10-2007

A Software-Based Trust Framework for Distributed Industrial Management Systems

Sheikh Iqbal Ahamed
Marquette University, sheikh.ahamed@marquette.edu

Mohammad Zulkernine
Queen's University - Kingston, Ontario

Steve Wolfe
Marquette University

Journal of Systems and Software, Vol. 80, No. 10 (October 2007): 1621-1630. DOI.

A Software-Based Trust Framework for Distributed Industrial Management Systems

Sheikh I. Ahamed
Department of Mathematics, Statistics, and Computer Science, Marquette University, Milwaukee, WI
Mohammad Zulkernine
School of Computing of Queen's University, Canada
Steve Wolfe
Ubicomp Research Lab, Mathematics, Statistics, and Computer Science, Marquette University, Milwaukee, WI

Abstract

One of the major problems in industrial security management is that most organizations or enterprises do not provide adequate guidelines or well-defined policy with respect to trust management, and trust is still an afterthought in most security engineering projects. With the increase of handheld devices, managers of business organizations tend to use handheld devices to access the information systems. However, the connection or access to an information system requires appropriate level of trust. In this paper, we present a flexible, manageable, and configurable software-based trust framework for the handheld devices of mangers to access

distributed information systems. The presented framework minimizes the effects of malicious recommendations related to the trust from other devices or infrastructures. The framework allows managers to customize trust-related settings depending on network environments in an effort to create a more secure and functional network. To cope with the organizational structure of a large enterprise, within this framework, handheld devices of managers are broken down into different categories based upon available resources and desired security functionalities. The framework is implemented and applied to build a number of trust sensitive applications such as health care.

Keywords

Security engineering, Trust management, Distributed industrial management systems

1. Introduction

Managers of business organizations often need to have access to information systems. They can make use of handheld devices and wireless networks to access the information systems of their organizations (Weiser, 1991, Gupta et al., 2001). As organizations rely more and more on geographically dispersed devices, there must be greater awareness about the trustworthiness and security of the communications that take place among the devices.[1]

Traditional security mechanisms (hard security) such as authentication and access control are not sufficient to protect resources by restricting unauthorized users in a distributed environment (Rasmusson and Jansson, 1996). In many cases, access control and authentication are required to be governed by the principles of "soft security" such as trust and reputation. Although the importance of trust is widely acknowledged, only a few research works have addressed it with appropriate level of importance, and trust is still an afterthought in most security engineering projects. "Trust is the extent to which one party is willing to depend on somebody, or something, in a given situation with a feeling of relative security, even though negative consequences are possible" (Jøsang et al., 2005, McKnight and Chervany, 1996). It is a "directional relationship between two parties that can be called trustor and trustee" (Jøsang et al., 2005). A trustor evaluates and makes decisions about the dependability of the trustee in a specific situation, while the trustee proves his or her dependability. The task of trust management includes building or defining trust levels, evaluation of trust levels, and making decisions based on the evaluated levels of trust.

One of the major problems in industrial security management is that most organizations or enterprises do not provide adequate guidelines or well-defined policies with respect to trust management *i.e*., identification and authentication. Most of them practice ill-defined and un-quantified methods for the assessment of the trustworthiness of the communicating parties. A number of process models and tools have been proposed for the efficient management of information security engineering tasks (Kim and Leem, 2004, Kim et al., 2004, Kim and Lee, 2005, Leem et al., 2005, Kim and Lee, 2006). However, none of them explicitly addresses the issues of trust management, where a number of users or devices of different levels of trust communicate with each other.

The deployment of handheld devices in a corporate or business environment requires serious thought and planning with respect to the levels of trust among the devices. It is very obvious that geographically dispersed devices or users invite additional challenges in terms of authentication, integration, and communication management. They require a higher degree of vigilance and security management. Essentially, the most important matter boils down to device authentication. In a traditional wired network or even an infrastructured wireless network, a single centralized device or a group of devices is responsible for performing this functionality. For instance, Public Key Infrastructure (PKI) provides a method of authentication using identity certificates that are issued by a trusted certification authority (Weise, 2006). Due to the nature of the mobility of

the managers with handheld devices, authentication schemes involving a centralized approach, like PKI, are not practical. Moreover, the devices or users have to adapt changes in the levels of trust through the use of automatic software tools.

In this paper, we propose a flexible software-based trust framework for a distributed industrial information system that can be customized to access the information system by its managers.[2] The paper identifies the essential characteristics of a managerial device to access distributed management information systems and presents the design of the framework. The steps of the framework are explained in detail using appropriate examples for different scenarios. It explains the categorization of managerial devices and the quantification of trust. A health care application is described which was developed based on the principle of this framework.

The major contributions of this work are the following. It provides a trust framework capable of managing the trust levels of rapid handheld devices for accessing distributed information systems. It allows for configuring a number of customizable options regarding the trust of the managerial devices. The more valid pre-configuration possible the more secure a network can be. The framework allows managers the ability to customize a network to take the advantage of the known characteristics of the network and any other available knowledge. It provides a more resilient network environment in a number of possible scenarios. When a direct analysis is not possible for a managerial device, other neighboring devices or infrastructures can provide this functionality and in doing so preserve battery life.

The overall implications of this work are as follows. The framework provides a set of methods for practicing trust management in a modern industrial security management setting, where a large number of handheld devices containing sensitive information are in use. It will in general help in strategic and tactical planning of a security engineer for industrial information security by minimizing the risk of exposure of assets and resources of the organization. Another important implication of this work is that it improves people's and organization's acceptance of wireless environments as a safe media for confidential interactions by managing devices' (people's) trust in each other.

Thus far, we have presented the motivation for the systematic management of trust in an organization. Section 2 identifies the characteristics of a managerial device to access distributed information systems and provides an overview of the design of the proposed trust framework. Section 3 discusses a number of related works with respect to security engineering methodology and trust management. Section 4 describes the management of trust in the framework by delineating the categories of devices and explaining the mechanisms involved. Section 5 explains the implementation and the evaluation of the framework. Section 6 summarizes the paper and identifies some future work.

2. Overview of the trust framework

2.1. Characteristics of a managerial device

The design of the proposed trust framework considers the following essential characteristics for a managerial device to access distributed management information systems:

(A1) Distributed: The trust framework must be distributed in some way. Since the managers are on move, a traditional or centralized approach is not suitable for this.
(A2) No pre-configuration necessary: Pre-configuration such as setting shared keys in multiple devices is a timely process and should not be a requirement for the deployment.
(A3) Capability to customize: Different managers can utilize the deployment environment to make the network more secure and functional. The trust framework should allow for security level customization.

(A4) Circumvent malicious recommendations: A recommendation-based trust framework should contain processes to identify and minimize malicious recommendations.
(A5) Prevent attack from variety of attack scenarios: An adaptive trust framework should perform reasonably well against adversary network models made of either a single device, a small number of devices, or a large number of devices.
(A6) Recalculation: The trust framework should involve a periodic update of trust values. Ideally, the time window between updates should be minimal to quickly respond to new threats.
(A7) Low memory footprint/resource usage: All mobile devices of the managers are typically constrained by battery life and computational power. Any trust service should make efficient use of resources to prevent unnecessary overhead.

2.2. Design overview

The trust framework is designed by taking into account the characteristics of a managerial device identified in Section 2.1, the organizational structure of a typical industrial management system, and network environments. The framework is intended for usage at application layer on top of an operating system. The trust system is designed to act as a thin client module residing with a group of services (such as security service) as shown in Fig. 1. These services are essential and available to all applications. The trust service can be used by both applications and security services or any other services. To allow appropriate access, the trust service resides below the applications of a manager's device and offers an interface to allow applications retrieve a trust value and deterministically allow, disallow, or limit the communications with the intended devices. Within the service itself, several key components make up the trust service (see Fig. 1). The *r*(*ecommendation*)*Value* manager estimates the recommendation values for all the devices that are used by the trust calculation component to calculate trust along with the recommendations based on the directly monitoring data of the device. The request/response handler requests and replies to the recommendation queries from the neighboring devices. The application request handler replies to the trust related queries from the applications. It can be used by the manager to customize the trust service of the existing network environment.

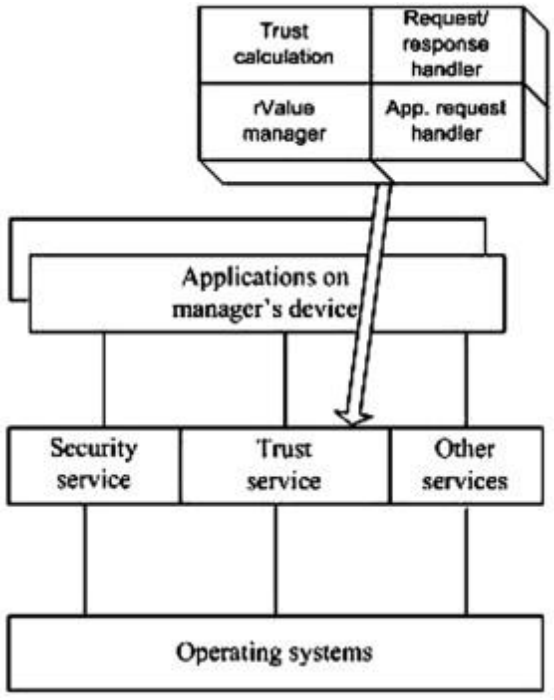


Fig. 1. Trust service on a device.

3. Related work

In this section, we discuss a number of security engineering methodologies and trust models in the context of our work. The trust models used for handheld devices are reviewed with respect to the characteristics identified in Section 2.1.

Extensive research have been performed on security engineering methodology by addressing the major issues that various types of industrial security management systems might face for their information security (Kim and Leem, 2004, Kim et al., 2004, Kim and Lee, 2005, Leem et al., 2005, Kim and Lee, 2006). Kim and Leem (2004) propose an information engineering methodology for security strategy planning. The methodology provides systematic steps and tools for planning and managing information security controls and for reconciling the security strategies with other strategies of an enterprise. Kim et al. (2004) propose a new architecture of authentication mechanism which is suitable for Digital Television Commerce (T-commerce) environments. The authentication mechanism allows users to order digital TV programs from their handheld devices. A cost benefit analysis methodology (a process model and analysis criteria) is proposed for the economic justification of the security investments of an organization. In Leem et al. (2005), an assessment methodology on the maturity level of information security management system is proposed. The assessment is based on the technical, managerial, and operational features of information security. Another security engineering methodology is presented based on a problem solving theory (Kim and Lee, 2006). The methodology allows requirements analysis and suggests a process model and components for ill-defined problems of information security. Leem and Kim (2002) present an integrated methodology for successful development and implementation of enterprise information systems. While the aforementioned works cover a wide range of industry applications, they do not address the issues with respect to trust – an important aspect of soft security (Rasmusson and Jansson, 1996). The secure service discovery within pervasive computing environment is addressed in (Zhu et al., 2005) but they do not address trust. Moreover, they also need to use infrastructure (high end servers and proxies).

A trust model-based qualitative risk management approach for distributed system security is proposed in (Lin and Varadharajan, 2006). The model uses trust assessment results not only to prevent the access of unwanted users but also to provide access permissions to the trusted users. The method can be employed to maximize the utility of distributed systems.

Some recent approaches (Yi and Kravets, 2002, Pirzada and McDonald, 2004, Luo et al., 2002) for ad hoc environments use reactionary distributed approach that shift the burden of authentication to the all or subset of the devices in the network. Distributed approaches can differ considerably amongst themselves. The Resurrecting Duckling model forces the network to form a hierarchical master/slave pairing (Stajano and Anderson, 1999). The slave receives authentication information from its master only. In contrast, most distributed approaches (Pirzada and McDonald, 2004, Luo et al., 2002) involve the device making a judgment based upon its observations and recommendations from the neighboring devices.

Abdul-Rahman and Hailes (1997) propose a distributed trust model for managing trust. Their approach provides a quantitative scheme for trust using distributed recommendations. Within their model, each device maintains a discrete trust value for the neighboring devices. The proposed values range from −1 (completely untrustworthy) to 4 (completely trusted). It requires trust to be transitive and devices make recommendations on another device's behalf. Recommendations are conditional based upon the recommending device's trust level. The trust value assigned by a device is computed based upon the recommended trust values and the trust values of their recommenders. Hence, it needs additional external processing support to deal with false or malicious recommendations. Moreover, it is not intrinsically designed to support an ad hoc network. Thus it is not suitable

for distributed information systems, where managerial devices are on move and want access in an ad hoc manner.

In self-securing ad hoc networks (Luo et al., 2002), devices are trusted unless there is a first-hand evidence or a sufficient number of neighboring devices stating otherwise. This research proposes a distributed authentication mechanism (A1, see Section 2.1) where authenticated devices posses a valid certificate. To obtain a certificate, a specified number of devices must vouch for the validity of that device. It hinges upon a localized trust scheme to determine whether to offer a partial certificate. In the absence of direct evidence or overwhelming recommendations from the neighboring devices, a device offers its partial certificate to any requesting device. Certificates are only temporary (A6) and eventually needs to be renewed. This re-certification mechanism ensures that a misbehaving device is not allowed to remain connected. The approach is targeted for purely ad hoc environments (A2) and requires that all devices perform monitoring and analysis.

Pirzada and McDonald (2004) propose a distributed trust model (A1) based upon direct monitoring in combination with the utility and importance of the situation. In this model, trust values form a continuous range from −1 to 1 representing complete distrust to complete trust respectively. The solution separates the trust calculation into numerous trust categories. The addition of importance and utility is included to account for the spontaneous nature of ad hoc networks. They explicitly contrast the differences between “managed” and “pure” ad hoc environments. Their solution is designed specifically for “pure” ad hoc environments, requiring no pre-configuration (A2). The calculation for trust contains a mechanism for devaluing the influence of malicious devices and recalculating the perceived trust (A4, A6). It lacks the support for different levels of managers when the managerial devices are mobile.

Sun and Song (2004) present a trust framework for an ad hoc networking environment based upon game theory and distributed algorithm principles. The framework calculates trustworthiness on the reputation value of the device, the environment, time, and other quantities (A6). In this model, reputation is composed of two parts: the action history of the device and the recommendation of other devices. Their trust framework requires that each device broadcasts its trustworthiness to the network upon entering. This feature creates a vulnerability to the malicious devices that misrepresent their trustworthiness to the network. Hence, it does not fit for managerial devices. Additionally, their proposal assumes that malicious devices cannot work together.

4. Management of trust in the framework

The aim of this work is to provide a configurable framework flexible enough to work efficiently in a mobile environment for the managers of the companies. The proposed framework consists of a number of major steps: categorizing managerial devices, quantifying trust, finding trust value, and calculating trust. The steps are shown in Fig. 2 and are explained in the following sections.

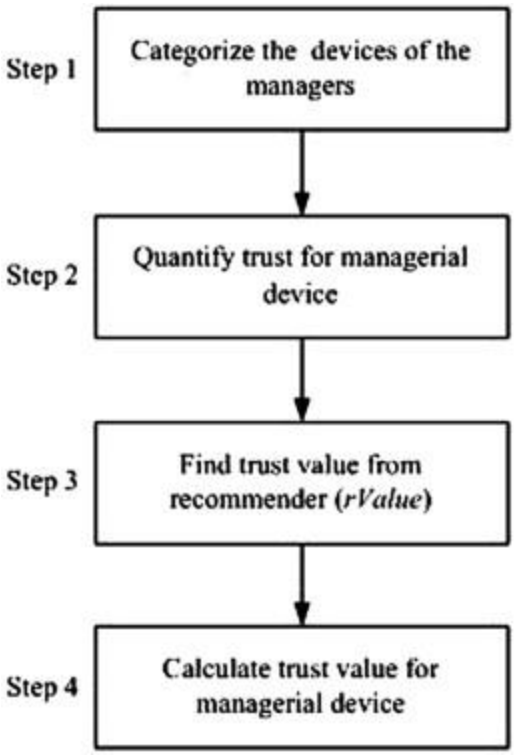


Fig. 2. Steps for the trust management.

4.1. Device categorization

Under the presented trust framework, devices within a mobile network where managers have access to information systems are divided into three categories. Categories are based on the desired amount of security functionality preformed by each device of the manager, available resources, and manager preferences and responsibilities. The three categories of devices, in order of provided security services, are top level, mid level, and low level. A brief summary of the categories is provided below.

4.1.1. Top level managerial devices

Top level managerial devices are configured to provide the maximum amount of security services since top level managers need maximum access. These devices directly monitor network traffic, analyze and make network wide recommendations. Additionally, top level managerial devices can perform security services for the neighboring mid or low level managerial devices. In this scenario, recommendations from a top level managerial device would be assigned higher (or total) authority over the recommendations from the other mid or low level managerial devices.

4.1.2. Mid level managerial devices

Mid level managerial devices perform security functionality only for themselves. They have the ability to monitor proximal network traffic, analyze for suspected attacks, and make recommendations to the neighboring devices. However, these devices do not analyze network traffic for other devices like top level managers and do not require a top level managerial device to analyze data. In terms of security, mid level managerial devices operate autonomously from the neighboring devices.

4.1.3. Low level managerial devices

Devices configured as dependant perform the most primitive security functionalities since low level managers do not have that much of authority. Low level managerial devices may perform minimal logging of network activity for further analysis. Additionally, these devices accept recommendations from the neighboring managerial devices. However, these low level managerial devices do not perform any direct monitoring or

analysis of network traffic. Most frequently, devices in this category are limited in computing power and battery life. To prioritize other processes, security functionalities are delegated to one or more higher level neighboring managerial devices. An ordinary worker can be treated as a low level manager depending on company policy.

4.2. Quantifying trust for managerial devices

Before proceeding into the calculation of trust for a managerial device, we again define the word "trust" in the context of this work, since it is used in various contexts in the literature (Jøsang et al., 2005, Rasmusson and Jansson, 1996, McKnight and Chervany, 1996). In this work, trust is the likelihood that a managerial device will not use network resources for malicious purposes. Similar to the trust model of Abdul-Rahman and Hailes (1997), we also consider trust as subjective and conditionally transitive (Wolfe et al., 2006). It means that the trust value device *A* recommends for device *B* may not be the same as the trust value *B* provides for *A*. Further, a device may recommend a trust value for another device. However, this recommendation must be taken along with the trust placed in its recommender. Once a trust value has been calculated, it is the responsibility of the application of the managerial device to decide what action to take.

In order for any model to work effectively, it requires some criterions for a managerial device to decisively allow, disallow, or limit communication with a neighboring device or information infrastructure. This is implemented by assigning a numeric value from one device to another. This numeric value represents how much a device "trusts" another device. It ranges from −1 to 1. A completely trusted device is provided a value of 1, and conversely, a distrusted device has a value of −1. The calculation of this value relies on direct monitoring, when available, and the recommendations of the neighboring devices. Additionally, the value of a recommendation includes the trust in its recommender.

4.3. Trust value of recommenders

A recommendation taken at face value is a somewhat naive approach. Even in mobile environments, a device should place more faith in the recommendations of the trusted managerial devices as opposed to the relatively new managerial devices or known malicious devices. In any event, the recommendation level should be assessed in combination with the amount of trust placed in its recommender.

The framework calculates a weighted value of a recommendation based upon the trust placed in the recommender by the device. The recommendation value for a device represents the weight of a recommendation from that particular device. This level is initially set in each managerial device for later manual and dynamic configuration based upon a history of established legitimate communication. The recommendation value (*rValue*) for a device is a numeric value between 0 to 1. A device with a recommendation value of 1 is highly trusted and its recommendation is the most influential. Conversely, a device with a recommendation value of 0 is disregarded.

4.3.1. Guidelines for setting rValue

A configurable *rValue* provides an opportunity for a manager to customize a network with prior knowledge. By weighing the recommendations, the manager makes the network more resilient against attacks from malicious devices. Moreover, this recommendation scheme allows for devices incapable of direct monitoring. However, the ability to weigh recommendations brings about the possibility of a highly trusted malicious device. In this event, a malicious device would be more empowered than in a traditional trust scheme. This possibility requires that a *rValue* should be elevated carefully. In addition to the knowledge regarding network environments, a manager should also take into account the physical security of the device.

4.4. Calculating trust for managerial device

The calculation of a managerial device's trust value is a combination of direct monitoring, recommended trust values, and the *rValues* of their recommenders. In the event of a direct evidence of any malicious use, a device's

trust value should be determined solely by the device itself. The significance of this evidence is left as a customizable property.

A more complicated problem occurs when a managerial device has no previous knowledge about the malicious behavior of a device. In this event, the calculation of trust comes from a combination of the recommendations from all the knowledgeable neighbors. A managerial device sends out a request for the recommendations for a device and calculates trust based upon the returned results. The trust is calculated using the following formula:

$$T_A(B) = \frac{(L_A(B) \cdot \mathrm{DF}) + \left[\mathrm{RF} \cdot \frac{\sum\left(T_{Di}(B) \cdot R_A(Di)\right)}{\sum\left(R_A(Di)\right)}\right]}{\mathrm{DF} + \mathrm{RF}}$$

$T_{Di}(B)$ trust value device *Di* places on device *B*

$L_A(B)$ monitoring value *A* places on *B*

$R_A(Di)$ recommendation value device *A* places on a recommendation from device *Di*

RF recommendation factor (significance)

DF direct factor (significance of direct evidence).

4.4.1. Trust calculation mechanism for a device

The calculation mechanism becomes effective when a new managerial device enters the network or at the time set for the next calculation of trust. The flowchart of Fig. 3 depicts the calculation of trust. Initially, the device checks direct monitoring data and queries remote machines for recommendations. The device waits for a specified time for responses to come back. Upon receiving a recommendation request, a neighboring device or an infrastructure checks its direct monitoring data. If data is found, the machine responds with that value, otherwise, it does not respond. The results of the recommendations are calculated using the formula presented earlier. If direct monitoring data is found, this information is used in the trust calculation. The importance of directly available data and its effect on the trust calculation should be higher, however, it is left to the administrator as a customizable option.

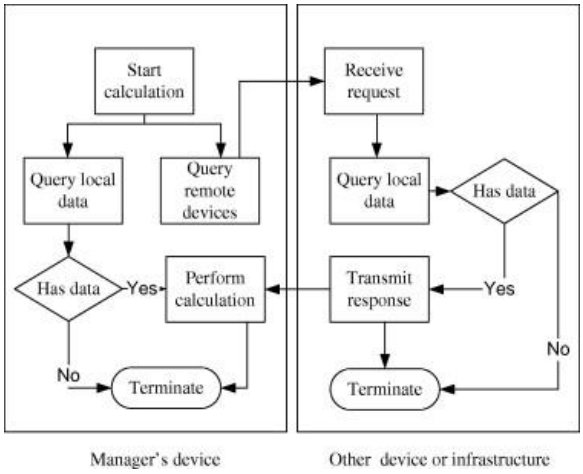


Fig. 3. Calculation of trust on a manager's device.

4.4.2. Interval of trust calculation

In order for a new managerial device to communicate, surrounding devices must perform a trust calculation on the device prior to the communication. Obviously, trust values should be recalculated periodically to ensure an

accurate reading. Performing this calculation too frequently can lead to undesirable network overhead. Conversely, performing this calculation too seldom may generate inaccurate trust values and allow malicious devices longer access to the network. The frequency of the calculation should be configured based upon the corresponding network conditions.

5. Implementation and example applications

A prototype has been implemented based on the proposed trust framework. The prototype implementation has used WINCE as the operating system, a Dell Axim X50v as PDA hardware platform, VC#.Net Compact Framework as programming language, mobile ad hoc mode of IEEE 802.11b as underlying wireless protocol, and SQLCE for database support.

The aim of this work is to provide a flexible framework allowing managerial devices to customize the trust scheme to their needs. The current implementation allows a number of configurable options. Table 1 provides the complete summary of the options by providing a description of their effect on the trust scheme and the corresponding default values.

Table 1. Customizable settings and default values

Property	**Description**	**Possible values**	**Initial value**
Device categorization (DC)	The category of security functionality performed.	Managerial, independent, dependent.	Independent
Recommendation trust value (rValue)	The trust value a device places in the recommendation from another device.	0 (insignificant) to 1 (dominating).	0.5
Established trust update (ETU)	The ability to increase trust relationships due to established legitimate communication.	Enabled, disabled.	Disabled
Trust upper limit (TUL)	The upper trust value a device can reach by established legitimate communication.	Any numeric value from −1 to 1.	0.5
Hierarchical mode (HM)	A dependent device establishes a master/slave relationship with an independent or a managerial device.	Enabled, disabled.	Disabled
Trust recalculation time period (TRTP)	The amount of time between calculating trust of neighboring devices.	Any possible time value.	60 s
Direct factor (DF)	The significance of direct evidence in trust calculation.	Any numeric value from 0 to 1.	0.9
Recommendation factor (RF)	The significance of recommendation in trust calculation.	Any numeric value from 0 to 1.	0.1
Initial trust value (ITV)	The trust value a device is assigned in the absence of recommendations or direct monitoring.	Any numeric value from −1 to 1.	0.5

We have tested the prototype against a number of adversary networks models that include a single malicious device, several malicious devices, and environments where legitimate devices are outnumbered by malicious devices. The following subsection describes an example attack scenario with numerous malicious devices. The framework has also been applied for a practical healthcare application as described in Section 5.2.

5.1. Numerous malicious devices

Network scenarios involving a large number of malicious devices create problems in the framework, where the trust calculation relies heavily upon recommendations. Any trust framework must have some mechanisms for dealing with false recommendations. For instance, three malicious devices (*M*1, *M*2, and *M*3) are deployed in a geographical area in an effort to create a coordinated attack against the existing devices *D*1, *D*2, and *D*3 (see Fig. 4). When device *D*1 is asked to recalculate the trust value of its neighbors, the following events transpire.

(a) For device *D*2, *D*1 checks its direct monitoring for direct evidence against *D*2.
(b) Finding no direct evidence, *D*1 requests recommendations pertaining to *D*2 from the neighboring devices.
(c) *D*3 responds favorably to *D*1's request.
(d) *M*1, *M*2, and *M*3 respond negatively and report malicious recommendations in an attempt to halt the communication between *D*1 and *D*2.
(e) After receiving the responses, *D*1 concludes that *D*2 cannot be trusted and limits or halts the communication.

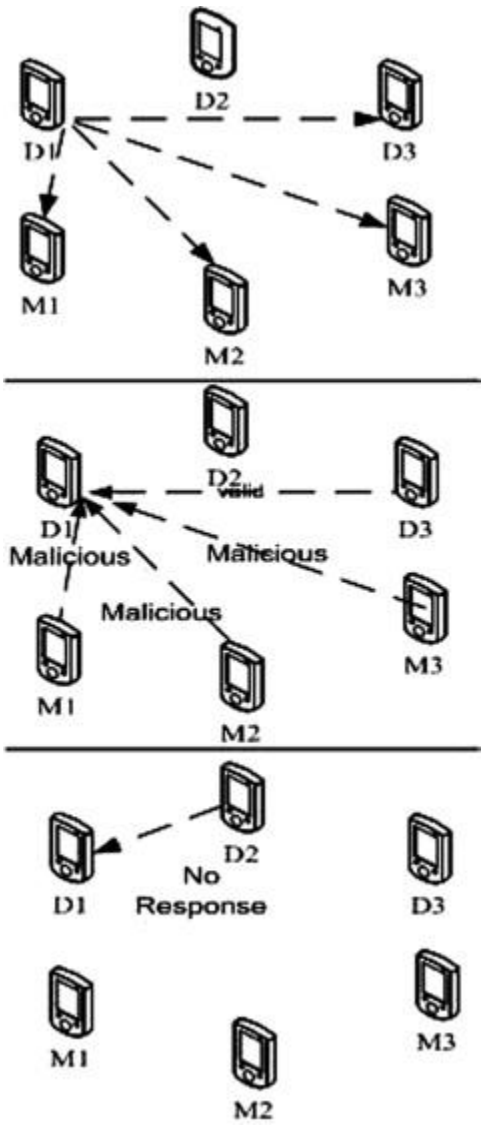


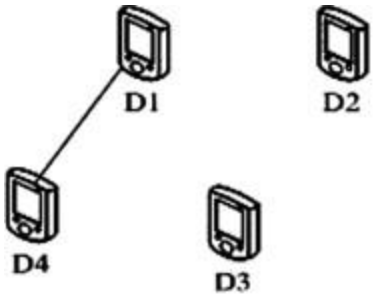
Fig. 4. Example of a large adversary model.

In this example, the three malicious devices have effectively used the trust mechanism as a weapon against the desired purpose of the network. A similar result could happen between *D*1 and *D*3, and *D*2 and*D*3 effectively halting all legitimate communications. The framework could resolve this problem or at least make the network more robust against this type of attack. Prior to the deployment, the administrator can elevate the recommendations of *D*1, *D*2, and *D*3 amongst each other. Other devices are provided the default recommendation value of the network. False recommendations from the other devices are taken at lower value than the recommendations from *D*1, *D*2, and *D*3. Depending upon the level of elevation, the framework could prevent this type of attack from an adversary model containing more malicious devices than the actually deployed devices.

5.2. A healthcare application

5.2.1. How to develop the trust framework

We have implemented a healthcare application (Sharmin et al., 2006) based on the proposed trust framework. We describe the implemented application by following the steps of the framework (see Fig. 2) for a particular scenario of a medical center. Suppose, Ms. Becky has been admitted in the gynecology department of a medical center for some complex problems, the exact etiology of which is still unclear. She is under the care of Dr. Fin, who has prescribed medication. Mary and Carla are on duty nurses, looking after Ms. Becky and other patients and managing the patient care system. However, the patient wants to consult with other physicians about her condition. In such a situation, rather than calling up various physicians from different sub-specialties, a handheld device can be used to broadcast the patient information to the PDAs of a set of selected physicians and nurses. This is more convenient, and it will save time significantly. Therefore, let us assume that the device *D*1 belongs to Ms. Becky, *D*2 belongs to Mary, *D*3 belongs to Dr. Fin, and *D*4 belongs to Carla (see Fig. 5). Suppose, Ms. Becky asks Carla about a medicine. Carla wants to look up Ms. Becky's information and contact Dr. Fin as Carla is managing the health care of Ms. Becky. Therefore, Ms. Becky's device needs to use our framework to calculate the trust so that she can ask Carla's device.



Trust table for D1

	Trust value	*rValue*
D2		0.8
D3		0.3
D4	0.14	-

Trust table for D2

	Trust value	*rValue*
D2		
D3		
D4	0.6	-

Trust table for D3

	Trust value	*rValue*
D2		
D3		
D4	-0.6	-

Fig. 5. Trust values in the health care application.

If we follow the steps of Fig. 2, we can divide the devices used in this example into two categories in Step 1: Dr. Fin and Ms. Becky's devices are top level managers and the other devices are mid level managers according to the framework. In Step 2, each trust value is quantified between −1 and +1. It is calculated based on the recommendations from the neighboring devices and available direct monitoring data. The *rValue*s obtained in Step 3 for *D*2 (Dr. Fin) and *D*3 (Mary) are 0.8 and 0.3 respectively, since Dr. Fin is a doctor and Mary is a nurse. The values are shown in the trust tables of Fig. 5.

In Step 4, we calculate the trust value for Carla (*D*4). *D*1 (Ms. Becky) having no trust value for *D*4 (Carla) polls its immediate neighbors and hears back from *D*2 (Dr. Fin) and *D*3 (Mary). *D*2 responds with 0.6 and *D*3 responds with −0.6 (see Fig. 5). The calculation of *D*1's trust in *D*4 occurs as follows:

$$\begin{aligned} T_{D1}(D4) = ((&L_{D1}(D4) * \mathrm{DF}) + (\mathrm{RF} * (T_{D2}(D4) * R_{D1}(D2) + T_{D3}(D4) \\ &* R_{D1}(D3)))/(R_{D1}(D2) + R_{D1}(D3))/(DF + RF)) \\ &= ((0 * 1) + (1 * (0.6 * 0.8 + -0.6 * 0.3))/(0.8 + 0.3)/(1 + 1)) = 0.14 \end{aligned}$$

5.2.2. How to use the trust framework

The above trust values obtained based on the trust framework are used in the healthcare application running on the PDAs of Ms. Becky, Carla, Mary, and Dr. Fin. The screen shots of the healthcare application are shown in Fig. 6. Using Step 4's calculation, Ms. Becky is able to communicate to Carla. The health care nurse Carla (as a manager) can view patient and hospital information based on our trust framework.

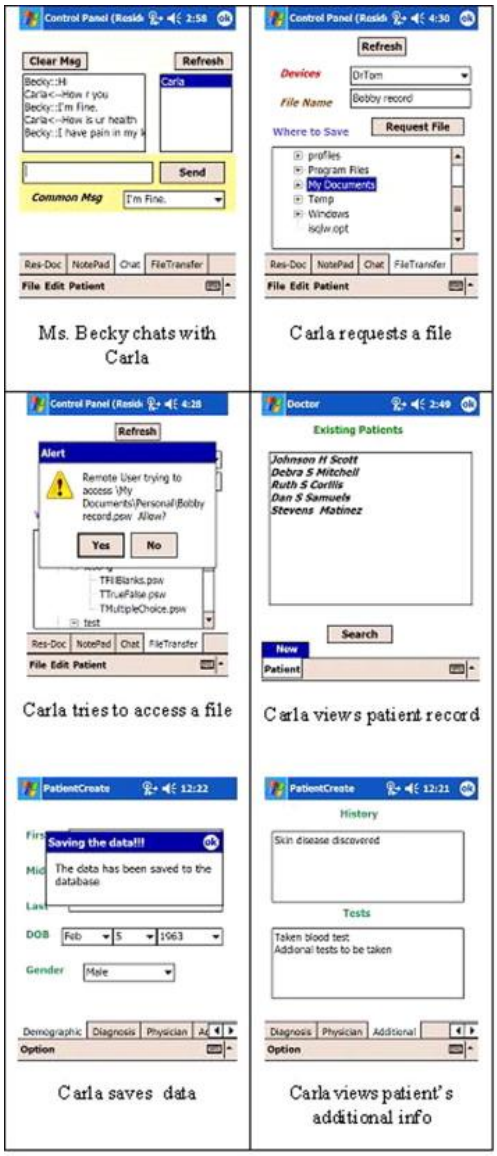


Fig. 6. Some screenshots of the healthcare application which uses the trust framework.

5.2.3. Efficiency of the trust framework

To show the effectiveness of the trust framework with respect to handling of power consumption, we have measured battery power of the managerial handheld devices. Power consumption is one of the most important performance metrics of handheld applications and services. In Fig. 7, PDA1, PDA2, and PDA3 are handheld devices (*D*1, *D*2, and *D*3) and belong to Ms. Becky, Mary and Dr. Fin respectively. Each PDA runs the framework and the healthcare application. Fig. 7 shows two cases of the remaining battery power for each PDA (Sharmin et al., 2006): the idle case (when the PDA is ON but inactive) and the active case (when the application and framework are running on the PDA). We observe that the rate of change in battery power for each device with and without running the application and the framework are almost the same. Hence, the healthcare application did not consume that much of battery power. The trust framework seems to be very power-conservative, and each device consumes the minimal amount of battery power possible.

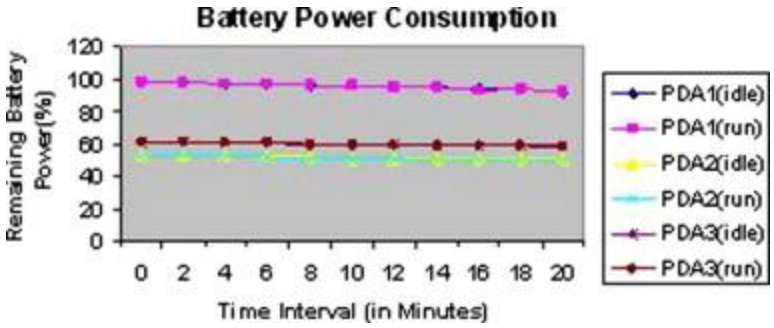


Fig. 7. Power consumption on three handheld devices before and after running the health care application and the trust framework.

6. Conclusions and future work

This paper presents a software-based flexible trust framework that is built for the efficient management of trust among the managerial handheld devices in a distributed industrial management setting. The paper also provides a survey on security engineering methodologies and trust models. The framework improves upon the existing methodologies of trust management with respect to a number of characteristics required for a managerial device identified in this research. The paper describes the management of trust in the framework in detail by categorizing devices, calculating trust, and facilitating trust-related communications. Each of the steps is explained using appropriate examples, while the framework is implemented on PDAs. A health care application is presented which was built based on the implemented trust framework. The experimental results indicate that the framework is efficient with respect to power saving – a very important issue for tiny handheld devices.

By enabling the administrator to customize the solution, the trust mechanism can be tailored to the environment of the deployment. The customizable options are initially set to the values ideal for a managerial device. Overall, the framework provides a general approach for making wireless communications more dependable. It also has the potential to provide a common platform for security engineers, software engineers, and business analysts in building more trustworthy and profitable distributed management information systems (Hussein and Zulkernine, 2007).

As of now, we have calculated and evaluated trust values only for small networks. Our immediate focus is to investigate the effectiveness and network overhead of the framework for various scales of enterprises from mid-size to large-size. More comprehensive evaluation results of various enterprises may indicate the scalability of the framework. The framework can be extended with the functionality that will allow security analysis to be outsourced to other more powerful and physically secure devices. We envisage that the proposed framework may be useful to integrate risk and security via trust and to automatically detect intrusions to industrial management systems (Kannadiga et al., 2005). We also aim to incorporate in the framework another very interrelated issue called “privacy”, which plays a major role in accessing various industrial information systems.

References

Abdul-Rahman and Hailes, 1997. Abdul-Rahman, A., Hailes, S., 1997. A distributed trust model. In: *Proceedings of the New Security Paradigms Workshop*, pp. 48–60.

Gupta et al., 2001. Gupta, S., Lee, W., Purukayastha, A., Srimani, P. (Eds.), 2001. *IEEE Personal Communications*. Special Issue on Pervasive Computing 8(4), 8–9.

Hussein and Zulkernine, 2007. M. Hussein, M. Zulkernine. **Intrusion detection aware component-based systems: a specification-based approach.** *Journal of Systems and Software*, 80 (2007), pp. 700-710

Jøsang et al., 2005. Jøsang, A., Keser, C., Dimitrakos, T., 2005. Can We Manage Trust? In: *Proc. of the Third International Conference on Trust Management (iTrust)*, Rocquencourt, France.

Kannadiga et al., 2005. Kannadiga, P., Zulkernine, M., Ahamed, S., 2005. Towards an intrusion detection system for pervasive computing environments. In: *Proc. of the International Conference on Information Technology*, pp. 277–282.

Kim and Lee, 2005. S. Kim, H.J. Lee. **Cost-benefit analysis of security investments: methodology and case study.** *Lecture Notes in Computer Science*, vol. 3482, Springer-Verlag (2005). pp. 1239–1248

Kim and Lee, 2006. S. Kim, H.J. Lee. **Security engineering methodology based on problem solving theory.** *Lecture Notes in Computer Science*, vol. 3983, Springer-Verlag (2006). pp. 639–648

Kim and Leem, 2004. S. Kim, C.L. Leem. **An information engineering methodology for the security strategy planning.** *Lecture Notes in Computer Science*, vol. 3043, Springer-Verlag (2004). pp. 597–607

Kim et al., 2004. S. Kim, H.J. Lee, C.L. Leem. **Architecture of authentication mechanism for emerging T-commerce environments.** *Lecture Notes in Computer Science*, vol. 3331, Springer-Verlag (2004). pp. 540–547

Leem and Kim, 2002. C.L. Leem, S. Kim. **Introduction to an integrated methodology for development and implementation of enterprise information systems.** *Journal of Systems and Software*, 60 (2002), pp. 249-261

Leem et al., 2005. C.L. Leem, S. Kim, H.J. Lee. **Assessment methodology on maturity level of ISMS.** *Lecture Notes in Computer Science*, vol. 3683, Springer-Verlag (2005). pp. 609–615

Lin and Varadharajan, 2006. Lin, C., Varadharajan, V., 2006. Trust based risk management framework for distributed system security – a new approach. In: *Proc. of the First International Conference on Availability, Reliability and Security*, Vienna, Austria, pp. 6–13.

Luo et al., 2002. Luo, H., Zerfos, P., Kong, J., Lu, S., Zhang, L., 2002. Self-securing ad hoc wireless networks. In: *Proc. of the Seventh International Symposium on Computers and Communications*, pp. 567–574.

McKnight and Chervany, 1996. McKnight, D.H., Chervany, N.L., 1996. The Meanings of Trust, Technical Report, MISRC Working Paper Series 96-04, Information Systems Research Centre, University of Minnesota.

Pirzada and McDonald, 2004. Pirzada, A., McDonald, C., 2004. Establishing trust in pure ad-hoc networks. In: *Proc. of the 27th Australian Conference on Computer Science*, 26, 47–54.

Rasmusson and Jansson, 1996. L. Rasmusson, S. Jansson. **Simulated social control for secure internet commerce.** *New Security Paradigms Workshop*, ACM Press, California, USA (1996), pp. 18-25

Sharmin et al., 2006. Sharmin, M., Ahmed, S., Ahamed, S., Haque, M., Khan, A., 2006. Healthcare aide: towards a virtual assistant for doctors, patients, nurses and resident doctors using pervasive middleware. In: *Proc. of the 1st Workshop on Ubiquitous and Pervasive Health Care*, pp. 490–495.

Stajano and Anderson, 1999. Stajano, F., Anderson, R., 1999. The resurrecting duckling security issues for ad-hoc wireless networking. In: *Proc. of the 7th International Workshop on Security Protocols*, pp. 172–182.

Sun and Song, 2004. Sun, H., Song, J., 2004. Strategy proof trust management in wireless ad hoc network. In: *Proc. of the IEEE Canadian Conference on Computer and Electrical Engineering*, Ontario, Canada, May 2004.

Weise, 2006. Weise, J., 2006. Public Key Infrastructure Overview, Sun BluePrints Online, http://www.sun.com/blueprints/0801/publickey.pdf (accessed in June 2006).

Weiser, 1991. M. Weiser. **The computer for the twenty-first century.** *Scientific American*, 265 (1991), pp. 66-75

Wolfe et al., 2006. Wolfe, S., Ahamed, S., Zulkernine, M., 2006. A trust framework for pervasive computing environments. In: *Proc. of the 4th ACS/IEEE International Conference on Computer Systems and Applications*, pp. 312–319.

Yi and Kravets, 2002. Yi, S., Kravets, R., 2002. Key management for heterogeneous ad-hoc wireless networks. In: *Proc. of the 10th IEEE International Conference on Network Protocols*, pp. 202–203.

Zhu et al., 2005. F. Zhu, M. Mutka, N. Lionel. **Facilitating secure ad-hoc service discovery in public environments.** *Journal of Systems and Software*, 76 (2005), pp. 45-54