Chapter 1

Distributed Trust Management Model for Wireless Sensor Networks

Janusz GÓRSKI[1], Alan TUROWER[2], Andrzej WARDZIŃSKI[3]

1. INTRODUCTION

Wireless sensor networks increase their significance in numerous application areas, including healthcare, defence, safety monitoring, environment monitoring and others. As sensor networks become more complex and provide more sophisticated services, the diversity of roles of network nodes increases, in addition to simple sensor nodes including: routers, heads, sinks and others. Such nodes, which mission is broader than just sensing the environment, can have considerable computing power and accomplish advanced tasks. As the size and complexity of the networks grows, managing them becomes more difficult to reconcile security with performance and flexibility. Moreover, individual nodes or sub-networks can be managed by different persons or organizations. Dependability of such networks becomes a difficult issue as in addition to technical imperfections and human faults, malicious actions have to be taken into account.

Ensuring security of sensor networks by just copying the best practices from the conventional networks is not practical as sensor nodes are subjected to severe limitations of their resources and cannot afford running sophisticated security mechanisms which are often significantly resource consuming. To cope with this problem we employ the concepts of trust and trustworthiness. Object A trusts object B if A makes some (positive) assumptions about the state and behaviour of B (for instance, A assumes that the data sent by B are genuine). B is trustworthy if A has in its disposal the evidence sufficient to justify its trust in B. With these definitions, we understand trust management as collecting the evidence about trustworthiness and based on this, making decisions about trust.

In sensor networks, trust management has a great importance as it provides for distinguishing between trustworthy and untrustworthy nodes which enables collaborative decisions leading to isolation and exclusion of the nodes with a very low level of trust.

---

[1] Gdańsk University of Technology, Faculty of Electronics, Telecommunication and Informatics, jango@eti.pg.gda.pl

[2] Gdańsk University of Technology, Faculty of Electronics, Telecommunication and Informatics, alan.turower@eti.pg.gda.pl

[3] Gdańsk University of Technology, Faculty of Electronics, Telecommunication and Informatics, andrzej.wardzinski@eti.pg.gda.pl

The objective of this chapter is to present a new trust management model for wireless sensor networks (WSN) together with the results of the analysis of its performance. The first version of the model was proposed for the ANGEL platform within the context of the wireless sensor networks architecture developed for healthcare and home environment monitoring [1]. The analyses presented in this paper are based on simulations performed with the help of a dedicated simulator.

1.1. RELATED WORKS

Trust management for complex wireless sensor networks is currently an area of active research. The main question is how network nodes should behave in order to effectively identify and isolate malicious nodes and to minimize *false positives* and *false negatives*.

There are two main architectures considered for trust models: centralized and distributed [12]. The first type distinguishes the Trust Authority (e.g. the central station) which manages trust relationships between nodes [10]. This solution is efficient and manageable, but it has problems with scalability and robustness.

A distributed trust model is considered to be suitable for large-scale sensor networks. Zhiying et al [12] find this model appropriate for sensor network security design because a node focuses on trustworthiness of its neighbours and can assess if these nodes obey agreed security policies. They propose a corresponding security framework with different security schemes. However their work does not take into consideration limited resources of nodes in sensor networks.

A hybrid approach is possible which combines the advantages of the centralized and distributed models. The network structure comprises two-levels – all nodes are divided into clusters and each cluster head is an element of so called backbone network, which enables communication of cluster heads with the central station. Song et al [11] propose to use the LEACH protocol by adding a trust management module. Boukerche et al [3] propose trust and reputation management scheme that uses mobile agents running on each node. In this model there is a central agent launcher responsible for generating and launching agents into the network. However, there is no central repository of trust, which makes trust exchange (if there are mobile nodes) significantly more difficult.

Zia [13] proposes the security framework to provide a comprehensive security solution against the known threats by integrating the reputation and trust management mechanism. In this concept nodes monitor their neighbouring nodes and rank the neighbours in terms of a trust vote.

Momani et al [9] have introduced a trust model and a reputation system for WSNs based on sensing continuous data. The trust model establishes the continuous version of the Beta reputation system [8], and a Bayesian probabilistic approach for mixing second-hand information from neighbouring nodes with directly observed information to calculate trust between nodes in WSNs is used.

Chen et al [5] propose a distributed agent-based trust management scheme where each agent node independently monitors the behaviour of the nodes within its radio

range and broadcasts their trust ratings. They also introduce a reputation based trust model using probability, statistics and mathematical analysis and have suggested a trust system to build up a reputation space and trust space in WSNs [6].

Our model assumes that the network is composed of *clusters*, each cluster having its *head* node and the nodes of a cluster communicating among themselves and with the cluster head. We propose a mechanism which enables each node to make autonomous decisions about trust based on the trustworthiness assessment of its neighbours. We see this model as applicable for the networks applying the LEACH protocol at the lower tier as well as at the higher tier (the backbone cluster).

1.2. PAPER STRUCTURE

The chapter is organized as follows. First we describe the proposed trust management mechanism. This is followed by an introduction of our dedicated simulator and presentation of the results obtained while using it to assess the potential of the proposed mechanism to detect malicious nodes. We finalize with discussion of the results obtained and presentation of the directions of further research.

2. THE PROPOSED TRUST MODEL

There are many definitions of trust. A dictionary definition states that trust is a belief or confidence in the honesty, goodness, skill or safety of a person, organization or thing [4]. Another definition [2] says that: trust is a bet that those entities, which you cannot control, will act in a predictable manner that is favourable to your cause. Generally, trust is a relation between the trustor and the trustee. We assume that each node of a network should be examined if it can be trusted and that all nodes should cooperate in that process [12, 7]. The objective of trust management system is to distinguish between trustworthy network nodes and untrustworthy ones. Then, the trustworthy nodes can cooperate to provide trustworthy network services and the untrustworthy nodes are excluded from the network.

Trustworthy network services can be provided if they are based on trustworthy information. Therefore we need a mechanism for assessing if a data item is trustworthy before it is subjected to further processing and passed through the network. Distrusted data are discarded and the trustworthiness assessment of the source of this data is being lowered. To limit the potential damage it is important to assess the trustworthiness as early as possible to prevent distrusted data from further processing. For large networks, centralized assessment by a dedicated node would lead to performance problems and excessive concentration of network traffic. Therefore, we assume that every node in the network is involved in trustworthiness assessment and the trust related decisions are localized.

For this paper we assume the following definition: *trust* is an act of acceptance of a message received from a network node which results from the assessment of the *trustworthiness* of the message and its source.

A network node acts as a *trustor* and a *trustee*:

- for outgoing communication, the node acts as a trustee – other nodes judge if it can be trusted,
- for incoming communication, a node acts as a trustor – it makes a real-time decision if the sender can be trusted.

Decision about trust is based on two complementary approaches:

- *Policy-based approach*: trustor evaluates trustworthiness of the trustee assessing the trustee's observed behaviour and its conformance with agreed policies, notably the security policy.
- *Reputation-based approach*: trustor takes into account information from other nodes if they trust the trustee – this information is called *recommendation*.

Decision about trust is made each time the trustor receives a message from any other node (the trustee). This decision is based on trustworthiness assessment of the sender of the message. The assessment is based on two pillars of evidence: the evidence resulting from the application of agreed security mechanisms (policy-based approach) and the evidence resulting from the recommendations received from the companion nodes (reputation-based approach). The following are examples of evidence considered while deciding about trusting or distrusting the trustee:

- formal correctness of the message, e.g. if the message is compliant with the network security policy;
- the right of the sender to send messages of a given type to the recipient node;
- message content check;
- sender's reputation; this reputation is calculated on the basis of the trustee's reputation maintained by the trustor and the recommendations received from other (trusted) nodes;
- recipient's reputation when the node acts as a router and it has to decide if the message is to be retransmitted.

The above mechanism is implemented in each network node and the nodes collectively perform the trust management process of the network.

Depending on the trust assessment result the trustor performs appropriate actions:

- if the trustee is trusted:
  - proceed with message processing,
  - raise reputation for the trustee;
- if the trustee is not trusted:
  - discard the message (do not process the data),
  - decrease the reputation for the trustee,
  - inform about the event if it is required.

Trust management requires that all nodes use the same scale for representing reputation. This scale is used while the network nodes exchange recommendations. There are four characteristic values related to this scale:

- *full (absolute) trust* – means that the node is (fully) trustworthy,
- *initial trust* – is the initial credit given to a node (for instance when the node joins the network and its trustors have yet no evidence related to its trustworthiness),
- *cut off point* – is the trustworthiness level below which the trust is not further justified (any messages from the node are discarded by the trustor),
- *distrust* – is the absolute distrust (minimal value on the scale).

Figure 1 presents a graphical form of the scale. Each new node in the network is credited with *initial trust* in the trustworthiness scale. This determines its initial reputation. Then, depending on its behaviour, its reputation can change. When the reputation drops below the cut off point, the node is perceived as untrustworthy and the messages received from this node are distrusted. In the model it is assumed that if node's reputation drops below the cut off point, it cannot regain trust unless the central network node (network head, service center etc.) resets its reputation to the initial level.

Each network node participates in trust management process and maintains data on the reputation of other nodes. The corresponding data structure is called *reputation table*. The trust management process policy requires that the network nodes obey the following rules:

- local reputation table can be sent on demand – for instance a new node joining the network asks its neighbours for recommendations;
- each node periodically broadcasts its local reputation table; the frequency depends on network traffic and reputation changes in the table;
- local reputation table or some of its entries can be reset as the result of a special command from the central node of the network (acting as the trust manager).

Trustee's reputation in the reputation table can be changed as a result of:

- assessment of a message send by the trustee and
- recommendation related to the trustee, received from other (trusted) node.

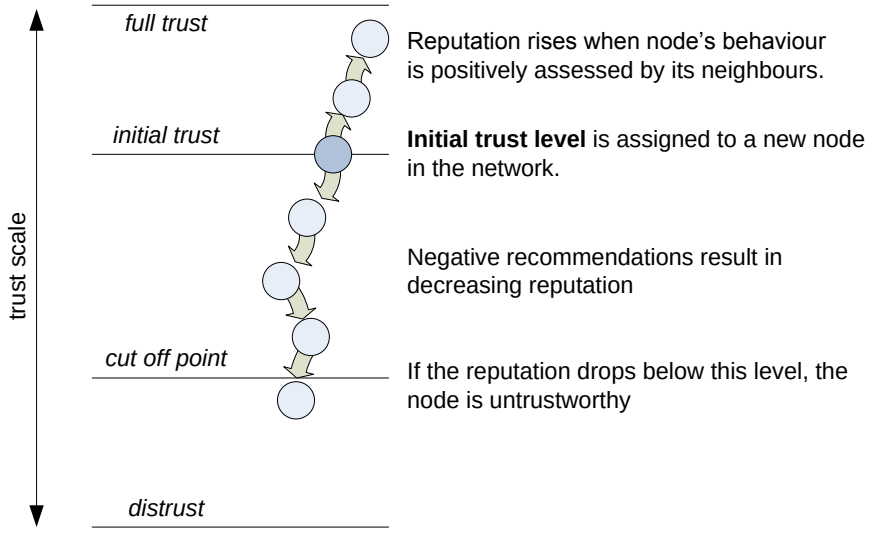

Fig 1. The proposed trust scale [7] with the initial trust level and possible changes

Recommendation is an entry of a local reputation table sent to another node. A node can recommend any other node except itself.

Reputation of a node sending a message depends on two factors: (1) assessment of the incoming message and other messages previously sent from this node and (2) received recommendations related to this node. The influence of these factors is characterized by the *Cooperation Factor* (CF). CF assumes values from 0 to 1, where 0 means that recommendations are discarded in reputation calculation and 1 means that reputation is solely based on recommendations. As a general rule, only these recommendations are taken into account that come from trusted nodes (the nodes of reputation higher than the cut off point).

Assuming that node A has N trusted neighbours and a trusted node B sends to A a recommendation for node C, A will recalculate reputation of C in accordance to the following formula:

$$Reputation_{\mathrm{C}} := Recommendation_{\mathrm{BtoC}} \times I + Reputation_{\mathrm{C}} \times (1 - I) \qquad (1)$$

where

$I = CF \times Reputation_{\mathrm{B}} \ / \ N$,

$Reputation_{\mathrm{X}}$ denotes reputation of node X,

$Recommendation_{\mathrm{BtoC}}$ – denotes the recommendation for C sent to A by B.

Reputation can also change in effect of the assessment of an incoming message. If node B sends a message to node A and A assesses that the message is against the assumed policies, B's reputation maintained by A is decreased by $change_{\text{negative}}$ factor:

$$Reputation_{\text{B}} -= Reputation_{\text{B}} \times change_{\text{negative}} \quad (2)$$

In case the message agrees the agreed policies, the reputation increases:

$$Reputation_{\text{B}} += (1 - Reputation_{\text{B}}) \times change_{\text{positive}} \quad (3)$$

For instance, $change_{\text{negative}} = 0{,}01$ means that the reputation will decrease by 1%.

3. THE SIMULATOR

To learn more about the properties of the proposed trust management model we developed a simple simulator. In particular, we were interested in the ability of the model to discover and isolate distrusted nodes.

We assume that the nodes are randomly dispersed in the space of the size X by X' units. We also assume that each node has a Y units range.

The simulator analyzes a network of ***n*** nodes and one sink (the central node of the network). The nodes are fixed (i.e. they do not change their position during simulation). Therefore, after initial distribution of nodes, some of them can be too far from the other nodes or the sink to communicate with. In such cases some nodes act as routers and pass the message to the recipient. We assume that each *benign* node can send an incorrect or broken message (it is called a *spoiled message*) with probability ***z***. Additionally, we assume that some nodes are *faulty* and in this case they send spoiled messages with probability ***w*** (where $w > z$) . At present, we assume that all spoiled messages are reliably detected by the receiving node and the messages containing recommendations are spoiled with probability ***z*** (as any other message).

The simulator assumes that the routing algorithms are in place. The route selection process takes into consideration reputations of the neighbours of a given node. If the reputation of node B as perceived by node A drops under the cut-off point, A will not pass messages to B. If all neighbours of A on a way to the sink are distrusted, the node A (and its sub network) is excluded from the whole network.

The simulator works in *simulation turns*, where each turn evaluates each network node. The nodes are permanently active, i.e. during a turn each of them sends one message to the sink (and resends the messages from other nodes, if it is a router node). At the end of the turn, the nodes exchange their reputation tables (issue recommendations) with their neighbours and update own reputation tables accordingly.

The messages sent are either accepted or discarded, depending on the sender's reputation.

The number of simulation turns necessary to transform the network from state S to state S' serves as a measure representing the distance between these two states. This distance tells us how long (in terms of message exchanges) we would have to wait to arrive in S', provided all nodes are still attempting to communicate with their neighbours (for instance, how long it will take for a network to detect and isolate all failed nodes).

4. SIMULATION EXPERIMENTS

During experiments we were considering networks containing faulty nodes.

The objective of the experiments was to verify:

- how many simulation turns are needed to detect the first faulty node,
- how many simulation turns are needed to detect all faulty nodes.

During experiments, the size of the area for nodes distribution was set to X = X' = 100 points (where point is a distance unit) and the node signal range was set to Y = 30 points. The reputation scale was the interval of real numbers [0..1] where

- *distrust* = 0;
- *full trust = 1;*
- *cut off point* = 0,2;
- *initial trust* = 0,5;

In each experiment, six networks of the following sizes were simulated: ***n*** **=** 20, 50, 100, 150, 200, 300. For each network size, the simulations were performed assuming the number of faulty nodes ranging from 1 to 10.

For each network size and for each number of faulty nodes, 100 different simulations were performed. Then, for each network size, the following parameters were calculated:

- **first node** – the average number of simulation turns necessary to detect the first faulty node,
- **all nodes** – the average number of simulation turns necessary to discover all faulty nodes.

The experiments' parameters are summarized in Table 1.

Table 1. Simulation parameter values for the experiments

simulation parameter	Experiment I	Experiment II
w	50%	70%
z	2%	2%
n	20 50 100 150 200 300	20 50 100 150 200 300
$change_{negative}$	0,8	0,8
$change_{positive}$	0,05	0,05
initial trust	0,5	0,5
cut off point	0,2	0,2

The results of Experiment I and Experiment II for networks of 100 nodes are shown in Figure 2.

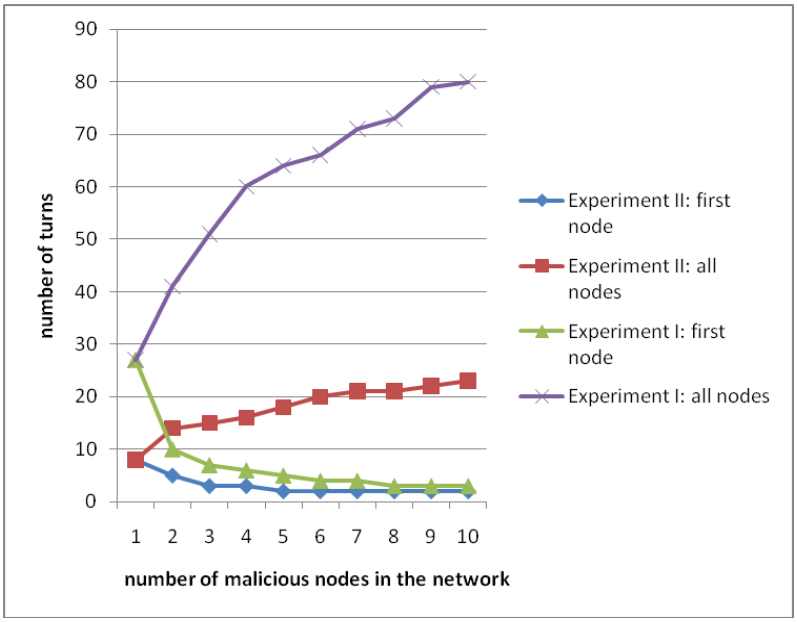

Fig. 2. Number of simulation turns needed to detect the first faulty node and all faulty nodes in the network of 100 nodes

The graphs for the simulation results for the other network sizes look similar.

From figure 2 we can see that number of simulation turns (the hereafter called *time delay*) needed to detect the first faulty node is rather weakly dependent on the total number of faulty nodes, whereas the time delay needed to detect all faulty nodes depends strongly on the total number of faulty nodes in the network and on the *node failure profile* (characterized by the parameter ***w***).

Figure 3 shows the *normalized time delay (the* time delay necessary to detect all faulty nodes normalized by the number of nodes in the network). We can observe that this parameter decreases significantly as the number of network nodes grows (up to 150 nodes) and then achieves a sort of saturation (the interval from 150 to 300 nodes). The value of this paramenter for e.g. the network of 100 nodes shows that the increase of the number of faulty nodes from 5 to 10 (100%) results in the increase of normalized time delay by 20%.

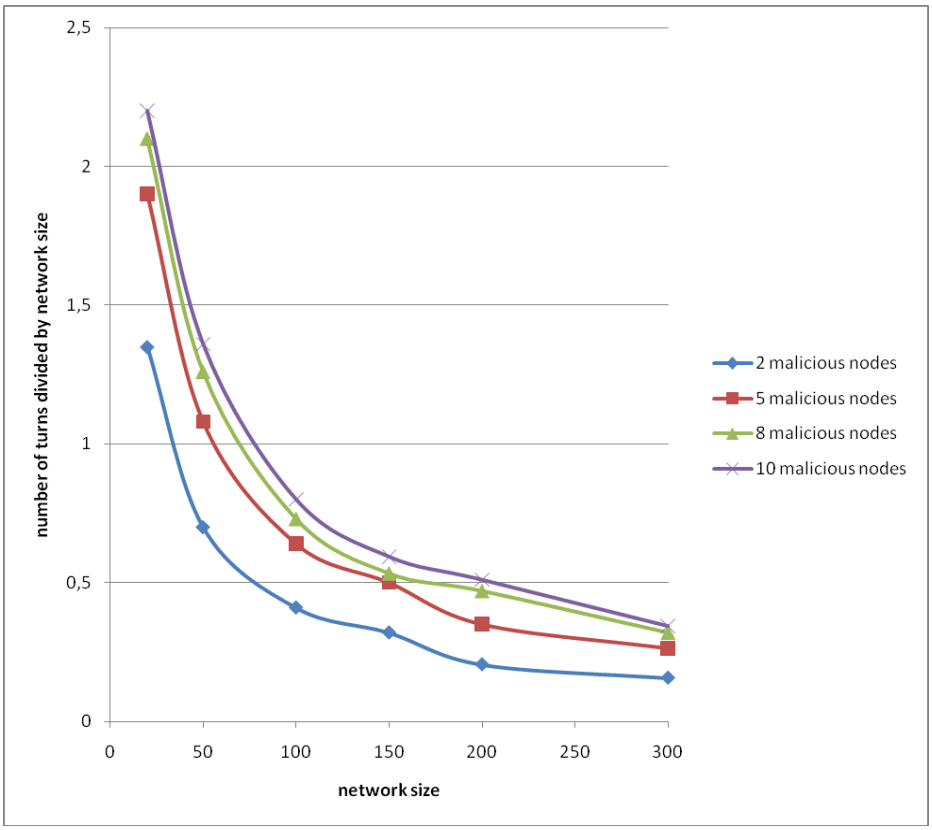

Fig. 3. The average number of simulation turns divided by the network size needed to detect all malicious nodes in Experiment II

5. CONSLUSION

The results of the experiments presented in this paper demonstrate the potential of the proposed trust management mechanism to detect and isolate faulty nodes in a sensor network. We measure the results in terms of simulation turns. During each turn each node attempts to send 'regular' messages to their neighbours and to exchange reputation messages. Therefore, the number of simulation turns tells us how many messages are needed to detect the first faulty node or to detect all faulty nodes. Bigger number of turns means that more message exchanges are necessary.

The experiments were focusing on the performance of the proposed mechanism with respect to a single cluster of nodes. For bigger networks, composed of many clusters, trust management would be implemented locally in each individual cluster and in addition in the cluster of heads (the upper tier). This would provide for scalability of the results presented in the paper.

The experiments were carried out with selected values of the simulation parameters. In further research we will investigate the performance of the proposed mechanism for larger networks and the sensitivity of the malicious nodes detection potential to the network parameters changes. We will also investigate how the proposed mechanism can be made immune to more sophisticated attacks where malicious nodes modify their behaviour on purpose to affect their reputation or cooperate to achieve their goals.

5. ACKNOWLEDGEMENT

This work was partially supported by the project ANGEL - *Advanced Networked embedded platform as a Gateway to Enhance quality of Life* (2005-IST-5-033506-STP).

REFERENCES

[1] *Angel System Specification*, ANGEL Project IST-5-033506-STP, Deliverable D1.2, 2007.

[2] BATT N. I., *Operational Trust: A New Look at the Human Requirement in Network Centric Warfare*, 9th International Command and Control Research and Technology , San Diego, 2004.

[3] BOUKERCHE A., LI X., *An Agent-based Trust and Reputation Management Scheme for Wireless Sensor Networks*, IEEE Globecom 2005, St. Louis, 28 Nov – 2 Dec 2005

[4] Cambridge Advanced Learner's Dictionary, http://dictionary.cambridge.org/

[5] CHEN H., WU H., ZHOU X., GAO C.: *Agent-based Trust Model in Wireless Sensor Networks*, in 8th ACIS Int.Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Qingdao, China, 2007.

[6] CHEN H., WU H., ZHOU X., GAO C.: *Reputation-based Trust in Wireless Sensor Networks*, in International Conference on Multimedia and Ubiquitous Engineering MUE'07, 2007.

[7] GÓRSKI J. et al.: *WSN Trust Management Model*, in "*Angel project report: WSN trust architecture and security protocols*", Deliverable 3.2., ANGEL Project, 2007.

[8] JOSANG A., ISMAIL R.: *The BetaReputation System*, in 15th Bled Electronic Commerce Conference, Bled, Slovenia, 2002.

[9] MOMANI M., CHALLA S.: *Trust Management in Wireless Sensor Networks*, Proc. of 5th ACM Conf. on Embedded Networked Sensor Systems, Sydney, Australia, November 6 – 9, 2007.

[10] PERLMAN R., *An overview of PKI trust models*, IEEE Network, vol. 13, no. 6, 1999.

[11] SONG F., ZHAO B., *Trust-based LEACH Protocol for Wireless Sensor Networks*, Future Generation Communication and Networking FGCN '08, 13–15 Dec 2008.

[12] ZHIYING Y., DAEYOUNG K., INSUN L., KIYOUNG K., JONGSOO J., *A Security Framework with Trust Management for Sensor Networks*, IEEE SecureComm, 5–9 Sept. 2005.

[13] ZIA T. A: *Reputaton-based Trust Managmenet in Wireless Sensor Networks*, Intelligent Sensors, Sensor Networks and Information Processing, 2008.