Nudging for Quantitative Access Control Systems

Charles Morisset, Thomas Groß, Aad van Moorsel, and Iryna Yevseyeva

Centre for Cybercrime and Computer Security,
Newcastle University, NE1 7RU, U.K.
firstname.lastname@newcastle.ac.uk

Abstract. On the one hand, an access control mechanism must make a conclusive decision for a given access request. On the other hand, such a mechanism usually relies on one or several decision making processes, which can return partial decisions, inconclusive ones, or conflicting ones. In some cases, this information might not be sufficient to automatically make a conclusive decision, and the access control mechanism might have to involve a human expert to make the final decision. In this paper, we formalise these decision making processes as *quantitative access control systems*, which associate each decision with a measure, indicating for instance the level of confidence of the system in the decision. We then propose to explore how nudging, i.e., how modifying the context of the decision making process for that human expert, can be used in this context. We thus formalise when such a delegation is required, when nudging is applicable, and illustrate some examples from the MINDSPACE framework in the context of access control.

1 Introduction

An *access control mechanism* (ACM) takes as input an access request, and returns a decision describing whether this request should be performed or not. We consider here an access request to contain all information that can be required to make the decision, following the attributed-based access control approach [22,23,7]. For instance, an RBAC request [9] would contain the user, the object, but also the roles of the user, the potential role hierarchy, the permission associated with the object, etc. In general, the final decision needs to be *conclusive*, i.e., either **accept** or **deny**, since the request either goes through or not.

In general, an ACM relies on one or several decision making processes, which indicate what decision should be made. Typically, an *access control policy* is a structured document associating each request with a decision, ranging from an access matrix [14] to sets of XACML policies [18]. Other decision processes can also be used, such as Machine Learning approaches [21,17] or Markov Decision Processes [15]. However, these decision processes do not always return a conclusive decision to the ACM, for instance: a policy might not be applicable to the

T. Tryfonas and I. Askoxylakis (Eds.): HAS 2014, LNCS 8533, pp. 340–351, 2014.

request, thus yielding the decision **na** [18]; missing attributes can create *indeterminacy*, and return a set of all possible decisions instead of a single one [7]; different decision processes can return conflicting decisions [2].

In such situations, the access control mechanism might be able to still make a conclusive decision, for instance by using resolving algorithms (such as the XACML permit-unless-deny and deny-unless-permit algorithms, which, by definition, can only return a conclusive decision), otherwise it might be required to involve a human expert, who will make the final decision.

In this paper, we introduce the notion of *quantitative access control system* (QACS), which represents a decision process, such that each decision is associated with some quantity, for instance indicating the weight of one decision over another. We then postulate that when no conclusive decision can be made autonomously, it might be possible to use the quantitative information to know in which direction the human expert should be *nudged*, following the observation that the way the information is presented to a decision maker has an influence on the final decision made [25]. Intuitively, if the access control mechanism believes that a request should be accepted more than it should be denied, but is not confident enough to make an autonomous decision, then the human expert can be nudged into accepting the request. This approach relies on the assumption that the human expert is more apt to resolve the uncertainty, and thus can ignore any nudge if needed, but could benefit from being provided with a *choice architecture* [24], i.e., a structured context in which the choice is made.

The main contribution of this paper are the following ones: we propose a general notion of access control mechanism, instantiated with several examples from the literature (Section 2); we describe the nudging approach in the context of access control (Section 3); and we propose an evaluation strategy for nudging, identifying the possible choices and their consequences, and we illustrate our approach with an example inspired from conference reviewing (Section 4). This paper is exploratory in nature, and focuses on presenting the different concepts within a common architecture, hopefully accessible to computer scientists, security experts and psychologists, with the intent to open the discussion on this research problem, rather than to bring a concrete solution for a specific problem.

2 Quantitative Access Control Mechanisms

We consider here the set of decisions $\mathcal{D} = \{\mathbf{accept}, \mathbf{deny}, \mathbf{na}\}$, and given a set of requests $\mathcal{R}$, a quantitative access control system (QACS) is a function $\kappa : \mathcal{R} \to (\mathcal{D} \to [0, 1])$, i.e., given a request, returns a function $\delta : \mathcal{D} \to [0, 1]$. In other words, a QACS associates each possible decision with a *quantity*, such that $\delta(\mathbf{accept}) + \delta(\mathbf{deny}) + \delta(\mathbf{na}) \leq 1$.

Our notion of QACS is inspired to some extent by subjective logic [13], which has been used, among others, in trust networks [12]. Indeed, in subjective logic, a truth value is given by a triple (b, d, u), where b represents the level of belief, d the level of disbelief and u the level of uncertainty, such that $b + d + u = 1$. However, our approach differs in that we do not impose the sum of all quantities to equal 1, and we consider the difference between this sum and 1 as the

measure of uncertainty. In other words, we could represent the function δ as a tuple (a, d, na, u), representing the quantities for **accept**, **deny**, **na** and for the uncertainty, respectively, such that $a + d + na + u = 1$. We do not consider here the composition of quantitative access decisions, and we leave for future work a further exploration of subjective logic, and other fuzzy logic in general, in the context of QACS.

We now present some concrete examples of QACS, based on majority-voting [20], Markov Decision Process [15] and machine learning [17]. These examples do not aim at representing an exhaustive list of such systems, but rather to illustrate their diversity.

2.1 Majority Voting Policy

Ni et al. propose in [20] the D-Algebra to encode the semantics of an access control model, where the underlying logics is not necessarily limited to the classical one, and can for instance be the Łukasiewicz one, which comes with a rational number interpretation. Hence, given a policy consisting of several sub-rules, the evaluation of this policy can be defined as true (or permit) if there are more rules evaluating to true than rules evaluating to false.

In particular, they propose an encoding of XACML [18], where each rule can evaluate to one of the following: P, D, NA, P-NA, D-NA, where P stands for Permit, D for Deny and NA for Not-Applicable[1], and given a policy consisting of several, respectively associate the values v_0, v_1, v_2, v_3 and v_4 for the number of rules evaluating for each decision. A policy evaluates to:

- P if $v_0 > v_1 + v_4$, i.e., when the number of permit is higher than the sum of deny and possible deny,
- D if $v_1 > v_0 + v_3$,
- P-NA if $v_0 \leq v_1 + v_4$ and $v_0 + v_3 > v_1 + v_4$,
- D-NA if $v_1 \leq v_0 + v_3$ and $v_1 + v_4 > v_0 + v_3$,
- NA otherwise.

We can define a quantitative access control system as the function

$$\delta(d) = \begin{cases} v_0 + v_3/2 & \text{if } d = \textbf{accept}, \\ v_1 + v_4/2 & \text{if } d = \textbf{deny}, \\ v_2 + v_3/2 + v_4/2 & \text{if } d = \textbf{na}, \end{cases}$$

More recently, Huth et al. introduced the Peal language [6,11], in which access control decisions are made based on numerical evidence, such as trust. In other words, each basic target identifying a condition of interest is associated with a quantity, which can be easily aggregated and selected through numerical operators. Although Peal somehow abstracts the different quantities in the final decision by using some thresholds, we could easily use them to define a QACS.

[1] This interpretation of the XACML decision set is somehow slightly between the standard set of XACML 2 and the extended set of XACML 3, but this discussion is outside of the scope of this paper and not particularly relevant for the notion of majority voting.

2.2 Markov Decision Process

A Markov Decision Process (MDP) [1] is a state machine, transitioning from one state to another through actions, such that transitions are probabilistic (i.e., given one state and one action, we know the probability of reaching each other state) and are associated with rewards. In this context, a *policy* is a function deciding which action to take in each state, and the *optimal policy* is that maximising the expected reward.

Martinelli and Morisset extended this notion in [15] with that of Access Control Markov Decision Process (AC-MDP), which, roughly speaking, is an MDP where each state contains both all relevant security information (levels of security, access matrix, roles, etc) and each action corresponds to a decision. For the sake of conciseness, we do not recall here the formal definition of an AC-MDP, but it is worth pointing out that Bellman's equations [1] allow us to return the expected reward of each decision in each state.

In general, the values of decisions as calculated by an AC-MDP need to be normalised in order to define a QACS, since they do not necessarily belong to $[0, 1]$. An interesting question is whether the normalisation should be done for each state or for the entire model. In the former case, we ensure that the sum of the values of all decisions equal 1, while in the latter case, the highest and lowest possible values in the entire model are likely not to be reachable in all states, meaning that the sum of the values of all decisions might not equal 1, which could account for uncertainty and/or non-applicability.

2.3 Classifier Based

Roughly speaking, a classifier can be seen has an hyper-plane, separating a set of data points into two distinct classes. Once this classifier is built, any new point is mechanically on one side or the other of the plane, and thus classified, assuming of course that the way the new point should be classified is somehow similar to the way the previous points have been classified.

Such an approach has been used in the context of access control policies [21], where a Role-Based Access Control policy is learned using different techniques. This approach is later refined in [17], showing that the distance to the hyper-plane can be used as a measure of uncertainty, the closer to the plane the higher the uncertainty. Following this idea, we could define a QACS which would return, for each decision, its normalised distance to the hyper-plane. Note that this approach is mostly tailored for only two decisions, **accept** and **deny**.

An important common aspect of the three examples described above is that they do not directly integrate an explicit notion of uncertainty, i.e., the sum of all decisions should normally equal 1. However, some of these examples could be extended to include such a notion, for instance, a majority-voting based QACS could consider any policy that could not be retrieved or evaluated at all as uncertain, or an MDP based QACS could compute the variance of the values of each decision in order to represent some notion of uncertainty, with the intuition that the higher the variance, the higher the uncertainty.

3 Nudging

Thaler and Sunstein define in [24] the notion of *choice architect*, who, in a system where its users have the choice between several options, has the responsibility for organizing the context in which these users make decisions. They give the example of a doctor, who must describe the alternative treatments available to a patient. In this case, the doctor does not make the decision for the patient, but presents the characteristics of each possible treatment, often by including some probabilities of success or failure. However, the way the probabilities are *framed*, i.e., emphasising the positive or negative side, can have an impact on the final decision made by the user [25]. For instance, presenting the survival rate of surgery rather than the mortality rate (one being simply the opposite of the other) often leads patients to prefer surgery to other treatments [19].

In the context of access control systems, the choice architect is the entity responsible to present the different possibilities when a decision has to be delegated. This entity is likely to be a human being with a good knowledge of the system, apt to understand the consequences of allowing or denying a given request. As described above, this *decision maker* is prone to bias, and might be influenced by the way the delegation is presented.

In this context, a nudge is "any aspect of the choice architecture that alters peoples behavior in a predictable way without forbidding any options or significantly changing their economic incentives" [24]. Some examples of nudges provided in this book include: *Give More Tomorrow*, where people tend to agree to increase donations in the future; *Filters for air conditioners; the helpful red light*, which proposes to have a red light notifying when the filter of an air conditioner should be changed; or *The Civility Check*, which would prompt the user with a warning when an email appears to be rude or inappropriate, in order to save the user from regretting to have sent it after a few hours or days.

A particularly important point of a nudge is the behavorial effect it uses in order to effectively influence people, and Dolan et al. define in [8] the MINDSPACE framework, recalled in Table 1, which presents nine classes of behavorial effects, based on the reason why each effect works.

Selecting a nudge requires a precise methodology [5], and we do not main here at defining which nudges or behavorial effects are the most relevant. However, intuitively speaking, some effects could be particularly worth exploring. For instance, consider the case where the decision is made by composing several sub-policies and by using a majority voting strategy (see Section 2.1), but a conclusive decision cannot be made autonomously because some sub-policies fail to evaluate. In order to influence the decision maker towards a decision, we could, according to the *Messenger* effect, indicate *who* issued the sub-policies returning that decision, assuming these entities are somehow trusted by the decision-maker.

Similarly, uncertainty in XACML can be due to missing attributes in the request, and XACML comes with an optional mechanism to identify which are those attributes. Returning these attributes to the decision maker, emphasizing

Table 1. The MINDSPACE framework for behavior change [8]

MINDSPACE cue	Behaviour
Messenger	We are heavily influenced by who communicates information to us
Incentives	Our responses to incentives are shaped by predictable mental shortcuts such as strongly avoiding losses
Norms	We are strongly influenced by what others do
Defaults	We go with the flow of pre-set options
Salience	Our attention is drawn to what is novel and seems relevant to us
Priming	Our acts are often influenced by sub-conscious cues
Affect	Our emotional associations can powerfully shape our actions
Commitments	We seek consistency with our public promises, and reciprocate acts
Ego	We act in ways that make us feel better about ourselves

their *salience*, could help justify why there is uncertainty about the request, and whether a conservative approach might be needed.

On a different aspect, Molloy et al. suggest in [16] to use market mechanisms in access control policies, showing that the *Incentive* effect can be used to enhance the decisions made by employees. This effect can be used at two different levels: the decision maker can receive some incentive to make the best decisions, possibly by adapting the incentive to the impact of the request, and the decision maker can take into account the incentive of the user for making the original request.

Another way of influencing the decision maker could be through the *Norm* effect, i.e., to compare how other decision makers would behave in the same situation. Such an approach has been proposed for instance in theorem-proving [3,10], by offering a user stuck with a proof the possibility of seeing different strategies followed by experts. In a similar fashion, a decision maker could be informed that in similar situations, known experts would accept or deny the request.

An important point to consider is that quite often, the choice architecture is not neutral to start with. For instance, the *Defaults* effect indicates that when facing different options, the one offered by default is more likely to be selected. Hence, proposing the decision maker with a pre-selected decision is not neutral. In addition, the *Salience* of the information plays an important role. It could for instance be the case that, when presenting an attribute request, the attributes presented first have an impact on the decision, e.g., presenting the attribute Secret first might lead the decision maker to be more conservative, while presenting the attribute Urgent first might lead the decision maker to be more flexible.

Clearly, the examples above are only intuitions of which nudges and effects could be useful, based on the idea that the decision maker is, in the end, a human being who needs to make a decision, and could be therefore influenced. It is also worth mentioning that nudges are often studied over an entire population rather than for a single individual, and little guarantee can be provided that a particular nudge will work for a given individual. Nevertheless, we believe that involving human decision makers in security mechanisms is required by the complexity of existing systems, especially when dealing with uncertainty, and

that any approach aiming at helping or influencing such a decision maker is worth exploring.

There is however a very important ethical aspect to consider. Indeed, it could be argued that if we merely provide the decision maker with more information, then we simply help making a rational decision; on the other hand, we can also influence the decision making process by adding, removing or changing some information. For instance, informing a decision maker that 90% of known experts would accept a particular request is very likely to have an impact, whether it is true or not. Highlighting or hiding some particular pieces of information might not be considered as lying, but can nevertheless highly influence the decision maker.

4 Nudging for Quantitative Decisions

Let δ be a function returned by a QACS for a given request. There are up to two choices to be made: first, whether the final decision can be made autonomously or should be delegated to the decision maker, and in the latter case, whether the decision maker should be influenced in some way.

4.1 When to Delegate

Perhaps the simplest case is when $\delta(\mathbf{accept}) = 1$ or $\delta(\mathbf{deny}) = 1$, i.e., when the QACS is effectively behaving as a regular ACM, and provides a conclusive decision, with no uncertainty. In this case, the QACS can make the decision autonomously, and there is no particular need for nudging.

Otherwise, the QACS could not reach a conclusive decision with certainty, in which case, it is hard to generalise whether the decision should be delegated to a human expert or not. For instance, consider the case $\delta(\mathbf{accept}) = 0.99$, $\delta(\mathbf{deny}) = 0$ and $\delta(\mathbf{na}) = 0.01$: one could argue that the decision **accept** can be taken autonomously. For instance, thresholds can be used to decide when the measure of a decision is good enough, in a similar way than when dealing with risk-based access control [4]. However, we could equally argue that in case of doubt, the decision should be delegated, especially if the impact of making the wrong decision is important.

4.2 Nudge Selection

Once we have decided that the decision should be delegated, the next question is to know whether the choice architecture of the decision maker should be organised in order to influence the outcome. For instance, if $\delta(\mathbf{accept}) + \delta(\mathbf{na}) + \delta(\mathbf{deny}) = 1$, then there is no uncertainty (i.e., there is no missing information) and if $\delta(\mathbf{accept}) = \delta(\mathbf{deny})$, then the QACS does not favour any particular conclusive decision. In this case, no particular nudge needs to be enforced, but it might be however worth notifying the decision maker that there is no uncertainty, meaning that there is just no helpful rule encoded in the QACS, or that there are equal chances of both decisions to be correct.

Table 2. Influenced Decision Maker (IDM) Versus Neutral Decision Maker (NDM)

	NDM correct	NDM incorrect
IDM = NDM	Non-blocking	Ineffective
IDM ≠ NDM	Counterproductive	Effective

On the other hand, if $\delta(\mathbf{accept}) + \delta(\mathbf{na}) + \delta(\mathbf{deny}) = 1$ and either $\delta(\mathbf{accept})$ or $\delta(\mathbf{deny})$ is strictly maximal, i.e., the value for a conclusive decision is strictly higher than the others, then it might be worth nudging the decision maker towards that decision. In particular, the value of each decision can be used to select a nudge with an appropriate "strength", i.e., with an appropriate chance of effectively influencing the decision maker.

For instance, in the case described above, $\delta(\mathbf{accept}) = 0.99$, $\delta(\mathbf{deny}) = 0$ and $\delta(\mathbf{na}) = 0.01$, the confidence that the request should be accepted is quite high, and for instance, a strong financial incentive could be offered as a nudge to the decision maker. On the other hand, in a case where $\delta(\mathbf{accept}) = 0.51$, $\delta(\mathbf{deny}) = 0.49$ and $\delta(\mathbf{na}) = 0$, we could simply make sure that **accept** is the first decision proposed to the decision maker. Clearly, such a flexibility in the nudge selection requires a detailed study of the effect of a catalogue of nudge proposed for a particular decision maker.

Finally, if $\delta(\mathbf{accept}) + \delta(\mathbf{na}) + \delta(\mathbf{deny}) < 1$, then there is some uncertainty in the QACS. In this case, in addition to nudging to the maximal conclusive decision, if any, it could also be worth highlighting this uncertainty, which could denote some problems in the system, such as an ongoing attack.

4.3 Evaluation of Nudging

Applying nudging in the context of access control naturally leads to the question of the evaluation of the approach, and whether we improve the situation or not. In order to define an evaluation model, let us assume an oracle, able to decide (possibly afterwards) if the final decision was correct or not[2].

In addition, we need to consider two different decision makers: the Neutral Decision Maker (NDM), who represents how the decision maker would have behaved without explicit nudging (we abuse the term neutral here, since, as we said above, defining a bias-free environment is not an easy task), and the Influenced Decision Maker (IDM), who represents how the decision maker behaves after being influenced by one or several nudges. In the following, we say that an IDM or an NDM are *correct* whenever they behave as the oracle. For the sake of simplicity, we also assume that the oracle returns a single decision, meaning that when two decision makers behave differently, at most one of them is correct.

Table 2 summarises the different possibilities when nudging. Note that this table does not directly depend on the actual decision taken by a decision maker,

[2] Clearly, such an oracle would not be available at run-time, otherwise it would be used in lieu of the ACM.

but rather on wether the IDM behaves similarly to the NDM, and whether the NDM was correct in the first place.

Roughly speaking, when the IDM behaves similarly to the NDM (first row), the nudge had no direct effect on the decision maker. Hence, two cases are possible: either the NDM was right in the first place, in which case the nudge is *non-blocking*, or the nudge is *ineffective*, as it failed to prevent the NDM from making the wrong decision. Two reasons can lead to the latter case: either a nudge leading to the correct decision was used but ignored by the decision maker, indicating that the nudge was not powerful enough, or the nudge was coinciding with the decision of the NDM, meaning that the QACS did not return a quantitative decision allowing to predict the correct decision. In either case, nudging is not worse than the neutral approach.

The main impact of nudging comes when the IDM behaves differently than the NDM (second row). Here again, two cases are possible: if the NDM was correct in the first place, then by assumption, the IDM is not correct, meaning that the nudge was *counterproductive*, or the NDM was incorrect, and therefore the IDM is correct, making the nudge *effective*. The second case is clearly the reason why we want to nudge in the first place, to help the decision maker to make the best decision. However, the first case is not to be ignored, and as with inefficient nudges, two reasons can lead to counterproductive nudges. First, the nudge could have been in the right direction, but confusing to the decision maker, who thus went against his own intuition and chose the incorrect decision (for instance by displaying an unexpected message box). Second, the QACS could have been wrong in predicting the decision, leading to intentionally nudging the decision maker away from his original decision, even though it was correct.

4.4 Example

As an example of our approach, let us consider a special case of access control policy with the reviewing mechanism of a conference: a paper submitted to a conference is eventually either accepted or denied, usually by considering quantitative decisions made by several reviewers, which fits with the idea of QACS.

For the sake of simplification, let us consider that each paper is reviewed by four different reviewers, each of them having up to 0.25 points to give for the paper, in a form of a triple (a, d, na), where a represents the number of points given to accept the paper, d the number of points given to reject the paper and na the number of points given to indicate that the reviewer is not apt to review the paper. A sum $a + d + na$ below 0.25 indicates some uncertainty from the reviewer. For instance, a review $(0.1, 0.1, 0.05)$ might indicate that the paper has both good and bad points, but that the reviewer is not a top expert in the field but has a good confidence; a review $(0.1, 0, 0.1)$ might indicate that the reviewer is not particularly confident, but found some good points; a review $(0, 0, 0)$ might indicate that the paper was not understood at all by the reviewer, etc.

In order to make a final decision, the triples returned by all the reviewers for each paper are added in a point-wise way, which creates a triple (a, d, na) corresponding to the final score of the paper. By construction, we have $a + d +$

$na \leq 1$, and therefore we can consider this score as returned by a QACS (note that a missing review would be automatically considered as a triple $(0, 0, 0)$, thus denoting full uncertainty). The Programme Committee Chair (PCC) of the conference sets up the rule that any paper with a score such that $a \geq 0.8$ is automatically accepted while any paper with a score such that $d \geq 0.8$ is automatically rejected; any other paper has to be processed by the PCC.

Neutral decision. If $a = d$, meaning that the paper has received an equal number of points for accepting and denying it, then no particular nudge is applied to influence one or the other decision, and the paper is presented in a neutral format (e.g., black text over white background). In addition, the names of the reviewers can be omitted, in order to remove Messenger influencer. If $a + d + na < 0.5$, then there is a lot of uncertainty about the paper, perhaps indicating that it is not well written, and an automatic spelling check could be performed, thus providing a quantitative indicator of the poor quality of the writing, if there are indeed many errors. If $na > 0.5$, it could be the case that the paper is off-topic for the conference, and the keywords of the paper could be highlighted, together with a comparison of the key-phrases of the paper (as Easychair [26] offers) with the call for papers of the conferences.

Nudging towards acceptance/rejection. If $a > d$ (and conversely when $a < d$), then the paper is presented in a positive format, for instance with a green background, and the names of the reviewers who accepted the paper are highlighted to the PCC. To some extent, Easychair is already using different colours to represent different potential decisions during the programme committee phase. Papers are also put in a by-default category, such that if no action is taken, the paper is accepted automatically. Finally, if the number of accepted papers is at this point lower than that of the previous edition of the conference, then the PCC can be reminded that the conference normally accepts more papers.

Altogether, one could argue that we are simply presenting the PCC with the most relevant information to make a final decision, without forcing her hand to accept or reject a paper, which is the general approach of nudging: organising the context in which the choice is made while leaving freedom of choice. It could be equally argued that, in practice, the PCC is influenced by multiple biases, some intentional (e.g., the colours of the options), some perhaps less (the order in which papers are presented can have an impact), and our approach could also provide a frame aiming at removing such unintentional biases. Finally, other aspects could be integrated in the nudging process, such as favouring more submissions coming from some countries, to encourage diversity, or even to favour less submissions coming from members of the programme committee.

5 Discussion

We have presented an abstract approach for nudging in the context of quantitative access control systems (QACS). This approach is based on two observations: human beings might be involved in the security decision making process, and

human beings can be influenced. The notion of QACS introduced in Section 2 is general enough to cover several kind of systems, and illustrates why a conclusive decision might not always be made autonomously, thus requiring the intervention of a human expert. We have seen in Section 3 that it could be possible to nudge the behaviour of this expert, using different techniques, for instance following the MINDSPACE framework, and we have discussed in Section 4 when nudging should be used and how to evaluate a nudging approach.

An initial observation can be taken from this discussion: a nudge can be inefficient or counterproductive both when the QACS is not accurate enough (and thus predicts the wrong decision) and when the nudge is not followed by the decision maker. Hence, nudging is not necessarily the best approach, and needs to be properly evaluated before being deployed. The example of the conference reviewing system could serve as an interesting basis for a study, especially since it is data that is often accessible to academics. However, this is not strictly speaking a security policy, and the effect of nudges in one context might not be applicable to others.

In addition to conducting several rigorous studies to evaluate nudging approaches in specific context, several leads are interesting to explore further. If the effects of a nudge can be quantified, then we can design an MDP to calculate the optimal decision at each step. However, the repeated usage of nudges leads to the *habituation* of this nudge for a user, and it might be worth considering using a nudge only when it is worth it, which could be done by integrating a notion of value in the above MDP. Finally, it could be worth studying how larger sets of decisions (e.g., including obligations) can impact the nudging approach, since the decision maker has more than two options to choose from.

Acknowledgements. This research is supported by EPSRC Grant EP/K006568 Choice Architecture for Information Security, part of the GCHQ/EPSRC Research Institute in Science of Cyber Security, and the authors warmly thank Pam Briggs, Lynne Coventry, Debora Jeske, Christopher Laing and James Turland for the fruitful discussions on nudging in the security context.

References

1. Bellman, R.: A markovian decision process. In. Univ. Math. J. 6, 679–684 (1957)
2. Bruns, G., Huth, M.: Access-control policies via belnap logic: Effective and efficient composition and analysis. In: Proc. of CSF 2008, pp. 163–176 (2008)
3. Bundy, A., Grov, G., Jones, C.: Learning from experts to aid the automation of proof search. In: AVoCS 2009, vol. CSR-2-2009, pp. 229–232 (September 2009)
4. Cheng, P.-C., Rohatgi, P., Keser, C., Karger, P.A., Wagner, G.M., Reninger, A.S.: Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In: Proceedings of S&P 2007, pp. 222–230. IEEE (2007)
5. Coventry, L., Briggs, P., Jeske, D., van Moorsel, A.: SCENE: A structured means for creating and evaluating behavioral nudges in a cyber security environment. In: Marcus, A. (ed.) DUXU 2014, Part I. LNCS, vol. 8517, pp. 229–239. Springer, Heidelberg (2014)

6. Crampton, J., Huth, M., Kuo, J., Morisset, C.: Policy-based access control from numerical evidence. Technical Report 2013/6, Imperial College London, Department of Computing (October 2013)
7. Crampton, J., Morisset, C.: PTaCL: A language for attribute-based access control in open systems. In: Degano, P., Guttman, J.D. (eds.) Principles of Security and Trust. LNCS, vol. 7215, pp. 390–409. Springer, Heidelberg (2012)
8. Dolan, P., Hallsworth, M., Halpern, D., King, D., Metcalfe, R., Vlaev, I.: Influencing behaviour: The mindspace way. Journal of Economic Psychology 33(1), 264–277 (2012)
9. Ferraiolo, D.F., Kuhn, D.R.: Role-based access control. In: Proceedings of the 15th National Computer Security Conference, pp. 554–563 (1992)
10. Freitas, L., Whiteside, I.: Proof Patterns for Formal Methods. In: Jones, C., Pihlajasaari, P., Sun, J. (eds.) FM 2014. LNCS, vol. 8442, pp. 279–295. Springer, Heidelberg (2014)
11. Huth, M., Kuo, J.: PEALT: A reasoning tool for numerical aggregation of trust evidence. Technical Report 2013/7, Imperial College London, Department of Computing (October 2013) ISSN 1469-4166 (Print), ISSN 1469-4174 (Online)
12. Jøsang, A., Hayward, R., Pope, S.: Trust network analysis with subjective logic. In: Proceedings of ACSC 2006, Darlinghurst, Australia, pp. 85–94 (2006)
13. Jøsang, A., Bondi, V.: Legal reasoning with subjective logic. Artificial Intelligence and Law 8(4), 289–315 (2000)
14. Lampson, B.: Protection. In: Proceedings of the 5th Annual Princeton Conference on Information Sciences and Systems, pp. 437–443. Princeton University (1971)
15. Martinelli, F., Morisset, C.: Quantitative access control with partially-observable markov decision processes. In: CODASPY 2012, pp. 169–180. ACM (2012)
16. Molloy, I., Cheng, P.-C., Rohatgi, P.: Trading in risk: Using markets to improve access control. In: NSPW (2008)
17. Molloy, I., Dickens, L., Morisset, C., Cheng, P.-C., Lobo, J., Russo, A.: Risk-based security decisions under uncertainty. In: Proceedings of CODASPY 2012, pp. 157–168. ACM, New York (2012)
18. Moses, T.: eXtensible Access Control Markup Language TC v2.0, XACML (2005)
19. Moxey, A., O'Connell, D., McGettigan, P., Henry, D.: Describing treatment effects to patients. Journal of General Internal Medicine 18(11), 948–959 (2003)
20. Ni, Q., Bertino, E., Lobo, J.: D-algebra for composing access control policy decisions. In: Li, W., Susilo, W., Tupakula, U.K., Safavi-Naini, R., Varadharajan, V. (eds.) ASIACCS, pp. 298–309. ACM (2009)
21. Ni, Q., Lobo, J., Calo, S., Rohatgi, P., Bertino, E.: Automating role-based provisioning by learning from examples. In: Proceedings of SACMAT 2009, pp. 75–84. ACM, New York (2009)
22. OASIS. eXtensible Access Control Markup Language (XACML) Version 3.0, Committee Specification 01 (2010)
23. Rao, P., Lin, D., Bertino, E., Li, N., Lobo, J.: An algebra for fine-grained integration of xacml policies. In: Proceedings of SACMAT 2009, pp. 63–72 (2009)
24. Thaler, R., Sunstein, C.: Nudge: Improving Decisions about Health, Wealth, and Happiness. Yale University Press (2008)
25. Tversky, A., Kahneman, D.: The framing of decisions and the psychology of choice. Science 211(4481), 453–458 (1981)
26. Voronkov, A.: Easychair. In: Kovacs, L., Kutsia, T. (eds.) WWV 2010. EPiC Series, vol. 18, p. 2. EasyChair (2013)