Engineering Adaptive Privacy: On the Role of Privacy Awareness Requirements

Inah Omoronyia[1], Luca Cavallaro[2], Mazeiar Salehie[2], Liliana Pasquale[2], Bashar Nuseibeh[2,3]
[1] School of Computing, University of Glasgow, UK
[2] Lero – The Irish Software Engineering Research Centre, University of Limerick, Ireland
[3] Department of Computing, The Open University, UK
inah.omoronyia@glasgow.ac.uk, {luca.cavallaro, mazeiar.salehie, liliana.pasquale, bashar.nuseibeh}@lero.ie

***Abstract*—Applications that continuously gather and disclose personal information about users are increasingly common. While disclosing this information may be essential for these applications to function, it may also raise privacy concerns. Partly, this is due to frequently changing context that introduces new privacy threats, and makes it difficult to continuously satisfy privacy requirements. To address this problem, applications may need to adapt in order to manage changing privacy concerns. Thus, we propose a framework that exploits the notion of privacy awareness requirements to identify runtime privacy properties to satisfy. These properties are used to support disclosure decision making by applications. Our evaluations suggest that applications that fail to satisfy privacy awareness requirements cannot regulate users' information disclosure. We also observe that the satisfaction of privacy awareness requirements is useful to users aiming to minimise exposure to privacy threats, and to users aiming to maximise functional benefits amidst increasing threat severity.**

***Index Terms*— Privacy, utility, selective disclosure, adaptation**

I. Introduction

Consumers and enterprises increasingly rely on mobile and ubiquitous applications, such as smart phones, to satisfy their social and business needs. This new generation of applications enable users to form localised, short- and long-lived groups or communities to achieve common objectives. These applications may need to gather and disclose users' sensitive information such as location, time, proximity to nearby services, and connectivity to other users. The exposure of such information in an unregulated way can threaten user privacy [1]. This calls for a more systematic approach for considering the privacy requirements of users in software applications. A representative class of such requirements is selective disclosure – deciding what information to disclose, in which context, and the degree of control an individual has over disclosed information [3].

A key determinant of selective disclosure is frequently changing context; e.g., changing time, location and group properties. These changes blur the boundary between public and personal spaces and may introduce unforeseen privacy threats [2]. Additionally, users may be unaware of when and for what purpose sensitive information about them is being collected, analysed or transmitted. This makes it even more difficult for users and applications to adapt in order to continue satisfying their privacy requirements.

In this paper, we present an adaptive privacy framework that aims to support the selective disclosure of personal information in software applications. We follow the popular MAPE (Monitor, Analyse, Plan and Execute) loop [8] for designing adaptive applications. Our framework consists of models, techniques and tools, and focuses on the role of *privacy awareness requirements* (PAR) that embody three concerns: (i) the identification of what attributes to monitor in order to detect privacy threats; (ii) the discovery of such threats before personal information is disclosed; and (iii) a utility for the severity of discovered threats, as well as for the benefit that can be derived if associated information is disclosed. We suggest and demonstrate that an advantage of this approach is that decisions on whether or not to disclose information can be made based on reliable knowledge of both its cost and benefit to the user of the application.

Our approach relies on software behavioural and context models, and the privacy requirements of users, to identify attributes to monitor in order to discover a privacy threat. A privacy threat is discovered by searching the history of system interactions that may affect the satisfaction of a user's privacy requirements. The severity of identified threat and the associated benefit of disclosure are determined by analysing the evolving properties of generated networks emerging from users' interactions. Subsequently, we investigate the relevance of such utility measure during the planning phase of software adaptation. Our approach for identifying monitored attributes had been implemented in an openly accessible automated environment [4]. In this paper, we evaluated our framework using a comparative study to examine the consequence of the satisfaction/failure of PAR during the planning phase of adaptive privacy. First, we showed that applications that fail to satisfy PAR are unable to manage privacy based on the utility of disclosure. Second, we showed that applications that satisfy PAR are able to regulate the disclosure of information with changing context.

The remainder of the paper is organised as follows. Section II presents some related work on privacy awareness and adaptation relevant to our overall approach, which is then presented in section III. Section IV describes the models that we use in our approach. Section V then presents PAR and the associated mechanisms used to derive monitoring and utility measures for adaptive privacy. Section VI focuses on experimental evaluation of our approach. Conclusions and further work are presented in section VII.

II. Related Work

The core contribution of our research is to show that capturing privacy awareness requirements in software systems can be used to better engineer adaptive privacy. In this section, we present a review of related work in the area of adaptive privacy, privacy engineering and privacy awareness upon which we build our approach.

A related notion of awareness requirements was described by Souza *et al.* [5] as a class of requirements that predicate the satisfaction of other requirements. Albeit not focused on privacy, the approach has been used [7] to specify the need for adaptation of software systems. In this paper we tailor the notion of awareness requirements, and propose the use of PAR for adaptive privacy.

Adaptive privacy was introduced by Schaub *et al.* [9] as a system's ability to preserve privacy in presence of context changes, by providing recommendations to the user or via automatic reconfiguration. This reconfiguration is essential since the boundary delimiting the decision of a user disclosing or withholding information changes with context [10]. Braghin *et al.* [11] used the concept of ambient to enforce privacy in a changing environment. This was achieved by using policies to define boundaries of information disclosure. Braghin *et al.* did not consider that changes in context may determine the need to change attributes that need to be monitored in the application, nor the notion of utility.

Privacy engineering was described by Spiekermann and Cranor [6] as a systematic effort to embed privacy concerns into the design of an application. Following this principle, Kalloniatis *et al.* [12] proposed a methodology to build privacy-preserving applications. Similarly, Liu *et al.* [13] proposed a requirement-driven development methodology for secure and privacy-preserving applications based on the i* framework. Barth *et al.* [14] used the contextual integrity framework as a means to design privacy-preserving applications. Contextual integrity features a context model and a formal model that uses temporal logic to define the communication between two entities, and how the disclosure of information takes place. These methodologies and framework did not consider that the system can adapt during its lifetime, consequently the constructs for the satisfaction of privacy requirements can also change.

Finally, Pötzsch [15] defines privacy awareness as an individual's cognition of who, when, which, what amount, and how personal information about his/her activity is processed and utilised. Pötzsch's view of privacy awareness helps to provide a set of constructs for building a context for which adaptive privacy can be assessed. An empirical study about the impact of this view of privacy awareness has been carried out [16]. However, privacy awareness constructs have not been investigated in adaptive privacy. We suggest that privacy awareness is critical to enable users and systems to gain sufficient knowledge about how to act in privacy sensitive situations. As they gain assurance that their privacy is broadly preserved and of the expected benefit of disclosure, they may consider forfeiting their privacy when engaging in some interactions.

III. Overall Approach and Motivating Example

The objective for adaptive privacy is to enable applications to detect privacy threats with changing context, and subsequently carry out adaptation actions to ameliorate the consequences of the threat. Our approach to achieve this is based on the rationale that for useful adaptation to occur in software systems, there needs to be monitoring, analysis, planning and execution [8]. These activities enable software systems to detect at runtime the changes in the system context and to appropriately react to them.

As shown in *Figure 1*, we propose that adaptive privacy firstly mandates that systems should be able to identify attributes to monitor in order to detect privacy threats (monitoring). Secondly, when privacy threats are discovered, systems should have an understanding of the consequence of the threat (analysis). It is the ability of systems to satisfy their monitoring and analysis needs that we characterise as PAR. The satisfaction of this requirement serves as a useful input into the system's ability to make informed decision on information disclosure (planning). Finally, based on disclosure decision, mitigation actions can be carried out to ameliorate the consequence of the threat (execution). Such mitigation actions can involve updating the behaviour of the system, changing the context of operations or users carrying out explicit mediation actions.

In this paper, we focus on the first three phases of the MAPE loop to highlight the PAR for making useful disclosure decisions. Specifically, PAR are the requirements that need to be satisfied by a system in order to meaningfully adapt the privacy of its users as context changes. In this research, we demonstrate that the core of such requirements include: the ability of systems to identify attributes to monitor in order to detect privacy threats; the discovery of a privacy threat before information about a user is disclosed; and the severity of the threat as well as the benefit of information disclosure amidst the discovered threat. In this paper, we first highlight the models useful for satisfying PAR (Section IV), we then present an approach to PAR analysis based on these models (Section

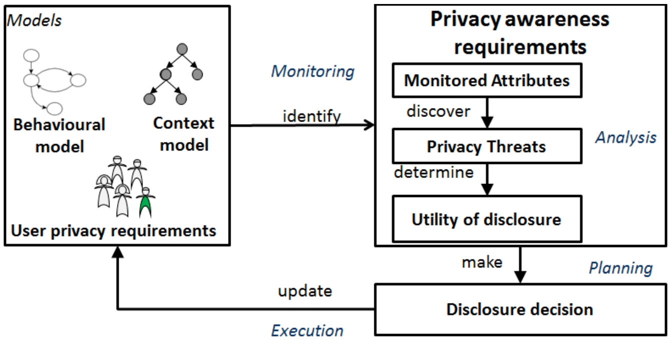

Figure 1 Adaptive privacy framework

V). Finally, we present an evaluation of the usefulness of PAR at the planning phase of adaptive privacy (section VI).

We achieve our objective using an example involving a track sharing application for runners and other outdoor activities. Similar examples of such application include B.iCycle (http://b-icycle.com), MyTracks (mytracks.appspot .com), etc. Typically, such applications enable a group of users to share live GPS tracks and performance statistics with fellow users and other agents such as their fitness instructors and physicians. For this example, privacy management includes the capability of users to decide the limits of information disclosure to other users – about their current location, distance, age, heart rate, burned calories, weight loss, etc. Effective adaptive privacy requires users sharing their outdoor activity experience to understand information flows, weigh the consequences of sharing information, and make informed, context-specific decisions to disclose or withhold information.

IV. Models for Satisfying PAR

As shown in figure 1, the three main models we use to enable the satisfaction of privacy awareness requirements include behavioural and context model, as well as a model representing user privacy requirements. We discuss each of these models in detail.

A. Context Model

A Context model is useful for identifying interactions that can occur between users, as well as the attributes of the users involved in achieving a common objective [17]. For adaptive privacy, context models represent attributes that are subsequently manipulated on by the software system in order to support the activity of the users as they interact with one another. The advantage of context models is their reuse for design of multiple systems.

In this research, we formally define a context model as a tuple: *CM = (D, Y, U, R, C)*. Where: *D* is a set of attributes; *Y* is a set of entities; *U* is a relation *U: Y x Y*, representing the association between entities, *R* is a relation $R\colon D \text{ x } 2^D$ between attributes, where 2^D is the power set of D, and *C* is a relation $C\colon Y \text{ x } 2^D$ that associates entities to attributes.

An instance of context model for the case study in Section III.B is shown in Figure 2. An entity in our model represents an object that can be described by a set of attributes. Attributes here are viewed as atomic variables that cannot be further decomposed. There are two kinds of entities: entities that characterise the environment of interaction (e.g. the *Location* entity, represented in light boxes in Figure 2), and entities representing roles of users in the system. As shown in the shaded boxes in Figure 2, we represent a set of users as agents. These agents interact to exchange information (attributes about an agent) with other agents. The agent responsible for sending the information is referred to as the *Sender*, while the receiving agent is the *Receiver*. The agent characterised by the sent information is the *Subject*. These interactions that occur between agents with a given role are modelled by the relation *U*. This relation also expresses how other entities relate with each other (e.g. *Track* and a *Location)* or between agent and other entities (e.g. *Subject* and *ActivityType*).

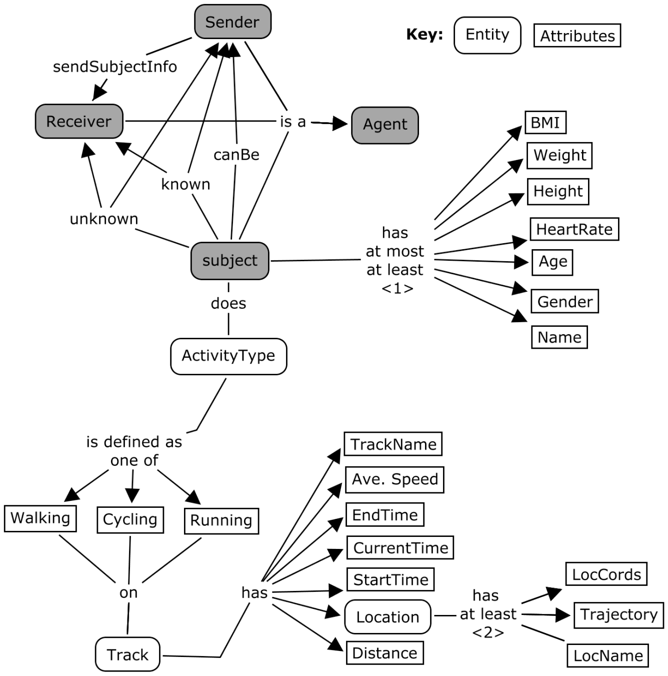

Figure 2 Example of a context model

Attributes can relate with each other as modelled in the inference relation *R.* Such interference relations are established rules that enable the deduction of previously unknown attribute from another disclosed attribute [17]. For example, the rule *LocCords* = Ave.*Speed* $\sqcup$ *TrackName* $\sqcup$ *StartTime*, infers that the location coordinates of a user can be inferred from the disclosure of the users average speed, track name and start time. Another example is the rule *BMI = Height* $\sqcup$ *Weight*. Finally, an entity is related to a set of attributes as expressed in the relation *C*. An example in Figure 2 is the relation between the entity *Location* and the attributes *LocCords*, *Trajectory*, and *LocName*. Generally, our approach to modelling context is amenable to other forms of inferences based on transitive, Euclidian or symmetric relations.

B. Behavioural Model

A Behavioural model represents the different states that an agent can reach, and the transitions required to enter such state. Formally, a behavioural model is a tuple: *B = (S, E, τ)*. Where: *S* is a set of states; *E* is a set of events, where an *event* is defined by a tuple $((D^n|Y), a_x,$ {*sent* | *received*}). In that tuple, D^n is a set of cardinality *n*, containing the attributes manipulated by the transition. The set D^n can be replaced by an entity that characterise the environment of interaction. The subject a_x identifies the agent, the event is referred to (i.e. an agent with type *Subject*), and *{sent|received}* expresses if the attributes are sent or received by the agent respectively. Finally, *τ* is the state transition relation $\tau\colon S \text{ x } E \cup \{\varepsilon\} \rightarrow S$. The occurrence of a transition with an *ε-event (ε*-transition) means that the agent that the behaviour is referred to does not send or receive any attribute in that transition.

An example of a behavioural model, represented as a Labelled Transition System (*LTS*), is shown in Figure 3. In Table 1, we illustrate the set of context attributes and the events associated to each transition of the *LTS* in Figure 3. For example, the transition t_4 in Table 1 from state 3 to 4 in Figure 3 is triggered by the event shareRaceResult. The attributes used by the event are shown in the right column of the table. These attributes are associated to the agent a_1, which in our example is the subject. Finally the event features the *sent* keyword, indicating that t_4 sends the attributes in the event.

Table 1 also features some ε-transitions that are signified by the representation → ε. Also note that there can be transitions not associated with context attributes (example t_8). For detecting privacy threats we focus only on the information that the various agents in the system may exchange. Consequently we consider t_1, t_2, t_3 and t_8 as ε-transitions, since they do not send or receive any attribute.

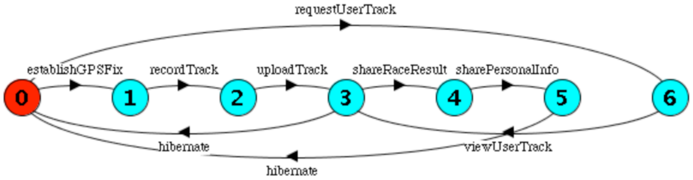

Figure3 A behavioural model B_1

Table 1 Attributes associated with state transitions (t_i)

(t)	*Event*	*Event ∪ {ε}*
t_1	establishGPSFix	{LocCords, StartTime, ActivityType} → ε
t_2	recordTrack	{CurrentTime, Ave.Speed, LocCords} → ε
t_3	uploadTrack	{AveSpeed, TrackName, LocCords, StartTime, EndTime} → ε
t_4	shareRaceResult	{AveSpeed, TrackName, LocCords, StartTime, EndTime, Weight}, a_1, sent
t_5	sharePersonalInfo	{Gender, Height, Age}, a_1, sent
t_6	requestUserTrack	{SubjectName}, a_1, sent
t_7	viewUserTrack	{AveSpeed, TrackName, StartTime, EndTime, LocCords , Weight, Height, Gender, Age}, a_1, sent
t_8	hibernate	{ } → ε

C. *User Privacy Requirements*

The User privacy requirements are individual expressions by agents to regulate the manner of information disclosure about their activity. We view the possible failure of a privacy requirement in a specific context as a *privacy threat* that triggers appropriate mitigation actions (see Section V.B).

We build on the view presented in [10] where the privacy of a subject is conditioned by the subject's (and other agents) past experiences and expectations of the future. In this way, we assume that privacy requirements are expressed as IF-THEN rules. The IF segment contains an event and the identity of its *Sender* or *Receiver*. If the event is a *sent* event, then a *Receiver* of the event is identified. Alternatively, if the event is a *received* event, then a *Sender* of the event is identified. The THEN part is a linear temporal logic (LTL) formula that captures the past experiences or future expectations of a subject by using past and future operators [18]. The LTL formula predicates about the values that attributes of the context model can assume, or the knowledge that can be gained by agents about a subject over time [19][20]. The formula, $K_{ai}\ d_{aj}$ is a modal representation of knowledge [20] that the agent a_i knows the value of the attribute *d* about a subject $a_{j.}$ Examples of privacy requirements for the selective disclosure of a subject's information can be stated as thus:

(PR$_1$) *IF Location, a_1, received, a_2*
THEN $\Diamond_{\mathrm{p}}$ StartTime $_{a1}$ < 21.00 hrs.

(PR$_2$) *IF Weight, a_1, sent, Receiver*
THEN $\Box\neg K_{Receiver}$ BMI$_{a1}$.

The symbols $\Diamond_{\mathrm{p}}$ and $\Box$ are the LTL operators eventually in the past and globally respectively as defined in Table 2. *PR$_1$* is active if a_2 sends a specific *Location* attribute about the agent a_1. In that case, the subject should have started his race before 21.00 hrs, at least once in the past. Similarly, *PR$_2$* is active, if the *Weight* of a_1 is received by any *Receiver*, then $\neg K_{Receiver}$ *BMI$_{a1}$*(i.e. the *Receiver* does not know the *BMI* of a_1) should hold globally (i.e. both in the past and in the future).

V. Analysing PAR

In this section, we describe different analysis used for the satisfaction of PAR based on models described in section IV.

A. *Identifying Attributes to Monitor*

For the satisfaction of PAR we first need to identify the attributes to monitor for the detection of privacy threats. We realise this by identifying the subset of attributes that are common to the context and behavioural model, and in the privacy requirement of the user. The identification process is composed of four steps:

1. For each privacy requirement *PR$_x$*, a couple *(d_x,a_x)* or *(y_x,a_x)* present in the event of the IF segment is collected in a set *M.* Where *d_x and y_x* identify an attributes or an entity referenced by the event in the IF part of *PR$_x$*, and a_x is the subject in the event.
2. For each couple (y_x, a_x) in *M*, the tuples c_{dx} contained in the relation *C* of the context model, and featuring y_x are selected. For each tuple c_{yx} in *C*, a new couple (d_j, a_x) is added to *M*, where *d_j is* an attribute in c_{yx}. Finally the couple (y_x, a_x) is removed from *M*.
3. For each couple (d_x, a_x) in *M*, the tuples r_{dx} contained in the relation *R* of the context model, and featuring d_x is selected. For each tuple r_{dx}, a new couple (d_j, a_x) is added to *M*, where d_j is an attribute in r_{dx}. It is noted here that each d_j can also be part of some other tuples in the relation *R*. In this case the step is repeated for those tuples.
4. Finally, for each couple (d_x, a_x) in *M*, the behavioural model of the agent a_x is considered. The transitions in the transition relation τ_{ax}, that are triggered by events associated to d_x are selected and associated to *PR$_x$*.

The output of these steps consists of the *monitored attributes* contained in the set *M*, and the *marked transitions* in the behavioural model of an agent that operate on monitored attributes. These marked transitions identify where specifically in the behaviour of an agent the monitoring needs to occur.

Considering PR_1 in Section IV.C, the IF part of the requirement contains the event (*Location*, a_1*, received)*. The first process step adds the couple *(Location,* a_1*)* to the set *M*. The second step considers those tuples in the relation *C* of the context model that include the *Location* entity. As showed in Figure 2, the tuples are *(Location,LocCords)*, *(Location,Trajectory)* and *(Location,LocName)*. The couples *(LocCords,* a_1*)*, *(Trajectory,* a_1*)*, and *(LocName,* a_1*)*, are then added to *M*. The third step considers the relation *R* in the context model and looks for attributes already in *M* also contained in a tuple in *R*. For instance, as highlighted in section IV.A, the *LocCords* attribute is included in the tuple *(LocCords, Ave.Speed, TrackName, StartTime)* of the relation *R*. Thus, the couples *(AveSpeed,* a_1*)*, *(TrackName,* a_1*)* and *(StartTime,* a_1*)* are added to *M.* The final step considers the transitions of the agent a_1 as shown in Table 1. The transitions t_4 and t_7 are marked since they contain the attributes *TrackName, LocCords* and *StartTime*.

B. Privacy Threats Detection

Privacy threats detection is aimed at discovering if an *interaction* between agents can result in the failure of a privacy requirement. We define an interaction as the exchange of information about a subject between a sender and a receiver. Given the set of monitored attributes and the associated transitions in agent behaviours, our approach analyses the interaction history of the system. The outcome ascertains the satisfaction or failure of a given privacy requirement in a moment of the system's history. In this subsection, we first characterise an interaction between two agents, followed by how a sequence of such interactions can be used to describe a history of the system. Finally, we demonstrate the detection of privacy threats based on the defined history.

1) Characterising an interaction between two agents: An interaction is a single information-flow between two agents' a_1 and a_2. Formally, such flow is defined as either of the following: *i)* a couple (t_s,t_r), where t_s is a transition $(s_a\ (d_1,d_2,\dots,d_y,\ a_x,\ sent)\ s_b)$ belonging to the transition function of a_1, and t_r is a tuple $(s'_a\ (d_1,d_2,\dots,d_y,\ s_x,\ received)\ s'_b)$ belonging to the transition function of a_2; *ii)* two couples (t_s,ε), (ε,t_r). The latter represents interactions where an agent sends information that is eventually received by a receiver in the system. We have assumed asynchronous interaction. It is possible that synchronous interaction can yield different behavioral runs and knowledge models

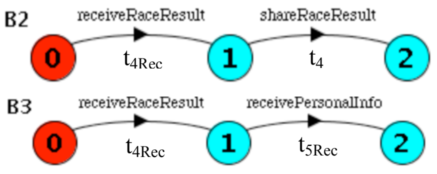

Figure 4 Behavioural models B_2 and B_3

For example, assume that in a system there exist an agent a_1 whose behavioural model is represented by B_1 in Figure 3, and a second agent a_2, whose behavioural model B_2 is presented in Figure 4. The transition t_4 in B_2 is identical to the transition t_4 in B_1, and the transition t_{4Rec} features a *receive* events with the same attributes as the event in t_4. Assuming the events in B_1 and in B_2 have the subject a_1 (i.e., a_1 is sending information about itself to a_2). Then the interaction between a_1 and a_2 is the tuple $(t_{4,\,a1};\ t_{4Rec,\,a2})$.

2) Defining the history of the system: A history of a system is a sequence of information-flows between agents in the system. The history keeps track in each time instant, of the predicates holding in that instant and the values of context attributes describing each subject. Formally, a history is defined as $H = h_1, h_2, \dots, h_n$. Each $h_{i\,\in[1,\,n]}$ is a history step described by the sets $(\sigma,\ \nu,\ \pi)$, where $\sigma = \{S_{a1},\dots,\ S_{am}\}$ contains the states of the agents in the system in that history step, v contains the contextual attributes values for each agent and π contains the predicates that hold in h_i. In *H*, the transition from h_i to h_{i+1} signifies an interaction between two agents in the system. That interaction changes the states of the agents involved in the transition and can modify the predicates holding in the arrival state.

Consider the partial history of a system involving interactions between a group of agents a_1, a_2, and a_3. An instance of that history is shown in Figure 5. The behaviour of a_1, a_2 and a_3 is as described in B_1, B_2 and B_3 of Figures 3 and 4 respectively. The behaviour of a_3 features the transition t_{4Rec}, which we assume identical to the one already introduced for a_2, and the transition t_{5Rec}, whose event has the same attributes as t_5 in Table 1, but is a *received* event. In h_1 of Figure 5, the agent a_1 is in state B1(3) as a result of *uploadTrack* event, while a_2 and a_3 are in Idle states (i.e., B2(0) and B3(0) respectively). The interaction $(t_{4,a1};\ t_{4Rec,a2})$ involving a_1 disclosing its weight to a_2, brings the history into h_2. At this point, a_2 knows the weight of a_1 (i.e. $K_{a2}Weight_{a1}$), since a_1 has sent that attribute in $t_{4,a1}$, and a_2 has received in in $t_{4Rec,a2}$. At h_3, a_2 sends a_1's *Weight* to a_3 via interaction $(t_{4,a2}\ ;\ t_{4Rec,a3})$. Consequently, the knowledge model (or what other agents know) about a_1 is $K_{a2}Weight\ _{a1}$ and

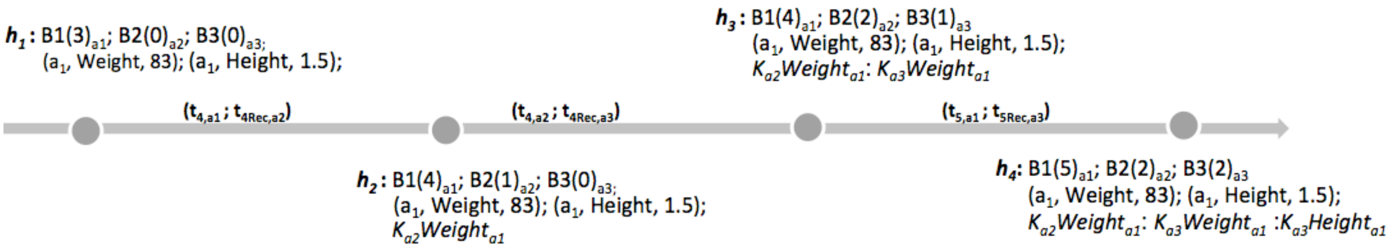
h1 : B1(3)a1; B2(0)a2; B3(0)a3;
(a1, Weight, 83); (a1, Height, 1.5);
(t4,a1 ; t4Rec,a2)
h2 : B1(4)a1; B2(1)a2; B3(0)a3;
(a1, Weight, 83); (a1, Height, 1.5);
Ka2Weighta1
h3 : B1(4)a1; B2(2)a2; B3(1)a3
(a1, Weight, 83); (a1, Height, 1.5);
Ka2Weighta1: Ka3Weighta1
(t4,a2 ; t4Rec,a3)
(t5,a1 ; t5Rec,a3)
h4: B1(5)a1; B2(2)a2; B3(2)a3
(a1, Weight, 83); (a1, Height, 1.5);
Ka2Weighta1: Ka3Weighta1 :Ka3Heighta1

Figure 5 Example of a history of a system involving interactions between agents A_1- A_3 where A_1 is the subject

$K_{a3}Weight_{a1}$. Finally, when the interaction ($t_{5,a1}$; $t_{5Rec,a3}$), involving a_1 disclosing its height to a_3 occurs, then $K_{a2}Weight_{a1}$, $K_{a3}Weight_{a1}$ and $K_{a3}Height_{a1}$ hold.

3) Discovering privacy threats from a history of the system: Given a history and a privacy requirement, a privacy threat is detected if the LTL formula in the THEN segment of the privacy requirement is not verified in the history. Note that only privacy requirements for which the IF segment matches the event associated with an incumbent interaction are considered.

The verification of a LTL formula on a finite history has been introduced in literature [18] [24]. In this research, we extend the semantics defined in [24] to introduce a three-valued logic. The semantics of our logic is given in Table 2. We assume the formula F on H is evaluated in the current step of history and denoted as H,i. Defined semantics for our logic has the following rationale: *I)* if the evaluation of a formula in the current instant i of the history of the system offers enough evidences that a formula F will be true (respectively false) in all the possible continuations of the history, then the formula is true (respectively false); *II)* otherwise the evaluation of the formula is inconclusive, and the formula will be *unknown*.

If the formula evaluated is true, then the privacy requirements are verified and the analysis detects no threat, in case the formula evaluated is false, then a privacy requirement is violated and the analysis consequently identifies a *privacy threat*. In case the formula evaluated is unknown then our analysis signals a *potential privacy threat*. This means that there are not enough evidences to conclude that the privacy requirements are violated. Such evidences can though be present in the following steps of the history. Consequently our analysis derives a new instance of the formula, scaled of one step in the future, and tries and verifies the new instance in the next history step.

Consider the history shown in Figure 5, also assuming the interactions that generated h_1 and h_2 have occurred. Then at h_3, PR_2 is active since its IF segment matches one of the events in the interaction ($t_{4,a1}$; $t_{4Rec,a2}$). The formula in the THEN segment is consequently evaluated in h_3. For the formula to be verified, the predicate $\neg K_{Receiver}BMI_{a1}$ should hold in every previous and future steps of the history. Furthermore, based on the inference rule $BMI = Height \cup Weight$, for $\neg K_{Receiver}\, BMI_{a1}$ to hold, then $\neg K_{Receiver}\ Weight_{a1}$ or $\neg K_{Receiver}Height_{a1}$ should hold, where *Receiver* identifies a generic agent receiving the information. If we assume that the interaction ($t_{5,a1}$;$t_{5Rec,a3}$) has not yet occurred, the predicate $K_{a3}Weight_{a1}$ holds at h_3. The information contained in the history up to h_3 is not sufficient to demonstrate that PR_2 will not hold in all the possible continuations of H, so our analysis marks it as unknown and signals a potential threat. The analysis will also continue to check the formula in the future, until further evidences can bring it to a conclusion. Those evidences are provided in h_4, where $K_{a3}Height_{a1}$ holds. In that step there are enough evidences to conclude that the condition does not hold in any possible continuation of H, and the analysis will consequently signal a *privacy threat*.

Table 2 LTL semantics on the system history H

Expression	*Semantics*
$H,i \vDash F$	F holds at the step i of H
$H, i \vDash LastTime\ F$	F holds at the step i-1 of H
$H, i \vDash \Diamond_p F$	$\exists\ 1 \leq j \leq i$ and H, $j \vDash F$
$H, i \vDash \Box_p F$	$\forall\ 1 \leq j \leq i$ and H,$j \vDash F$
$H, i \vDash Next\ F$	*Unknown* and *Check* H, i+1 $\vDash F$
$H, i \vDash \Diamond_f F$	True if $H, i \vDash F$. Otherwise *Unknown* and Check H, i+1 $\vDash \Diamond_f F$
$H, i \vDash \Box_f F$	False if $H, i \nvDash F$. Otherwise *Unknown* and Check H, i+1 $\vDash \Box_f F$
$H, i \vDash \Box F$	False if $(H, i \nvDash F) \vee \exists\ (1 \leq j < i$ and H, $j \nvDash F)$ Otherwise *Unknown* and Check H, i+1 $\vDash \Box F$

C. Determining Utility of Disclosure

When a privacy threat is discovered, the decision to disclose or not disclose needs to be made. A vital input into this decision making process is the insight on the severity of the discovered threat and the benefit of disclosure amidst the discovered threat. The utility of disclosure is then the difference between benefit of disclosure and the severity of discovered privacy threat. In this research, we determine the utility of disclosure using the properties of the network generated from the history of the system. For example, the shaded section of Figure 6 is the network generated using the history of the system for a_1-a_3 shown in Figure 5. Here, an undirected and unweighted network is assumed. In this network, nodes are agents, while a link between nodes is a relation formed between an information sender and a receiver. Specifically, we determine the benefit of disclosure based on the *clustering coefficient* of the generated network, while the severity of the threat is determined based on the *degree centrality* of receiving agent.

The clustering coefficient of an agent in a network is the ratio of the number of actual links between the agent's neighbours and the possible number of links. The overall clustering coefficient of the network is then an average of the clustering coefficients of each agent in the network. Clustering coefficient typically describes the concentration of the neighbourhood of an agent in the network. Such concentration has been used as an indicator to show the extent to which an agent shares common properties and/or plays similar roles with other agents in the network [21]. For example, assuming a_1, a_2 and a_3 have interacted with other agents as demonstrated in Figure 6. The dotted link illustrates a scenario where the agent a_1 is to disclose the attribute x to a_{R1} or a_{R2}. The clustering

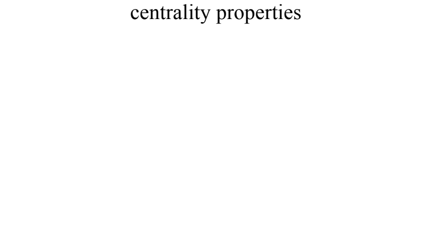

Figure 6 a_1 disclosing x to a_{R1} or a_{R2} having different clustering and centrality properties coefficient of a_1 if the attribute x is sent to the receiver $a_{R1}$is 0.5. Alternatively, the clustering coefficient if sent to $a_{R2}$is 0.3. Thus, using the clustering coefficient as a measure of benefit for the network in Figure 6, it will therefore be more beneficial if a_1 discloses x to a_{R1} compared to a_{R2}.

The degree centrality of an agent describes the number of direct links that an agent has with other agents. Generally, it has been shown that an agent with higher degree centrality can gain access to and/or influence over others. Such agents can also serve as a source or conduit for larger volumes of information exchange with other agents [21]. Thus, a receiver with higher degree centrality stands a greater chance of disseminating inappropriately disclosed information to more agents, hence representing a higher threat severity. Using the illustration in Figure 6, the degree centrality for a_{R1} is 0.25 and hence it represents a lesser threat severity, compared to a_{R2} which is 0.5. Given that a potential privacy threat is distinguished from a privacy threat that can be proven from existing history, a damping factor is applied to the former. Such factor ensures that a potential privacy threat does not have equal measure of severity compared to a realised one.

Overall, the utility of disclosure is then difference between benefit and threat severity. The utility of disclosure if a_1 discloses the attribute x to a_{R1} is 0.25 while that of $a_{R2}$is -0.2. Thus, depending on a utility value, users or systems may consider disclosing or withholding specific information. In this manner, they are either forfeiting or reinforcing their privacy when engaging in some interactions.

VI. Evaluation of PAR in Disclosure Decision Making

We conducted an evaluation to demonstrate if PAR can be a useful input to the planning part of adaptive privacy. One way of evaluating this outcome is to use specific adaptation strategies that regulate an acceptable utility, threat severity or benefit thresholds for a subject. The effectiveness of these strategies can then be checked for scenarios where the failure/satisfaction of PAR are realised.

Table 3 illustrates the set of strategies used in this study. *Case*$_1$ represents an agent that does not satisfy PAR. For this case, an agent initiates privacy management without an understanding of threat severity, benefit or the ultimate utility of disclosure. *Case*$_2$ represents an agent satisfying PAR and triggering the adaptation action that terminates all interactions with other agents once the utility reaches the value 0. At this point, for |H| >0, privacy threat severity (*TS*) equals benefit. Practically, *Case*$_2$ represents an agent disassociating itself from a group objective because the benefit does not exceed the privacy threat resulting from information disclosure. For *Case*$_3$, the agent satisfies PAR and triggers the adaptation action similar to *Case*$_2$ once *TS* is greater than a specific threshold (Th_i). Practically, *Case*$_3$ represents an agent disassociating itself from a group objective irrespective of the benefit derived, because a specific threat severity level is reached. Finally, *Case*$_3$ is similar to *Case*$_2$, but the adaptation action triggered is not to disclose information to other agents with specific properties. For this evaluation, the property we examine is the number of neighbours (n) of the receiving agent. Practically, *Case*$_4$ is a strategy that does not stop the increase in threat severity, but curbs the rate at which the increase occurs, while still deriving some utility.

Table 3 Adaptation triggers and disclosure decision

	Trigger	*Action*
Case$_1$	No-PAR	-
Case$_2$	Utility = 0	Terminate all disclosure
Case$_3$	$TS > Th_i$	Terminate all disclosure
Case$_4$	Utility = 0	Terminate only messages to receiver with more than n neighbours in the network

Based on the four highlighted cases, we conduct an experimental study that evaluates the following research questions: ***RQ***$_1$**:** What is the difference between an agent that does not satisfy PAR (i.e. No-PAR) and an agent that does (i.e. With PAR)? To address this question, we consider *Case*$_1$ (No-PAR) and *Case*$_2$ (with PAR). ***RQ***$_2$**:** Is there any advantage that *Case*$_3$ has over *Case*$_4$ or vice-versa. Considering that *Case*$_3$ and *Case*$_4$ manipulate on varying *Th* and *n* values, the aim is to understand the impact of these variations on utility. Finally, ***RQ***$_3$ investigates the impact that the number of agents in a group has on the utilities of *Case*$_3$ and *Case*$_4$.

A. Experimental Setup

In this evaluation we used Netlogo [23] (a programmable modelling environment for simulating natural and social phenomena) to simulate interactions across nine groups of agents. The number of agents in each group ranged from 10-250. This choice was inspired by studies on the average number of links that an agent has with other agents in typical group networks where agents are human [25]. The behaviour of each agent is a variant of the behavioural model shown in Figure 3. During the simulation, multiple interactions can occur over a single link. Thus the actual number of links is less than the total number of simulated interactions. Each group had a single subject with multiple senders and receivers. Thus, every interaction either involved a subject sending information about itself, or another agent information sending information about the subject. Also, the simulation of interactions followed the power-law distribution, which is typical in group networks where a small number of agents have very large number of links. Our simulation of interactions and subsequent networks were tailored to closely resemble mobility based networks where links arise mainly from spatial or temporal proximity of agents.

For each group, we associated PR_2 (Section IV.C) to the selected subject. We assumed a conservative scenario where every interaction resulted in a privacy threat. Furthermore, we fixed *Th* = 0.12, and evaluated *Case*$_3$ for *Th*, *Th**2, *Th**4, *Th**6, *Th**8 and *Th**10 respectively. Similarly, we fixed *n* =8, and evaluated *Case*$_4$ for *n, n*+2, *n*+4, *n*+6, *n*+8 and *n*+10 respectively.

B. Findings and Lessons Learned

The topology metrics for each group network based on the above experimental setup are shown in Table 4. $TS_{no\text{-}PAR}$ refers to privacy threat severity where PAR are not satisfied. The 2nd and 3rd columns show that the number of agents in a group increases with increasing links between agents. Similarly, the number of interactions required for privacy threat severity to reach 1 for a scenario where PAR are not satisfied, increases with the number of agents. This is because given the power-law and increasing the number of agents in a group, one would expect a decrease in the degree centrality of most receiving agents, and hence in the associated privacy threat severity. In the remaining of this subsection we used this result to address each of the designated research questions. While the results generated for the different groups were related, for space limitations, we show results for only group 7 in RQ_1 and RQ_2. In RQ_3, we then demonstrate the impact of the number of agents in a group on the effectiveness of adaptation actions.

Table 4 Simulated network topology metrics

Group	*No. gents*	*No. links*	*No. Interactions* ($TS_{noPAR} = 1$)
1	10	29	58
2	25	104	238
3	50	282	697
4	75	521	1339
5	100	788	2065
6	125	1095	2911
7	150	1441	3874
8	200	2189	5968
9	250	3170	8611

1) RQ_1 findings: Figure 7 shows the plots of threat severity, benefit and utility for $Case_1$ and $Case_2$ for a subject in Group 7. In both cases, it can be seen that the benefit of information disclosure reaches a tipping point and gradually decreases thereafter. This outcome is explained considering that the more a subject engages in interactions and form new links with other agents, the less likely the neighbours of the subject will have links to each other. This results in a lower benefit measure. In contrast, given No-PAR for $Case_1$, the threat severity continues to increase, with a continuous decrease in the utility. This continuous increase in severity results from the receiving agent having more neighbours with increasing interactions. For $Case_2$ with PAR, as shown in Figure 7, the subject is able to curb the continuous increase in threat severity.

In summary, the core distinction between an agent with PAR and No-PAR is that agents with PAR can regulate information disclosure. Such regulation is based on tolerable levels of threat severity or on the minimum expected utility. For $Case_2$, this has been achieved by terminating all subsequent interactions at the point where utility reaches 0. However, terminating all subsequent interactions can be viewed as an extreme risk-averting behaviour which can hinder agents from reaping the benefit of disclosure. Thus, a more appropriate scenario falls somewhere between extreme cases of risk-taking and averting. We investigate these scenarios in *RQ2* with varying values of *Th* and *n*.

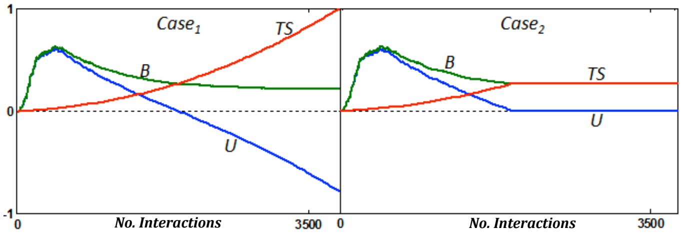

Figure 7- A plot of threat severity (TS), Benefit (B - Clustering Coeff.) and Utility (U) vs history of system (h_i) for agents=150.

2) RQ_2 findings: Figure 8 illustrates the impact of a subject terminating information disclosure to receiving agents with *n* neighbours in a group ($Case_4$). It is noticed here that increasing *n* results in tending utility towards what we observed in the No-PAR case. Again, this outcome can be explained by the assumption of power-law distribution where a smaller number of receiving agents will have high *n*. Thus, at higher *n* values this adaptation action is less effective. The outcome for $Case_3$ involving a different adaptation action is depicted in Figure 9. Here, all interactions are terminated when *TS* is more than a specified *Th*. For this case, the subject does not need to wait until utility = 0. As a result, the utility curves associated with the two lower thresholds (Th_1 and Th_2) outperform the utility of $Case_2$. Similar to $Case_{4,}$ as *Th* values increase, the utility for the subject tends towards the No-PAR utility.

To achieve a better understanding of the statistical significance of the results generated for varying *Th* and *n* values, we ran a non-parametric ANOVA test. The choice of non-parametric ANOVA was because the utility data of different cases did not have Normal distribution. Figure 10 shows the Box plots for the different utility function of $Case_1$ - $Case_4$. The Kruskal-Wallis test showed significant difference between these cases, but pair-wise comparison revealed that the three higher *Th* values for $Case_3$ (i.e., Th_4, Th_5 and Th_6) are not statistically different neither from No-PAR nor from each other. Conversely, Th_1, Th_2 and Th_3 are statistically different from No-PAR and each other. This outcome suggests that for a risk-averting behaviour, Th_1 is better than Th_2 and Th_3. Conversely, for a risk-taking behaviour, Th_3 is the limit

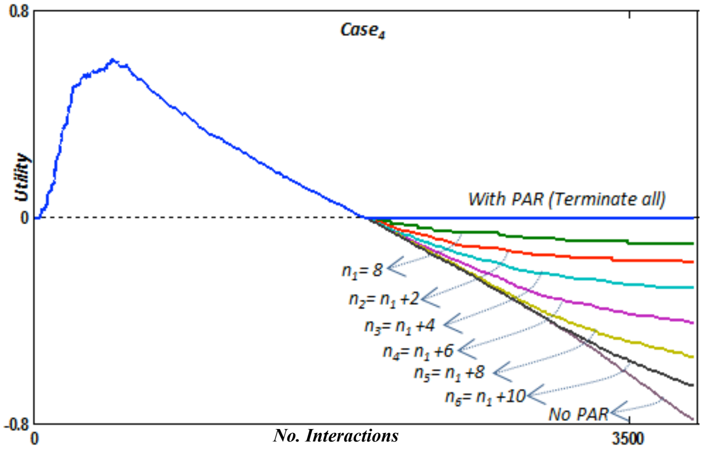

Figure 8- Utility of terminating disclosure to agents with different number of links (*n*) – agents=150.

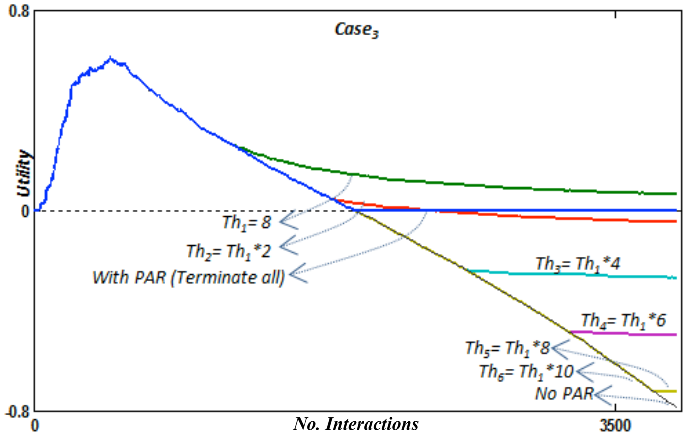

Figure 9- Utility of terminating disclosure to all agents when threat severity is more than a certain threshold (*Th*)

(compared to Th_4, Th_5 and Th_6) for which if applied, then some benefit can be derived without being similar to No-PAR.

Again, the two higher *n* values for $Case_4$ (i.e., n_5 and n_6) are not significantly different from neither No-PAR nor from each other. Conversely, $n_1 - n_4$ are statistically different from No-PAR as well as each other. This outcome suggests that for a risk-averting behaviour, n_1 is better than $n_2 - n_4$. Conversely, for a risk-taking behaviour, n_4 is the limit (compared to n_5 and n_6) for which if applied, then some benefit can be derived without reaching the No-PAR utility.

In summary, for a risk-averting behaviour, it can be said that lower values of *Th or n* are better than higher values. In contrast, for a risk-taking behaviour, the intension is for a subject to accommodate the decline in the utility and the increase in threat severity in order to leverage on the benefit. Then there is a limit to which such a subject can risk information disclosure, over which it is as good as the subject not satisfying PAR.

RQ_3 findings: Table 5 shows a comparison of $Case_3$ and $Case_4$ against No-PAR for different groups considered in this study. This table illustrates that for groups 1, 2 and 3 consisting of 10, 25 and 50 agents respectively, none of the adaptation actions in $Case_4$ was significantly different from the No-PAR. From groups 4 to 9, the outcome showed a different trend of decreasing numbers of *n* that was not significantly different from the No-PAR case. The rationale for this outcome is derived from the view that for a specific *n* value, and as the number of agents in a group increases, there is also an increased likelihood of the number of receiving agents whose neighbours would be more than *n*. Conversely, for a smaller group of agents (groups 1, 2 and 3), smaller values of *n* are required to achieve a statistically different utility. A pattern similar to $Case_4$ is also roughly observable for $Case_3$ that involves varying *Th* values across groups. As such, for groups with smaller number of agents, smaller values of *Th* are required to achieve a statistically different utility.

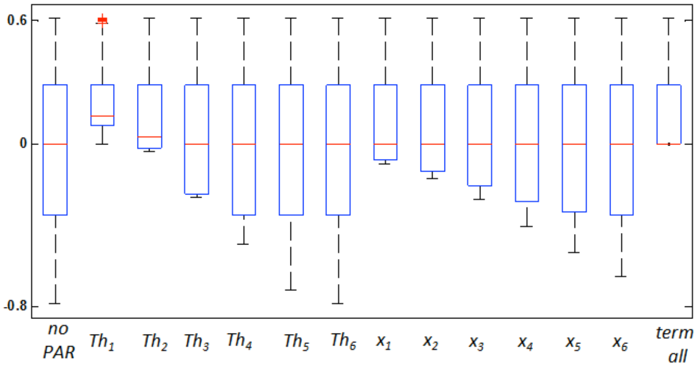

Figure 10- Box plots of different *Th* and *n* values - agents=150

Table 5 $Case_3$ and $Case_4$ vs no-PAR for different groups

	Not significantly different from No PAR	
Group	*$Case_4$*	*$Case_3$*
1	$n_1, n_2, n_3, n_4, n_5, n_6$	Th_3, Th_4, Th_5, Th_6
2	$n_1, n_2, n_3, n_4, n_5, n_6$	Th_3, Th_4, Th_5, Th_6
3	$n_1, n_2, n_3, n_4, n_5, n_6$	Th_3, Th_4, Th_5, Th_6
4	n_2, n_3, n_4, n_5, n_6	Th_4, Th_5, Th_6
5	n_3, n_4, n_5, n_6	Th_4, Th_5, Th_6
6	n_4, n_5, n_6	Th_4, Th_5, Th_6
7	n_5, n_6	Th_4, Th_5, Th_6
8	n_6	Th_4, Th_5, Th_6
9	-	Th_4, Th_5

The key observation is that *n* and *Th* are mutable factor that change depending on the number of agents in the group. As the number of agents in a group increases, the resulting utility of an adaptation action for a specific *n* and *Th* also become more significantly different from No-PAR.

VII. Conclusion and Further Work

In this paper, we presented an adaptive privacy framework that enables the runtime selective disclosure of personal information. Our approach is based on the rationale that for the appropriate disclosure of information from a sender to a receiver, some privacy awareness requirements (PAR) need to be satisfied. Such requirements underpin the ability of applications to identify the attributes to monitor in order to detect privacy threats, the discovery of a privacy threat before information is disclosed, and an understanding of the utility of disclosure, which includes the severity of the threat as well as the benefit of disclosure in the face of the discovered threat.

We evaluated our framework from two viewpoints. First, we showed that applications that fail to satisfy PAR are unable to regulate information flow based on the utility of disclosure. Secondly, we showed that applications that satisfy PAR can regulate the disclosure of information. We demonstrated the usefulness of PAR that crosscuts a spectrum, where at one end is risk-aversion (with the aim of user applications minimising exposure to privacy threats), and the other end is risk-taking (with the objective of maximising benefit amidst increasing threat severity and declining utility). Although we used a single motivating example to evaluate our approach, we suggest that our approach is generalisable to other domain where disclosure can be modelled as the transfer of information between agents.

The key benefit of our approach to engineering adaptive software is that PAR can serve as useful input into the planning and execution phases of the adaptation cycle. PAR are useful for planning as it provides an understanding of the utility of information disclosure. We also expect PAR to be useful for execution. This is because it provides a rationale for software models that need to change in-order to preserve privacy. While we have not focused on the semantics of such change, we expect it to involve an adaptation manager carrying out some actions. These include altering or removing disclosure behaviour by updating the LTS representation of the application. Other execution approaches may include refining user privacy requirements, or learning new inference rules in the context model to subsequently enable better adaptation. Another benefit of our approach to software engineering is that it relies on a minimal subset of general software engineering models. For example, there is no need to explicitly model a malicious user in order to discover privacy threats.

Generated interaction networks used in the evaluation of our approach closely resemble random networks that are typical of mobile applications. Further work will be required to generalise our approach to other forms of networks such as the small-world or networks where the power-law distribution is not assumed. Furthermore, our framework is extensible to utilise richer models of context. Thus, we aim to investigate other models that addresses possible uncertainties that can be introduced by mobility and changing context. The wider question that our framework poses is that of engineering existing legacy application into an adaptive privacy protecting one. In future work, we plan to investigate an aspect-oriented and component based approach to adaptive privacy that is amenable to legacy systems. Finally, we intend to investigate the notion of entropy and transience in adaptive privacy. This is necessary because the sensitivity of information may decay over time for a number of reasons. These include the transient nature of the knowledge of human agents, and the disclosed information becoming out of context or more inaccessible over time.

Acknowledgment

This work was supported, in part, by SFI grant 10/CE/I1855 (for CSET2) to Lero, Microsoft SEIF Award (2011) and the European Research Council.

References

[1] M. Langheinrich., A Privacy Awareness System for Ubiquitous Computing Environments, Ubiquitous Comp., Lec. Notes in Comp. Sc. G. Borriello and L. Holmquist, eds., 315-320, 2002.

[2] S. Lahlou, M. Langheinrich and C. Röcker, Privacy and trust issues with invisible computers, Com. ACM, 48(3), 59-60, 2005.

[3] T. Tun, A. Bandara, B. Price, Y. Yu, C. Haley, I. Omoronyia and B. Nuseibeh, Privacy Arguments: Analysing Selective Disclosure Requirements for Mobile Applications. RE'12., 2012

[4] I. Omoronyia, L. Pasquale, M. Salehie, L. Cavallaro, G. Doherty and B. Nuseibeh, Caprice: A tool for engineering adaptive privacy, Tool demo track, ASE 2012

[5] V. Silva Souza, A. Lapouchnian, W. N. Robinson, and J. Mylopoulos. Awareness requirements for adaptive systems. In *SEAMS*, 60–69, 2011.

[6] S. Spiekermann, and L. Cranor, Engineering Privacy, Sof. Eng., IEEE Trans on, 35(1) 67-82, 2009.

[7] V. Silva Souza. A requirements-based approach for the design of adaptive systems. In ICSE, 1635–1637, 2012.

[8] J. Kephart and D. Chess. The vision of autonomic computing. Computer, 36(1):41–50, 2003.

[9] F. Schaub, B. Könings, M. Weber, and F. Kargl. Towards context adaptive privacy decisions in ubiquitous computing, PERCOM Workshops, 2012.

[10] L. Palen and P. Dourish. Unpacking privacy for a networked world. In Proceedings of the SIGCHI, pages 129–136. 2003.

[11] C. Braghin, A. Cortesi, and R. Focardi. Information flow security in boundary ambients. Info. and Comp., 206(2):460–489, 2008.

[12] C. Kalloniatis, E. Kavakli, and S. Gritzalis. Addressing privacy requirements in system design: the PriS method. Requirements Engineering, 13(3):241–255, 2008.

[13] L. Liu, E. Yu, and J. Mylopoulos. Security and privacy requirements analysis within a social setting. In RE, 151–161, 2003.

[14] A. Barth, A. Datta, J.C. Mitchell, and H. Nissenbaum. Privacy and contextual integrity: Framework and applications. In IEEE Symposium on Security and Privacy, 15–pp., 2006.

[15] S. Pötzsch, Privacy Awareness: A Means to Solve the Privacy Paradox?, The Future of Identity in the Info. Society, IFIP Advances in Inf. and Commun Tech. V. Matyáš, S. Fischer-Hübner, D. Cvrcek et al., eds., 226-236: Springer Boston, 2009.

[16] C.E. Wills and M. Zeljkovic. A personalized approach to web privacy: awareness, attitudes and actions. Information Management & Computer Security, 19(1):53–73, 2011.

[17] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, A survey of context modelling and reasoning techniques, Pervasive and Mobile Computing, 6(2):161-180, 2010.

[18] A. Pnueli, The temporal logic of programs. FOCS, 1977, 46–57.

[19] R. De Landtsheer and A. van Lamsweerde.. Reasoning about confidentiality at requirements engineering time. In ESEC/FSE-13., 41-49. 2005

[20] R. Fagin, J. Halpern, Y. Moses, and M. Vardi, Reasoning about Knowledge. MIT Press, 1995.

[21] M. Newman, Networks: An Introduction. 2010.

[22] B. Newell. Rethinking Reasonable Expectations of Privacy in Online Social Networks. Richmond Journal of Law and Technology. 16, 1–61, 2011.

[23] M. Dickerson, Multi-agent simulation and NetLogo in the introductory computer science curriculum. Journal of Computing Sciences in Colleges, 27(1), 2011.

[24] H. Barringer, A. Goldberg, K. Havelund, and K. Sen. Rule-based runtime verification. In Verification, Model Checking, and Abstract Interpretation, 2004, 277-306.

[25] J. Ugander, B. Karrer, L. backstrom, and C. Marlow, The anatomy of the Facebook Social Graph. In Proceedings of CoRR, 2011.