TRAAC: Trust and Risk Aware Access Control

Chris Burnett, Liang Chen, Peter Edwards and Timothy J. Norman
Department of Computing Science, University of Aberdeen, UK
{cburnett, l.chen, p.edwards, t.j.norman}@abdn.ac.uk

***Abstract*—Systems for allowing users to manage access to their personal data are important for a wide variety of applications including healthcare, where authorised individuals may need to share information in ways that the owner had not anticipated. Simply denying access in unknown cases may hamper critical decisions and affect service delivery. Rather, decisions can be made considering the risk of a given sharing request, and the trustworthiness of the requester. We propose a trust- and risk-aware access control mechanism (TRAAC) and a sparse zone-based policy model, which together allow decision-making on the basis of the requester's trustworthiness with regards to both the information to be shared, and the completion of obligations designed to mitigate risk. We formalise our approach and compare it with an existing approach that does not model trust through simulation.**

I. Introduction

As the volume and richness of personal data grows, so do the possibilities for its exploitation, both for legitimate and malicious purposes. From the security and privacy perspective, there is a need for mechanisms to allow individuals to self-manage access to their personal data [1]. Many fine-grained authorisation models have been proposed for the protection of personal data, the most important being those based on roles [2], relationships [3] and general attributes [4].

In this paper, we propose a novel access control approach which is initially risk-taking and permissive. This approach is applicable to domains with the following characteristics:

- **Policy coverage:** it is difficult to define policies which cover all the possible circumstances in which access should be granted or denied.
- **Dynamicity:** the context within which access control decisions are made changes with such frequency that static policies become quickly outdated.
- **Denial risk:** inappropriately denying an access request that should have been granted (due to inadequate policy coverage) could lead to potentially harmful consequences.
- **Delegation:** it may be necessary for data owners to delegate the ability to share information on their behalf.

The healthcare domain exhibits all of these characteristics [5]. The information requirements of various roles involved in a patient's care are complex and subject to change, as are the individuals playing those roles (policy coverage, dynamicity). As a result, it is difficult to define policies which adequately cover all possibilities. In addition, overly strict or cautious policies could prevent information from being available when it is needed (i.e. denial risk). Finally, patients will not typically be able to anticipate the ways in which their information may need to be shared, and so must delegate the ability to share their data to individuals trusted to make those decisions on their behalf. Here, notions of trust and risk have significant roles to play. Not all domains have these characteristics, however, social networks, for example, are dynamic but rarely involve a denial risk, and it is usually possible to specify simple policies which cover all the possible information sharing actions.

We will use the following scenario to illustrate our approach. A patient, named Alice, is suffering from chronic depression and is taking part in an intervention which involves her doctor, Bob, and pharmacist, Charlie, with the aim of promoting patient compliance with therapies including drugs and exercise. To monitor compliance, and the effectiveness of the programme, Alice collects activity data on her smartphone, based on GPS traces. Her smartphone also records a mood diary and data about her sleeping patterns, using accelerometer data. This is then used by Bob and Charlie to make decisions about Alice's care. In some cases, Bob may need to share data with other healthcare practitioners. Alice is aware of this, and trusts Bob to make sharing decisions on her behalf, but does not know about the potential recipients of her data, or the conditions under which it might be shared.

Our contribution is twofold. Firstly, we present a zoned policy model for describing sparse privacy policies. Secondly, we present a model of trust-based permissive access control, and show how established trust assessment methods can be used to both compute the risks of information disclosure dynamically, and to mediate the effectiveness of strategies for mitigating those risks. Prior to formalising and evaluating our model of trust- and risk-aware access control (RAAC), we provide a brief overview of the relevant background research into RAAC and trust assessment.

II. Background

A. Risk-aware access control

Risk-aware access control (RAAC) is a novel access control paradigm that was proposed to address the increasing need to securely share information in dynamic environments [6]. RAAC introduces an authorisation decision function that makes decisions based on dynamic assessments of how much risk is incurred by allowing some access, which could lead to information disclosure or modification. This risk-aware approach is more flexible and adaptive than traditional policy-based approaches, which use fixed, pre-defined authorisation policies to make decisions, and do not consider contextual risk.

RAAC is a permissive authorisation model in the sense that some risky or exceptional access requests are allowed,

provided that the risk of permitting access is below a risk threshold. *Risk mitigation methods* can be employed to bring perceived risk within acceptable thresholds. These are generally obligatory actions that are imposed on the requesting user or the system itself, to be completed either before or after the request is granted. In this paper, we will consider obligations to be completed *after* an access is granted; as we will show, this introduces a need for trust.

In summary, there are two key steps involved in developing a RAAC model: risk conceptualisation and risk mitigation. Concerning the former, researchers have attempted to incorporate the concept of risk into existing authorisation models such as multi-level security [7], [8] and RBAC [9], [10], or to quantify the risk for supporting authorisation decision-making in a specific domain (e.g. clinical information systems [11]).

The authorisation decision-making mechanism proposed by Dimmock [10] conducts risk analysis for all the possible outcomes from granting an access request. This risk analysis considers the trustworthiness of the requester, the cost of potential outcomes, and pre-defined risk-based security policies. However, this approach does not provide a means for learning or updating trust beliefs. Ni *et al.* [8] make use of fuzzy inference techniques to estimate risk for access requests. The authors explore a number of appropriate fuzzy operations that can be used in the inference process to make risk-aware authorisation decisions. Wang and Jin [11] propose an approach to computing risk based on the "need-to-know" principle. Their approach requires a *relevance function* which rates requests as high risk if the requested information is irrelevant for the specified purpose.

With regard to risk mitigation, a counterpart to risk assessment, there exists some work in RAAC [7], [9] that provide sophisticated treatment of risk mitigation actions that are required to be performed by the system after risky access is granted. However, to our knowledge, little work has focused on the use of user obligations as risk mitigation methods with exception of Chen *et al.* [12]. This work explored some simple incentive mechanisms (specifically, the enforcement of *risk budgets*) for users to fulfil obligations in order to control and account for the risk incurred by granting access. However, none of the above approaches consider that users may not be reliable in fulfilling their obligations or sharing information appropriately. In this paper, we build upon this in proposing a novel approach to computing and mitigating risk based on trust assessment.

B. Trust assessment

Trust assessment mechanisms are used to help determine how trustworthy some potential partners are with respect to some *issue* in a given context. While trust can be defined in many ways, we define it as the *subjective probability* with which an individual believes another will behave as expected when some issue is delegated. By allowing a requester to access or share some data, an owner is essentially *delegating* to the requester the task of adhering to his privacy preferences, and exposing himself to the risk that the requester will not behave appropriately.

For our purposes, probabilistic computational trust models [13] are appropriate. While the approach presented here can be used with any probabilistic trust model, we adopt Jøsang's widely used Subjective Logic model [14], which is well known, and based on Bayesian principles. To facilitate a clearer discussion of our approach, we present a brief overview of this model here.

1) Opinion representation: An opinion held by an individual w about another individual u regarding issue i is represented by a tuple:

$$\omega^w_{u:i} = \langle b^w_{u:i}, d^w_{u:i}, u^w_{u:i}, a^w_{u:i} \rangle \quad \text{where } b^w_{u:i} + d^w_{u:i} + u^w_{u:i} = 1, \quad \text{and } a^w_{u:i} \in [0,1]. \tag{1}$$

Here, $b^w_{u:i}, d^w_{u:i}$ and $u^w_{u:i}$ respectively represent the degrees of *belief*, *disbelief*, and *uncertainty* regarding the proposition that u will behave as we expect with regard to some issue. The base rate parameter $a^w_{u:i}$ represents the *a priori* degree of trust w has about u, before any evidence has been received.

2) Forming and updating opinions: Opinions are formed and updated based on observations of past performance using two parameters $r^w_{u:i}$ and $s^w_{u:i}$, capturing, respectively, the number of positive and negative experiences observed by w about u with respect to i. With these two parameters, w's opinion about u is computed as follows.

$$\begin{aligned} b^w_{u:i} &= \frac{r^w_{u:i}}{(r^w_{u:i} + s^w_{u:i} + 2)} \\ d^w_{u:i} &= \frac{s^w_{u:i}}{(r^w_{u:i} + s^w_{u:i} + 2)} \\ u^w_{u:i} &= \frac{2}{(r^w_{u:i} + s^w_{u:i} + 2)} \end{aligned} \tag{2}$$

Therefore, for an initial opinion with no evidence, $b^w_{u:i} = 0$, $d^w_{u:i} = 0$, $u^w_{u:i} = 1$ and $a^w_{u:i} = 0.5$.

3) Trust ratings: Given an opinion computed through Equation 2, a single-valued *trust rating*, which can be used to rank and compare individuals, can be obtained as follows.

$$P(\omega^w_{u:i}) = b^w_{u:i} + a^w_{u:i} \cdot u^w_{u:i} \tag{3}$$

We write $P(\omega^w_{u:i} | r^w_{u:i}, s^w_{u:i}, a^w_{u:i})$ to denote a rating that has been derived from r and s evidence parameters and an *a priori* rating $a^w_{u:i}$ using Equations 2 and 3.

III. Trust- and risk-aware access control

We will now formalise our model for trust and risk-aware access control (TRAAC).

A. A zoned policy model

We assume the existence of a set of objects O and a set of users U. An object $o \in O$ represents a piece of data to which access is controlled by the system. Each object o is owned by exactly one owner $w \in U$. Let $O_w \subseteq O$ denote the set of objects owned by user w. We associate each object with a *policy* $\pi_o^w : V \rightarrow \{\mathsf{share}, \mathsf{read}_u, \mathsf{read}_s, \mathsf{deny}, \mathsf{undefined}\}$, where $V = U \setminus \{w\}$. This function divides users seeking to access o into zones, where a user can only be in one zone at a time. The owner w of o initialises the policy for o by placing users into share, read_u, and deny zones. Let Π^w denote the set of all policies belonging to a user w.

Users in the deny zone are not allowed to read o. Users in the read_u and read_s zones can, where read_u contains users who have been explicitly granted read access by the data owner. Users in the share zone can read o, and share o with others. The read_s contains users who can read o because of a sharing action performed by another individual from the share zone. Users in the undefined zone are those for whom no zone has been explicitly defined by w. The owner w of an object is a special case; w can add or remove individuals from any zone, for any object that she owns. For convenience, we may use read to refer to users who are in either read_u or read_s.

We consider two kinds of requests. Firstly, *read* requests, defined as $read(u, o)$, denote a request of some user u to access an object o. Our above policy model is used to evaluate read requests. Formally, a request $read(u, o)$ is granted if $\pi_o^w(u) \in \{\mathsf{read}, \mathsf{share}\}$ and denied otherwise (i.e. $\pi_o^w(u) \in \{\mathsf{undefined}, \mathsf{deny}\}$).

During the system's operation, the owner's initial policy with respect to a particular object is subject to change, as a result of sharing actions requested by users in its share zone. Such *share* requests are defined as $share(u, v, o)$, and denote the request of a user u to share o with another user v. The effect of successfully granting $share(u, v, o)$ is to move v into the read_s zone for o. We assume that the policies of a data owner are visible to requesters. However, in reality, requesters may not be aware of all of an owner's policies.

The presence of the undefined zone means that some sharing actions have no explicit decision defined for them in advance. In this case, the system must consider its *trust* in the requester, and the impact of this on the *risk* of the request. We now discuss how these values may be computed to support decision-making about undefined zone sharing requests.

B. Computing trust

We consider trust in another with respect to some *issue*. Here, we consider two kinds of issues when evaluating sharing requests, and so two kinds of trust:

- **Sharing trust (*ST*):** trust that a requester will share information appropriately and not violate prohibitions;
- **Obligation trust (*OT*):** trust that a requester will complete obligations assigned to him.

We use ST to compute the risk associated with allowing a user to share some information with another. We use OT to mediate the effects of risk mitigation strategies, which will be discussed in Section III-D.

1) Updating sharing trust: In order to update trust, it is necessary to evaluate events and determine whether they constitute a trust violation or not. To do this, we introduce a history of the events that have occurred, where events of interest are instances of information sharing. Let H_{share} be a time-ordered sequence of sharing events, whose members are instances of *requested* $share(u, v, o)$ actions. We restrict a data owner w's view of the global history to only those elements regarding the data he owns, so that $H_{share}^w = \{share(u, v, o) \in H_{share} \mid o \in O_w\}$.

Each data owner w has an *evaluation function* for the sharing trust under consideration. $E_{ST}^w : H_{share}^w \rightarrow \{\mathsf{pos}, \mathsf{neg}, \mathsf{none}\}$ determines whether an observed sharing action $share(u, v, o)$ necessitates a positive or negative sharing trust update, or no update. When data is shared into the undefined zone, the trust model must make an assumption about whether this represents a trust violation or not, until such time as the data owner explicitly specifies a zone for the recipient. These assumptions can be user-defined for each object. We capture this via a function $A_w : O_w \rightarrow \{\mathsf{pos}, \mathsf{neg}, \mathsf{none}\}$ which specifies how to treat the sharing of an object into its undefined or read_s zones. This means an individual v who is in the read_s zone (i.e. some other individual has shared o with them, but this has not been explicitly approved by the data owner) is still considered as being in the undefined zone for the purposes of updating trust about some u who wishes to share o with v.

The behaviour of the E_{ST}^w function is defined as follows:

$$E_{ST}^w(share(u, v, o)) = \begin{cases} \mathsf{pos} & \text{if } \pi_o^w(v) \in \{\mathsf{share}, \mathsf{read}_u\} \\ \mathsf{neg} & \text{if } \pi_o^w(v) \in \{\mathsf{deny}\} \\ A_w(o) & \text{otherwise} \end{cases} \tag{4}$$

Positive trust updates should occur when a user behaves in a way which is expected by the data owner, with regard to sharing actions in the undefined zone. Requests to share information with individuals in the share and read_u zones result in a positive update, as this represents explicitly allowed sharing which brings some benefit to the data owner. Also, the *state* of being in the share zone and refraining from sharing information which violates the policy should increase trust. To address this, we adopt an "innocent until proven guilty" approach, that is, individuals in the share zone who have not yet violated any policy should receive a positive update to reflect this. We do this by giving a positive trust update for each sharing zone an individual is in, where no violations have occurred. That is, $\delta_u^w = |\{\pi_o^w \mid \pi_o^w \in \Pi^w\}|$ such that $\pi_o^w(u) = \mathsf{share}$, and for all $share(u, v, o) \in H_{share}^w, \pi_o^w(v) \neq \mathsf{deny}$. Requests to share information with individuals in the deny zone can be considered a breach of trust, even though they are denied, and will result in a negative trust update.

Now, using a user's policy set Π^w, sharing history H_{share}^w and evaluation function E_{ST}^w, we can arrive at a trust value for a given user. Let $G(u, H_{share}) = \{share(u', v, o) \in$

$H_{share} \mid u' = u\}$ denote the elements of a history H_{share} involving u sharing some information with some recipient. Using Equation 4, the evidence parameters $r^w_{u:ST}$ and $s^w_{u:ST}$ for sharing trust can be computed as follows:

$$r^w_{u:ST} = |\{\rho \in G(u, H^w_{share}) \mid E^w_{ST}(\rho) = \mathsf{pos}\}| + \delta^w_u \quad (5)$$

$$s^w_{u:ST} = |\{\rho \in G(u, H^w_{share}) \mid E^w_{ST}(\rho) = \mathsf{neg}\}| \quad (6)$$

We then compute $P(\omega^w_{u:ST})$ using Equations 2 and 3.

2) Updating obligation trust: Let Obs denote the global set of all obligations. We define a function $f : Obs \to \{\mathsf{active}, \mathsf{satisfied}\}$ that indicates the current status (active or satisfied) for each obligation. Let H_{assign} be a time-ordered sequence of obligation assignment events, whose members are of form $assign(u, ob, o)$ and denote that a user u was assigned an obligation $ob \in Obs$, which was generated as part of a risk mitigation strategy for sharing some object o. We write $H^w_{assign} = \{assign(u, ob, o) \in H_{assign} \mid o \in O_w\}$ to denote parts of the global history relevant to the data w owns.

Similarly to sharing trust, $E^w_{OT} : H^w_{assign} \to \{\mathsf{pos}, \mathsf{neg}\}$ determines whether an obligation has been satisfied or not.

$$E^w_{OT}(assign(u, ob, o)) = \begin{cases} \mathsf{pos} & \text{if } f(ob) = \mathsf{satisfied} \\ \mathsf{neg} & \text{otherwise} \end{cases} \quad (7)$$

In contrast to sharing trust, we adopt an "guilty until proven innocent" approach for updating obligation trust, that is, users who accept an obligation after a sharing request was granted should receive a negative update until the obligation is satisfied. In this way, we address users who accept many obligations, but do not satisfy them. In practice, the f function could be extended in various ways to allow more fine-grained assessment of the state of obligations (e.g. deadlines).

Let $G(u, H_{assign}) = \{assign(u', ob, o) \in H_{assign} \mid u = u'\}$. Then we have:

$$\begin{aligned} r^w_{u:OT} &= \left|\{\rho \in G(u, H^w_{assign}) \mid E^w_{OT}(\rho) = \mathsf{pos}\}\right| \\ s^w_{u:OT} &= \left|\{\rho \in G(u, H^w_{assign}) \mid E^w_{OT}(\rho) = \mathsf{neg}\}\right| \end{aligned} \quad (8)$$

As before, we compute $P(\omega^w_{u:OT})$ using Equations 2 and 3.

C. Computing risk

We adopt a common decision-theoretic view of *risk* [15], and define it in terms of the expected loss in the event of unwanted disclosure after granting a sharing request. Generally, risk is defined as the product of the probabilities of undesirable events and the loss that would be sustained if those events occurred.

Loss can be difficult to define for non-economic domains. In healthcare, the loss in the case of information disclosure could include emotional damages (e.g. loss of partner, family problems, stigmatisation) as well as financial ones (e.g. loss of job, insurance premium increase). We adopt a proxy for loss based on the *sensitivity* of information, assuming that the more sensitive some information is perceived to be, the greater the subjective loss would be in the event of its disclosure [7]. Such categorisations of the sensitivity of information have been investigated in the healthcare domain to aid privacy management [1].

We define the *sensitivity categories* as a strict totally ordered set C. That is, for all $c, c' \in C$, either $c < c'$ or $c > c'$. Each user u has a mapping $m_u : O_u \to C$ which assigns a sensitivity category to each object. We write $\mathbb{R}_{[0,1]}$ for the set of real numbers between 0 and 1; that is, $\mathbb{R}_{[0,1]} = \{x \in \mathbb{R} \mid 0 \leqslant x \leqslant 1\}$. Let $L : C \to \mathbb{R}_{[0,1]}$, be a loss function which assigns a real loss value to the unwanted disclosure of information in each sensitivity category.

Given a request $r = share(u, v, o)$, where w is the owner of o, we can now define risk as:

$$Risk(r) = \begin{cases} 0 & \text{if } \pi^w_o(v) \in \{\mathsf{share}, \mathsf{read}_u\} \\ 1 & \text{if } \pi^w_o(v) \in \{\mathsf{deny}\} \\ (1 - P(\omega^w_{u:ST})) \cdot L(m_w(o)) + Risk_s & \text{otherwise} \end{cases} \quad (9)$$

It can be seen that the risk associated with request $share(u, v, o)$ is 0 if $\pi^w_o(v) \in \{\mathsf{share}, \mathsf{read}_u\}$. In contrast, if $\pi^w_o(v) \in \{\mathsf{deny}\}$, then the risk is 1. By the risk-aware authorisation function defined in Section III-D, request $share(u, v, o)$ will be allowed if its associated risk is 0, and will be denied when the risk equals to 1. However, if $\pi^w_o(v) \in \{\mathsf{undefined}, \mathsf{read}_s\}$, then the risk is determined by u' trustworthiness in sharing information and loss value associated with o. In addition, we assume that risks from other sources, such as inappropriate disclosure due to system security vulnerabilities, are accounted for elsewhere. Since these risks are not depend on the individual making the access request, we consider them as a given constant $Risk_s$ (system risk).

D. Checking sharing requests

Given the level of request risk, based on the trustworthiness of the requester and the sensitivity of the requested information, we can now define our mechanisms for checking information sharing requests.

1) Risk intervals: We assume the existence of a risk domain $\mathcal{D}$ that is defined to be $\mathbb{R}_{[0,1]}$. We write $[d, d')$ to denote the *risk interval* $\{x \in \mathcal{D} : d \leqslant x < d'\}$. We then define a *risk mitigation strategy* to be a list of interval-obligation pairs, that is $\psi = [(d_0, \bot), (d_1, ob_1), \ldots, (d_{n-1}, ob_{n-1}), (d_n, \bot)]$, where $0 = d_0 < d_1 < \cdots < d_n \leqslant 1$ and $ob_i \in Obs$ is an obligation that the requesting user should fulfil. We write $\bot$ to denote a null obligation, which indicates that the requester is not required to do anything.

Informally, if the risk associated with a sharing request $share(u, v, o)$ is d, then u is required to perform the obligation corresponding to the interval containing d. We write $\psi.d_n$ to denote the start point of the most risky interval, $[d_n, 1]$. We write Ψ to denote a set of risk mitigation strategies.

We associate each sensitivity category with a risk mitigation strategy, defined by an *injective* function $\lambda : C \to \Psi$. In other

Algorithm 1 Adjusting risk intervals

Require: ψ a risk mitigation strategy
Require: $P(\omega^w_{u:OT})$ the obligation trust of u
Ensure: ψ', an adjusted risk mitigation strategy

1: **function** ADJUSTRI($\psi, P(\omega^w_{u:OT})$)
2: $\psi = [(0, \bot), (d_1, ob_1), \ldots, (d_n, \bot)]$
3: $\psi' \leftarrow [(0, \bot)]$
4: **for** $i = 1$ **to** n **do**
5: $d'_i \leftarrow (d_i - (1 - P(\omega^w_{u:OT})) \cdot (d_i - d'_{i-1}))$
6: add (d'_i, ob_i) to ψ'
7: **end for**
8: **return** ψ'
9: **end function**

words, λ maps any two distinct elements of C to distinct elements of Ψ. Specifically, given two sensitivity categories c and c', if $c < c'$, then we require that $\lambda(c).d_n > \lambda(c').d'_n$. This means the more sensitive an object is perceived to be, the lower the acceptable risk threshold for granting requests regarding sharing the object would be.

Notice that when a user u fulfils an assigned obligation, the system would perceive that this action mitigates the risk of granting u's prior sharing request. On the other hand, if u fails to complete an assigned obligation, then the risk mitigation strategy can be considered unsuccessful, in that the initial risk has not, in fact, been mitigated. Therefore, we now describe how u's trustworthiness in fulfilling obligations determines the effectiveness of the risk mitigation strategy for her subsequent requests.

2) Authorisation and risk mitigation: Formally, given a sharing request $r = share(u, v, o)$, where w is the owner of o, we can derive the sensitivity category associated with o by computing $c = m_w(o)$, and the risk mitigation strategy for o by $\lambda(c)$. Let $P(\omega^w_{u:OT})$ denote the trust w places in u in terms of fulfilling obligations (defined in Equation 8).

We derive a new risk mitigation strategy $\lambda(c)'$ by dynamically adjusting risk intervals in $\lambda(c)$ on the basis of $P(\omega^w_{u:OT})$, as illustrated in Algorithm 1. The algorithm describes the process by which the start point d_i of each interval $[d_i, d_{i+1})$ is shifted downs to d'_i. The shifting distance for each d_i is determined by two factors: $P(\omega^w_{u:OT})$ and distance between d_i and d'_{i-1} (the last point that has been shifted). If a user has a very high trust rating (e.g. 1), then these intervals do not change, that is $d'_i = d_i$ for all $i \in [0, n]$. Conversely, if u has a poor history of carrying out obligations (e.g. 0), then all intervals $[d_i, d_{i+1})$, where $0 \leqslant i < n$, disappears, because it is completely cover by $[d'_n, 1]$, where $d'_n = 0$. Informally, the effect of Algorithm 1 is to reduce the risk which can be allowed by the risk mitigation strategies; the less trustworthy a requester is with regard to obligations, the lower the thresholds for using risk mitigation strategies will be. For example, for a completely untrusted agent, no risk mitigation will be available.

We now define an $Auth$ function that evaluates the sharing request r on the basis of the adjusted risk mitigation strategy $\lambda(c)'$ as follows:

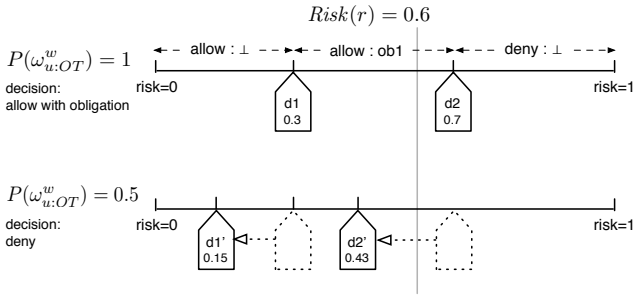

Fig. 1: Risk mitigation through dynamic risk intervals

$$Auth(r) = \begin{cases} (\mathsf{allow}, \bot) & \text{if } Risk(r) \in [0, d_1) \\ (\mathsf{allow}, ob_i) & \text{if } Risk(r) \in [d_i, d_{i+1}),\ 1 \leqslant i < n \\ (\mathsf{deny}, \bot) & \text{otherwise} \end{cases} \tag{10}$$

In other words, the $Auth$ function associates unconditional allow and deny responses with the least $[0, d_1)$ and most $[d_n, 1]$ risky intervals respectively. Otherwise, the request r is permitted but u is obliged to perform obligation ob_i when $Risk(r)$ belongs to $[d_i, d_{i+1})$, where $1 \leqslant i < n$.

Fig. 1 gives an example of how Algorithm 1 and $Auth$ work. A given request r with $Risk(r) = 0.6$, where the obligation trust of the requester $P(\omega^w_{u:OT}) = 1$ (completely trusted), will fall into the second risk interval $[d_1, d_2)$ and will be granted together with the obligation ob_1. However, when the requester is moderately trusted, that is, $P(\omega^w_{u:OT}) = 0.5$, the thresholds d_1 and d_2 will be shifted to $d'_1 = 0.15$ and $d'_2 = 0.43$ respectively. Now the request falls into the *last* risk interval $[0.43, 1]$, and will be denied.

IV. EVALUATION

In order to assess the effectiveness of a trust-based approach, we compare our model, which uses trust and risk assessment and dynamic risk intervals, against the model described by Chen *et al.* [12], which does not consider trust, and uses fixed risk intervals.

A. Experiment Design

We employ a simulated agent society, where 40 *requesters* make requests to share information belonging to 400 *owners* with a number of *recipients*, where recipients are drawn from the same pool as requesters. Requesters vary in competence both with regard to sharing information (referred to as sharing competence, or SC) and fulfilling obligations (referred to as obligation competence, or OC). We define four behavioural profiles from which requesters are generated (Table Ia).

Agents with high SC are more likely to share with recipients in the read_u and share zones. When they share into the $\mathsf{undefined}$ zone, the recipients are likely to be those which the data owner will later add to the read_u zone. To model this, we

TABLE I: Experimental parameters

(a) Requester profiles

id	SC	OC	Count
GG	0.8	0.5	10
GB	0.8	0.1	10
BG	0.3	0.5	10
BB	0.3	0.1	10

(b) Risk intervals

id	Interval	Decision
i_1	[0, 0.2)	$(\mathsf{allow}, \perp)$
i_2	[0.2, 0.6)	$(\mathsf{allow}, email)$
i_3	[0.6, 1]	$(\mathsf{deny}, \perp)$

(c) Parameter settings

Parameter	Value	Description
N_{steps}	500	Number of time steps
N_{runs}	100	Number of runs in each condition
N_{own}	400	Number of data owners
N_{req}	40	Number of requesters
$InitialBudget$	10	Initial risk budget
$BudgetDec$	1	Budget decrement
$a^w_{u:ST}$	1	Sharing trust prior
$a^w_{u:OT}$	1	Obligation trust prior
$P_{timeout}$	0.1	Obligation time-out probability

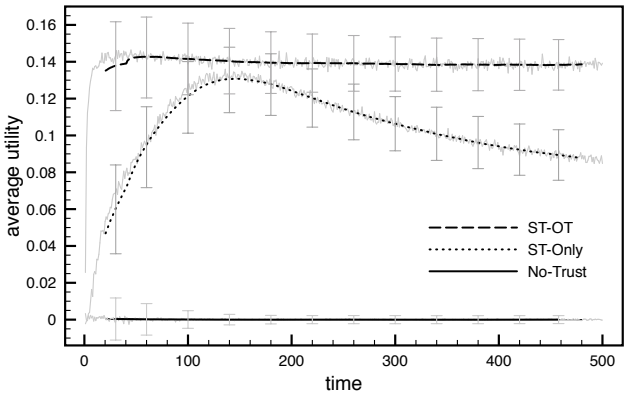

Fig. 2: Average utility gain of agents in each time step

separate the undefined zone into two zones, $\mathsf{undefined}_{good}$ and $\mathsf{undefined}_{bad}$. When an agent selects a recipient to share with, the recipient will be drawn from the owner's read_u, share, or $\mathsf{undefined}_{good}$ zones with probability SC, and from other zones with probability $1 - SC$.

Each owner's policy is initialised by randomly assigning recipients to zones. Then, in each time step, we generate a request for each owner, drawing a requester from that owner's share zone. The requester then selects a recipient to share some information with. The sensitivity category of this information randomly set to either $high$, $medium$ or low, with an associated disclosure loss of 1, 0.5 and 0.2 respectively. The sharing request is handled according to the experimental condition, using the risk intervals and obligations in Table Ib. All requesters initially have a risk budget of 10. The system may assign an obligation $email$ to the requester, reducing the budget by $BudgetDec = 1$. In each time step, each requester will complete an outstanding obligation with probability OC, and each outstanding obligation will "time-out" with a probability of 0.1, resulting in a permanent loss of risk budget.

We employ three experimental conditions:

- **No-Trust**: no sharing or obligation trust assessment is performed. The risk of a request to share information with sensitivity c in this condition is simply $L(c)$. This is equivalent to assuming $P(\omega^w_{u:ST}) = 0$ for every $u \in U$. Risk budgets are used to support authorisation decision-making.
- **ST-Only**: sharing trust, and the risk of granting a sharing request is computed based on the requester's past actions and the sensitivity of information. Risk budgets are used as before.
- **ST-OT**: both sharing and obligation trust assessment is performed. Dynamic risk intervals and obligation trust are used, and risk budgeting is disabled (by setting $BudgetDec = 0$).

For the *ST-Only* and *ST-OT* settings (which employ trust models), the $a^w_{u:ST}$ (*a priori*) parameter is important. Setting this to 1 will result in a sceptical agent, while setting to 0 will result in a naive agent. We set $a^w_{u:ST} = 1$ for all agents, representing that agents are initially optimistic at the outset. Similarly for obligation trust, we set $a^w_{u:OT} = 1$, as this means that the risk intervals will be unaffected when an owner has no evidence about a requester.

We score model performance in the following way. A model receives utility $U(c)$ when category c is shared in the $\mathsf{undefined}_{good}$ zone. When c is shared to $\mathsf{undefined}_{bad}$, $U(c)$ is deducted. Our simulation runs over 500 time steps. Each condition was run 100 times, with the global utility results for each time step averaged. Table Ic summaries all the parameter settings for our simulation.

B. Results

Fig. 2 shows the average utility gain of agents over time, in each experimental condition. Since all penalties and rewards are in the range $[0, 1]$, the global average utilities are quite small. In the *No-Trust* condition, the model averages around 0, allowing equal numbers of good and bad requests. This is expected, as the model does not learn, and instead pessimistically assumes that $P(\omega^w_{u:ST}) = 0$ for all requesters. In the *ST-Only* setting, we see that performance gradually climbs to around 0.13, before falling away. The *ST-OT* model performs best, rapidly reaching 0.14 and remaining there. This means that the model quickly and consistently rejects requests from untrustworthy agents.

We hypothesised that the *ST-OT* model would perform best, as it can more rapidly respond to requesters with low OT by adjusting risk intervals, while the *ST-Only* model can only fully block requesters who have no risk budget. Furthermore, users with high ST but low OT (i.e. the GB profile) will receive more favourable treatment with the *ST-OT* model than with *ST-Only*. These requesters will fall into low risk intervals, and will not be affected by their poor obligation performance in *ST-OT*; however, they are blocked by depleting their risk budget under *ST-Only*.

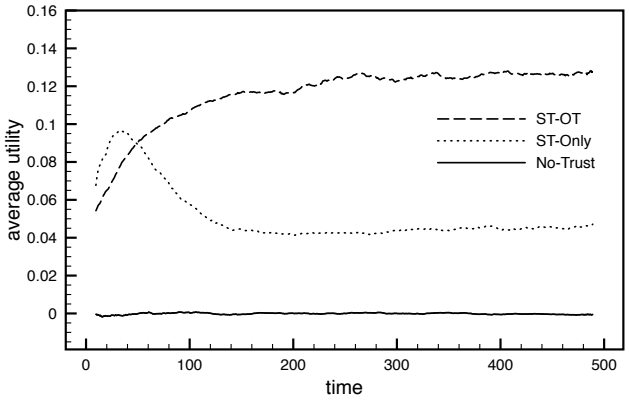

Fig. 3: Average utility gain, $\alpha^w_{u:ST} = 0$

This also explains the falling-off of utility for *ST-Only* models. Here, requesters who deplete their risk budget are unable to make further requests, which is the only way to obtain further obligations to rebuild their budget again. Requesters who deplete their budget are then "locked out" of the system, including those who have high ST.

To assess the effect of the sharing trust prior parameter $\alpha^w_{u:ST}$, we ran our experiments with $\alpha^w_{u:ST} = 0$ (Fig. 3). We found no change in the *No-Trust* condition. However, the *ST-Only* model now receives a rapid spike of early high performance, quickly reaching 0.1, but falling off quickly to around 0.04. The *ST-OT* model, on the other hand, now improves more slowly, taking much longer to converge to 0.14.

One possible reason for this may be that, with $\alpha^w_{u:ST} = 0$, *ST-OT* owners are more sceptical, over-estimating risk, and initially reject more high-risk requests from individuals in $\mathsf{undefined}_{good}$. This appears to indicate that an optimistic value for $\alpha^w_{u:ST}$ is appropriate, when there is an even distribution of good and bad agents (GG and BG). The *ST-Only* model appears to perform more poorly, as agents from the GB class (high ST, low OT) are "locked-out", reducing the number of good requesters, while the dynamic risk intervals used by the *ST-OT* model do not exclude these agents.

V. Discussion

As we have shown, using trust and loss to compute risk can be beneficial. Our TRAAC model performs a similar function to risk budgets by adjusting the effectiveness of risk mitigation strategies. Users will eventually be denied access once they are not trusted enough to perform obligations, and fall into the deny interval. With sufficient knowledge of a domain, risk budgets and decrements could be designed to out-perform our approach. However, dynamic risk intervals have the advantage of not requiring explicit budget parameters, and so are useful when these values are not known.

Our zoned policy model facilitates iterative policy authoring and refinement. Users can inspect and audit the system's decisions and classify them as good or bad. While our approach makes assumptions and sharing decisions on behalf of the user, the user always retains the ability to define fixed groups of individuals who absolutely can, or cannot access some information. The ability of users to audit system behaviour is critical if such mechanisms are to be accepted, and this goal can be supported by the capture of *provenance* [16]. Our model considers the trustworthiness of requesters, but does not consider aspects of the *context* when making trust updates, such as the reason behind a particular violation. For example, a doctor may be bound by legal or ethical responsibilities to share information in certain situations. Sometimes, it is appropriate to *mitigate* trust update based on the context of a violation, without necessarily changing the data owner's policy. Capturing rich provenance about the contexts and reasons behind violations, we can avoid forming trust assessments which do not reflect the underlying trustworthiness of a requester.

While we explicitly assume that owners' policies are visible to requesters, policies themselves could be considered as sensitive information. By assuming visibility of policies, we can ascribe "blame" for violations to the requester, who has either willingly violated or neglected to check the owner's policy beforehand. If we assume that policies are not necessarily visible, then requesters could violate them without realising. This could lead to unfair situations. For example, a requester could violate a policy and become less trustworthy, but would not have caused a violation had the policy been visible to him. Weakening this assumption and maintaining fairness is an interesting future challenge.

We have considered classes of users who can read information, and users who can both read and share with others. However, we represent sharing as the ability to add others to the read zone. In reality, information can always be leaked beyond the system boundaries, and so a "read-only" zone is unrealistic, and so it is more realistic to assume that anyone who can read some data can also share it. This introduces the possibility of sharing *chains*, where information is disseminated around a network by users with sharing ability. Previous work has considered ways in which trust can be calculated when tasks can be sub-delegated, creating chains of trust [17], and similar techniques could be applied to computing information sharing risk when there is a potential for information to be disseminated within a network.

Interestingly, the model presented here permits sub-delegation of risk-mitigation obligations. There is no requirement that the individual who accepts an obligation as part of a risk mitigation strategy be the same individual who eventually completes it. Deposited risk budget is always returned to the individual who accepted an obligation on its completion. However, in our mechanism, the individual who accepts an obligation receives an immediate trust reduction, but only the individual who actually performs the obligation receives the associated positive trust update. This disincentivises sub-delegation. While this scheme seems intuitive, and may be desirable, many others are certainly possible. These sub-delegation issues are an interesting avenue for future work.

Through our requirement that a data item have only one

owner, we are making a strong assumption. In many domains, including healthcare, a data item could have multiple stakeholders, whose policies have some influence over what can be done with that data. Consider a data item having multiple stakeholders, whose zoned policies are initially identical. Initially, the authorisation decision function would result in a "consensus" decision. However, the stakeholders could have different attitudes to risk, and different experiences with requesters in other settings. As time goes on, the decisions returned by each stakeholder for a given request would begin to diverge. In this case, we may consider negotiation, argumentation and conflict resolution approaches to reaching consensus.

Finally, it is important to give consideration to ways in which such a system may be open to manipulation. One potential weakness is our assumption of a single ST trust model for all data items belonging to an owner. This means that a requester's trustworthiness is assumed to be an intrinsic variable, not dependent on the information he can share. As a result, trustworthiness can be "transferred" across data items, leading to faster bootstrapping of the owner's trust. However, since data items can belong to different sensitivity categories and can have different disclosure losses, it is possible for requesters to become highly trusted by exhibiting good ST and OT behaviour with requests to low-sensitivity information [18], in order to gain access to higher-sensitivity information, with the goal of exploiting it.

This could be addressed by maintaining separate trust models for each sensitivity category. Such a model is used when assessing a request for a data item for which there is no evidence, but for which evidence exists about other data items in the same sensitivity category. Another interesting approach may be to apply *stereotypical* trust models, which can learn generalised assessments of trustworthiness based on attributes of a request [19], such as requester features (such as roles or qualifications), contextual attributes (such as the place and time of the request) or social attributes (such as relationships between the requester and other individuals or groups).

VI. Conclusion

We have presented a trust- and risk-aware access control model for controlling the data sharing activities of users. Our approach considers two kinds of trust in the requester: sharing trust and obligation trust.

Moreover, we have proposed a sparse zoned policy model for personal data, which provides a far greater coverage of policy than existing models with the inclusion of share and undefined zones. We evaluated our approach by simulation, and our results suggest that considering both kinds of trust, with dynamic risk intervals, results in more desirable access control behaviour than using fixed risk budgets, or not considering trust at all.

Our model is primarily motivated by the privacy requirements of patients' healthcare records. However, we believe that they can be adapted for many other situations, particularly where both inappropriate sharing, and failing to share properly, entail significant risks.

Acknowledgements

This research is supported by the award made by the RCUK Digital Economy and Energy programmes to the TRUMP UK-India project (award reference: EP/J00068X/1) and the RCUK Digital Economy programme to the dot.rural Digital Economy Hub (award reference: EP/G066051/1).

References

[1] K. Caine and R. Hanania, "Patients want granular privacy control over health information in electronic medical records," *Journal of the American Medical Informatics Assoc.*, vol. 20, no. 1, pp. 7–15, 2013.

[2] T. Wang, M. Srivatsa, and L. Liu, "Fine-grained access control of personal data," in *Proc. of the 17th ACM Symposium on Access Control Models and Technologies*, 2012, pp. 145–156.

[3] P. W. L. Fong, "Relationship-based access control: Protection model and policy language," in *Proc. of the 1st ACM Conf. on Data and Application Security and Privacy*, 2011, pp. 191–202.

[4] S. Barker, "Personalizing access control by generalizing access control," in *Proc. of the 15th ACM Symposium on Access Control Models and Technologies*, 2010, pp. 149–158.

[5] T. Trojer, B. Katt, T. Schabetsberger, R. Breu, and R. Mair, "Considering privacy and effectiveness of authorization policies for shared electronic health records," in *Proc. of the 2nd ACM SIGHIT Intl. Health Informatics Symposium*, 2012, pp. 553–562.

[6] JASON Program Office, "Horizontal integration: Broader access models for realizing information dominance," MITRE Corporation, Technical Report JSR-04-132, 2004.

[7] P.-C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger, "Fuzzy multi-level security: An experiment on quantified risk-adaptive access control," in *Proc. of the 28th IEEE Symposium on Security and Privacy*, 2007, pp. 222–230.

[8] Q. Ni, E. Bertino, and J. Lobo, "Risk-based access control systems built on fuzzy inferences," in *Proc. of the 5th ACM Symposium on Information, Computer and Communications Security*, 2010, pp. 250–260.

[9] L. Chen and J. Crampton, "Risk-aware role-based access control," in *Proc. of the 7th Intl. Workshop on Security and Trust Management*, 2011, pp. 140–156.

[10] N. E. Dimmock, "Using trust and risk for access control in global computing," University of Cambridge, Tech. Rep., August 2005.

[11] Q. Wang and H. Jin, "Quantified risk-adaptive access control for patient privacy protection in health information systems," in *Proc. of the 6th ACM Symposium on Information, Computer and Communications Security*, 2011, pp. 406–410.

[12] L. Chen, J. Crampton, M. J. Kollingbaum, and T. J. Norman, "Obligations in risk-aware access control," in *Proc. of the 10th Annual Conf. on Privacy, Security and Trust*, 2012, pp. 145–152.

[13] I. Pinyol and J. Sabater-Mir, "Computational trust and reputation models for open multi-agent systems: A review," *Artificial Intelligence Review*, vol. 40, no. 1, pp. 1–25, 2013.

[14] A. Jøsang, R. Hayward, and S. Pope, "Trust network analysis with subjective logic," in *Proc. of the 29th Australasian Computer Science Conf.*, 2006, pp. 85–94.

[15] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*, 2nd ed. Springer-Verlag, 1985.

[16] T. D. Huynh, M. Ebden, M. Venanzi, S. D. Ramchurn, S. J. Roberts, and L. Moreau, "Interpretation of crowdsourced activities using provenance network analysis," in *Proc. of the 1st AAAI Conf. on Human Computation and Crowdsourcing*, 2013.

[17] C. Burnett and N. Oren, "Position-based trust update in delegation chains," in *Proc. of the 16th Intl. Workshop on Trust in Agent Societies*, 2013.

[18] K. J. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Computing Surveys*, vol. 42, no. 1, pp. 1:1–1:31, 2009.

[19] C. Burnett, T. J. Norman, and K. P. Sycara, "Stereotypical trust and bias in dynamic multiagent systems," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 2, pp. 26:1–26:22, 2013.