Security Provisioning in Pervasive Environments Using Multi-objective Optimization

Rinku Dewri, Indrakshi Ray, Indrajit Ray, and Darrell Whitley

Colorado State University, Fort Collins, CO 80523, USA
{rinku,iray,indrajit,whitley}@cs.colostate.edu

Abstract. Pervasive computing applications involve information flow across multiple organizations. Thus, any security breach in an application can have far-reaching consequences. However, effective security mechanisms can be quite different from those typically deployed in conventional applications since these mechanisms are constrained by various factors in a pervasive environment. In this paper, we propose a methodology to perform a cost-benefit analysis under such circumstances. Our approach is based on the formulation of a set of constrained multi-objective optimization problems to minimize the residual damage and the cost of security provisioning. We propose the use of workflow profiles to capture the contexts in which a communication channel is used in a pervasive environment. This is used to minimize the cost that the underlying business entity will have to incur in order to keep the workflow secure and running.

Keywords: Security, Pervasive computing, Multi-objective optimization.

1 Introduction

Pervasive computing aims at making the presence of computing machinery so transparent that their very presence becomes imperceptible to the end user. These applications involve interacting with heterogeneous devices having various capabilities under the control of different entities. Such applications make use of workflows that are mostly automated and do not require much human intervention. For critical applications, the workflows pass on sensitive information to the various devices and make crucial decisions. Failure to protect such information against security breaches may cause irreparable damages.

Pervasive computing applications impose a number of unique constraints that make choosing the appropriate security mechanisms difficult. Interoperability of the heterogeneous devices must be taken into account while selecting security mechanisms. Resource consumption is a very important consideration as this directly relates to the up-time and maintainability of the application. The cost of deployment must also be considered. Thus, an overall picture illustrating the cost-benefit trade-offs in the presence of these constraints is needed before a final decision can be made.

Unfortunately, security threats in a pervasive environment are very application-dependent. Thus, it is not possible to give a solution that is satisfactory for all

S. Jajodia, and J. Lopez (Eds.): ESORICS 2008, LNCS 5283, pp. 349–363, 2008.

pervasive applications. For example, if a communication is between a mobile device and a base station, then a particular type of authentication protocol may be appropriate, whereas if the communication is of an ad-hoc nature, the same protocol may be inappropriate. Further, the communication between two devices can be deemed sensitive or non-sensitive depending on the context under which the communication is taking place. Finally, the resource constraints varies for different scenarios and prohibit the usage of the same security measures even for the same class of threats.

Context based security provisioning has earlier been proposed to adapt the security level of a communication to the sensitivity of the information being exchanged [1,2,3]. Nonetheless, exploring the aforementioned trade-off options becomes difficult when contexts are introduced. Contexts are dynamic in nature and proactive analysis do not always capture all possible scenarios. However, it is important to realize that pervasive environments set up with a specific application in mind has predefined ways of handling the different scenarios that can appear during the lifetime of the environment. It is therefore possible that these scenarios be represented in a concise way and subjected to security evaluation techniques. This is a good beginning since an organization has a concrete understanding of the assets it has and the points of immediate interest usually involve the likelihood of potential damages to these known assets.

In this paper, we formalize these issues and identify possible resolutions to some of the decision making problems related to securing a pervasive environment. We propose the use of workflow profiles to represent the business model of a pervasive environment and compute the cost associated with the maintenance of the workflow. We then perform a multi-objective analysis to maximize the security level of the workflow and minimize the cost incurred thereof. The multi-objective formulations take into account the energy constraints imposed by devices in the environment. We demonstrate our methodology using an evolutionary algorithm in a pervasive healthcare domain.

The rest of the paper is organized as follows. Section 2 presents the related work in this field. An example healthcare pervasive environment along with the concept of workflow representations is presented in Sect. 3. Security provisioning in the workflow is discussed in Sect. 4. Section 5 presents the cost model for the workflow. Section 6 discusses the multi-objective problems, results of which are presented in Sect. 7. Finally, Sect. 8 concludes the paper.

2 Related Work

Security provisioning in pervasive environments is an open field for research. Campbell et al. present an overview of the specific security challenges that are present in this field [4] and describe their prototype implementation of a component-based middleware operating system. A more specific formulation for security provisioning in wireless sensor networks is presented by Chigan et al. [5]. Their framework has an offline security optimization module targeted towards maximizing a *security provision index*. Ranganathan et al. propose some

meta-level metrics to gauge the ability of different security services in a pervasive environment [6].

Dependability issues related to the application of pervasive computing to the healthcare domain is discussed by Bohn et al. [7]. They argue that the healthcare domain can serve as a benchmark platform for pervasive computing research. They point out the security issues relevant to the setup of such a platform. Similar practical issues are also investigated by Black et al. [8].

An example usage of context information in pervasive applications is presented by Judd and Steenkiste [1]. Their approach allows proactive applications to obtain context information on an user's current environment and adapt their behavior accordingly. The use of context information for security provisioning is proposed by Mostéfaoui and Brézillon [2]. They propose using contextual graphs to appropriately decide on the security policies to enforce. Further reasoning on the contributions of combining context and security is provided by the same authors in [3]. Sanchez et al. propose a Monte Carlo based framework to model context data and evaluate context based security policies [9].

3 The Pervasive Workflow Model

Security threats in a pervasive environment are application-dependent. Consequently, business models investing in any kind of a pervasive computing paradigm will highly benefit if formalisms are derived to enable a "case-by-case" study of the problem of security provisioning. We therefore discuss our approach using an example healthcare application.

3.1 A Pervasive Health Care Environment

The pervasive healthcare environment consists of devices that measure the vital signs of patients, location sensors that locate mobile resources, location-aware PDAs carried by health-care personnel, and back-end systems storing and processing records of patient data. The devices are connected through wired or wireless medium. The application consists of different workflows that get triggered by various events. The following example specifies the workflow that handles the situation when an unanticipated change occurs in a patient's vital signs (VS) monitor.

Case 1: The VS monitor tries to detect the presence of the doctor within a wireless communicable distance. If the doctor is present, he can make suggestions which may or may not be based on the patient report stored at the back-end. He may also decide to request the assistance of a nurse, who is located with the help of the network infrastructure. In case of an emergency, the same infrastructure is used to notify the emergency service.

Case 2: If a doctor cannot be located nearby, there is a search for a nurse. The nurse may have the requisite skills to take care of the situation, perhaps with information obtained from the back-end system. If not, the nurse requests the network infrastructure to locate a remote doctor. The remote doctor can

then make his suggestions to the nurse or directly interact with the monitoring devices using the network. Possibilities are also that the doctor feels the need to be immediately with the patient and informs the emergency service on his way. *Case 3:* If a nearby doctor or a nurse cannot be located, the VS monitor communicates with the network infrastructure to locate a remote doctor. The doctor, once located, can remotely interact with the monitoring equipments, or decide to attend to the situation physically, often asking for assistance from a nurse. Emergency services are notified on a need basis. Also, on the event that the network is unable to locate the doctor, it informs the emergency service.

3.2 Computing and Communication Infrastructure

A pervasive application requires communication between different devices with varying degrees of processing power and resource constraints. We classify these devices into three categories: *adapters*, *composers*, and *back-end*. *Adapters* are devices with low processing capabilities driven by a battery source or a wired power supply. They are responsible for collecting raw sensor data and forwarding it to another suitable device. A limited amount of processing can also be performed on the collected data before forwarding them. *Composers* have medium processing capabilities and may have a fixed or battery driven power source. They interact with the adapters and interpret much of the data collected by them, most likely with aid from the back-end. The *back-end* has high processing capabilities driven by a wired power supply. Databases relevant to the pervasive environment reside in the back-end. Figure 1 depicts the typical interactions that can happen between the three device classes.

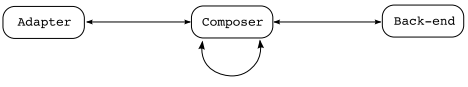

Fig. 1. Component interactions in a pervasive environment

Examples of adapters in the pervasive healthcare environment are the devices that monitor a patient's vital signs and location sensors present in the facility that helps discover a mobile resource. A composer can be a location-aware PDA carried by a doctor or a nurse, a laptop, a data relay point present as part of the network infrastructure, or the system monitored by the emergency personnel. The back-end in this example are data servers used to store patients' medical records or the high-end systems available to perform computationally intensive tasks. Note that the back-end may not be reachable directly by all composers. In such cases, a composer (a personnel's PDA, for example) will first communicate with another composer (a data relay point perhaps) which will then route the request (may be using other data relay points) to the back-end system. Adapters may communicate using a wired/wireless medium with the data relay points, which in turn communicate over an infrastructure network to

the back-end system. The composer class comprising mainly of handheld PDAs and similar devices communicate wirelessly with adapters and other composers.

3.3 Workflow

A workflow captures the various relationships between the participating nodes and provides a concise representation of the different contexts under which the different nodes communicate with each other.

Definition 1. ***(Workflow)*** *A workflow is a tuple* $\langle N, E, n \rangle$ *representing one or more execution paths of a business model, where* N *is a multiset of nodes representing the devices in the application,* E *is a set of ordered pairs of the form* $(n_s, n_d) \in N \times N$ *denoting the communication links, and* $n \in N$ *is a source node that triggers the workflow.*

A workflow can also be visualized in terms of transfer of work between devices. A workflow is a meta-level representation of the order in which different devices participate to achieve one or more business goals. A *feasible execution path* in such a representation resembles the order of participation of the nodes to achieve one of the goals. For example, to find the nearest doctor, transfer of work progresses as VS Monitor→Data Relay Point→Location Sensor→Data Relay Point→ ... →Doctor, and signifies an *execution path.* Although this transfer of work involves multiple location sensors and multiple data relay points, the workflow representation does not make a distinction among them. This is primarily because the objective behind the usage of a communication link between a data relay point and a location sensor is fixed (find a doctor) and hence all such links are assumed to exchange information with the same level of sensitivity.

3.4 Context

Although a workflow helps identify the different communication links used as part of different execution paths, it does not provide any information on the frequency with which a particular channel is used. This frequency estimate is required to determine the rate of power consumption of the two participating devices in the link. When security measures are placed in these links, the rate of power consumption will increase depending on the computational requirements of the algorithms. Here we have an inherent conflict. Heavily used communication channels should ideally have security measures with low computing requirements in order to reduce the power consumption at the two participating devices. At the same time, strong security measures are perhaps required to safeguard the huge amount of information flowing through the channel. Quite often this is hard to achieve since strong security measures typically are computation intensive algorithms.

Definition 2. ***(Context)*** *A context is a prefix of some execution path through a workflow. It is associated with a probability estimate vector that gives the likelihood of the context changing into other contexts.*

A context specifies the different nodes that are traversed when following a particular execution path. We use the notation $\mathcal{C}_v$ to represent a context with the current node v. In some cases, we use a more informal notation and describe contexts by just concatenating the names of the nodes. For example, for the context $ABCDC$, the current node is C and the node has been reached by following the path $A \rightarrow B \rightarrow C \rightarrow D \rightarrow C$. We use '|' to denote the operator to concatenate a node to a context, resulting in a new context. Note that a context may be a part of multiple execution paths. Context-based probability estimates will help capture the likelihood with which a context can change. This is described next.

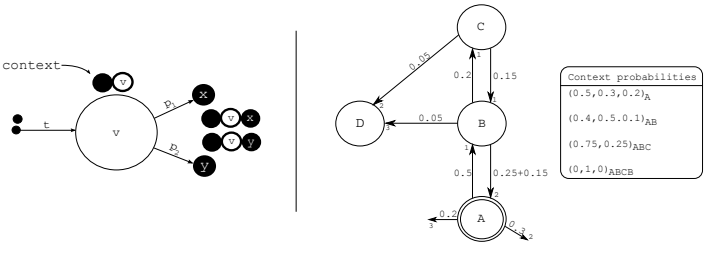

Fig. 2. Left: Context-based probability estimates. Right: Workflow probability calculation.

Consider Fig. 2 (left); let $\mathcal{C}_v$ be the context and let t be its probability of occurrence. We assign a probability on each outgoing edge signifying the chances with which the current context changes. The context becomes $\mathcal{C}_v|x$ with probability p_1 and $\mathcal{C}_v|y$ with probability p_2.

For a node v with k outgoing edges numbered in a particular order, the context-based probability vector $(p_1, p_2, \ldots, p_k)_{\mathcal{C}_v}$ gives the probabilities on the outgoing edges in the same order when the current context is $\mathcal{C}_v$. The probability values for a given context can be obtained from audit logs collected over a period of time signifying how often did v act as an intermediate node in the communication between two neighbor nodes under a particular communication sequence.

It is important to realize that a workflow by itself does not reveal the feasible execution paths. It is only by including context information that the feasible execution paths get defined. Context-based probability estimates tell us if an adjacent node can become a part of a particular execution path. If all probabilities on the outgoing edges of a particular node are zero, then the execution does not proceed and the current context at that point terminates as a feasible execution path. Hence, by defining a set of possible contexts, along with their context-based probability estimates, we have a precise way of defining which execution paths are feasible. We call this set the *context set* of the workflow.

Definition 3. ***(Context Set)*** *A context set for a workflow is a set of contexts that are all possible prefixes of all feasible execution paths.*

The context set contains only those contexts for which there is non-zero probability of transitioning into another context. To infer the feasible execution paths from a given context set, we start at the source node and check to see if the current context (the source node alone at this point) is present in the context set. We can move onto an adjacent node if the probability on the edge to it is non-zero. The current context then changes to a different one for each reachable node. The process is repeated for all such nodes until a node is reached where the current context is not present in the context set. Such a context is then a feasible execution path.

Note that context-based probability estimates provide the probability with which an outgoing edge will be used in the current context. This probability does not, as yet, provide an estimate of the overall frequency with which the edge is used in the workflow. We shall show in Sect. 5 how the context set is used to compute the effective probability with which a particular communication link in the workflow gets used.

4 Security Provisioning

The problem of security provisioning in a workflow involves the identification of potentially damaging attacks and the security mechanisms that can be adopted to protect against such attacks. Depending on the nature of communication between two nodes, a subset of the known attacks can be more prominent than others in a communication channel. Besides the ability to defend against one or more attacks, the choice of a security mechanism is also dependent on the resources it consumes during execution.

Definition 4. ***(Security Mechanism)*** *Given a set of A attacks, denoted by* $a_1, a_2, \ldots, a_A$*, a security mechanism* S_i *is a boolean vector* $[S_{i1}, S_{i2}, \ldots, S_{iA}]$*, where* S_{ij} *is* 1 *if it defends against attack* a_j*,* 0 *otherwise.*

An attack in this definition refers to the consequence of a malicious activity. A security mechanism is capable of preventing one or more attacks. For a given set of N_S security mechanisms, we have a *coverage matrix* defining which attacks are covered by which mechanisms. Further, each mechanism has an associated power consumption rate, denoted by SMC_i, where $1 \leq i \leq N_S$.

To facilitate the enforcement of different security mechanisms along different communication links, we augment each edge on the workflow with an *attack vector* specifying the attacks which are of concern in the link.

Definition 5. ***(Attack Vector)*** *An attack vector on an edge of the workflow is a boolean vector of size A with the* j^{th} *component being either* 1 *or* 0 *based on whether attack* a_j *is plausible on the edge or not.*

Given an attack vector, it is possible that no single security mechanism can provide the required defenses against all attacks of concern on the edge. Multiple mechanisms have to be selected such that they can collectively provide the coverage for the attacks.

Definition 6. ***(Security Control Vector)*** *A security control vector* $SV_e = [SV_{e1}, SV_{e2}, \ldots, SV_{eN_S}]$ *on the edge* e *is a boolean vector, where* SV_{ei} *is* 1 *if the security mechanism* S_i *is chosen,* 0 *otherwise.*

For a particular security control vector SV_e, the attacks collectively covered by SV_e is computed from the expression $\bigvee_i (SV_{ei} \cdot S_i)$, for $i = 1, \ldots, N_s$. The 'dot' operator here indicates scalar multiplication with a vector and '$\vee$' signifies the boolean OR operation. The resultant of this expression is a boolean vector of size A and signifies which attacks are covered by the combination of the security mechanisms. We shall call this vector the *covered attack vector* of the edge on which the security control vector operates.

If AV_e and CAV_e are the attack vector and covered attack vector on an edge e, then the hamming distance between the zero vector and the vector $AV_e \wedge \neg CAV_e$, '$\wedge$' and '$\neg$' signifying the boolean AND and NOT operators respectively, computes the number of attacks initially specified as plausible on the edge but not covered by any security mechanism in the control vector. We shall denote this quantity by $H(AV_e, CAV_e)$.

5 Cost Computation

In this study, we are interested in the cost that an organization has to incur to keep a particular workflow running. To this effect, we consider the *cost of maintenance*. The cost of maintenance relates to the expenses that an organization has to incur to hire personnel for regular maintenance rounds, purchase supplies and hardware support equipments, or may be due to losses arising from downtime in services during maintenance. A reduction in the cost is possible if the workflow can be engineered to run for a longer time between two maintenance rounds. We realize that the contributing factors appear with different magnitudes and in different dimensions, often with different levels of impact, and are hence difficult to combine into one cost measure. We choose to adapt Butler's Multi-attribute Risk Assessment model [10,11] to cater to these difficulties. Butler's framework enables an aggregated representation of the various factors dominating the business model of an organization.

Maintenance cost estimates obtained using Butler's method is used along with frequency estimates obtained from the workflow to determine the total maintenance cost incurred to keep the workflow running. Before we can do so, the probability estimates on the communication links have to be aggregated to determine the overall frequency with which the link is used. This is done by calculating the *effective probability* of usage of a communication link on the event the workflow gets triggered.

Refer to Fig. 2 (left). Since the probabilities on the outgoing edges are decided independent of each other, the effective probability on the two outgoing edges can be calculated as $p_1 t$ and $p_2 t$ respectively. Also, since the probabilities on an outgoing edge is only dependent on the current context, effective probabilities accumulating on an edge from different contexts can be summed together to give

Algorithm 1. EffectiveProbability

```
Global Initialization: Effective probability of all edges is 0.0
Data: Context set C, Context based probability estimates, Workflow graph
Input: Context c, Probability t
if c ∈ C then
    h ⟵ current node in context c
    for each edge e outgoing to node g from h do
        p_e ⟵ probability of going to g from h in the context c
        add p_e t to effective probability of edge e
        EffectiveProbability(c|g,p_e t)
    end
end
return
```

the probability with which the edge is used. Figure 2 (right) shows the calculation of the effective probabilities for a small workflow. The workflow has node A as the source node. The outgoing edges from the source node capture all possible situations that can occur when the workflow is triggered. The given context probabilities are ordered according to the numbering on the outgoing edge at the current node. Algorithm 1 when instantiated with arguments $(A, 1.0)$ recursively computes the effective probabilities on the edges of the workflow given the context probabilities shown in the figure. We denote the effective probability on an edge e by p_e. Once the effective probabilities on each edge of the workflow are calculated, the number of times a particular edge is used can be found by multiplying the number of times the workflow is triggered with the effective probability on the edge.

Let P_i and MC_i be the power capacity and total maintenance cost of the i^{th} unique device respectively. Let $\{e_1, e_2, \ldots, e_k\}$ be the set of edges that connect the nodes corresponding to this device with other nodes. Let T_i be a constant power consumption by the device and $PC_{ij} = \sum_{l=1}^{N_S}(SV_{e_j l} \times SMC_l)$ the power consumption rate of the device because of the security mechanisms in place on edge $e_j; j = 1, \ldots, k$. If the workflow is triggered F times, then edge e_j with effective probability p_{e_j} will be used $f = F \times p_{e_j}$ times. Hence, the total power consumed at device i after the security provisioning on edge e_j is expressed as $(T_i + PC_{ij}) \times f$. A maintenance will be required for the device every $\sum_{j=1}^{k} \frac{(T_i+PC_{ij})\times f}{P_i}$ times of usage. The total maintenance cost for the workflow is,

$$TMC = \sum_i (MC_i \times \sum_{j=1}^{k} \frac{(T_i + PC_{ij}) \times f}{P_i})$$

6 Problem Formulation

The multi-objective formulations presented here are intended to help analyze the trade-offs resulting from the selection of a particular set of security control vectors for the different communication links in a workflow and the corresponding cost of maintenance. To begin with, we decide on a subset $E_p \subseteq E$ of edges on the workflow that are subjected to the security provisioning procedure.

The first optimization problem we consider is the determination of security control vectors for each edge in E_p in order to minimize the cost of maintenance

and the total number of attacks left uncovered on the edges of the workflow, i.e. minimize $\sum_{e \in E_p} H(AV_e, CAV_e)$.

Problem 1. Find security control vectors for each edge $e \in E_p$ that minimizes TMC and minimizes $\sum_{e \in E_p} H(AV_e, CAV_e)$.

Although the total number of uncovered attacks provide a good idea about the potential exploits still remaining in the workflow, the damages that can result from the exploitation of the uncovered attacks can be more than that could have resulted from the covered attacks. The choice of security control vectors based on the number of covered attacks can thus be a misleading indicator of the assets that the employed security mechanisms helped protect. To this effect, instead of minimizing the number of uncovered attacks, the second formulation incorporates the minimization of the total potential damage that can result from the uncovered attacks. To facilitate the computation of the total potential damage, we modify the attack vector to indicate the damages possible instead of just a boolean indicator of whether an attack is plausible in an edge or not. The j^{th} component of the attack vector is then a real valued quantity signifying the *potential damage cost* if attack a_j is not covered by a security mechanism on the edge. The quantity can be zero if the corresponding attack is not of concern on the particular edge. We do not focus on the cost models that can be adopted to estimate such damage levels. Butler's framework is a good starting point in this direction.

Problem 2. Find security control vectors for each edge $e \in E_p$ that minimizes TMC and minimizes $\sum_{e \in E_p} \langle AV_e, \neg CAV_e \rangle$, *where $\langle , \rangle$ signifies the scalar product operation.*

An assumption implicit in the above two formulations is that every security mechanism is capable of running in all the devices present in the workflow. This is not true when there exists devices with very low power capabilities and not all security mechanisms can be supported by them. The existence of such devices impose the constraint that certain security mechanisms can never be placed on certain communication links. Thus, we extend Problem 2 to generate solutions that are feasible within such constraints. Let $n_{s,e}$ and $n_{d,e}$ denote the two communicating devices on the edge e. For the security mechanisms to be able to execute, the total power consumed by the mechanisms in place on this edge has to be less than the minimum of the power capacities of the two participating devices. The optimization problem is then formulated as follows.

Problem 3. Find security control vectors SV_e for each edge $e \in E_p$ which minimizes TMC and minimizes $\sum_{e \in E_p} \langle AV_e, \neg CAV_e \rangle$, *satisfying the constraints* $\sum_{i=1}^{N_S} (SV_{ei} \times SMC_i) \leq min(P_{n_{s,e}}, P_{n_{d,e}})$ *, for all edges $e \in E_p$.*

For the final problem, we explore the scenario when the adopted security mechanisms are not robust enough and are prone to failures. Non-robust security control vectors suffer from the drawback that a failure in one of the security mechanisms can heavily increase the number of uncovered attacks or the total potential damage in the workflow. Robust solutions, on the other hand, are

able to contain such increase within a pre-specified acceptable level. We first introduce the notion of a *failure radius* r which signifies the number of security mechanisms that can fail at a time. For a given failure radius, we can specify an acceptable level D of increase in the total number of uncovered attacks, or the total potential damage, in the event of failure. The robust version of Problem 3 is then stated as follows.

Problem 4. Find security control vectors SV_e for each edge $e \in E_p$ which minimizes TMC and minimizes $PD = \sum_{e \in E_p} \langle AV_e, \neg CAV_e \rangle$, *satisfying the constraints* $\sum_{i=1}^{N_S} (SV_{ei} \times SMC_i) \leq min(PC_{n_{s,e}}, PC_{n_{d,e}})$ *, for all edges* $e \in E_p$ *and the constraint that the maximum increase in* PD, *resulting from at most* r *security mechanism failures, does not exceed* D.

We employ the Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) [12] to solve the four multi-objective problems presented. A solution to a problem is represented by a boolean string generated by the concatenation of the security control vectors for each edge. We identified 23 different communication links of interest in the example healthcare workflow and 8 different security mechanisms, giving us a solution encoding length of $8 \times 23 = 184$. The parameters of the algorithm are set as follows: population size = 300, number of generations = 1000, crossover probability = 0.9, mutation rate = 0.01, and binary tournament selection.

Due to the non-availability of standard test data sets, the experiments performed involve hypothetical data. Nonetheless, the analysis do not make any reference to the absolute values obtained for the objectives from the optimization. The observations reveal what kind of cost-benefit information can such an analysis provide irrespective of the exact numerical values of the quantities.

7 Results and Discussion

The trade-off solutions obtained when minimizing the number of uncovered attacks and the total maintenance cost are shown in Fig. 3 (left). More number of attacks can be covered by enforcing a properly chosen subset of the security mechanisms, although resulting in heavy power utilization to support them. The cost of maintenance thus increase when lesser number of attacks are left uncovered. This observation conforms to our intuitive understanding. However, although no two solutions in the solution set (non-dominated front) are comparable in terms of their objective values, all solutions from the set do not fare equally well. Note that the number of attacks covered by a solution has no information in it about the total damage that it helped contain. This prompts us to identify the *line of shift* where the damage cost possible from uncovered attacks becomes more than that from covered ones.

A graphical illustration of the line of shift is shown in Fig. 3 (right). The figure shows the ratio of uncovered damage cost to covered damage cost. Any solution beyond the line of shift signifies a higher uncovered damage cost. Observe that a substantial number of solutions can exist beyond this line. If a decision maker's

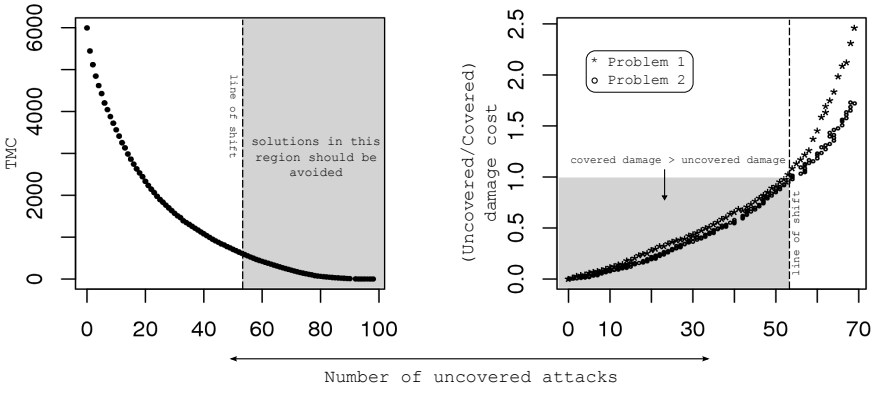

Fig. 3. Left: NSGA-II solutions to Problem 1. Right: Ratio of uncovered and covered damage cost for solutions obtained for Problem 1 and 2. The line of shift shows the point beyond which the uncovered damage is more than the covered damage.

solution of choice lies beyond the line of shift, it is advisable that the process of security provisioning be rethought.

In terms of problem formulation, Problem 2 takes a damage-centric view of security and explicitly considers the total uncovered potential damage cost as an objective. Interestingly, this formulation can result in solutions with a lower uncovered to covered damage ratio for a given number of attacks left uncovered (Fig. 3 (right)). A lower ratio indicates that the fraction of damages covered is much more than that uncovered. Hence, a Problem 2 solution is better than a Problem 1 solution since it gives the added benefit of having a lower uncovered to covered damage ratio. In this sense, solving Problem 2 can be a better approach even when the view of security is attack-centric.

Figure 4 (left) shows the trade-off solutions when the uncovered damage cost is considered as one of the objectives. The non-dominated front is concave in structure with three identifiable regions of interest. Type I and Type III regions correspond to solutions where one of the objectives has a faster rate of decay than the other. From a cost of maintenance point of view, the trade-off nature in these regions signify that a decision maker can generate better outcome in one objective without much degradation on the other. This is quite difficult to perceive without having a global view of the interaction present between the two cost measures. The choice of a solution in the Type II region signify a good balance between the two cost factors. However, the number of solutions lying in each of these regions can vary significantly. Figure 4 (right) shows the same non-dominated front when certain devices have a much higher cost of maintenance compared to others. Observe that the Type I and Type III regions become more prominent in this front. This gives a decision maker better avenues to argue the selection of a solution biased towards a particular objective. Further, often solutions appear as part of a disconnected region of the non-dominated front (Fig. 4 (inset-right)). Such regions can be of special interest to a decision maker

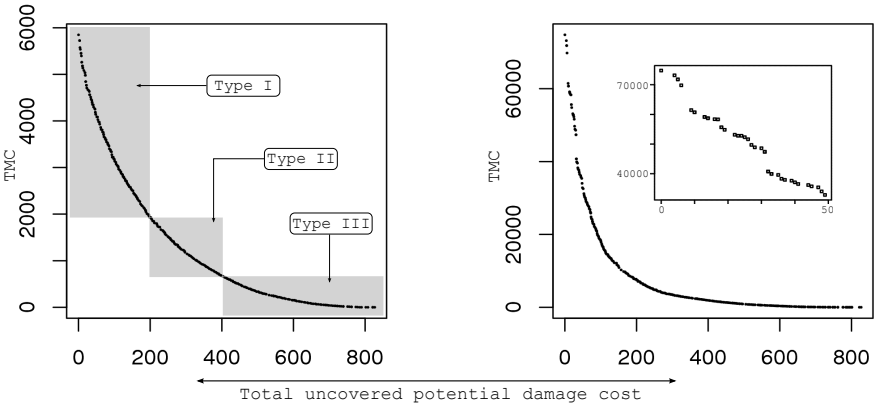

Fig. 4. NSGA-II solutions to Problem 2. Left: Solutions when maintenance cost of the devices are comparable. Right: Solutions when some devices have comparatively higher maintenance cost.

since disconnected solutions indicate that a change can be obtained in one objective by sacrificing a negligible value in the other.

The power capacity of a device restricts the usage of all possible subsets of the security mechanisms in the device. Figure 5 (left) illustrates how the non-dominated front from Fig. 4 (left) changes when the feasibility constraints are considered. The entire non-dominated front shifts to the right and clearly marks a region where no solution can be obtained. This in turn indicates the unavoidable damage cost that remains in the workflow. It is important that a business entity investing in a pervasive setup is aware of this residual cost in the system. This cost provides a preliminary risk estimate which, in the worst case, can become a matter of concern. If the unavoidable potential damage is too high, the setup will be running under a high risk of collapse.

The next step to the analysis involves the sensitivity of the solutions towards failure. The robustness analysis of a solution in Fig. 5 (left-inset) indicates that the uncovered potential damage cost can increase considerably for a failure radius of only 1. At this point, a decision maker can perform such analysis on every solution of interest and choose a feasible one. However, such analysis are cumbersome and no control is possible on the actual amount of increase in the cost that an organization can sustain in the event of failure. Problem 4 alleviates this situation with a robust formulation of Problem 3.

Figure 5 (right) shows the robust solutions obtained for varying levels of acceptable cost increase. The increase in the potential damage cost stay within this level in the event of a failure of at most one security mechanism. Depending on the nature of the problem, obtaining solutions with small values of D may not be possible at all. Thus, obtaining a certain level of robustness for a given level of security is not always feasible. However, there could be areas where experimenting with different values of D can be beneficial in understanding how

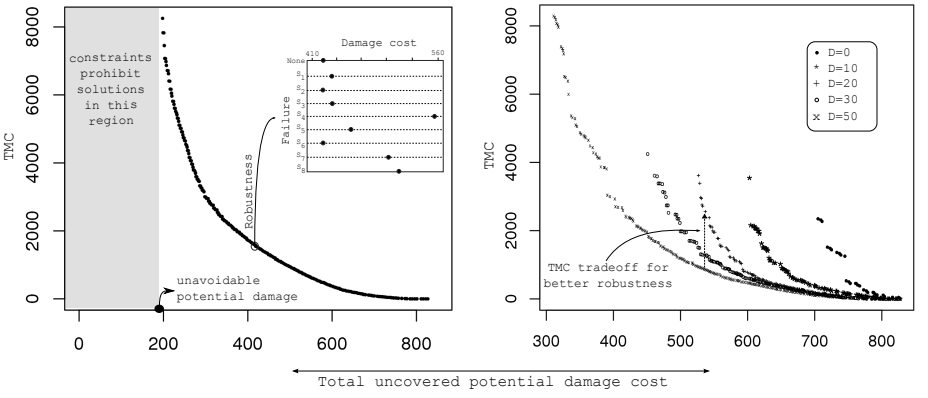

Fig. 5. Left: NSGA-II solutions to Problem 3. Constraints on the usage of security mechanisms result in an unavoidable potential damage. Inset figure shows the sensitivity of a solution to security mechanism failures. Right: NSGA-II solutions to Problem 4 for *failure radius* = 1. Robustness can be improved at the cost of higher maintenance cost.

the cost of maintenance changes with changing levels of robustness. As is seen in this example, the increase in the cost of maintenance is much higher when moving from a robustness level of $D = 30$ to 20 than moving from $D = 50$ to 30.

8 Conclusions

In this paper, we address the problem of optimal security provisioning in pervasive environments under the presence of energy constraints. We adopt a workflow model to represent the different contexts under which a communication is established between two devices. We provide a formal statement of the problem of security provisioning and define a series of multi-objective optimization problems to understand the trade-offs involved between the cost of maintenance and the security of a pervasive setup.

Our analysis reveals important parameters that a business entity should be aware of before investing in the setup. First, the definition of "security" in the formulated problems plays an important role. Often, an attack-centric view of security is not enough and emphasis must be paid rather to a damage-centric view. Good solutions protecting against more attacks do not necessarily protect higher asset values. Also, the distribution of these solutions on the objective space provide invaluable clues to a decision maker on the amount of security gains possible across different levels of cost. The presence of energy constraints results in an unavoidable potential damage always residual in the system, early estimates on which can help the business entity invest better in risk mitigation strategies. Risk estimates also depend on the robustness of a chosen solution. Our robust formulation enables one to control the changes that can occur in the event of security mechanism failure and explore the costs involved.

We acknowledge that the presented work involves various other areas of research that require equal attention. The modeling of the different cost factors is a crucial aspect without which optimization formulations are difficult to transition to the real world. As immediate future work, we shall explore the possibility of modifying the optimization framework to work on workflow models that can be broken down into sub-workflows for scalability.

Acknowledgment

This work was partially supported by the U.S. AFOSR under contracts FA9550-07-1-0042 and FA9550-07-1-0403. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies of the U.S. Air Force or other federal government agencies.

References

1. Judd, G., Steenkiste, P.: Providing Contextual Information to Pervasive Computing Applications. In: PerCom 2003, pp. 133–142 (2003)
2. Mostéfaoui, G.K., Brézillon, P.: Modeling Context-Based Security Policies with Contextual Graphs. In: PerCom 2004, pp. 28–32 (2004)
3. Mostéfaoui, G.K., Brézillon, P.: Context-Based Constraints in Security: Motivation and First Approach. Electronic Notes in Theoretical Computer Science 146(1), 85–100 (2006)
4. Campbell, R.H., Al-Muhtadi, J., Naldurg, P., Sampemane, G., Mickunas, M.D.: Towards Security and Privacy for Pervasive Computing. In: Okada, M., Pierce, B.C., Scedrov, A., Tokuda, H., Yonezawa, A. (eds.) ISSS 2002. LNCS, vol. 2609, pp. 1–15. Springer, Heidelberg (2003)
5. Chigan, C., Ye, Y., Li, L.: Balancing Security Against Performance in Wireless Ad Hoc and Sensor Networks. In: VTC 2004, vol. 7, pp. 4735–4739 (2004)
6. Ranganathan, A., Al-Muhtadi, J., Biehl, J., Ziebart, B., Campbell, R., Bailey, B.: Towards a Pervasive Computing Benchmark. In: PerCom 2005, pp. 194–198 (2005)
7. Bohn, J., Gärtner, F.: H.Vogt: Dependability Issues in Pervasive Computing in a Healthcare Environment. In: SPC 2003, pp. 53–70 (2003)
8. Black, J.P., Segmuller, W., Cohen, N., Leiba, B., Misra, A., Ebling, M.R., Stern, E.: Pervasive Computing in Health Care: Smart Spaces and Enterprise Information Systems. In: MobiSys 2004 Workshop on Context Awareness (2004)
9. Sanchez, C., Gruenwald, L., Sanchez, M.: A Monte Carlo Framework to Evaluate Context Based Security Policies in Pervasive Mobile Environments. In: MobiDE 2007, pp. 41–48 (2007)
10. Butler, S.A.: Security Attribute Evaluation Method: A Cost-benefit Approach. In: ICSE 2002, pp. 232–240 (2002)
11. Butler, S.A., Fischbeck, P.: Multi-attribute Risk Assessment. In: SREIS 2002 (2002)
12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)