World Applied Sciences Journal 29 (Computer Sciences, Engineering and Its Applications): 234-240, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.29.csea.2274

Efficient Node Replica Detection in Wireless Sensor Networks

R. Udayakumar, V. Khanaa and K.P. Thooyamani

School of Computing Science, Bharath University - 73, India

Abstract: Wireless sensor nodes lack hardware support for tamper resistance and are often deployed in unattended environments, thus leaving them vulnerable to capture and compromise by an adversary. In a node replication attack, an adversary can easily capture even a single node and inserts duplicated nodes at any location in the network. If no specific detection mechanisms are established, the attacker could lead many insidious attacks such as subverting data aggregation protocols by injecting false data, revoking legitimate nodes and disconnecting the network if the replicated nodes are judiciously placed at chosen locations.Without an effective and efficient detection mechanism, these replicas can be used to launch a variety of attacks that undermine many sensor applications and protocols. In this paper, we present a novel distributed approach called Localized Multicast for detecting node replication attacks. The efficiency and security of our approach are evaluated via simulation. Our analysis and simulations demonstrate our protocol is effective even when there are a large number of compromised nodes and at the same time achieves a higher probability of detecting node replicas

Key words: WSNS · Node Joins · Multicasting,replication · Data aggregation

INTRODUCTION

Wireless sensor networks (WSNs) are composed of a large number of low-cost, low-power and multi-functional sensor nodes that communicate at short distance through wireless links. They are usually deployed in an open and uncontrolled environment where attackers may be present. Due to the use of low-cost materials, hardware components are not tamper resistant and an adversary could access a sensor's internal state. Thus an adversary could access a sensor's internal state. An adversary can easily capture a single node, replicate it indefinitely and insert duplicated nodes at any location in the network. Node replication attack occurs when a single identity is used by multiple nodes simultaneously in the network. If no specific detection mechanisms are set up, the attacker could lead many insidious attacks such as subverting data aggregation protocols by injecting false data, revoking legitimate nodes and disconnecting the network if the replicated nodes are judiciously placed at chosen locations.

The replication attack consists in introducing new malicious nodes with existing identities in the network. There exist two main categories for replicated node detection algorithms: *centralized* and *distributed* algorithms. When centralized detection is used, each sensor node transfers its neighbors list to the base station seen as a central entity which can then filter out replicated nodes and can revoke them by a sample flooding in the network [1, 2]. This solution has several drawbacks as the single point of failure (the Base Station) and especially high communication costs. Hence, a distributed solution is desirable.

Distributed approaches for detecting node replications are based on storing a node's location information at one or more witness nodes in the network. When a new node joins the network, its location claim is forwarded to the corresponding witness nodes. If any witness receives two different location claims for the same node identity (ID), it will have detected the existence of a replica and can take appropriate actions to revoke the node's credentials. Distributed protocols are much

Corresponding Author: R. Udayakumar, School of Computing Science, Bharath University - 73, India.

promising due to the distributive nature of sensors networks. The main idea here consists in network broadcasts: each single node in the network broadcasts its location (i.e. its identity) to the whole network and save the location claims of its direct neighbors; then if it receives a conflicting claim, it revokes the offending node [3, 4].

The basic challenge of any distributed protocol in detecting node replicas is to minimize communication and per node memory costs while ensuring that the adversary cannot defeat the protocol. A protocol that deterministically maps a node's ID to a unique witness node would minimize both communication costs and memory requirements per node, but would not offer enough security because the adversary would need to compromise just a single witness node in order to be able to introduce a replica without being detected [5].

The set of witnesses is uniformly chosen from the whole network due to the usage of a pseudorandom function, the inputs of which include the identity of the node, the number of locations (of witnesses) that have to be generated by any neighbor of this node that decides to forward the location claim and a random number rand which is changed per iteration. Therefore, there exists a dilemma in selecting an appropriate value of the number of locations (of witnesses) that have to be generated so as to achieve the balance between efficiency and robustness against node compromise [6].

In this paper, we present a novel distributed protocol for detecting node replication attacks that takes a different approach for selecting witnesses for a node. In our approach, which we call Localized Multicast, the witness nodes for a node identity are randomly selected from the nodes. Our approach first deterministically maps a node's ID to one or more cells and then uses randomization within the cell(s) to increase the resilience and security of the scheme[7]. One major advantage of our approach is that the probability of detecting node replicas is much higher than the previous detection approaches. We describe and analyze two variants of the Localized Multicast approach:

Single Deterministic Cell (SDC) and Parallel Multiple Probabilistic Cells (P-MPC), Which as their name suggests differ in the number of cells to which a location claim is mapped and the manner in which the cells are selected. We evaluate the performance and security of these approaches via simulation.

Background and Prior Work

Goals: For a given sensor network, we would like to detect a node replication attack, i.e., an attempt by the adversary to add one or more nodes to the network that use the same ID as another node in the network. The methods of detecting node replication can be divided into two categories: centralized and distributed. Ideally, we would like to detect this behavior without centralized monitoring, since centralized solutions suffer from several inherent drawbacks. The scheme should also revoke the replicated nodes, so that non faulty nodes in the network cease to communicate with any nodes injected in this fashion [8].

Here we are using one of the distributed approaches, which we call as *Localized Multicast*, the witness nodes for a node identity are randomly selected from the nodes. Our approach first deterministically maps a node's ID to one or more cells and then uses randomization within the cell(s) located within a geographically limited region (referred to as a cell). Our analysis and simulations demonstrate our protocol is effective even when there are a large number of compromised nodes and at the same time achieves a higher probability of detecting node replicas [9].

Sensor Network Environments: A sensor network typically consists of hundreds, or even thousands, of small, low-cost nodes distributed over a wide area [10]. The nodes are expected to function in an unsupervised fashion even if new nodes are added, or old nodes disappear (e.g., due to power loss or accidental damage). While some networks include a central location for data collection, many operate in an entirely distributed manner, allowing the operators to retrieve aggregated data from any of the nodes in the network. Furthermore, data collection may only occur at irregular intervals.

For example, many military applications strive to avoid any centralized and fixed points of failure. Instead, data is collected by mobile units (e.g., unmanned aerial units, foot soldiers, etc.) that access the sensor network at unpredictable locations and utilize the first sensor node they encounter as a conduit for the information accumulated by the network. Since these networks often operate in an unsupervised fashion for long periods of time, we would like to detect a node replication attack soon after it occurs. If we wait until the next data collection cycle, the adversary has time to use

its presence in the network to corrupt data, decommission legitimate nodes, or otherwise subvert the network's intended purpose.

Protocol Framework: In this section, we present the system, network and adversary models assumed in our work, as well as the notation and symbols used in the paper.

System and Network Model: We consider a sensor network with a large number of low-cost nodes distributed over a wide area. In our approach, we assume the existence of a trusted base station and the sensor network is considered to be a geographic grid, each unit of which is called a cell. Sensors are distributed uniformly in the network. New sensors may be added into the network regularly to replace old ones.

Each node is assigned a unique identity and a pair of identity-based public and private keys by an offline Trust Authority (TA). In identity-based signature schemes, the private key is generated by signing its public key (usually a hash on its unique identity) with a master secret held only by the TA. In other words, to generate a new identity-based key pair, cooperation from the TA is a must. Therefore, we assume that adversaries cannot easily create sensors with new identities in the sense that they cannot generate the private keys corresponding to the identities claimed and thus fail to prove themselves to the neighbors during the authentication of the location claims.

We require that, when a node is added into the network, it needs to generate a location claim and broadcast the claim to its neighbors. Each neighbor independently decides whether to forward the claim with a given probability. For those neighbors that plan to forward the claim, they determine the destination cell(s) according to the output of a geographic hash function, which uniquely maps the identity of the sender of the location claim to one or a few of the cells in the grid. Then, the claim is forwarded to the destination cell(s) using a geographic routing protocol such as GPSR.

Adversary Model: In examining the security of a sensor network, we take a conservative approach by assuming that the adversary has the ability to surreptitiously capture a limited number of legitimate sensor nodes. We limit the percentage of nodes captured, since an adversary that can capture most or all of the nodes in the network can obviously subvert any protocol running in the network. Having captured these nodes, the adversary can employ arbitrary attacks on the nodes to extract their private information. For example, the adversary might exploit the unshielded nature of the nodes to read their cryptographic information from memory.

The adversary could then clone the node by loading the node's cryptographic information onto multiple generic sensor nodes. Since sensor networks are inherently designed to facilitate ad hoc deployment, these clones can then be easily inserted into arbitrary locations within the network, subject only to the constraint that each inserted node shares at least one key with some of its neighbors. We allow all of the nodes under the adversary's control to communicate and collaborate, but we make the simplifying assumption that any cloned node has at least one legitimate node as a neighbor.

Notation: In Table 1, we list the notation and symbols used in this paper.

The Localized Multicast Approach for Detecting Node Replications: We have designed two variants of the Localized Multicast approach, specifically *Single Deterministic Cell (SDC)* and *Parallel Multiple Probabilistic Cells (P-MPC)*.

Single Deterministic Cell: In the Single Deterministic Cell scheme, a geographic hash function is used to uniquely and randomly map node L's identity to one of the cells in the grid. For example, given that the geographic grid consists of a×b cells, a cell at the a′ th row and the b′ th column (where a′_ {1…a}, b∈′_ {1,..b}) is uniquely identified as c (where c = a′.b + b′). By using one way hash function H ( ), node L is mapped to a cell C, where c= [H (ID_L) mod (a.b)] +1.

The format of the location claim is:

$$[ID_L, l_L, SIG_{SKL}(H(ID_L \| l_L))]$$

where k denotes the concatenation operation and lL is the location information of L, which can be expressed using either the two-dimension or three-dimension coordinate.

When L broadcasts its location claim, each neighbor first verifies the plausibility of l_L (e.g., based on its location and the transmission range of the sensor) and the validity of the signature in the location claim.

Table 1: Notations and Symbols

n	The number of sensors in the network
s	The number of sensors in a cell
L	The node sending the location claim
ID_i	The identity of a sensor i
l_i	The location information of a sensor i
d	The number of L's neighbors
p_f	The probability that any neighbor of L decides to forward the location claim from L
r	The number of L's neighbors that forward the location claim from L
w	The number of the witness nodes that store the local claim from L
p_s	The probability that a sensor in the cell stores the location claim
t	The number of sensors that have been compromised by adversaries
x	The number of sensors with the same identity (including the compromised sensor and its replicas)
PK_i, SK_i	The public key and the private key of a sensor i
$H()$	A collision-free one-way hash function
$SIG_{SK}(M)$	A message M signed by a key SK

In identity-based signature schemes, only a signature generated with the private key corresponding to the identity claimed can pass the validation process. Thus, adversaries cannot generate valid signatures unless they compromise the node with that identity.

Each neighbor independently decides whether to for-ward the claim with a probability p_f. If a neighbor plans to forward the location claim, it first needs to execute a geographic hash function to determine the destination cell, denoted as C. The location claim is then forwarded toward cell C.

Once the location claim arrives at cell C, the sensor receiving the claim first verifies the validity of the signature and then checks whether cell C is indeed the cell corresponding to the identity listed in the claim message based on the geographic hash function. If both the verifications succeed, the location claim is flooded within cell C. Each node in the cell independently decides whether to store the claim with a probability p_s. Note that the flooding process is executed only when the first copy of the location claim arrives at cell C and the following copies are ignored. As a result, the number of witnesses in the cell w is $s.p_s$ on average, where s is the number of sensors in a cell.

Whenever any witness receives a location claim with the same identity but a different location compared to a previously stored claim, it forwards both location claims to the base station. Then, the base station will broadcast a message within the network to revoke the replicas.

An example of blocking attack against the SDC approach is shown in Fig. 1. Cell C1 and C2 are the deterministic cells for the identity ID_{C1} and ID_{C2}, respectively and B is an area in which all the nodes have been compromised (referred to as a black hole). In this

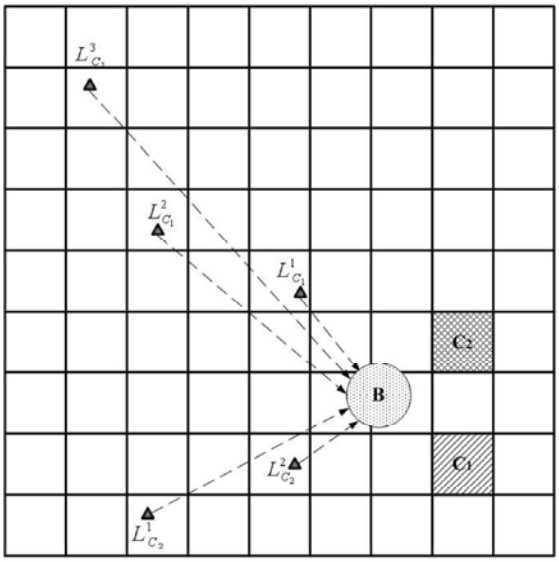

Fig. 1: The blocking attacks

example, three replicas (i.e., L^1_{C1}, L^2_{C1} and L^3_{C1}) claiming the same identity that is mapped to cell C1 are added to the network sequentially, with a certain time interval between any pair of consecutive joins.

As shown in Fig. 1, two replicas (i.e.,L^1_{C2} and L^2_{C2}) claiming the same identity that is mapped to cell C_2 are inserted into the network and their location claim blocked by hole B.

Parallel Multiple Probabilistic Cells: In the SDC approach, all the location claims are first forwarded from the neighbors of L to a deterministic cell. Therefore, there is a high probability that these forwarding paths intersect with each other. In particular, when L and the destination cell (i.e., cell C) are far from each other, there is a high probability that all the location claims will pass through one or a small set of nodes of size y. Therefore, the adversary only needs to compromise one or y nodes per replica so as to block the forwarding of a location claim. Hop-by-hop watchdog monitoring may help mitigate this attack. However, it will fail if all or most of the neighbors of an intersection point are compromised.

Another potential risk is that a smart adversary can take advantage of the knowledge that the destination cell for a given identity is deterministic and launch a blocking attack. Informally, after compromising a small set of sensors denoted as V, the adversary can generate replicas of members in V and deploy them in such a way that all the location claims of these replicas are forwarded through members of V.

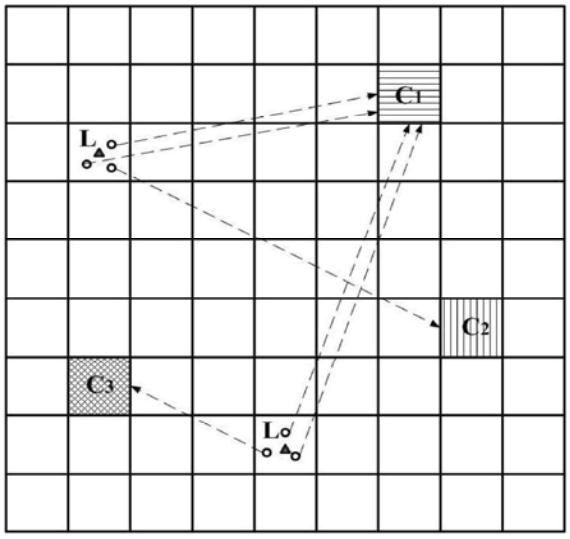

Fig. 2: The parallel multiple probabilistic cells approach

Like SDC, in the P-MPC scheme, a geographic hash function is employed to map node L's identity to the destination cells. However, instead of mapping to a single deterministic cell, in P-MPC, the location claim is mapped and forwarded to multiple deterministic cells with various probabilities

Once the location claim arrives at cell Cj, the sensor receiving it first verifies whether Cj is a member of C which can be calculated based on the geographic hash function and the identity listed in the claim message. In addition, this sensor needs to verify the validity of the signature in the location claim. If both the verifications succeed, the claim is flooded within the cell and probabilistically stored at w nodes in the same manner as in the SDC scheme.

For example, in Fig. 2, there are two replicas with the same identity in the network. In this example, an identity is mapped to three cells (i.e., C1; C2; C3) with different probabilities (i.e., pc1 > pc2 > pc3). The neighbors of one replica forward the location claims to cell C1 and C2, while the neighbors of the other replica forward the location claims to cell C1 and C3. Therefore, any witness node with cell C1 can detect the node replication.

Analysis

Single Deterministic Cell (SDC) Scheme

Security Analysis: The metrics used to evaluate the security of the SDC scheme are:

- The probability of detecting node replication when versaries put x replicas (including the compromised node) with the same identity into the network, which is denoted as p_{dr}.
- The probability that adversaries control all the witnesses for a given identity after compromising t nodes, which is denoted as p_{ts}.
- The probability that adversaries control all the witnesses for at least one identity after compromising t nodes, which is denoted as p_{tm}.

The latter two metrics estimate the risk that an adversary controls all the witnesses for a node and can thus launch a node replication attack without being detected.

Detecting Replicas: Unlike the Random Multicast and Line-Selected Multicast algorithms where the nodes storing the copies of a location claim are chosen randomly from the whole network, in SDC such nodes are chosen randomly from a small subset of all the nodes in the network, i.e., the nodes in the destination cell determined by the geographic hash function. In addition, since the location claim will be flooded within the destination cell, the SDC scheme can always detect any pair of nodes claiming the same identity.
In other words, $p_{dr} = 100\%$ in SDC, when $r > 0$ and $w > 0$.

Efficiency Analysis: The metrics used to evaluate the efficiency of the SDC Scheme includes:

- The average number of packets sent and received while propagating the location claim, which is denoted as n_f.
- The average number of copies of the location claims stored on a sensor, which is denoted as n_s.

The former is to measure the communication cost, while the latter is to estimate the memory overhead. We do not explicitly consider the computation cost (i.e., verifying that the location claim is generated by an entity which holds the private key corresponding to the identity listed in the claim), since every forwarding node needs to execute such a verification and thus it is proportional to the communication cost. In other words, the higher the communication cost, the higher the computation cost.

Parallel Multiple Probabilistic Cell (P-MPC) Scheme: In this section, we analyze the security and efficiency of the P-MPC scheme.

Security Analysis: For simplicity, in this section we assume that the number of neighbors (r) forwarding the location claim is a fixed number. We assume that the adversary creates x-1 replicas of a given compromised

node with id IDL and deploys them in the network. We assume that adversaries do not reposition the compromised node, l1 and the replicas are added in sequence from l2 to lx. Let pir denote the probability that the node replication attack is not detected by our scheme after the ith node with the same identity has been added to the network.

Detecting Replicas: Let C_{s1} denote the set of all combinations of choosing 1 to v -1 elements from C, i.e., the set of cells to which ID_L is mapped. If the node replication attack is not detected when the adversary adds replica l_2 to the network, it implies that the location claims for l_2 have been forwarded to a set of cells, none of which contains any node storing a location claim from l_1 Let C_{e1} denote a subset of the cells in C that do not store the location claims of l_1. Let $p_{i,\ 1}$ denote the probability that the location claim of l_1 is forwarded to all the cells in C except the cells in C_{e1}, which is an element of Cs1. Let $p_{i,2}$ denote the probability that the location claim of l_2 is forwarded to any cell(s) in C_{e1}. Therefore, we have:

$$P_{2r}= \Sigma \Sigma\ p_i.\ p_{ij,\ 1}.p_{ij,\ 2}$$

Now, we consider further the case that the adversary adds l3 to the network. Let C_{s1b} denote the set of all the combinations of choosing 2 to v-1 elements from C. For a given C_{e1} _ C_{s1b}, let C_{s2} denote all the combinations of choosing 1 to jC_{e1}j-1 elements from C_{e1}. We denote C_{e2} as the set of cells that store the location claim from l_2 but not l_1 and C_{e2} _ C_{s2}. Let pi denote the probability that the location claim of l_1 is forwarded to all the cells in C except the cells in C_{e1}, which is an element of Cs1b. Let $p_{ij,1}$ denote the probability that the location claim of l2 is forwarded only to all the cells in C_{e2}. Let $_{pij,2}$ denote the probability that the location claim of l_3 is forwarded to any cell(s) in C_{e1} except those in C_{e2}. Thus, we have:

$$P_{3r} = \Sigma \Sigma\ p_i.p_{ij,\ 1}.p_{ij,\ 2}$$

Let r =3 and v=3. In Table 2, we show the estimated success rate of detecting node replications under different settings of p_{ci} according to (5) and (6). According to Table 2 (where "Set." is a short notation for "Setting"), the P-MPC scheme can achieve a very high replica detection rate, even when an identity is mapped to three destination cells. Moreover, we notice that the larger the differences between the probabilities p_{cis}, the higher is p_{ir}.

Table 2: Detection Rates When There Are 2 or 3 Nodes with the Same Identity, Given Different Settings of the Distribution of Forwarding Probabilities

	p_{c1}	p_{c2}	p_{c3}	$1-p_{2r}$	$1-p_{3r}$
Set. I	80%	15%	5%	99.77%	100%
Set. II	70%	20%	10%	99.38%	100%
Set. III	50%	30%	20%	98.88%	99.98%

Table 3: Probability that the Adversary Controls All w Witnesses for a Given Identity after Compromising t_ Nodes in a Cell of Size s in the P-MPC Scheme (s=100, w=5, tΔ=30)

	Set. A	Set. B	Set. C	Set. D	Set. E
Set. I	9.69e-04	2.04e-05	1.73e-06	6.39e-06	2.37e-07
Set. II	2.37e-04	5.08e-06	5.36e-07	5.11e-05	1.51e-05
Set. III	7.01e-05	1.60e-06	3.72e-07	7.01e-05	7.01e-05

Evaluation: We evaluated the performance and security of our schemes. In addition, we also investigated security and efficiency of our approach under different settings, such as different probabilities of forwarding location claims.

Metrics: We used the following metrics to compare the schemes:

Communication overhead: We measured the total number of packets sent and received per node for running the replica detection algorithm when n nodes are added to the network. We denote this metric as n_f.

Success rate in detecting replicas: We measured the probability of detecting a replica, when there are two sensors with the same identity in the network, i.e., p_{2r}.

CONCLUSION AND FUTURE WORK

In this paper, we proposed two variants of the Localized Multicast approach for distributed detection of node replication attacks in wireless sensor networks. Our approach combines deterministic mapping (to reduce communication and storage costs) with randomization (to increase the level of resilience to node compromise). Our simulation result shows that our schemes are more efficient in large-scale sensor networks, in terms of communication and memory costs. Moreover, the probability of replica detection in our approach is higher than that achieved in these two algorithms. Our results show that we can gain the benefits of an detecting the node replications while node replication attacks attack. Our preliminary analysis also shows that, our approaches are more robust than RED against selective node compromise and the communication and memory overheads of our approaches are similar or slightly higher

than that of RED. One of our future works is to simulate the RED protocol and then have a more detailed comparison of efficiency based on empirical results.

REFERENCES

1. Conti, M., R. Di Pietro, L.V. Mancini and A. Mei, 2007. "A Randomized, Efficient and Distributed Protocol for the Detection of Node Replication Attacks in Wireless Sensor Networks," Proc. ACM MobiHoc, pp: 80-89.
2. Zhu, B., V.G.K. Addada, S. Setia, S. Jajodia and S. Roy, 2007. "Efficient Distributed Detection of Node Replication Attacks in Sensor Networks," Proc. 23rd Ann. Computer Security Applications Conf. (ACSAC '07), 2007.
3. Choi, H., S. Zhu and T.F. La Porta, "SET: Detecting Node Clones in.
4. Udayakumar, R., V. Khanaa and K.P. Kaliyamurthie, 2013. Optical Ring Architecture Performance Evaluation using ordinary receiver, Indian Journal of Science and Technology, ISSN: 0974-6846, 6(6): 4742-4747.
5. Udayakumar, R., V. Khanna, T. Saravanan and G. Saritha, 2013. Retinal Image Analysis Using Curvelet Transform and Multistructure Elements Morphology by Reconstruction, Middle-East Journal of Scientific Research, ISSN:1990-9233, 16(12): 1798-1800.
6. Udayakumar, R., V. Khanna, T. Saravanan and G. Saritha, 2013. Cross Layer Optimization For Wireless Network (Wimax), Middle-East Journal of ScientificResearch,ISSN:1990-9233,16(12):1786-1789.
7. Saravanan, T. and R. Udayakumar, 2013. Optimization of Machining Hybrid Metal matrix Composites using desirability analysis, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1691-1697.
8. Saravanan, T., V. Srinivasan and R. Udayakumar, 2013. Images segmentation via Gradient watershed hiera.rchies and Fast region merging, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1680-1683.
9. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Application of Soft Computing Techniques in weather forecasting: Ann Approach, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1845-1850.
10. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Improving Web Information gathering for personalised ontology in user profiles, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1675-1679.