# A Unified Trust Model for Pervasive Environments – Simulation and Analysis

**Hamed Khiabani** [1]**, Norbik Bashah Idris** [1] **Jamalul-lail Ab Manan** [2]
[1] Advanced Informatics School, University Technology Malaysia
Kuala Lumpur, Malaysia
[e-mail: hamed.khiabani@ieee.org, norbik@ic.utm.my]
[2] Strategic Advanced Research Cluster, MIMOS Berhad
Kuala Lumpur, Malaysia
[e-mail: jamalul.lail@mimos.my]
*Corresponding author: Jamalul-lail Ab Manan

*Received January 10, 2013; revised April 30, 2013; accepted July 5, 2013; published July 30, 2013*

## *Abstract*

Ubiquitous interaction in a pervasive environment is the main attribute of smart spaces. Pervasive systems are weaving themselves in our daily life, making it possible to collect user information invisibly, in an unobtrusive manner by known and even unknown parties. Huge number of interactions between users and pervasive devices necessitate a comprehensive trust model which unifies different trust factors like context, recommendation, and history to calculate the trust level of each party precisely. Trusted computing enables effective solutions to verify the trustworthiness of computing platforms. In this paper, we elaborate Unified Trust Model (UTM) which calculates entity's trustworthiness based on history, recommendation, context and platform integrity measurement, and formally use these factors in trustworthiness calculation. We evaluate UTM behaviour by simulating in different scenario experiments using a Trust and Reputation Models Simulator for Wireless Sensor Networks. We show that UTM offers responsive behaviour and can be used effectively in the low interaction environments.

***Keywords:*** Unified Trust Model, Smart Spaces, Remote Attestation, Pervasive Computing, Wireless Sensor Network, Trusted Platform Module

A preliminary version of this paper appeared in IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, April 20-23, Perth, Australia. This version includes the formalization of components and results of simulation & analysis. This research is a collaborative program between University Technology Malaysia and Advanced Analysis and Modeling Cluster, MIMOS Berhad. The financial support by MIMOS Berhad for this work is gratefully acknowledged.

**http://dx.doi.org/10.3837/tiis.2013.07.003**

## 1. Introduction

Pervasive computing [1] is an emerging research field that initiates innovative concepts and ideas into other disciplines such as *Smart Spaces* and *Internet of Things*. It provides ambient services and applications that allow users, devices, and applications in different physical locations to communicate unobtrusively. In a pervasive computing environment, the devices are interconnected and embedded in physical objects to collect, process, and transport information with the least human participation.

Security is an important issue in such a decentralised environment since devices need to autonomously distinguish peers and then interact amongst them, without any human intervention. Since pervasive systems do not have any central control and the users are not predetermined, conventional access control mechanisms like authentication and authorisation are not suitable for pervasive environments. Such environments require a security architecture based on trust to handle security and privacy problems [2], [3]**.**

In pervasive computing environments, devices tend to interact without prior knowledge of each other and meanwhile need to distinguish each other autonomously without human intervention. The most noticeable properties of pervasive environments compared to other computer science domains are ubiquity, invisibility, intimate data gathering and sharing [4], [5]. As it is clear, the pervasive computing properties raised several trust issues, for example, invisible sensing of communication between two devices might happen even without user trusting any of these communication endpoints as well as the endpoints themselves. In such a decentralised environment, unprecedented data sharing could possibly allow unwanted information flow between heterogeneous entities. Therefore, providing automatic (and invisible) determination of user oriented trust calculation system is a must for any pervasive environment.

Similar to human society the most relevant sources of information to calculate trustworthiness of an entity may come from the historical transactions of that entity with other parties. We noted that the reliability of the results of these calculations depends on the accuracy of the trust models in calculating the trustworthiness and the variety of parameters taken into account.

By calculating the trustworthiness, a pervasive device can estimate as accurate as possible its peer's "honesty" before interaction occurs. In general, trust management through trustworthiness calculations, enhances security and privacy of devices in pervasive computing environments, and hence improves the efficiency and quality of the communications among devices.

One of the significant application scenarios for our research is *Healthcare Smart Space*. Implementing healthcare applications over pervasive computing infrastructure gives a considerable improvement in terms of quality, productivity and agility. The pervasive and ambient healthcare environment encompasses human beings, sensors, wearable devices, physical access controls and hospital building, clinics. Various stakeholders utilise the sensitive and confidential health information which is dynamically collected together with their contextual data. This could potentially generate new threats that do not exist in preceding systems. Since the pervasive nodes are mobile and join/leave the network continuously, we further need an additional trust evaluation schema for these tokens instead of central authentication and authorisation system. A precise and accurate trust determination mechanism to the entities within this environment can effectively help to improve the usability and acceptance factors. Alemdar and Ersoy have done a very comprehensive survey of the application of pervasive healthcare systems in wireless sensor networks and have provided a detailed analysis of its benefits and challenges [6].

The remainder of this paper is organised as follows. In section 2, we analyse previous related trust models. We explain our approach and propose our own trust model, its parts and properties in section 3. In section 4, the proposed model is analysed and evaluated by simulating it in different pervasive scenarios of the Wireless Sensor Networks. Finally, we conclude the paper and outline our research direction for the future.

## 2. Related Works

Generally, a trust relationship involves two parties: a trustor and a trustee. The trustor is the entity that holds confidence and reliance on the integrity and reliability of another one, which is the object of trust - the trustee [7].

Since pervasive entities are constantly changing, trust determination is not simply a static and trivial process. To overcome this problem several trust models have been proposed each of which focusing on one of the following trust dimensions:

- History: experience of one entity about its past interaction with its peer.
- Recommendation: experience of other entities.
- Context: comprises a set of attributes that represent the conditions under which an interaction between two entities takes place. These conditions may be the identity of the entities, their positions and the time of interaction.

The recent trust models have evolved from single-dimension trust determination models to multi-dimension trust determination models. These models merge the above mentioned dimensions to achieve more accurate trustworthiness [2], [7]–[13].

In pervasive computing environments, because of the ad-hoc nature of interactions between devices and large number of possible devices willing to communicate, while development of trust-negotiation protocols are critically required, attesting trustworthiness of the devices could be useful [7], [14].

There are many suitable hardware and software properties that can be remotely attested using TPM [15] to vouch for trustworthiness of hardware devices in pervasive environments, which namely support for mandatory controls, trusted run-times, hardware-based software integrity reporting, secure storage, support for non-repudiation in transactions, tamper-resistance, domain-based isolation, network-immunity, self-healing mechanisms, theft-deterrence and fail-secure capabilities.

Gómez Mármol and Martínez Pérez in [16] categorised the trust and reputation models for distributed environments namely WSN, P2P, Ad-hoc and Agent-based networks and compared them to come up with a standard approach in defining the models. Khiabani *et al.* in [17] reviewed and summarised the existing trust models in a tabular format based on the factors that should be considered in evaluating trust; history, recommendation, context and platform. To recap, in all these approaches, trust is influenced by direct experiences, recommendations, and context (e.g., situation, risk, time, etc.). A number of trust models support the dynamics of trust. So far, some basic elements of context such as time and context similarity have been considered. However, none of the existing works on pervasive environments give a common and unified consideration of all factors that influence the trust, and to the best of our knowledge none of them attest the platform integrity to enhance the trust establishment approach. This has been our main motivation for our proposed model presented in this paper.

In the research field of embedded systems and resource-constrained devices, which are the basic components of pervasive systems, a few researchers have proposed the idea of embedding TC component in cluster heads (or group heads) of wireless sensor networks [18], [19]. Krauß et al. proposed a lightweight attestation protocol that can be used to verify platform integrity of TC-enabled cluster heads [18]. Yang et al. proposed the same idea of a heterogeneous network whereby the nodes are divided into clusters and each cluster has a cluster head that possesses more resources and is equipped with a TPM to enforce trust mechanisms in the network [19]. The problem of these solutions is that in some situations the cluster heads are not within proximity of the sensor radio.

## 3. Proposed Model

In our previous work, we introduced Unified Trust Model (UTM) [20] and pointed out its components. In this section, we extend our previous work and formalise the components and properties of the process for calculating entities' trustworthiness.

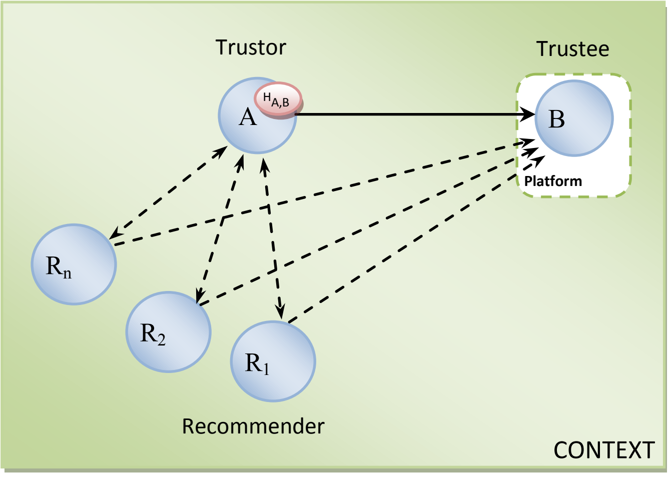


**Fig. 1.** Trust Relationship Formation in Proposed Model.

The UTM considers history, recommendation, context, and platform properties as factors for calculating trustworthiness of entities. **Fig. 1** depicts the formation of trust relationship for an entity in our proposed model. The trustworthiness of entity *B* can be evaluated by entity *A* by considering the history of past interactions as $H_{A,B}$ or recommendations of other entities in the same environment as $R_1$ to $R_n$ by considering communication context and platform properties of *B*.

The following notation introduces the entire trust of A in B ($T^{A,B}$) with respect to the four parameters that are interpreted in the next subsections.

$$T^{A,B} = f\left(T^{A,B}_{Platform}, T^{A,B}_{History}, T^{B}_{Recommendation}, Context\right)$$

It is equal to a parametric function, called $f$ , which is partially formalised and ultimately unified in the equations (7) and (8). The four parameters are platform properties($T^{A,B}_{Platform}$), history ($T^{A,B}_{History}$), recommendation ($T^{B}_{Recommendation}$), and context. All the four parameters are described and formalised in sections 3-2 to 3-5 respectively.

In the following subsection, we describe in more detail the factors considered in the trust calculation of the UTM and formalise their role in deducing trustworthiness.

## 3.1. Initialization Process

In our unified trust establishment for pervasive systems scenario, we need to specify a protocol for initializing a trust value for parties that are without any previous interactions.

Uncertainty happens when a fresh party joins a network and the other parties (its peers) have no knowledge about its trust level. One possible solution can be, by considering a coefficient of contextual state or assigning a value based on authenticated identity. In our proposed model, a scaled value of attested properties is used to initialise the trust evaluation process against fresh parties. It can also be combined in a unified trust calculation formula which considers the history, recommendation, context, and platform properties in calculating trustworthiness.

## 3.2. Platform

With the emergence of Trusted Computing (TC) [21] and its implementation like TPM (Trusted Platform Module) [15] and MTM (Mobile Trusted Module) [22], it is possible to remotely attest security properties of a remote entity.

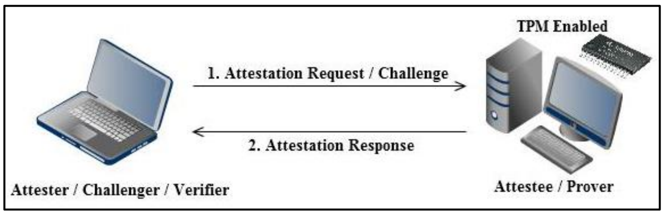

TPM Enabled
1. Attestation Request / Challenge
2. Attestation Response
Attester / Challenger / Verifier
Attestee / Prover

**Fig. 2.** Basic Remote Attestation Process.

The process of providing trustworthiness of a platform by reporting the status of its properties to the remote party is called *Remote Attestation* (**Fig. 2**). Several remote attestation techniques that leverage the capabilities of the TPM have been developed. The type of attestation that is coined by the TCG is commonly known as *Binary Attestation* [23]. To overcome the shortcomings of the binary attestation, Sadeghi and Stüble [24] introduced Property-based Attestation (PBA). Using PBA, attester verifies whether a platform conforms to particular security requirements without revealing the specific configuration information. So, the attestation of a platform is only based on the *properties* that the platform offers, but it still relies on Trusted Third Party (TTP) to match software identity to security properties. However, Chen et al. [25] presented a new PBA approach that does not depend on a TTP.

PBA gives a lot of benefits compared to the other techniques. It preserves privacy and is flexible and appropriate to a heterogeneous environment like our scenario. In our trust model, we took up this remote attestation technique and assumed that the attested properties are quantified between 0 and 1.

Therefore, a weighted mean concept is adopted to extend the quantified attested properties to a value between 0 and 1. Then, the trust value based on Platform Attestation of $B$ to $A$ ($\mathrm{T}_{\mathrm{Platform}}^{\mathrm{A,B}}$) can be obtained using equation (1) in which $p_i$ and $w_i$ are particular property of $B$ and its corresponding weight, respectively.

$$\mathrm{T}_{\mathrm{Platform}}^{\mathrm{A,B}} = \frac{\sum_{\mathrm{i=1}}^{|\mathrm{P}|} \mathrm{w_i p_i}}{\sum_{\mathrm{i=1}}^{|\mathrm{P}|} \mathrm{w_i}} \tag{1}$$

Where

$$P = \{(p_i, w_i) | i \in \mathbb{N}, w_i \in \mathbb{Z}\ and\ w_i \geq 0, p_i \in \{0,1\}\}$$

So $P$ is a set of properties that can be attested and their corresponding weight. The cardinality of $P$ shows the total number of properties that are going to be attested to. The result of Platform Attestation is a real number between 0 and 1 and properties that are more important have more contributions to the final value.

$$T_{Platform}^{A,B} \in \{x \in \mathbb{R} | x \geq 0\ and\ x \leq 1\}$$

### 3.3. History

Experience refers to the direct communication history between two entities that contains entities' information about each other. It is also called history of interactions.

Successful and unsuccessful transactions can be a good measurement to define experience of two transacting entities. The following parametric function notation introduces the parameters that are taken to account to measure entities' experience:

$$T_{History}^{A,B} = f(\mathrm{S_i}, \mathrm{U_i})$$

Where $S_i$ shows the number of successful interaction at time *i* for party *A* with *B*, and $U_i$ shows the number of unsuccessful interactions at time *i* for party *A* with *B*.

We adopted *Logistic Function* or *Pearl Curve* (named after U.S. demographer Raymond Pearl), to map the growth and decay of trust based on the past interactions (historical data). So in this model, the historical trust at time $i$ will be calculated by using equation (2).

$$T_i = \frac{1}{1 + \alpha e^{-\beta(\mathrm{S_i} - \mathrm{U_i})}} \tag{2}$$

Where $e$ is the Euler's constant, $\alpha$ and $\beta$ are the coefficients that independently control the location and shape of the Pearl Curve respectively. The curve is symmetrical about the inflection point and $\alpha$ and $\beta$ can be adjusted in accordance with the context of interaction environment. After having a successful or unsuccessful interaction, the historical trust value can be calculated by using equation (3) or (4) respectively.

$If\ s_{i+1} = (s_i + 1) and\ u_{i+1} = u_i$

$$T_{i+1} = \frac{1}{1 + \alpha e^{-\beta(f^{-1}(T_i)+1)}} = \frac{e^{\beta} T_i}{(e^{\beta} - 1) T_i + 1} \tag{3}$$

$If\ s_{i+1} = s_i\ and\ u_{i+1} = u_i + 1$

$$T_{i+1} = \frac{1}{1 + \alpha e^{-\beta(f^{-1}(T_i)-1)}} = \frac{T_i}{(1 - e^{\beta}) T_i + e^{\beta}} \tag{4}$$

The context of an interaction affects the effect of an experience in the calculation of trustworthiness between two parties. For example, consider entity *A* had a successful academic interaction with entity *B* but both have an unsuccessful financial interaction. If *A* is currently pursuing the trustworthiness of academic interactions with *B*, we should consider more positive effect for previous academic transaction in calculating *B* trustworthiness for academic interactions.

We consider three coefficients to reflect the contextual effect of successful and unsuccessful interactions in calculating entities experience trustworthiness.

$$T_{History}^{A,B} = f(S_t,\ U_t, \alpha, \beta_s, \beta_u)$$

Where α reflects the current contextual status of the entity while $\beta_s$ and $\beta_u$ are showing the contextual state of previous successful and unsuccessful transactions respectively. We can use the above coefficient to achieve following context-based trustworthiness calculation formula:

$$T_{History}^{A,B} = \frac{1}{1 + \alpha e^{-(\beta_s S_t - \beta_u U_t)}} \tag{5}$$

The above formula would lead to a trust value between 0 and 1 which reflects the historical context-based trust of an entity A on another entity *B*.

Reputation reflects the overall history of a party, and we need a third trusted party to keep the history of all entities' previous transactions for calculating the trustworthiness, but this is not applicable to decentralised pervasive environments.

### 3.4. Recommendation

The third factor which shapes a partial trust of entities to each other is recommendation of other entities. Recommendation of entity *R* about entity *B* on request of entity *A* can shape the trust of *A* on *B* based on previous trust level of *A* on *R* as well as the context of the recommender. For instance, if a university professor gives a recommendation on another professor it would be more trustable than a recommendation of a student about the same professor.

An entity *R* might recommend another entity *B* for an entity *A* if and only if *R* had some *history* with both *B* and *A*. Therefore, we can extend the historical trustworthiness calculation formula to calculate overall historical and recommendation trust of party *A* into party *B* as follows:

$$\Delta T_{His.\ \&\, Rec.}^{A,B} = \Delta T_{History}^{A,R} \times \Delta T_{\text{History}}^{R,B}$$

We may generalise the above formula for a chain of trusted entities as follows:

$$\Delta T_{H\,\&\,R}^{A,B} = \sum_{i=1}^{n} \left(\Delta T_{History}^{A,i} \times \Delta T_{\text{History}}^{i,B}\right) \tag{6}$$

While *n* is the maximum number of recommenders.

The above formula considers the overall effects of history. The contextual effects have been considered in the above formula in calculation of the *history* of the trust since α, $\beta_s$ and $\beta_u$ reflect the contextual status of the entity requested for recommendation.

### 3.5. Context

The context can be divided into two types based on their ontological texture, *Unified* and *Break-apart* texture. In unified ontology texture, all contexts are interconnected with each other and the user contextual status changes gradually while he is transferring from one context into another. In break-apart ontology texture, the texture is divided into several separated contexts so we assume the user should leave one context to join another. **Fig. 3** shows the unified and break-apart contexts.

We extend the equation (6) to include contextual effects in calculating entities overall trustworthiness. We use $\int$ in unified contexts and $\sum$ in break-apart contexts as per formulas (7) and (8) respectively.

$$\Delta T_{H\&R\&C}^{A,B} = \int_{\mathrm{Context}_1}^{\mathrm{Context}_n} \Delta \mathrm{T}_{\mathrm{H}\&R}^{\mathrm{A,B}} \tag{7}$$

$$\Delta T_{H\&R\&C}^{A,B} = \sum_{\mathrm{Context}_1}^{\mathrm{Context}_n} \Delta \mathrm{T}_{\mathrm{H}\&R}^{\mathrm{A,B}} \tag{8}$$

The above formulas consider the overall effects of context, recommendation, and history in calculation of the entity *A* trust on entity *B*.

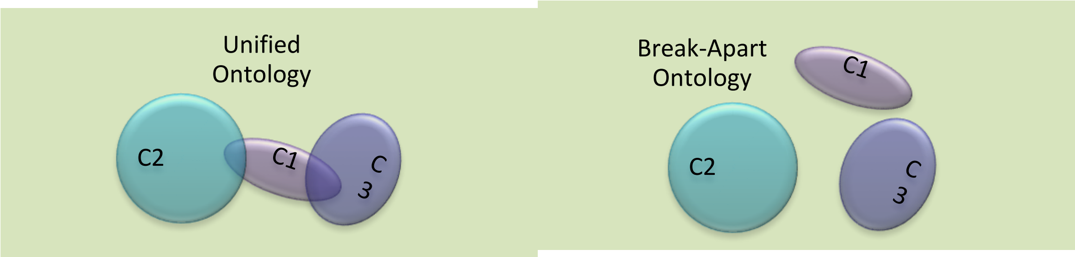


**Fig. 3.** Unified and Break-Apart Contexts.

## 4. Simulation and Results

In this section, we describe our simulation approach and analyse the simulation results from the experiments performed in order to test the accuracy and performance of our proposed model.

We evaluated our model in different aspects. In this research work, we used TRMSim-WSN [26] simulation platform as our testing environment and we implemented and integrated our model into it. TRMSim-WSN (Trust and Reputation Models Simulator for Wireless Sensor Networks) is an open-source Java-based simulation framework [27] intended to evaluate and compare trust and/or reputation models for distributed environments. So far a number of studies have done their experiments using TRMSim-WSN.

We pursued the standardisation approach introduced by Gómez Mármol and Martínez Pérez in [16] which breaks down the trust evaluation process into five components. These components are information gathering, scoring and ranking, entity selection, transaction, and rewarding and punishing. The core procedure of the simulator follows the Algorithm 1.

### 4.1. Simulation Settings

For our simulation experiments, we used a flat, rectangular area of 100 by 100 units. In the simulator, the maximum and minimum number of the sensors and the radio range of every sensor determine the link density of the network. When a new WSN is created the total number of nodes is randomly selected, i.e. a number between maximum and minimum number of nodes. The average number of the neighbours of each node can be obtained by $n\pi R^2/100^2$ where n and R is the total number and radio range of sensors respectively. So, for example, if we want every sensor to have an average of five direct neighbours, the radio range would be $100\sqrt{5/n\pi}$.

In this simulation environment, the percentage of the nodes that we want to act as a client can be adjusted. The total number of clients is obtained after randomly creating them up to the specified percentage and the rest of the nodes will act as a server. Similarly, the percentage of the servers who do not offer the required service and only relay the transaction can be adjusted. From the rest of the servers who offer the required service, we can set the percentage of them that behave maliciously and provide the required service defectively or unsatisfactorily. The remaining are the benevolent servers. Among the benevolent servers, we can set the number of TC-enabled nodes that have a TPM chip inside and can attest their trustworthiness to the other peers.

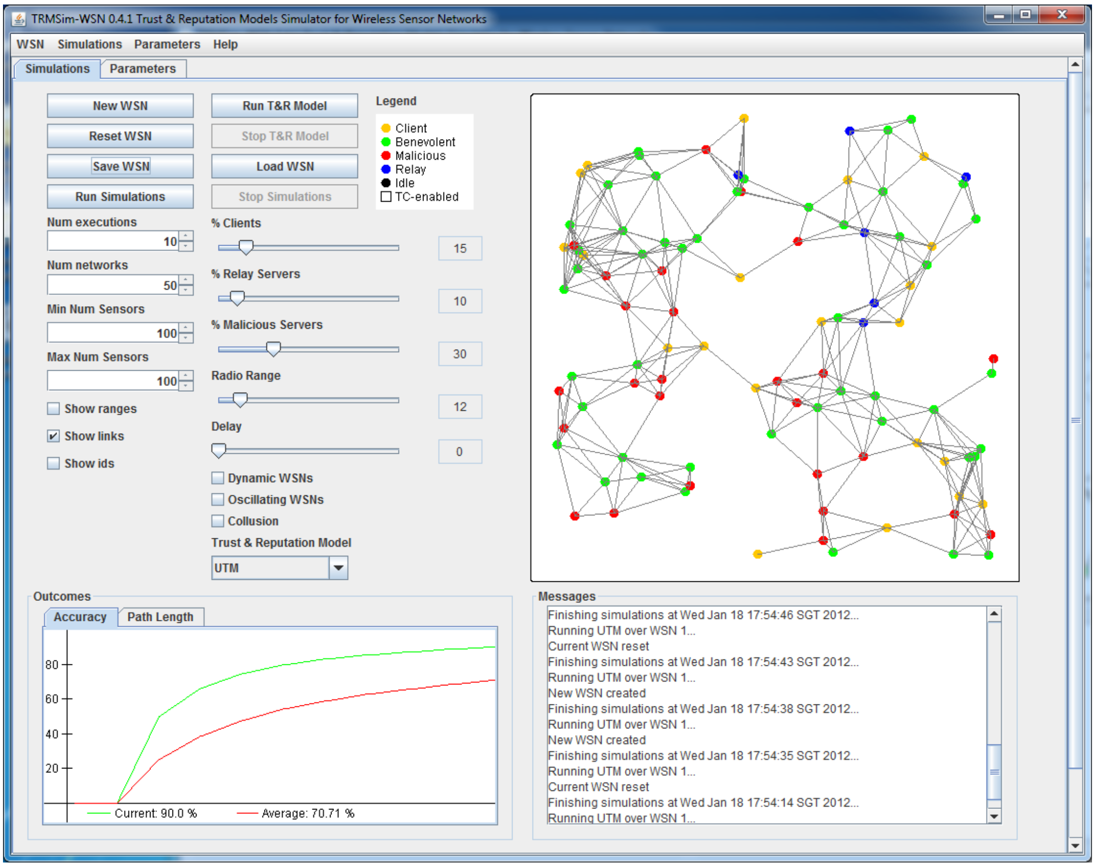
Algorithm 1. Simulation Algorithm

```
for i=1 to Number_ of_ Networks do
    CreateNewNetwork (NumSensors, RadioRange, %Clients, % RelayServers,
     %MaliciousServers)
    for j=1 to Number_ of_ Executions do
        Gather Information()    // find all the possible paths to the reachable servers
                                   who offers the desired service.
        MaxTrust = 0
        for path :  All_Path_to_Servers do        // Score and Ranking
            if  CalculateTrust(path.server) > MaxTrust then
                MostTrustworthyPath = Path
            End if
        end for
        PathLength = Length(MostTrustworthyPath)
        If  PerformTransaction(MostTrustworthyPath) = IsSatisfied then
        // Perform Transaction
            Satisfaction=1
            //The next trust value will be calculated using formula (3)
        else
            Satisfaction=0
            //The next trust value will be calculated using formula (4)
        end if
    end for
    Accuracy = Percentage of benevolent servers selected and transaction was successful
end for
```

In order to simulate the UTM, we need to set the number of executions, which is the number of times each client requests for the service. The number of different random WSNs needs to be set as well.

Two security threats are included in the simulator to test the accuracy of the trust model, oscillating behaviour and collusion. A good trust model should quickly respond to these threats and avoid selecting a malicious node as the most trustworthy one. For simplicity, in this simulation, when the oscillating behaviour option is selected, every 20 executions of trust model each malicious server becomes benevolent (in reality this would not be constantly 20 executions, rather it would be also random) and the same percentage of previous malicious servers are randomly selected to be malicious now. When collusion option is selected, every malicious sensor gives the maximum rating for every other malicious sensor, and the minimum rating for every benevolent one.

Fig. 4 depicts the simulation settings and results for an example experiment that we launched our UTM trust model over 50 random WSNs. In each run, we initiate a new WSN network that is composed of 100 nodes, randomly generated and randomly positioned. The probability of the clients was set to 15%. The maximum percentages of the relay, malicious and benevolent servers are 8.5% (85×10), 22.95% (85×90×30) and 53.55% (85×90×70) respectively. The node transmission distance is set to 12 units in this experiment, so every node has 4-5 direct neighbours on average. We executed the UTM trust model 10 times over the WSN and the average percentage of times that the model selected a benevolent server as the most trustworthy one (accuracy) was 70.71%.



Num executions: 10

Num networks: 50

Min Num Sensors: 100

Max Num Sensors: 100

□ Show ranges

☑ Show links

□ Show ids

% Clients: 15

% Relay Servers: 10

% Malicious Servers: 30

Radio Range: 12

Delay: 0

□ Dynamic WSNs

□ Oscillating WSNs

□ Collusion

Trust & Reputation Model: UTM

Current: 90.0 %

Average: 70.71 %

```
Finishing simulations at Wed Jan 18 17:54:46 SGT 2012...
Running UTM over WSN 1...
Current WSN reset
Finishing simulations at Wed Jan 18 17:54:43 SGT 2012...
Running UTM over WSN 1...
New WSN created
Finishing simulations at Wed Jan 18 17:54:38 SGT 2012...
Running UTM over WSN 1...
New WSN created
Finishing simulations at Wed Jan 18 17:54:35 SGT 2012...
Running UTM over WSN 1...
Current WSN reset
Finishing simulations at Wed Jan 18 17:54:14 SGT 2012...
Running UTM over WSN 1...
```

Fig. 4. TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks.

## 4.2. Experiments and Results

We designed and performed three different experiments called “normal executing”, “oscillating” and “number of interactions effect”. For each experiment, we carried out the experiment with and without having TC-enabled nodes and then measured the accuracy of the model and the average length of the path that go to the selected benevolent servers. The accuracy in this work is defined as the percentage of selecting an actually trustworthy server and correspondingly the length of the path to that server which represent its performance and energy consumption of the nodes. **Table 1** summarises the parameters used to perform these experiments.

**Table 1.** Parameters Value

| Parameter | Value |
|---|---|
| Number of Executions | 100 |
| Number of Networks | 100 |
| Number of Clients Probability | 15% |
| Number of Relay Servers Probability | 5% |
| Number of Malicious Severs Probability | {20%, 40%, 60%, 80%} |
| Minimum Number of Sensors | {500, 300, 100, 50} |
| Maximum Number Sensors | {500, 300, 100, 50} |
| Radio range | {5, 7, 13, 18} |
| TC_enabled Nodes Percentage | {0, 0.1} |
| TC_enabled Nodes Weight | 0.8 |
| Window Size | 5 |
| Alpha & Beta | 1.0 |

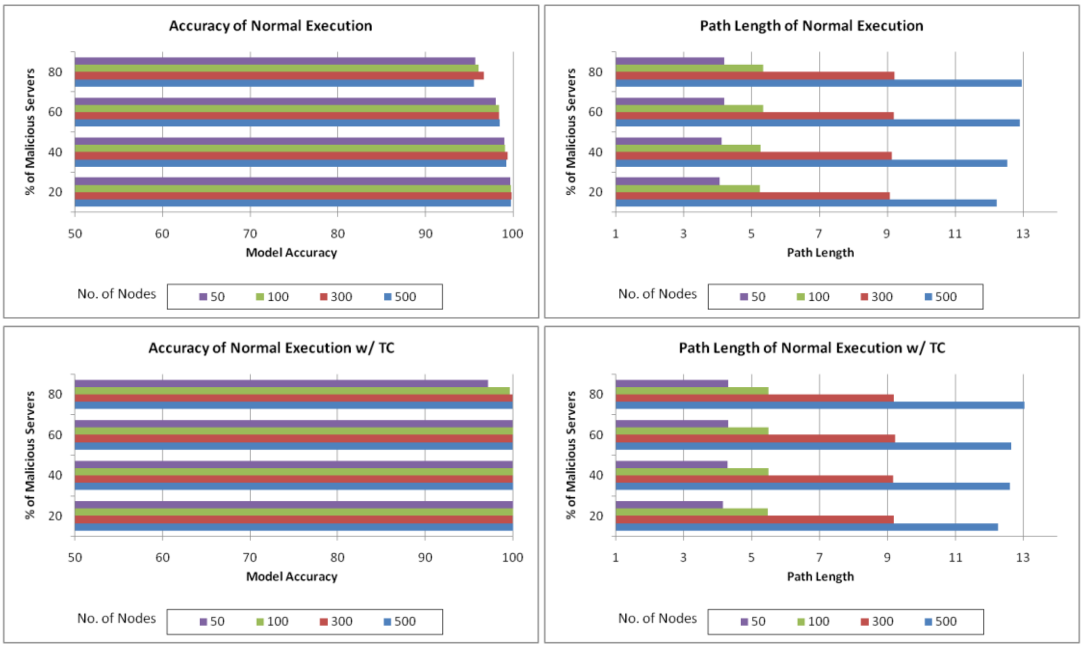
**Experiment One** – UTM trust model in normal execution

In the first experiment, we executed the UTM trust model 100 times over 100 different random WSNs composed by 50, 100, 300 and 500 sensors with transmission distance of 18, 13, 7 and 5 units respectively. The maximum percentage of the clients was set to 15% and the probability of relay servers from the rest of the sensors was 5%. In order to test the robustness of the UTM, for each WSN series, we increased the probability of malicious servers from 20% to a maximum of 80% and kept all the other settings as mentioned above.

The results are indicated in **Fig. 5**. In analysing the result, it can be seen from this figure that the model has a quite good accuracy (higher than 95% even when the probability of malicious server is 80%) regardless the size of the network. The model accuracy also diminishes slightly as the number of malicious servers increases. The experiment proves that the model is fully scalable in terms of accuracy. Let us analyse the performance which is represented by the average length of the path that goes to the most trustable server. In the simulation, the average length of the path increases as the network size grows. As the TC-enabled nodes are gradually added to the network (right graph) the model accuracy became more stable. As the network increases in size and the probability of malicious servers also increases. In most of the occasions, a client almost would never select a compromised node. From the figure, the performance of the model, either without (left graph) or with (right graph) TC-enabled nodes are almost similar, i.e. means that there is no performance penalty on using TC-enabled platforms.

**Experiment Two** – UTM trust model with malicious nodes oscillating

In the second experiment, we tested the resilience of the proposed model by evaluating the accuracy of the model when the nodes do not exhibit the same behaviour during the experiment, and the servers oscillate between benevolent state and malicious state. This oscillating behaviour is typical of a malicious attacker wanting to have access to private information within the pervasive environment. In our simulation, the fluctuation pattern is fixed (again this is meant to simplify the simulation environment) and occurs every 20 executions and the nodes that change their behaviour are randomly selected. However the percentage of malicious nodes remains same after this fluctuation. We run the model 100 round over 100 different random WSNs composed by 50, 100, 300 and 500 sensors with transmission distance of 18, 13, 7 and 5 units respectively. The maximum percentage of the clients was set to 15% and the probability of relay servers from the rest of the sensors was 5%. The probability of malicious servers was gradually increased from 20% to a maximum of 80% and all the other settings were kept as mentioned above.


Accuracy of Normal Execution

Path Length of Normal Execution

Accuracy of Normal Execution w/ TC

Path Length of Normal Execution w/ TC

**Fig. 5.** UTM Accuracy and Path Length without and with TC-enabled Nodes in Normal Executing Experiment.

As it can be observed in **Fig. 6**, our model responded reasonably well against this kind of threat. In **Fig. 6**, without having TC-enabled nodes in the network (left graph), the greater the percentage of malicious nodes is, the worse the results are, for instance, response to malicious attacks becomes less accurate. Although the accuracy of the model gets worse as the percentage of malicious servers increases, it is still within acceptable margin (higher than 70% even when the probability of malicious server is 80%). When the network is supplemented with the TC-enabled nodes (right graph) the model reacts very well to this threat, and the model accuracy becomes very stable and constant. In the experiment, it is obvious that adding more TC-enabled platforms could potentially improve response to malicious attacks. While the network increases in size and the probability of malicious servers raises, the TC-enabled nodes are able to avoid the oscillation threat to success. From the performance point of view, as per the previous experiment, there is no performance penalty for using TC-enabled platforms.

**Experiment Three** –UTM trust model with number of interactions effect

In the third experiment, we evaluated the effect of interactions count on the accuracy of the model. We executed the model over 200 different random WSNs composed of 100 sensors with transmission distance of 13 units. The maximum percentage of the clients was set to 15% and the probability of relay servers from the rest of the sensors was 5%. In order to test the impact of changes in the number of interactions in our model, we carried out the model 5, 10, 20, 30 and 100 times over the same WSN and for each of them we increased the probability of malicious servers from 20% to a maximum of 80%.

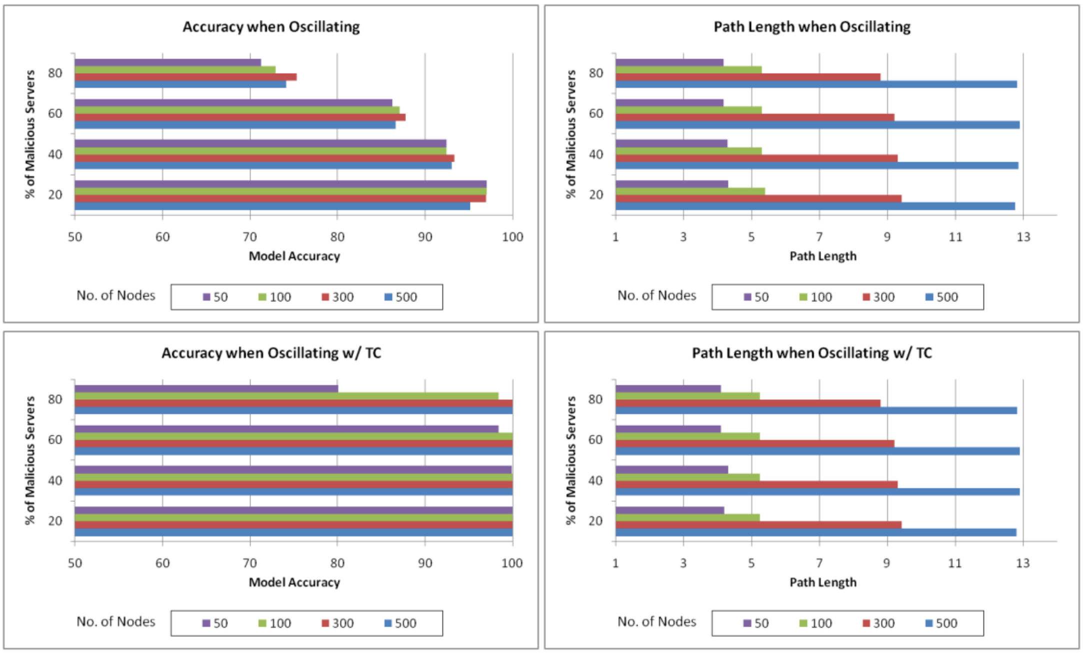


**Fig. 6.** UTM Accuracy and Path Length without and with TC-enabled Nodes in Oscillating Experiment.

**Fig. 7** depicts the outcome of this experiment. It can be seen from **Fig. 7** that, when we have no TC-enabled nodes (left graph), and when the number of the malicious servers remains equal, the more the interactions, the better accuracy we get. It is obvious that when number of interactions is less than 10, the model is vulnerable against high number of compromised nodes. However, when we utilise the TC-enabled nodes in our experiment (right graph), it can be seen that the clients can reach the desired service almost every time even by having a few interactions.

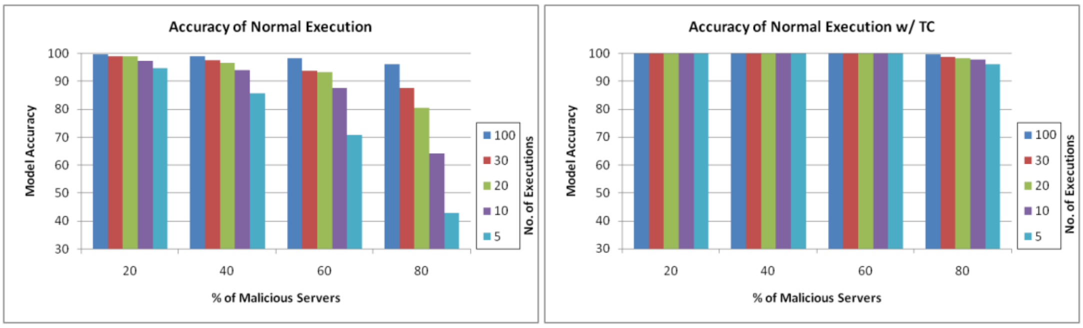


**Fig. 7.** No. of Interactions Effect on UTM Accuracy without and with TC-enabled Nodes in Normal Executing Experiment.

**Fig. 8** reflects the outcome of the above mentioned experiment while the nodes oscillate between benevolent and malicious (i.e. When malicious nodes try to attack the network). The rest of the experiment remains same and similar inference can be drawn.

We deduce that in the environments where new parties join and leave frequently (this scenario is pertinent in a pervasive environment), and the nodes have a low number of interactions, our model by leveraging the TC-enabled nodes demonstrate a very good accuracy even when the number of the compromised nodes is high i.e. when their behavior fluctuates between malicious and benevolent.

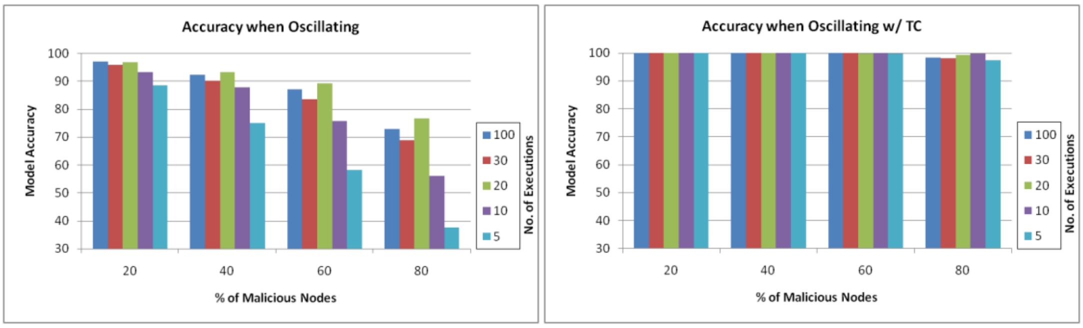


**Fig. 8.** No. of Interactions Effect on UTM Accuracy without and with TC-enabled Nodes in Oscillating Experiment.

## 5. Conclusion and Future Works

In this paper, we highlighted the need for a comprehensive trust model which unifies several factors in trustworthiness calculation. We have elaborated Unified Trust Model (UTM) which calculates entities trustworthiness based on history, recommendation, context and platform integrity measurement (used in remote attestation), and formally uses these factors in trustworthiness calculation. We also described how TPM can be incorporated for measuring trustworthiness of a party. We performed simulation and analysis of the proposed model in different scenarios of the Wireless Sensor Networks and demonstrated its accuracy and performance. From the analysis of the experiments, we deduced that by including TC-enabled nodes in the proposed model have notably increased the accuracy of trust evaluation in pervasive environments and can effectively increase the response of the model to malicious node attacks. This has been demonstrated to be true even if the number of transactions is low and if the nodes frequently join and leave the network. In conclusion, we have evaluated UTM by simulating in different experiments and showed that the model offers responsive behaviour and can be used effectively in the low interaction environments. As for our future work, we will compare our model and its simulation results with the other related models.

## References

[1] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 75–66, 1991. Article (CrossRef Link)

[2] T. Sun and M. K. Denko, "Performance Evaluation of Trust Management in Pervasive Computing," in *Proc. of the 22nd International Conference on Advanced Information Networking and Applications*, 2008, pp. 386–394. Article (CrossRef Link)

[3] L. Kagal, T. Finin, and A. Joshi, "Trust-Based Security in Pervasive Computing Environments,"

*Computer*, vol. 34, no. 12, pp. 154–157, 2001. Article (CrossRef Link)

[4] S. Lahlou, M. Langheinrich, and C. Röcker, "Privacy and trust issues with invisible computers," *Communications of the ACM*, vol. 48, no. 3, pp. 59–60, 2005. Article (CrossRef Link)

[5] M. Langheinrich, "Privacy by Design - Principles of Privacy-Aware Ubiquitous Systems," in *Proc. of the 3rd international conference on Ubiquitous Computing*, Atlanta, Georgia, USA, 2001, pp. 273–291.

[6] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, Oct. 2010. Article (CrossRef Link)

[7] Z. Yan and S. Holtmanns, "Trust Modeling and Management: from Social Trust to Digital Trust," in *Proc. of Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions*, IGI Global, 2008, pp. 290–323. Article (CrossRef Link)

[8] C. T. Nguyen, O. Camp, and S. Loiseau, "A Bayesian network based trust model for improving collaboration in mobile ad hoc networks," in *Proc. of the 2007 IEEE International Conference on Research, Innovation and Vision for the Future*, 2007, pp. 144–151. Article (CrossRef Link)

[9] C. T. Nguyen and O. Camp, "Using Context Information to Improve Computation of Trust in Ad Hoc Networks," in *Proc. of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication*, 2008, pp. 619–624. Article (CrossRef Link)

[10] K. Sarkio and S. Holtmanns, "Tailored trustworthiness estimations in Peer-to-Peer networks," *International Journal of Internet Technology and Secured Transactions*, vol. 1, no. 1/2, pp. 95 – 107, 2007. Article (CrossRef Link)

[11] S. Holtmanns and Z. Yan, "Context-Aware Adaptive Trust," *Developing Ambient Intelligence*, Springer Paris, 2006, pp. 137–146. Article (CrossRef Link)

[12] Z. Liu, A. W. Joy, and R. A. Thompson, "A Dynamic Trust Model for Mobile Ad Hoc Networks," in *Proc. of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems FTDCS 2004*, 2004, pp. 80–85.

[13] Y. Wang and V. Varadharajan, "Trust2: Developing Trust in Peer-to-Peer Environments," in *Proc. of the 2005 IEEE International Conference on Services Computing - Volume 01*, 2005, pp. 24–34. Article (CrossRef Link)

[14] K. Ranganathan, "Trustworthy Pervasive Computing: The Hard Security Problems," in *Proc. of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004, pp. 117–121. Article (CrossRef Link)

[15] TCG, "Trusted Computing Group - Trusted Platform Module," 2011. [Online]. Available: http://www.trustedcomputinggroup.org/developers/trusted_platform_module. [Accessed: 07-Dec-2011].

[16] F. Gómez Mármol and G. Martínez Pérez, "Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems," *Computer Standards & Interfaces*, vol. 32, no. 4, pp. 185–196, Jun. 2010. Article (CrossRef Link)

[17] H. Khiabani, J. Ab Manan, and Z. M. Sidek, "A Study of Trust & Privacy Models in Pervasive Computing," in *Proc. of International Conference for Technical Postgraduates (TECHPOS 2009)*, Kuala Lumpur, 2009. Article (CrossRef Link)

[18] C. Krauß, F. Stumpf, and C. Eckert, "Detecting node compromise in hybrid wireless sensor networks using attestation techniques," in *Proc. of the 4th European conference on Security and privacy in ad-hoc and sensor networks*, Berlin, Heidelberg, 2007, pp. 203–217. Article (CrossRef Link)

[19] Y. Yang, J. Zhou, R. H. Deng, and F. Bao, "Better security enforcement in trusted computing enabled heterogeneous wireless sensor networks," *Security and Communication Networks*, vol. 4, no. 1, pp. 11–22, 2011. Article (CrossRef Link)

[20] H. Khiabani, Z. M. Sidek, and J. Ab Manan, "Towards a Unified Trust Model in Pervasive Systems," in *Proc. of 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, Perth-Australia, 2010, pp. 831–835. Article (CrossRef Link)

[21] TCG, "Trusted Computing Group," 2011. [Online]. Available: http://www.trustedcomputinggroup.org/. [Accessed: 07-Dec-2011].

[22] TCG, "Trusted Computing Group - Mobile Trusted Module," 2011. [Online]. Available: http://www.trustedcomputinggroup.org/developers/mobile. [Accessed: 07-Dec-2011].

[23] R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn, "Design and implementation of a TCG-based integrity measurement architecture," in *Proc. of the 13th conference on USENIX Security Symposium - Volume 13*, Berkeley, CA, USA, 2004, pp. 16–16.

[24] A.-R. Sadeghi and C. Stüble, "Property-based attestation for computing platforms: caring about properties, not mechanisms," in *Proc. of the 2004 workshop on New security paradigms*, New York, NY, USA, 2004, pp. 67–77. Article (CrossRef Link)

[25] L. Chen, H. Löhr, M. Manulis, and A.-R. Sadeghi, "Property-Based Attestation without a Trusted Third Party," *Information Security*, vol. 5222, T.-C. Wu, C.-L. Lei, V. Rijmen, and D.-T. Lee, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 31–46. Article (CrossRef Link)

[26] F. Gómez Mármol and G. Martínez Pérez, "TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks," in *Proc. of IEEE International Conference on Communications (ICC '09)*, 2009, pp. 1–5. Article (CrossRef Link)

[27] F. Gómez Mármol, "TRMSim-WSN - Trust and Reputation Models Simulator for Wireless Sensor Networks," 2012. [Online]. Available: http://ants.dif.um.es/~felixgm/research/trmsim-wsn/. [Accessed: 21-Jan-2012].

**Norbik Bashah Idris** was Founder and Director of the Centre for Advanced Software Engineering, University of Technology Malaysia and is currently a professor in the Advanced Informatics School of the same university. He received his first of many graduate honours in Australia and then received his Ph.D in the area of IT security from the University of Wales, United Kingdom. He has been Chairman of the ICT Security Standardization Committee for the Malaysian Government EG-project, consultant and advisor to ministries, government agencies and private sector, and a regular speaker at seminars, conferences, TV at both national and international levels. Norbik was awarded the first Distinguished Senior IT Security Professional for Asia-Pacific by ISC2 in 2007 and the 2007 Innovative Entrepreneur of the Year by the Malaysia Chambers of Commerce. He was also a top-nominee for the Ernst & Young Technology Entrepreneur for 2007. As an academician, Norbik lectures and researches mainly in ICT Security and Software Engineering.

**Jamalul-Lail Ab Manan gra**duated from University of Sheffield, UK with a Bachelor in Electrical Engineering (BEng). He pursued his Master of Science (MSc) in Microprocessor Engineering at University of Bradford, UK and PhD in Communications Engineering at University of Strathclyde, Glasgow, UK. He is currently leading the Cryptography Laboratory at Advanced Analysis and Modeling (ADAM) Cluster, MIMOS Berhad. He has 17.5 years of experience in teaching subjects in Electrical and Electronics Engineering, Microprocessor Engineering and Computer Science. He has more than 13 years of industrial experience as a Network Engineer, Senior Manager, Senior Vice President in ICT based government companies in Malaysia. His current research focus is on Trusted Computing, Privacy Enhancing Technologies and Cryptography.

**Hamed Khiabani** is currently pursuing his PhD degree in Computer Science (information Security) in the University of Technology, Malaysia (UTM) while attached to MIMOS Berhad under a research collaboration program. He received his BEng. of Computer Engineering (Hardware) in 1995 from the Iran University of Science and Technology (IUST) and his MSc. in Computer Engineering in 1998 from the Azad University-Science & Research Campus. He is a member of International Electrical and Electronics Engineers (IEEE), IEEE Computer Society, and IEEE Communications Society. He is currently active in the cyber security and digital forensics research community and his main research interests include trusted computing and trust management in pervasive computing environment.