A software-based trust framework for distributed industrial management systems

Sheikh I. Ahamed a,*, Mohammad Zulkernine b, Steve Wolfe a

a Marquette University, WI 53233, USA
b Queen’s University, Kingston, Ont., Canada K7L 3N6

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 managers 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 (Rasmussen 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.

0164-1212/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2007.01.043
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 policy 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 infrastructure 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. 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 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 the 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 rrecommendationValue 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.

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 infra-
structure (high end servers and proxies).

A trust model-based qualitative risk management
approach for distributed system security is proposed in
(Lin and Varadarajan, 2006). The model uses trust assess-
ment 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 environ-
ments 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 consider-
ably 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 mak-
ing a judgment based upon its observations and recommenda-
tions 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 recommend-
dations. Within their model, each device maintains a dis-
crete trust value for the neighboring devices. The proposed
values range from –1 (completely untrustwor-
thy) to 4 (completely trusted). It requires trust to be transi-
tive 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 recom-
manded trust values and the trust values of their recom-
manders. 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 other-
wise. This research proposes a distributed authentication
mechanism (A1, see Section 2.1) where authenticated
devices possess 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 recommenda-
tions from the neighboring devices, a device offers its par-
tial 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 combina-
tion 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 respec-
tively. 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 dif-
fferences between “managed” and “pure” ad hoc environ-
ments. Their solution is designed specifically for “pure”
ad hoc environments, requiring no pre-configuration
(A2). The calculation for trust contains a mechanism for
devailing the influence of malicious devices and recalculat-
ing the perceived trust (A4,A6). It lacks the support for dif-
ferent 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 calcu-
lates 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 his-
tory of the device and the recommendation of other
devices. Their trust framework requires that each device
broadcasts its trustworthiness to the network upon enter-
ing. This feature creates a vulnerability to the malicious
devices that misrepresent their trustworthiness to the net-
work. Hence, it does not fit for managerial devices. Addi-
tionally, their proposal assumes that malicious devices
cannot work together.
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, re-
commendations from a top level managerial device would
be assigned higher (or total) authority over the recommen-
dations from the other mid or low level managerial devices.

4.1.2. Mid level managerial devices
Mid level managerial devices perform security function-
ality 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 autonom-
ously 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 rec-
commendations from the neighboring managerial devices.
However, these low level managerial devices do not per-
form 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 pro-
cesses, 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 Jans-
son, 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 transi-
tive (Wolfe et al., 2006). It means that the trust value
A recommends for device B may not be the same as the
trust value B provides for A. Further, a device may recom-
mand a trust value for another device. However, this rec-
ommendation 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 criteria 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 recom-
endation based upon the trust placed in the recommender
by the device. The recommendation value for a device rep-
resents the weight of a recommendation from that particu-
lar 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 from 0 to 1. A device with a recommenda-
tion value of 1 is highly trusted and its recommendation

Please cite this article in press as: Ahamed, S.I. et al., A software-based trust framework for distributed ..., J. Syst. Software (2007),
doi:10.1016/j.jss.2007.01.043
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 weighting 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 weight 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:

\[ T_A(B) = \frac{(L_A(B) \cdot DF) + \sum RF \cdot \frac{(T_D(B) \cdot R_A(Di))}{\sum R_A(Di)}}{DF + RF} \]

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.

4.5. 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.
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.

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 pro-

![Fig. 4. Example of a large adversary model.](image-url)
562 provided the default recommendation value of the network.
563 False recommendations from the other devices are taken
564 at lower value than the recommendations from D1, D2,
565 and D3. Depending upon the level of elevation, the frame-
566 work could prevent this type of attack from an adversary
567 model containing more malicious devices than the actually
568 deployed devices.

569 5.2. A healthcare application

570 5.2.1. How to develop the trust framework

571 We have implemented a healthcare application (Shar-
572 min et al., 2006) based on the proposed trust framework.
573 We describe the implemented application by following
574 the steps of the framework (see Fig. 2) for a particular sce-
575 nario of a medical center. Suppose, Ms. Becky has been
576 admitted in the gynecology department of a medical center
577 for some complex problems, the exact etiology of which is
578 still unclear. She is under the care of Dr. Fin, who has pre-
579 scribed medication. Mary and Carla are on duty nurses,
580 looking after Ms. Becky and other patients and managing
581 the patient care system. However, the patient wants to con-
582 suit with other physicians about her condition. In such a
583 situation, rather than calling up various physicians from
584 different sub-specialties, a handheld device can be used to
585 broadcast the patient information to the PDAs of a set
586 of selected physicians and nurses. This is more convenient,
587 and it will save time significantly. Therefore, let us assume
588 that the device D1 belongs to Ms. Becky, D2 belongs to
589 Mary, D3 belongs to Dr. Fin, and D4 belongs to Carla
589 (see Fig. 5). Suppose, Ms. Becky asks Carla about a med-
590 icine. Carla wants to look up Ms. Becky’s information and
591 contact Dr. Fin as Carla is managing the health care of Ms.
592 Becky. Therefore, Ms. Becky’s device needs to use our
593 framework to calculate the trust so that she can ask Carla’s
594 device.

595 If we follow the steps of Fig. 2, we can divide the devices
596 used in this example into two categories in Step 1: Dr. Fin
597 and Ms. Becky’s devices are top level managers and the
598 other devices are mid level managers according to the
599 framework. In Step 2, each trust value is quantified
600 between −1 and +1. It is calculated based on the recom-
601 mendations from the neighboring devices and available
602 direct monitoring data. The rValues obtained in Step 3
603 for D2 (Dr. Fin) and D3 (Mary) are 0.8 and 0.3 respec-
604 tively, since Dr. Fin is a doctor and Mary is a nurse. The
605 values are shown in the trust tables of Fig. 5.

606 In Step 4, we calculate the trust value for Carla (D4). D1
607 (Ms. Becky) having no trust value for D4 (Carla) polls its
608 immediate neighbors and hears back from D2 (Dr. Fin)
609 and D3 (Mary). D2 responds with 0.6 and D3 responds
610 with −0.6 (see Fig. 5). The calculation of D1’s trust in
611 D4 occurs as follows:

612 \[ T_{D1}(D4) = (L_{D1}(D4) \times DF) + (RF \times (T_{D2}(D4) \times R_{D1}(D2)
613 + T_{D3}(D4) \times R_{D1}(D3))/(R_{D1}(D2)
614 + R_{D1}(D3))/DF + RF) \]

615 \[ = (0 \times 1) + (1 \times (0.6 \times 0.8 + -0.6 \times 0.3)/
616 \times (0.8 + 0.3)/(1 + 1)) \]

617 \[ = 0.14 \]

5.2.2. How to use the trust framework

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

5.2.3. Efficiency of the trust framework

626 To show the effectiveness of the trust framework with
627 respect to handling of power consumption, we have mea-
628 sured battery power of the managerial handheld devices.
629 Power consumption is one of the most important perform-
630 ance metrics of handheld applications and services. In
631 Fig. 7, PDA1, PDA2, and PDA3 are handheld devices
632 (D1, D2, and D3) and belong to Ms. Becky, Mary and
633 Dr. Fin respectively. Each PDA runs the framework and
634 the healthcare application. Fig. 7 shows two cases of the
635 remaining battery power for each PDA (Sharmin et al.,
636 2006): the idle case (when the PDA is ON but inactive)
uncorrected proof

636 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.

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, in press).

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”,

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

Fig. 7. Power consumption on three handheld devices before and after running the health care application and the trust framework.
which plays a major role in accessing various industrial information systems.

References


Sheikh I. Ahamed is a faculty member in the Department of Mathematics, Statistics, and Computer Science at Marquette University, USA. He is also the director of the Ubicomp research lab at Marquette University. He is a member of the IEEE, ACM, and the IEEE Computer Society. He received his B.Sc. in computer science and engineering from the Bangladesh University of Engineering and Technology, Bangladesh in 1995. He completed his Ph.D. in Computer Science from Arizona State University, USA in 2003. His research interests are in ad hoc networks, middleware for ubiquitous/pervasive computing, sensor networks, pervasive healthcare and RFID. He has also developed several pervasive computing systems using PDAs, Cell phones, RFID, and Motes on 802.11, Bluetooth and Zigbee wireless networks. He serves regularly on international conference program committees in software engineering and pervasive computing such as COMPASS 04, COMPASS 05, COMPASS 06, SAC 07 and ICP 07. He is the Program Co-Chair of International Workshop on Security, Privacy, and Trust for Pervasive Computing Applications (SPTPA 2006). He can be contacted at iq@mcs.queensu.edu; http://www.mcs.queensu.edu/~iq.

Mohammad Zulkernine is a faculty member of the School of Computing of Queen’s University, Canada, where he is leading the Queen’s Reliable Software Technology (QRST) research group. He received his B.Sc. degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology in 1993. He received an M.Eng. degree in Computer Science and Systems Engineering from Muroran Institute of Technology, Japan in 1998. He received his Ph.D. degree from the Department of Electrical and Computer Engineering of the University of Waterloo, Canada in 2003, where he belonged to the university’s Bell Canada Software Reliability Laboratory. His current research focuses on software engineering (software reliability and security) and network security (intrusion detection). His research has been funded by a number of provincial and federal funding agencies and industry: Bell Canada, Canada Foundation for Innovation (CFI), Ontario Ministry of Economic Development and Trade (MEDT), National Science and Engineering Research Council of Canada (NSERC), and Mathematics of Information Technology and Complexity Systems (MITACS), Canada. He has been very active in the international research community by organizing conferences and workshops. He is also cross-appointed in the Department of Electrical and Computer Engineering of Queen’s University, and he is a licensed professional engineer of the province of Ontario, Canada. He is a member of the IEEE, ACM, and the IEEE Computer Society. He can be reached at muziker@cs.queensu.ca; http://www.cs.queensu.ca/~muziker.

Steve Wolfe is a graduate student and member of Ubicomp research lab in the department of Mathematics, Statistics, and Computer Science at Marquette University, USA. He has received his B.Sc. degree in Computer Science. His field of interest encompasses pervasive security, trust model in pervasive computing and wireless sensor network. His contact address is swolfe@mcs.queensu.edu.