[5] where truth-telling is proved to be dominant. Like CILE, CDLE provides payment to the elected node and the payment is based on a per-packet price that depends on the number of votes the elected node gets.

## 4 Leader Election Algorithm

While designing leader election algorithm we have to consider following 1)Design messages that establishes election procedure 2)consider addition and deletion of nodes in the network 3)consider performance overhead while designing.

### 4.1 Objectives and Assumptions

To design leader election algorithm following requirements are needed1)For protecting the nodes, each and every nodes should be monitored by leader.2)To balance resource consumption , the overall cost of analysis should be minimized.

Leader election algorithm has the following assumptions about nodes 1)Every node must aware of its 2 hop neighbours 2)Loosely synchronization between nodes -Every node should maintain (public, private) key pair for establishing secure communication3)Every node should aware of addition and removal of nodes.

### 4.2 Leader Election

Four types of messages are used in election mechanism. Hello-Used to initiate election process, Begin-Election-Used to tell the cost of node, Vote-used by node to elect leader, Acknowledgment-Send by leader to broadcast payment. The following are the notations used for describing algorithm Service-table(k)=It contains list of all nodes that vote for node k, Reputation-table=Every node keeps the record of reputation of all the nodes, Neighbors(k)=List of nodes neighbor to node k, Leadernode(k)=the id of k's leader, Leader(k)= boolean set to true if it is leader, other wise false.

Each and every node sends hello message to all its neighbors and starts timer T1.The hello message contains hash value of cost of analysis. After the timer is expired each node checks hello message. Nodes from whom hello messages are not received are excluded.

Algorithm 1 (Executed by every node)

/* On receiving *Hello*, all nodes reply with their cost */

1. **if** (received *Hello* from all neighbors) **then**
2. Send *Begin-Election* (IDk, costk);
3. **else if**(neighbors(k)=Ø) **then**
4. Launch IDS.
5. **end if**

Up on receiving hello message each node send begin-election message and starts timer T2.Begin-election

 

contains cost of analysis of node.If there is no neighbors then it launches its own Ids.
Algorithm 2 (Executed by every node)
/* Each node votes for one node among the neighbors */
1. **if** ($\forall$ n $\epsilon$ neighbor(k), $\exists$ i $\epsilon$ n : $c_i \leq c_n$) **then**
2. send Vote($ID_k$, $ID_i$ , $cost_{j \neq i}$);
3. *leadernode(k)*:= i;
5. **end if**

After timer T2 is expired , then each node compares the hash value of hello and value received by begin election to verify cost of analysis. Then each node calculates least cost value of neighbors and send vote for node say i.The vote message contain ID of source node,Id of proposed leader and second least cost among neighbors.The the node sets node I as leader.The Second least cost is needed by leader for payment.If the node has leat cost among all the neighbors, then votes to itself and starts timer T3.
Algorithm 3 (Executed by Elected leader node)
/* Send Acknowledge message to the neighbor nodes */
1. Leader(i) := TRUE;
2. Compute Payment, $P_i$;
3. $update_{service-table}$(i);
4. $update_{reputation-table}$(i);
5. Acknowledge = $P_i$ + all the votes;
6. Send Acknowledge(i);
7. Launch IDS
After T3 is expired it calculates payment and send acknowledgment message to all the nodes that vote for it. The acknowledgement message contains payment.

**4.2.1 Adding a Node**

If a new node is entered in to network, it can either launches its own ids or become a normal node to leader node. Four messages are needed to add node to network. They are hello, status, join, acknowledgment. New node sends hello message to all the neighbor nodes. Hello message is same as previous one. Up on receiving hello neighbors send status message. The status contain cost if it is leader node, otherwise it contains the ID of leader node
Algorithm 4 (Executed by neighboring nodes)
/* The neighboring nodes send 'Status' to new node */
1. **if** (leader(k) = TRUE) **then**
2. *Status* := Costk;
3. **else**
4. *Status* := *leadernode(k)*;
5. **end if;**
6. send Status(k, n);
On receiving status message it sends join to leader node. If two of the neighbors are leader nodes then it send join to any of two depending on physical location. We assume that new node has no interest to be a leader because it will not receive any payment. If the new node has least cost it can either launches or send join to leader. On receiving join it sends acknowledgment to new node. It contains payment.

**4.2.2 Removing a Node**

A node may be removed from the network due to many reasons such as mobility, battery depletion. In such a cases neighbor nodes needs to reconfigure the network. Dead message is circulated among neighbor nodes to confirm its deletion, on receiving dead it checks weather it is leader node. If it is leader node then it announces for election. If it is normal node, then leader node updates its serving list.
Algorithm 5 (Executed by neighboring nodes)
/* The neighboring nodes reconfigure the network and */
/* declare new election if necessary*/
1. **if** (leadernode(k) = n) **then**
2. leadernode(k):= NULL;
3. $update_{reputation}$(k);
4. send Begin − Election as in Algorithm 1;
5. **end if;**
6. **if** (leader(k) = TRUE) **then**
7. **if** (n $\epsilon$ service(k)) **then**
8. $update_{service}$();
9. **end if;**

 

10. **end if;**

**Conclusion**

Both approaches random based and connectivity based aimed at electing the leader, but it has some dis-advantages. In both approaches, if the elected leader is selfish nodes it does not provide ids (Intrusion Detection service) service to entire cluster. Hence each node need to run its own ids, lower energy nodes get die by running its own ids results in partition of network. Hence, effective solution is introduced based on mechanism design. It securely elects leader and the elected leader provides ids service for intrusion detection to entire cluster. To honestly participate in every election round, mechanism design provides payments to elected leader. The payments is in the form of reputation. The payments are based on vcg mechanism. More the payment the leader gets it can analyze more number of packets for intrusion detection. Mechanism design reduce the reputation of node when it does not provide ids service. Thus it is the optimal solution to balance the resource consumption ( Energy level) of all the nodes to prolong the life time of manet.

**References**

[1] F. Anjum and P. Mouchtaris, Security for Wireless Ad Hoc Networks. John Wiley and Sons, Inc., 2007.
[2] P. Brutch and C. Ko, “Challenges in Intrusion Detection for Wireless Ad-Hoc Networks,” Proc. IEEE Symp. Applications and the Internet (SAINT) Workshop, 2003.
[3] Y. Huang and W. Lee, “A Cooperative Intrusion Detection System for Ad Hoc Networks,” Proc. ACM Workshop Security of Ad Hoc and Sensor Networks, 2003.
[4] O. Kachirski and R. Guha, “Efficient Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks,” Proc. IEEE Hawaii Int’l Conf. System Sciences (HICSS), 2003.
[5] A. Mas-Colell, M. Whinston, and J. Green, Microeconomic Theory. Oxford Univ. Press, 1995.
[6] P. Morris, Introduction to Game Theory, first ed. Springer, 1994.
[7] H. Otrok, N. Mohammed, L. Wang, M. Debbabi, and P. Bhattacharya, “A Game-Theoretic Intrusion Detection Model for Mobile Ad-Hoc Networks,” J. Computer Comm., vol. 31, no. 4, pp. 708-721, 2008.
[8] P. Krishna, N.H. Vaidya, M. Chatterjee, and D.K. Pradhan, “A Cluster-Based Approach for Routing in Dynamic Networks,” Proc. ACM SIGCOMM Computer Comm. Rev., 1997.
[9] L. Hurwicz and S. Reiter, Designing Economic Mechanisms, first ed. Cambridge Univ. Press, 2008.
[10] J. Shneidman and D. Parkes, “Specification Faithfulness in Networks with Rational Nodes,” Proc. ACM Symp. Principles of Distributed Computing, 2004.

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:
http://www.iiste.org

## CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

**Prospective authors of journals can find the submission instruction on the following page:** http://www.iiste.org/journals/ All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

## MORE RESOURCES

Book publication information: http://www.iiste.org/book/

Academic conference: http://www.iiste.org/conference/upcoming-conferences-call-for-paper/

## IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar