Toward a Trust Management Model for a Configurable Body Sensor Platform

Vinh T. Bui
Dept. of Mathematics and Computer Science
TU Eindhoven
t.v.bui@tue.nl

Johan J. Lukkien
Dept. of Mathematics and Computer Science
TU Eindhoven
j.j.lukkien@tue.nl

Richard Verhoeven
Dept. of Mathematics and Computer Science
TU Eindhoven
p.h.f.m.verhoeven@tue.nl

ABSTRACT

Body Sensor Networks (BSNs) are used for diverse applications ranging from monitoring for medical purposes, sport coaching to computer gaming. This leads to viewing the BSN as an application platform and requires the capability to dynamically extend and configure BSN applications, including reprogramming the sensor nodes of the BSN. We propose a trust management model along with a component-based architecture to maintain the BSN as trustworthy platform under such changes in applications. With this model the trustworthiness can be autonomically evaluated and monitored at component, application, and system levels with respect to properties of security, dependability, and performance. The usage of the model is shown with an example of an ECG monitoring application.

1. INTRODUCTION

Body Sensor Networks (BSNs) consist of wearable or even implantable electronic devices known as body sensors which record human body functions including physiological, emotional, and spatial aspects. BSNs can be used for a wide range of applications from monitoring for medical purposes and sports coaching to computer gaming [10]. While various types of BSNs have been developed most of the proposed BSNs are special purpose platforms on which the applications are installed during deployment and never updated.

In the VITRUVIUS project[1], we aim to develop a body sensor *platform*, on which several BSN applications are installed dynamically, which is self-contained and which can connect to backend systems, e.g., of a hospital. In a sense it is similar to modern smart phones or other smart devices that admit installing applications, albeit that a BSN is a distributed system. The BSN consists of sensors together with a more powerful device, called the *body hub*, that is capable of storing data and running application-specific components (see Figure 1). In our system, applications, or parts of applications are uploaded as packages containing four parts: modules for the sensors, signal processing modules, database tables and behavioral instructions for the body hub (including rules for acting upon the measured data). Installation of such a package leads to the configuration of the BSN towards a specific service.

Uploaded components may intentionally or unintentionally affect system properties like dependability and security: new components might leak information, affect the overall resource distribution amongst components, or jeopardize the correct execution of the device. There also exists other potential threats to the system security (e.g., wrong data from sensors, spoofing of sensors, and eavesdropping), which were described and analyzed in [2]. The system requires, therefore, a mechanism to assert and manage the trustworthiness of the components and also the system (BSN) in which the components are running.

We propose a trust management model and method which is able to evaluate and monitor the trustworthiness at component, application, and system levels. From these evaluation results and from a given trust policy, subsequent actions are decided and performed in order to preserve system trustworthiness with respect to quality properties like dependability, security, and performance [3].

2. RELATED WORK

Trust plays an important role in many applications including content selection for Web documents, medical systems, mobile code, and electronic commerce [4, 5]. Grandison *et al.* [6] have presented trust specification and management, as well as several solutions for Internet applications.

In WSNs, trust management is specifically useful to manage the trustworthiness of interactions among network entities. A security framework with trust management, i.e. establishment of a trustworthy network environment, is used to secure sensor networks [11]. A distributed trust model enabling recommendation-based trust and trust-based recommendation has been proposed, to build reasonable trust relationships among network entities. A trust model without a central trust authority in order to establish trust for WSNs has been introduced in [7]. The model combines several kinds of trust values together, including the direct trust value established between a node that initiates the cooperation and a target node that provides a service; and the indirect trust value established by a third-party node. These models, however, focus more on the trust relationship during the interaction process among the sensor nodes than the

[1]The VITRUVIUS project is supported by the Dutch Ministry of Economic Affairs under the Innovation Oriented Research Program (IOP GenCom).

BODYNETS 2011, November 07-08, Beijing, People's Republic of China

DOI 10.4108/icst.bodynets.2011.247072

trust management with respect to the system properties and behavior of the application components running on the sensor nodes.

Being different from the other models above, the Trust4All project [9] has introduced a trust management framework for component-based applications, which supervises the system's existing trustor-trustee relationships at application level and preserves the overall system level of dependability and security. This work is a conceptual starting point for our trust management model.

3. SYSTEM ARCHITECTURE

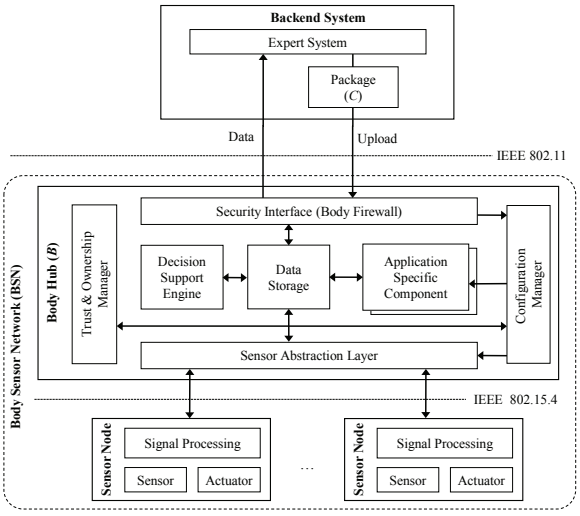

Figure 1: The VITRUVIUS system architecture. The package containing component C is uploaded and installed on the body hub for a specific task.

The VITRUVIUS system architecture is shown in Figure 1. The body hub controls the BSN and is the primary access point. The BSN may connect to the backend systems through the Internet, employing a secure connection between the *expert system* on the backend system and the *security interface* on the body hub. Through this connection, the backend system communicates with the BSN for the purposes of retrieving data and uploading the components. The security interface shields the body hub from the outside world and limits access to privacy-sensitive data according to the authorization level of the requesting party.

The backend system hosts the expert system, which is capable of generating application-specific components (either in the form of native code, interpreted code, rule set, or the configuration of the application) automatically from the level a doctor is working. The component then is included into a software package together with its certificates and quality profile, and is uploaded to the body hub.

The body hub receives data from the sensors, stores, and processes it according to the instructions from the loaded components. The body hub must be capable of autonomously responding to abnormal conditions found in the data, for example, by contacting the user or the expert system in case of an emergency. To support the upload and configuration processes in a secure and trust preserving manner, the body hub has the *security interface*, the *configuration manager*, and the *trust and ownership manager* modules. The security interface module provides means for component download and certificate verification. The configuration manager provides means for component installation and configuration. The trust and ownership manager is responsible for evaluating and monitoring the trustworthiness and ownership.

The sensor nodes extract information from the body via dedicated sensors and send it to the body hub. A program running on the sensor nodes determines the data which is sent to the body hub. Our sensor platform allows programming and configuring sensors over the air using code and data received in the uploaded package by the body hub.

4. TRUSTWORTHINESS PROBLEM

The decision which the system needs to take upon loading a package is considered. The concerns play at the level of a component, of the application which the component is part of, and of the entire system. Potential problems upon installing and executing a particular component are

- The functionality of the component or the computing accuracy is not acceptable.
- The new component may disable other components or processes running on the system by excessive use of system resources, e.g., CPU time or main memory.
- Unauthorized access or leakage of private information. For example, a gaming application may access or leak epilepsy monitoring information.
- Damage to the data or even to the system execution. For example, the new component may modify or remove data files, or already-resident executables.

In response to the need to verify trustworthiness of components or their sources, some approaches use digital certificates [6]. This type of trust is based on certification of the trustworthiness by a third party, so trust would be based on criteria relating to the set of certificates presented by the component to the system. The certificates can also play a role in addressing security concerns (e.g., component identity authentication), but they cannot address all aspects. The components may be authored by someone and deployed without the certificates, e.g., the gaming applications. Moreover, the decision about whether the component can be run is based on the current status or context of the system which the component author cannot predict.

5. TRUST MANAGEMENT MODEL

Before beginning the discussion on the trust management model, the definition of trust must be clarified. Many definitions of trust and its related concepts are established in the literature [6, 1]. There, however, is no consensus on what trust is and some definitions are very domain specific. In our system trust is defined as follows. *Trust is the degree to which a trustor has a justifiable belief that, in a given context, a trustee will live up to a given set of statements about its behavior.* In this definition, trustor and trustee can be any entity (e.g., a user, a computer, or a component); 'justifiable' refers to the ability to explain the reasoning or computation. According to this view, trust is a value (or a vector) in the range 0 through 100% determined by trustor, trustee, and context. The context includes elements that determine the trustee's behavior (e.g., available resources or competing applications) as well as information that influ-

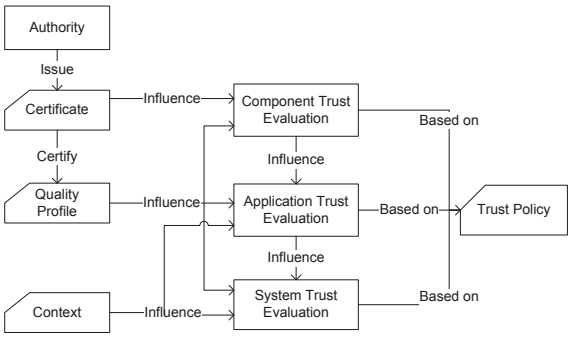

Figure 2: The factors related to the trust evaluation at component, application, and system levels.

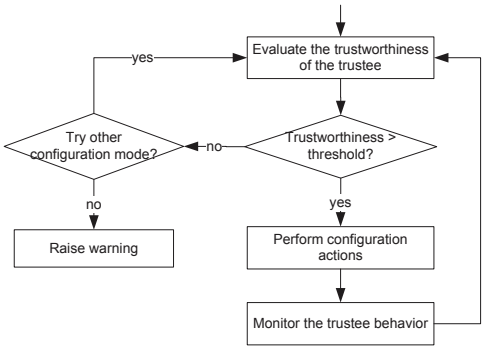

Figure 3: The trust management model with trust evaluation, monitoring, and controlling processes.

ences the trustor's judgement, e.g., history. Trust can be computed per property of interest.

For download and execution of components, the acceptance procedure of a component relies on the computation of trust values that is built up from three levels. At the component level, there must be sufficient trust that the component behaves according to its specification. The specification of a component is simplified by a vector of properties, named *quality profile*, that comprises relevant functional and extra-functional properties. Elements of this trust evaluation are *certificates* signed by third-party authorities and certifying the correctness and accuracy of the quality profile. In this way verification is delegated to the authority allowing the trustor to trust the trustee on behalf of the authority. This trust evaluation together with the component's quality profile is input to a trust evaluation of the application where the component becomes part of. This latter evaluation is compared to a threshold to examine if there is sufficient belief that the application satisfies the requirements. In this way trust is balanced with *risk*, represented by the threshold. The threshold is set depending on the application and its operation mode.

At the system level, a trust evaluation is performed for the system extended with the new component. System level concerns are evaluated here, like resource availability and system integrity, and other context-related concerns. The relevance of certain extra-functional properties of a component will depend on this context and can be captured by a weight factor, which gives the relative importance of these properties to the trustor. For example, if there is plenty of memory available on the platform it is not very relevant how accurate the memory usage specification of a component is. Figure 2 shows the factors related to trustworthiness and the trust evaluations at the three levels of component, application, and system.

5.1 Trust evaluation

In this paper, we propose and evaluate a trust value vector at the system level. The method is presented below, using the following context and terminology.

1. Component C is uploaded to body hub B. Thus, the trustor is B and the trustee is C, the current context of B is $B.S$, and the trust policy is denoted by $B.TP$;
2. QA is the set of quality attributes qa_i $(i = 1, .., n)$ of C; these quality attributes can be taken from the quality profile of C or from monitoring the behavior of C.
3. W is the set of weights w_i $(i = 1, .., n)$ of the quality attributes. In particular, $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$.
4. tv is the trust value of C considered at the system level.

Evaluation method

1. Convert values of the quality attributes to the same scale, using the proportional scoring method [8]:
   (a) Determine a range $[qa_i^{min}, qa_i^{max}]$ for each quality attribute qa_i. qa_i^{min} and qa_i^{max} can be selected based on context $B.S$ and trust policy $B.TP$;
   (b) Compute score $\widehat{qa}_i$ for each quality attribute qa_i, $\widehat{qa}_i \in [0, 100]$:

$$\widehat{qa}_i = 100 \times \frac{qa_i - qa_i^{min}}{qa_i^{max} - qa_i^{min}} \quad (1)$$

2. Compute weight w_i for each quality attribute qa_i based on system context $B.S$:

$$w_i = f_w(qa_i, B.S) \quad (2)$$

3. Compute trust value tv:

$$tv = \sum_{i=1}^{n} w_i \times \widehat{qa}_i \quad (3)$$

5.2 Trust Management

In addition to trust prediction and evaluation at the component installation time, the system further monitors the behavior of the component execution in order to keep checking the compliance of the component's actual behavior against its quality profile. The monitoring process also monitors the application and system behavior, and triggers the trust re-evaluation process if necessary, for example in case of a misbehaving component or low system performance.

We further consider to provide control mechanisms for the trust management. The component and application may have multiple configuration modes of operations and each mode has different performance and resource usage. For example, the 'fast mode' of a heart-rate detection component has lower accuracy but uses less resource than the 'accurate mode'. At a certain moment, if the trustworthiness of the component is less than the threshold, the trust management tries a different configuration mode of the component and re-evaluates its trustworthiness. In case the trustworthiness is satisfied, the new configuration mode is applied. Otherwise, a warning is raised if there is no suitable mode. Depending

Mode	ECG logging	RR interval	Privacy
ECG sensor node			
CPU	2%	30%	80%
RAM	100 B	30 B	300 B
ROM	30 B	400 B	8000 B
Network	400 B/s	10 B/s	40 B/i
Storage	n.a.	n.a.	90 KB/i
Current	10mA	1mA	6mA
Body hub			
Quality	100%	15%	80%
Storage	4 KB/s	40 B/s	120 B/i
Retention	7 days	2 days	5 days
Access	Direct	Cond.	Cond.

Table 1: Modes of operation for ECG monitoring.

on the trust policy, the system also might perform configuration actions like suspending or removing the component from the system. The trust management model with processes of trust evaluation, monitoring, and controlling are presented in Figure 3.

5.3 An example

The BSN can include an ECG sensor, such that a component can monitor a patient and report relevant data to a doctor. Although the doctor would prefer the most accurate information (the raw ECG signal at 250Hz), other concerns might forbid this (long battery life) and the patient might require other properties (privacy guarantees). The monitoring component provides multiple modes of operations to allow a trade-off between accuracy and privacy, as indicated in Table 1. In addition, the different modes can be used to adjust the monitoring according to available resources.

In the ECG logging mode, the raw ECG signal is recorded by the body hub and made accessible to the doctor. Only few resources are required on the ECG sensor node, but the network communication and body hub are used intensively. The recorded raw ECG signal causes a privacy concern.

In the RR interval mode, the ECG sensor node analyzes the signal to determine the intervals between R peaks, which are recorded and analyzed by the body hub. The body hub informs the doctor when an incident occurs and provides access to a subset of the data, which reduces the privacy concerns.

In the privacy mode the ECG sensor node performs a full analysis of the ECG signal and is able to detect relevant incidents. When an incident is detected, the raw ECG signal is stored locally and the body hub is informed. Upon request, the doctor is able to retrieve the ECG data from the sensor node. Since the body hub does not store ECG related data directly, the privacy concerns are further reduced.

Based on the system context, the trust value of a certain mode of a component will differ. When there is considerable radio interference, the trust value of the ECG logging mode will be affected. Packet loss can affect the reliable transport of the ECG data and results in retransmissions. As a result, the recorded quality might not be 100% and the derived battery life time cannot be guaranteed.

When a patient has a heart disease that generates many incidents, the privacy mode might not be able to fulfill its specification, as the available storage on the sensor node might be insufficient to provide a retention of 5 days.

When there are privacy concerns, the trust policy is configured such that the weight factor for access, retention and storage attributes is high. Depending on the heart disease, the doctor might require a different quality level of the recorded data and adjust the related weight factor.

When a life threatening condition is detected or when the doctor requests a daily raw ECG recording for some period, the system context can change such that privacy or battery concerns are no longer taken into account.

6. CONCLUSION AND FUTURE WORK

A trust management model together with an example for a configurable body sensor platform is introduced. The trust management is used to enhance system properties like dependability and security under dynamic changes in applications, e.g., installation of a new component.

Current work is on how to specify the trust evaluation and management models more precisely and on creating an experimental evaluation. In the experiment, budget-based resource monitoring and control will be integrated.

7. REFERENCES

[1] A. Abdul-Rahman and S. Hailes. A distributed trust model. In *Proc. of the 1997 workshop on New security paradigms*, pages 48–60, New York, USA, 1997. ACM.

[2] S. Amini, R. Verhoeven, J. Lukkien, and S. Chen. Toward a security model for a body sensor platform. In *Proc. 29th IEEE International Conference on Consumer Electronics*, 2011.

[3] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secur. Comput.*, 1:11–33, Jan. 2004.

[4] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis. Secure Internet programming. chapter The role of trust management in distributed systems security, pages 185–210. London, UK, 1999.

[5] J. Feigenbaum and P. Lee. Trust management and proof-carrying code in secure mobile-code applications. In *DARPA Workshop on Foundations for Secure Mobile Code*, 1997.

[6] T. Grandison and M. Sloman. A survey of trust in Internet applications. *IEEE Communications Surveys and Tutorials*, 3(4), 2000.

[7] G. Han, D. Choi, and W. Lim. A reliable approach of establishing trust for wireless sensor networks. In *Proc. IFIP International Conference on Network and Parallel Computing Workshops*, pages 232–237, 2007.

[8] C. W. Kirkwood. *Strategic Decision Making: Multiobjective Decision Analysis with Spreadsheets.* Duxbury Press, 1996.

[9] G. Lenzini, A. Tokmakoff, and J. Muskens. Managing trustworthiness in component-based embedded systems. *Electronic Notes in Theoretical Computer Science*, 179:143–155, 2007.

[10] B. Lo and G. Z. Yang. Key technical challenges and current implementations of body sensor networks. In *Proc. 2nd International Workshop on Body Sensor Networks*, Apr. 2005.

[11] Z. Yao, D. Kim, I. Lee, K. Kim, and J. Jang. A security framework with trust management for sensor networks. In *Workshop of Security and Privacy for Emerging Areas in Communication Networks*, pages 190–198, 2005.