

# Exploring rationality of self awareness in social networking for logical modeling of unintentional insiders

Florian Kammüller and Chelsea Mira Alvarado

Middlesex University London and  
Technische Universität Berlin

f.kammueLLer@mdx.ac.uk | CA936@live.mdx.ac.uk

**Abstract.** Unawareness of privacy risks together with approval seeking motivations make humans enter too much detail into the likes of Facebook, Twitter, and Instagram. To test whether the rationality principle applies, we construct a tool that shows to a user what is known publicly on social networking sites about her. In our experiment, we check whether this revelation changes human behaviour. To extrapolate and generalize, we use the insights gained by practical experimentation. Unaware users can become targeted by attackers. They then become unintentional insiders. We demonstrate this by extending the Isabelle Insider framework to accommodate a formal model of unintentional insiders, an open problem with long standing.

## 1 Introduction

The privacy paradox [3] shows that humans can be on one side quite concerned about security and privacy in general but when it comes to their own behaviour they seem to ignore any caution and freely spread their private data into public cloud based social network services, like Facebook, Twitter or Instagram. Assuming that humans are rationally acting beings has led to quite successful models and prediction in economics using what is termed Rational Choice Theory (RCT) [43]. Sociologists have transferred RCT more generally to social interaction forming what is known as *exchange theory*. We want to test this theory on the privacy paradox and use the results to improve automated logical verification of social networks. We consider a dynamic system research approach more suitable. The Isabelle Insider framework permits modeling and analyzing dynamic state transition. Thus we can reason on actions and their effects. Methodologically, we thus follow the action research approach [39] interleaving empirical research with interventions, here, practical implementations and verification.

This paper first presents an empirical study on increasing privacy awareness for the construction of a social self awareness tool for social networks. It uses assumptions from RCT testing and highlighting the significance of applying this theory. RCT can be considered as a follow up theory of Max Weber's sociological explanation which has strongly inspired the human actor model of

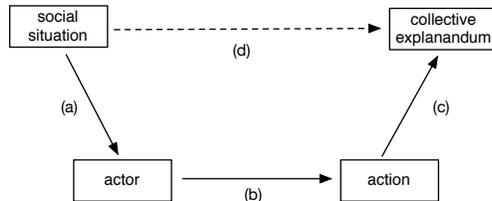
Isabelle’s Insider framework. Consequently, it appears natural to use the RCT interpretation found in the empirical study to extend the human model in the Isabelle Insider framework. Moreover, it turns out that the RCT interpretation of social awareness allows to model unintentional insiders a challenge hitherto unanswered.

This paper first gives some background from sociology about RCT and the Isabelle Insider framework (Section 2). Section 3 presents the tool based study on privacy awareness in social networks and the influence of RCT giving some insights into the requirements, design, testing and evaluation of the tool and the key findings in the RCT interpretation with respect to privacy awareness. This section is based on the Bachelor of Science dissertation of one of the authors [1]. Section 4 then continues to transfer the experimental findings into extending the Isabelle Insider framework and illustrating them on the case study. The Isabelle sources are publicly available on github [22].

## 2 Background

### 2.1 Social Explanation and Rational Choice Theory

Rational choice theory is based upon the assumption that complex social phenomena can be explained by individual actions that constitute them. This philosophy now coined *methodological individualism* holds that: ‘The elementary unit of social life is the individual human action. To explain social institutions and social change is to show how they arise as the result of the action and interaction of individuals’ [9]. Seemingly very close to *methodological individualism*, is what was originally conceived by Max Weber [46,45] as ‘understanding explanation’ (*Verstehendes Erklären*) sketched in Figure 1. Despite these similarities, RCT



**Fig. 1.** Max Weber’s sociological explanation model: a macro-micro-macro-level-transition explaining sociological phenomena by breaking down the global facts from the macro level (a) onto a more refined local view of individual actors at the micro-level (b). Finally those micro-steps are generalized and lifted back on the macro-level (c) to explain the global phenomenon (d).

is more extreme in only considering rational actions. John Scott explains in his

critical overview over RCT [43]: ‘what distinguishes RCT [...] is that it denies the existence of any kind of action other than the purely rational [...]’. We draw from Scott’s overview to contrast and provide the right context for our work. He quite critically highlights the limitations of RCT in particular when branching out from economics and applying RCT more generally to sociology. According to Scott, Homans [12] was a “pioneering figure” in establishing rational choice theory in sociology setting up the basic framework of *exchange theory* which can be understood as RCT for social interaction. In this framework, money and market mechanisms of economic theories are replaced by human resources as time, information, approval and prestige. Besides pioneering RCT, Homans additionally grounded exchange theory on assumptions that he drew from behaviourist psychology. While the methodological individualism of rational choice theories starts from individuals’ actions and sees all social phenomena reducible to these actions, Homans went one step further into explaining them. For him it was necessary to reduce these actions to conditioned psychological responses. In brief, human behaviour is like animal behaviour not free but determined by rewards and punishment. This reinforcement is called ‘conditioning’ and determines human behaviour. Behaviour can thus be studied purely externally and needs no inspection of internal mental states.

While others rejected Homans’ claims about this explanation of human behaviour – and even Homans came to see it as inessential – for our formal model of awareness and unintentional insiders it is very helpful. In Section 4, when we formalize the taxonomy extracted from the experimental work into Isabelle, we model human behaviour in the sense of ‘conditioning’. We actually do model the internal state of the actors although Homans considered this as unnecessary but our model permits dynamic state inspection including psychological disposition of human actors.

## 2.2 Isabelle Insider framework

The Isabelle Insider framework [5,31] has also been inspired by Max Weber and methodological individualism. In mapping this fundamental philosophy to logic, this framework follows a common introductory textbook for sociologists by Hartmut Esser [10] written in the spirit of Popper’s critical rationalism. This offers an approach to understand sociological experiments in a formal way using a logical view on explanation by the logicians Hempel and Oppenheim [11]. In addition, the Isabelle Insider framework uses a taxonomy provided in [41] which is founded on empirical and psychological studies of counterproductive workplace behaviour. In Section 4, we will in more detail present the details of how the human disposition and its effects to the environment are modeled in Isabelle and how this model is now extended to accommodate the unintentional insider.

Isabelle is an interactive proof assistant based on Higher Order Logic (HOL). Application specific logics are formalized into new theories extending HOL. They are called object-logics. Although HOL is undecidable and therefore proving needs human interaction, the reasoning capabilities are very sophisticated supporting “simple”, i.e., repetitive, tedious proof tasks to a level of complete au-

tomation. The use of HOL has the advantage that it enables expressing even the most complex application scenarios, conditions, and logical requirements and HOL simultaneously enables the analysis of the meta-theory. That is, repeating patterns specific to an application can be abstracted and proved once and for all. An object-logic contains new types, constants, and definitions. These items reside in a theory file. For instance, the file `UnintentionalInsider.thy` contains the object-logic for unintentional insiders described in the following paragraphs. This Isabelle Insider framework is a *conservative extension* of HOL. This means that our object logic does not introduce new axioms and hence guarantees consistency. Conceptually, new types are defined as subsets of existing types and properties are proved using a one-to-one relationship to the new type from properties of the existing type.

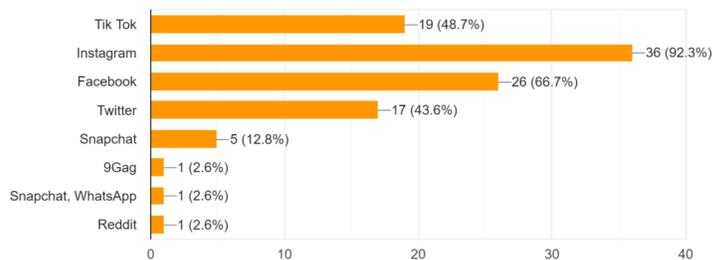
We are going to use Isabelle syntax and concepts in the presentation of the Isabelle Insider framework and will explain them when they are used.

### 3 Social Networks and Privacy Awareness

#### 3.1 Requirements analysis and design of social awareness tool

A questionnaire was created in order to research about public attitudes to internet security amongst social media users. Quantitative and qualitative data are obtained through this method, allowing more time for analysis of the results and how the results can be used to create a prototype. Answers to ‘How many different social media apps/websites do you use every day?’, show that 84.6% of 39 responses use more than 3 different forms of social media every day. This shows the commonality and reliance of social media in everyday lives and how many different apps can hold information about you.

Do you use any of these social media platforms regularly?  
39 responses



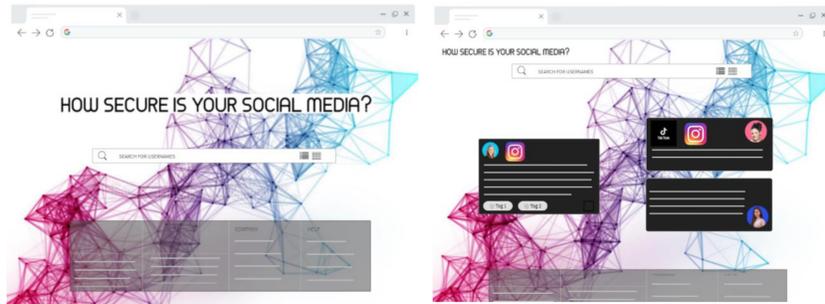
The above chart show the most common applications that people use, Instagram, Facebook and Tik Tok have shown to be the most common platforms. Each of which have shown to have breaches in misuse of personal data they have collected from their users.

How many people have their accounts on private? The majority, 56.4%, say that only some are private, meaning that the users have chosen to only privatise one or more of their accounts but have left others to be able to be accessed

by the public. Are these users aware of how much information they have put out publicly? Surprisingly, the most common answers are completely aware or somewhat aware. Of the amount. 59% have said that they have knowledge of information they have posted publicly but leave room for uncertainty as to how much is actually available to the public. This shows a slight concern from users in their social media behaviour.

### 3.2 Testing and Evaluation

The left side of Figure 2 shows the design of the search page which focuses on clear minimalistic aesthetic to display clear concise information, which will be easily accessible by all. The website title placed at the top middle and highlights the purpose of the website. The search bar is in the middle of the page, letting the user know that the tool only has one importance and should not show otherwise. The user will not be lost when navigating the website, easing user comfort. The bottom grey section reflects basic information on the importance of internet security and what the website aims to show. User inputs through the search bar and uses the search button. The right side of the figure displays what



**Fig. 2.** Interface of social network awareness tool

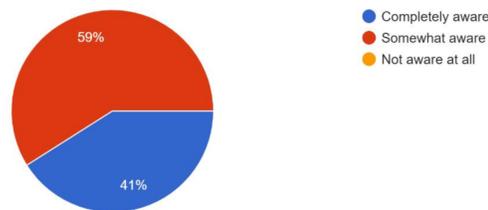
the user sees when they have searched a username. Profiles are created of the available information such as other social media accounts that are linked to the username searched. The profiles are highlighted by the black boxes they are in that contrast the white background, allowing less crowded visuals which may have disorientated people.

For the implementation, we used opensource API's. An API is an application programming interface that allows computers to send signals and receive data in return. This enables specific queries and actions to be retrieved. APIs need keys allowing access to sensitive data whilst also protecting important and sensitive data that cannot be accessed by any user. All social networks allow developers to apply for API keys, allowing APIs to be used for projects. The API allowed

us to retrieve that data necessary enabling to connect to the internet and use genuine social network server data. Users are able to search any username on any social network and retrieve related information. The API's proved to be the best solution for this project as we could acquire the necessary data and use it to create a summarized only profile. The results are thus inherently genuine reflecting real world scenarios.

### 3.3 Key findings and RCT interpretation of privacy awareness

How aware do you think you are of how much information you have posted publicly?  
39 responses



Most importantly, the last question investigates the issue of social media behaviours. It asks whether they would personally change their online behaviours if they were able to see what others could see about them. From responses to this question on the questionnaire, many have shown concern about what others could obtain from the information they post online and would immediately act on this by privatising their social media. Consequently, this shows the importance of a tool that helps people become more aware of their online behaviours.

It matches the rationality assumption of RCT and proves that creating awareness changes the users attitude. This creates a potential for improved privacy in social networks and how awareness could change the risk of attacks on privacy.

## 4 Modeling Unaware Social Network Users and Unintentional Insiders in Isabelle

The state based dynamic semantics of the Isabelle infrastructure framework allows expressing how awareness dynamically changes the global policy and thus how a change on awareness eliminates the risk. We also show how to integrate awareness into the notion of insiderness thus extending the Isabelle Insider framework to unintentional insiders based on the findings of our experiment with the social awareness tool.

### 4.1 Infrastructures, Policies, Actors in Isabelle

The Isabelle Infrastructure framework supports the representation of infrastructures as graphs with actors and policies attached to nodes. These infrastructures are the *states* of the Kripke structure.

The transition between states is triggered by non-parameterized actions `get`, `move`, `eval`, and `put` executed by actors. Actors are given by an abstract type `actor` and a function `Actor` that creates elements of that type from identities (of type `string` written `''s''` in Isabelle).

```
typedecl actor
type_synonym identity = string
consts Actor :: string  $\Rightarrow$  actor
```

Note that it would seem more natural and simpler to just define `actor` as a datatype over identities with a constructor `Actor` instead of a simple constant together with a type declaration like, for example, in the Isabelle inductive package by Paulson [42]. This would, however, make the constructor `Actor` an injective function by the underlying foundation of datatypes therefore excluding the fine grained modelling that is at the core of the insider definition: in fact, the core insider property `UasI` (see below) defines the function `Actor` to be injective for all except insiders and explicitly enables insiders to have different roles by identifying `Actor` images.

To represent the macro level view seeing the actor within an infrastructure, we define a graph datatype `igraph` (see below) for infrastructures. This datatype has generic input parameters that are going to be supplied as concrete parts of an application infrastructure on instantiation of an `igraph`. They represent the actual location graph, the actors in each locations, their roles, credentials and psychological disposition (see following subsection) and the locations' state.

```
datatype igraph = Lgraph (location  $\times$  location)set
                                location  $\Rightarrow$  identity set
                                actor  $\Rightarrow$  (string list  $\times$  string list)
                                actor  $\Rightarrow$  actor_state
                                location  $\Rightarrow$  string list
```

Consider here the social network case study as an example.

```
ex_graph  $\equiv$  Lgraph
  {(aphone,instagram), (bphone,instagram)}
  ( $\lambda$  x. if x = aphone then {'Alice'} else
    (if x = bphone then {'Bob'} else {}))
  ex_creds ex_locs
```

Policies specify the expected behaviour of actors of an infrastructure. Atomic policies of type `apolicy` describe prerequisites for actions to be granted to actors given by pairs of predicates (conditions) and sets of (enabled) actions:

```
type_synonym apolicy = ((actor  $\Rightarrow$  bool)  $\times$  action set)
```

For example, the `apolicy` pair  $(\lambda x. \text{has } (x, \text{'PIN'}), \{\text{move}\})$  specifies that all actors who know the PIN are enabled to perform action `move`.

Infrastructures combine an infrastructure graph of type `igraph` with a policy function that assigns local policies over a graph to each location of the graph, that is, it is a function mapping an `igraph` to a function from `location` to `apolicy set`.

```
datatype infrastructure = Infrastructure igrph
                                [igrph, location] => apolicy set
```

For our social network example, the initial infrastructure contains the above graph `ex_graph` and the local policies defined shortly.

```
sn_scenario ≡ Infrastructure ex_graph local_policies
```

The function `local_policies` gives the policy for each location `x` over an infrastructure graph `G` as a pair: the first element of this pair is a function specifying the actors `y` that are entitled to perform the actions specified in the set which is the second element of that pair.

```
local_policies G x ≡
case x of
  aphone => {(\ y. has G (y, 'aPIN')), {put,get,move,eval}}
| bphone => {(\ y. has G (y, 'bPIN')), {put,get,move,eval}}
| instagram => {(\ y. ∈ {Actor 'Alice', Actor 'Bob'},
                {put,get,move,eval})}
| _ => {}
```

We define the behaviour of actors using a predicate `enables`: within infrastructure `I`, at location `l`, an actor `h` is enabled to perform an action `a` if there is a pair `(p,e)` in the local policy of `l - delta I l` projects to the local policy – such that that action `a` is in the action set `e` and the policy predicate `p` holds for actor `h`.

```
enables I l h a ≡ ∃ (p,e) ∈ delta I l. a ∈ e ∧ p h
```

For example, the statement `enables I l (Actor 'Bob') move` is true if the atomic policy `(\x. True, {move})` is in the set of atomic policies `delta I l` at location `l` in infrastructure `I`. Double quotes as in `'Bob'` create a string in Isabelle/HOL.

## 4.2 Modelling the human actor and psychological disposition

The human actor’s level is modeled in the Isabelle Insider framework by assigning the individual actor’s psychological disposition to each actor’s identity.

```
datatype actor_state = State psy_state motivations
```

There are selector functions `motivation` and `psy_state` to project the components from an `actor_state` element. The psychological state of an actor is not determined using the formal system but we use here empirical facts as input as for example our own studies from Section 3 or other sociological findings, like [41]. The formal representation of *Psychological State* is a simple enumeration datatype distinguishing the “normal” state of happiness from one in which the actor is alerted or “suspicious”.

```
datatype psy_states = happy | suspicious
```

The element on the right hand side are the two injective constructors of the new datatype `psy_states`. They are simple constants, modeled as functions without arguments. Motivation plays a vital role in RCT and as Homans observed the strongest one is that humans seek approval (which is only excluded by a state of mind that corresponds to complete detachment which we abbreviate as “zen”).

```
datatype motivations = approval_hungry | zen
```

The types for psychological state and motivations allow defining the users state of unawareness by a predicate.

```
definition unaware :: actor_state  $\Rightarrow$  bool
  unaware a  $\equiv$  motivation a = {approval_hungry}  $\wedge$  happy = psy_state a
```

### 4.3 Privacy by labeling data and state transition

The Decentralized Label Model (DLM) [40] introduced the idea to label data by owners and readers. We use this idea and formalize a new type to encode the owner and the set of readers of a data item.

```
type_synonym dlm = actor  $\times$  actor set
```

Labelled data is then just given by the type `dlm  $\times$  data` where `data` can be any data type.

The abstract state transition provided in the underlying Kripke structure theory is instantiated in the infrastructure model by an inductive definition of a state transition relation  $\rightarrow_n$  over infrastructures. A set of inductive rules defines this transition relation  $\rightarrow_n$  relative to characteristics of the current state. These characteristics can exploit the information encoded into the infrastructure as well as the enables predicate to express how the next infrastructure state evolves from the current one. We show here the rules for put and get as they suffice to illustrate how to model the social network application scenario.

**The put pata rule** assumes an actor `h` residing at a location `l` in the infrastructure graph `G` and being enabled the put action. In addition, the psychological state `pgra G h` needs to be unaware. Here we add the newly extended option for the human actor model to the semantic rule as precondition thus stating that only unaware users put their data onto the graph. If infrastructure state `I` fulfills those preconditions, the next state `I'` can be constructed from the current state by adding the data item `((Actor h, hs), n)` at location `l`. The addition is given by updating (using `:=`) the existing data storage `lgra G l` at location `l` with the singleton set `{((Actor h, hs), n)}`. Note that the first component `Actor h` marks the owner of this data item as `h`.

```
put:
G = graphI I  $\Rightarrow$  h @G l  $\Rightarrow$  enables I l (Actor h) put  $\Rightarrow$ 
unaware (pgra G h)  $\Rightarrow$ 
I' = Infrastructure
```

```

(Lgraph (gra G)(agra G)(cgra G)(pgra G)
 ((lgra G)(l := (lgra G l ∪ {(Actor h, hs), n}))))
(delta I)
⇒ I →n I'

```

**The get data rule** resembles the put data rule in many parts. However, here an actor  $h$  accesses data in a remote location  $l'$  and adds it to the data in his current location  $l$ . This copying of data is only permitted if the current location  $l'$  of the data enables  $h$  to **get** and if the list of readers  $hs$  in the data item  $((Actor\ h',\ hs),\ n)$  contains the entry **Actor**  $h$  or if the accessing actor is  $h$  herself.

```

get_data:
G = graphI I ⇒ h @G l ⇒ enables I l' (Actor h) get ⇒
((Actor h', hs), n) ∈ lgra G l' ⇒ Actor h ∈ hs ∨ h = h' ⇒
I' = Infrastructure
(Lgraph (gra G)(agra G)(cgra G)(pgra G)
 ((lgra G)(l := (lgra G l ∪ {(Actor h', hs), n}))))
(delta I)
⇒ I →n I'

```

The global policy is ‘only the owner and friends can access the data on the cloud’ using for example the definition of **friends** as  $\{\text{''Alice''}, \text{''Bob''}\}$ .

```

global_policy I a ≡ a ∉ friends
→ ¬(enables I instagram (Actor a) get)

```

We can prove that Bob is enabled to get (Alice’s data) at instagram if Bob is specified as a reader in an application scenario where Alice sets the label parameter  $hs$  in a put action accordingly. So, using the features of attack tree analysis of the Isabelle Insider framework, we can formally prove such statements. However, we are interested in investigating negative effects of unawareness and how a change of human behaviour may improve the situation. Therefore, we use the representation of human factors and (malicious) Insiders in the Isabelle Insider framework, integrating the existing notion of malicious insiders and extending them to include also unintentional insiders.

#### 4.4 Representing human factors and insiders

The Isabelle Insider framework defines “[a]n insider [as] a trusted user of a system who behaves like an attacker abusing privileges thereby bypassing security controls” [24]. This definition leads to the notion of an insider as an attacker formally represented as an actor **Eve** who is a malicious “evil” actor outside some set of actors within the system. Actors are represented as having a unique identity as well as a role of actor which normally is the same as their identity unless impersonation happens. Insiderhood is now represented by explicitly identifying the actor **Eve** with privileged users. Thus the malicious actor **Eve** can act like an inside actor. So far, the Isabelle Insider framework has rooted insiderhood

on a taxonomy from the insider threat literature based on psychological studies [41]. Thus, insidership was uniquely determined by the description of an insider as a system actor turning bad as a consequence of susceptible dispositions and triggering events leading to a “tipping point”.

Technically, we model this explicit yet flexible impersonation of privileged users inside the system by a function `Actor` that maps identities to roles. In places where an impersonation is deemed feasible the function may map the identity of the “evil” actor `Eve` to the same role as that of a privileged user inside the system. For all other identities that are not compromised the function actor maps these identities exclusively to roles in the system, that is, for these identities `Actor` is injective:  $id_0 \neq id_1 \Rightarrow \text{Actor } id_0 \neq \text{Actor } id_1$ .

Here, we want to extend this classical view of an intentional insider to that of an unintentional insider [44]. As Matt Bishop puts it “[i]n many cases, unintentional insider attacks are as dangerous as deliberate insider attacks; preventing them adds more complexity to an already, difficult problem. Any approach therefore must have not only a technical aspect (detecting the attack), but also a non-technical aspect (detecting the problem), which includes consideration of social, political, legal, and cultural influences, among others” [4].

We remain in the spirit of this design decision of representing the human actor but extend it with awareness and thus unintentional insidership. In the following we retrace the steps of the formal insider model as originally conceived in the Isabelle Insider framework highlighting the additions and extensions to accommodate unintentional insiders.

#### 4.5 Integrating Unaware with Malicious Insiders

For the integration of unintentional insiders with the existing the malicious insiders, e.g. [24], we extend the definitions of the types `motivations` and `psy_state` given in Section 4.2. The values for the malicious insider are based on a taxonomy from psychological insider research by Nurse et al. [41].

```
datatype psy_states = ... | depressed | disgruntled | angry | stressed
```

Another example is `motivation` for malicious insiders ranging far [41].

```
datatype motivations = ... | financial | political | revenge
                        | fun | competitive_advantage | power | peer_recognition
```

The transition to become an insider is represented by a *catalyst* that tips the insider over the edge so he acts as an insider formalized as a “tipping point” predicate.

```
definition tipping_point :: actor_state  $\Rightarrow$  bool
tipping_point a  $\equiv$  motivation a  $\neq$  {}  $\wedge$  motivation a  $\neq$  {approval_hungry}
 $\wedge$  happy  $\neq$  psy_states a
```

To embed the fact that the attacker is an insider, the actor can then impersonate other actors. This assumption entails that an insider `Actor 'Eve'` can act like their alter ego, say `Actor 'Charlie'` within the context of the locale. This is realized by the predicate `UasI`.

$$\text{UasI } a \ b \equiv (\text{Actor } a = \text{Actor } b) \wedge \\ \forall x \ y. \ x \neq a \wedge y \neq a \wedge \text{Actor } x = \text{Actor } y \longrightarrow x = y$$

Note that this predicate also stipulates that the function `Actor` is injective for any other than the identities `a` and `b`. This completes the `Actor` function to an “almost everywhere injective function”. Insiderness can now be defined as a rule that is triggered by conditions that may be valid in a state of the infrastructure. For the malicious insider, this condition has been the “tipping point” for an actor’s state (given here as the parameterized as `a`). To integrate insiderness to unintentional insiders, we simply add `unawareness` as an additional sufficient condition to the rule.

$$\text{Insider } a \ C \ as \equiv \text{tipping\_point } (as \ a) \vee \text{unaware } (as \ a) \\ \longrightarrow (\forall b \in C. \text{UasI } a \ b)$$

Although the above insider predicate is a rule, it is not axiomatized. It is just an Isabelle definition, that is, it serves as an abbreviation. To use it in an application, like the auction protocol, we can use this rule as a local assumption in theorems or using the `assumes` feature of locales [32]).

Based on the state transition and the above defined `sn_scenario`, we define the first Kripke structure.

$$\text{sn\_Kripke} \equiv \\ \text{Kripke } \{ I. \text{sn\_scenario} \rightarrow^* I \} \{ \text{sn\_scenario} \}$$

#### 4.6 Attack: Eve can get data

How do we find attacks? The key is to use invalidation [30] of the security property we want to achieve, here the global policy. Since we consider a predicate transformer semantics, we use sets of states to represent properties. The invalidated global policy is given by the following set `ssn`.

$$\text{ssn} \equiv \{x. \neg (\text{global\_policy } x \text{ ''Eve''})\}$$

The attack we are interested in is to see whether for the scenario

$$\text{sn\_scenario} \equiv \text{Infrastructure } \text{ex\_graph } \text{local\_policies}$$

from the initial state `Isn`  $\equiv \{\text{sn\_scenario}\}$ , the critical state `ssn` can be reached, that is, is there a valid attack  $(\text{Isn}, \text{ssn})$ ?

For the Kripke structure

$$\text{sn\_Kripke} \equiv \text{Kripke } \{ I. \text{sn\_scenario} \rightarrow^* I \} \text{Isn}$$

we first derive a valid and-attack using the attack tree proof calculus.

$$\vdash [\mathcal{N}_{(\text{Isn}, \text{SN})}, \mathcal{N}_{(\text{SN}, \text{ssn})}] \oplus_{\wedge}^{(\text{Isn}, \text{ssn})}$$

The set `SN` is an intermediate state where `Alice` moves to `instagram` to then put her data `''Alice's_diary''` there.

The attack tree calculus [17] exhibits that an attack is possible.

`sn_Kripke ⊢ EF ssn`

The attack tree formalisation in the Isabelle Infrastructure framework provides adequacy, that is, Correctness and Completeness theorem for the relationship between attack trees and the CTL statement [17]. We can thus simply apply the Correctness theorem `AT_EF` to immediately prove CTL-EF statements. This application of the meta-theorem of Correctness of attack trees saves us proving the CTL formula tediously by exploring the state space in Isabelle proofs. Alternatively, we could use generated code for the function `is_attack_tree` in Scala [22] to check that a refined attack of the above is valid.

## 5 Conclusions

### 5.1 Related work on awareness

Awareness contributes to having knowledge of something; thus, security awareness could be considered as a cognitive behavioural response to security and understanding its consequences. Some studies investigate this possible understanding of internet and cyber security awareness, such as Bulgur [6]. Korovessis et al. [34] introduces a “toolkit approach to information security awareness and education”, whilst focusing on organisations and the importance of user training by introducing a toolkit. Training in this sense, focuses on teaching skills to safeguard information. They completed a string of surveys, focus groups and interviews with different participant groups and ages to establish the effectiveness of the toolkit. Results showed that the prototype was successful in establishing awareness, however limitations were shown through the delivery of the approach as the kit was not accessible to everyone. Kruger and Kearney [36] establish a model prototype for assessing informational security awareness. The model focuses on knowledge, attitude, and behaviour. As stated by Lacey [38], the gap in internet security is not the technology, but fundamentally the awareness in people. The effectiveness of the approach is assessed by the resulting attitudes and behaviour to the topic.

Bada, Sasse and Nurse [2] also investigated through a psychological perspective where lack of motivation lead to poorly designed security systems and poor security compliance. The study results showed a raised awareness and had positive effects on creating a “security minded culture”. By introducing human factors to awareness campaigns, the results deemed more positive, showing us that security awareness can be increased if the tool used is more personal and relatable. Bada, Sasse and Nurse [2] provide a literature based survey on the effectiveness of campaigns on human behaviour comparing cyber security awareness campaigns in Africa and UK. They review Dolan et al’s nine critical factors which influence and change human behaviour. Although these factors provide an even finer granularity of categorizing human motivations, they are aligned with the psychological characterization by Homans [12] that we use as a basis for our model. While our work uses an experimental approach, their survey [2] also leads them to conclude that “security education has to be more than

providing information to users – it needs to be targeted, actionable, doable and provide feedback”. Our approach is aligned with their findings, since our security tool and modeling enables a “targetted, actionable, doable” analysis of a social network leading to feedback to the user.

Labuschagne et al [37] proposed a game hosted by social networking sites to increase security awareness. The game uses social networks, something that is accessible by those at home and at work. Lack of security knowledge is what makes people vulnerable and unable to protect their information, an idea clearly stated by Kritzinger and von Solms [35], with internet becoming so involved in personal lives, it is paramount that the tools to raise awareness should be accessible to all. The approach utilises a medium that is popular, therefore accessible. Whilst a prototype has not been created, the approach must be analysed to see if it would utilise the increase of public awareness to internet security. In this scenario the game that is hosted by social media sites is an approach that would possibly be interacted with by the younger audience, producing a limitation as to its non-inclusive medium, leaving a numerous amount of the public not being educated to. Jemal Abwajy 2012 [18] concludes that combined delivery methods of text, video and game would be a more suitable approach to deliver security awareness, rather than individual as it creates an inclusive audience.

## 5.2 Related work on Isabelle Insider and Infrastructure framework

A whole range of publications have documented the development of the Isabelle Insider framework. The publications [29,30,31] first define the fundamental notions of insiderness, policies, and behaviour showing how these concepts are able to express the classical insider threat patterns identified in the seminal CERT guide on insider threats [7]. This Isabelle Insider framework has been applied to auction protocols [25,26]. An Airplane case study [23,24] revealed the need for dynamic state verification leading to the extension of adding a mutable state. Meanwhile, the embedding of Kripke structures and CTL into Isabelle have enabled the emulation of Modelchecking and to provide a semantics for attack trees [16,15,14,17,20]. Attack trees have provided the leverage to integrate Isabelle formal reasoning for IoT systems as has been illustrated in the the CHIST-ERA project SUCCESS [8] where attack trees have been used in combination with the Behaviour Interaction Priority (BIP) component architecture model to develop security and privacy enhanced IoT solutions. This development has emphasized the technical rather than the psychological side of the framework development and thus branched off the development of the Isabelle *Insider* framework into the Isabelle *Infrastructure* framework. Since the strong expressiveness of Isabelle allows to formalize the IoT scenarios as well as actors and policies, the latter framework can also be applied to evaluate IoT scenarios with respect to policies like the European data privacy regulation GDPR [18]. Application to security protocols first pioneered in the auction protocol application [25,26] has further motivated the analysis of Quantum Cryptography which in turn necessitated the extension by probabilities [21,19,13].

Requirements raised by these various security and privacy case studies have shown the need for a cyclic engineering process for developing specifications and refining them towards implementations. A first case study takes the IoT health-care application and exemplifies a step-by-step refinement interspersed with attack analysis using attack trees to increase privacy by ultimately introducing a blockchain for access control [20]. This formalisation of secure distributed data labels has given rise generalising to sets of blockchain for Inter-clockchain protocols [28]. First ideas to support a dedicated security refinement process are available in a preliminary arXiv paper [33] but the first to fully formalize the RR-cycle and illustrate its application completely is the application to the Corona-virus Warn App (CWA) [27].

### 5.3 Discussion and Outlook

We have presented a pragmatic action research study into awareness in social networks. User awareness interviews have given evidence to design, implement, and test a web-based tool enabling to show the user how much is known about her. This feedback leads users to be more cautious and not give private data to social networks. In our research, we have followed the action research methodology, that is, we have used quantitative and qualitative research with practical interventions which consisted in implementing a web application tool based on social network APIs for feedbacking to users what is visible of their data. In addition, we mechanized formal modeling and analysis for social network scenarios including human actors in Isabelle. For the latter application, we have used the Isabelle Insider framework to provide a dynamic logic model enabling (1) formally reproducing the experimental scenario and (2) embedding the notion of awareness in the general security notion of insiderness. We have thus linked up social network analysis to formal security engineering and provided a novel formal notion of unintentional insiderness.

## References

1. C. M. Alvarado. Privacy and risk on the internet: A social search tool for social networking users to increase self-awareness in information sharing, 2021. BSc CS dissertation, Middlesex University London.
2. M. Bada, A. Sasse, and J. R. C. Nurse. Cyber security awareness campaign: Why do they fail to change behaviour? In *International Conference on Cyber Security for Sustainable Society*, pages 118–31, 2015. arXiv.org 1901.02672.
3. S. B. Barnes. A privacy paradox: Social networking in the united states. *First Monday*, 11(9), Sep. 2006.
4. M. Bishop, K. Nance, and J. Clark. Inside the insider threat (introduction). In *Proceedings of the 50th Hawaii International Conference on System Sciences*, page 2637, Jan. 2017.
5. J. Boender, M. G. Ivanova, F. Kammüller, and G. Primiero. Modeling human behaviour with higher order logic: Insider threats. In *STAST'14*. IEEE, 2014. co-located with CSF'14 in the Vienna Summer of Logic.

6. B. Bulgurcu, H. Cavusoglu, and I. Benbasat. Information security policy compliance: an empirical study of rationality-based beliefs and information security awareness. *MIS Quarterly*, 34(3):523–548, 2010.
7. D. M. Cappelli, A. P. Moore, and R. F. Trzeciak. *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)*. SEI Series in Software Engineering. Addison-Wesley Professional, 1 edition, Feb. 2012.
8. CHIST-ERA. Success: Secure accessibility for the internet of things, 2016. <http://www.chistera.eu/projects/success>.
9. J. Elster. *The Cement of Society*. Cambridge University Press, 1989.
10. H. Esser. *Soziologie – Allgemeine Grundlagen*. Campus, 1993.
11. C. G. Hempel and P. Oppenheim. Studies in the logic of explanation. *Philosophy of Science*, 15:135–175, April 1948.
12. G. Homans. *Social Behaviour: Its Elementary Forms*. Routledge and Kegan Paul, 1961.
13. F. Kammüller. Formalizing probabilistic quantum security protocols in the isabelle infrastructure framework. Informal Presentation at Computability in Europe, CiE 2019.
14. F. Kammüller. Formal models of human factors for security and privacy. In *5th International Conference on Human Aspects of Security, Privacy and Trust, HCII-HAS 2017*, volume 10292 of *LNCS*, pages 339–352. Springer, 2017. Affiliated with HCII 2017.
15. F. Kammüller. Human centric security and privacy for the iot using formal techniques. In *3d International Conference on Human Factors in Cybersecurity*, volume 593 of *Advances in Intelligent Systems and Computing*, pages 106–116. Springer, 2017. Affiliated with AHFE’2017.
16. F. Kammüller. A proof calculus for attack trees. In *Data Privacy Management, DPM’17, 12th Int. Workshop*, volume 10436 of *LNCS*. Springer, 2017. Co-located with ESORICS’17.
17. F. Kammüller. Attack trees in isabelle. In *20th International Conference on Information and Communications Security, ICICS2018*, volume 11149 of *LNCS*. Springer, 2018.
18. F. Kammüller. Formal modeling and analysis of data protection for gdpr compliance of iot healthcare systems. In *IEEE Systems, Man and Cybernetics, SMC2018*. IEEE, 2018.
19. F. Kammüller. Attack trees in isabelle extended with probabilities for quantum cryptography. *Computer & Security*, 87, 2019.
20. F. Kammüller. Combining secure system design with risk assessment for iot healthcare systems. In *Workshop on Security, Privacy, and Trust in the IoT, SPTIoT’19, colocated with IEEE PerCom*. IEEE, 2019.
21. F. Kammüller. Qkd in isabelle – bayesian calculation. *arXiv*, cs.CR, 2019.
22. F. Kammüller. Isabelle Insider and Infrastructure framework with Kripke structures, CTL, attack trees, security refinement, and examples including IoT, GDPR, QKD, and social networks, 2021. Available at <https://github.com/flokam/IsabelleAT>.
23. F. Kammüller and M. Kerber. Investigating airplane safety and security against insider threats using logical modeling. In *IEEE Security and Privacy Workshops, Workshop on Research in Insider Threats, WRIT’16*. IEEE, 2016.
24. F. Kammüller and M. Kerber. Applying the isabelle insider framework to airplane security. *Science of Computer Programming*, 206, 2021.

25. F. Kammüller, M. Kerber, and C. Probst. Towards formal analysis of insider threats for auctions. In *8th ACM CCS International Workshop on Managing Insider Security Threats, MIST'16*. ACM, 2016.
26. F. Kammüller, M. Kerber, and C. Probst. Insider threats for auctions: Formal modeling, proof, and certified code. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, 8(1), 2017.
27. F. Kammüller and B. Lutz. Modeling and analyzing the corona-virus warning app with the isabelle infrastructure framework. In *20th International Workshop of Data Privacy Management, DPM'20*, volume 12484 of *LNCS*. Springer, 2020. Co-located with ESORICS'20.
28. F. Kammüller and U. Nestmann. Inter-blockchain protocols with the isabelle infrastructure framework. In *Formal Methods for Blockchain, 2nd Int. Workshop, colocated with CAV'20*, Open Access series in Informatics. Dagstuhl publishing, 2020. To appear.
29. F. Kammüller and C. W. Probst. Invalidating policies using structural information. In *IEEE Security and Privacy Workshops, Workshop on Research in Insider Threats, WRIT'13*, 2013.
30. F. Kammüller and C. W. Probst. Combining generated data models with formal invalidation for insider threat analysis. In *IEEE Security and Privacy Workshops, Workshop on Research in Insider Threats, WRIT'14*, 2014.
31. F. Kammüller and C. W. Probst. Modeling and verification of insider threats using logical analysis. *IEEE Systems Journal, Special issue on Insider Threats to Information Security, Digital Espionage, and Counter Intelligence*, 11(2):534–545, 2017.
32. F. Kammüller, M. Wenzel, and L. C. Paulson. Locales – a sectioning concept for Isabelle. In Y. Bertot, G. Dowek, A. Hirschowitz, C. Paulin, and L. Thery, editors, *Theorem Proving in Higher Order Logics, 12th International Conference, TPHOLS'99*, volume 1690 of *LNCS*. Springer, 1999.
33. F. Kammüller. A formal development cycle for security engineering in isabelle, 2020.
34. P. Korovessis, S. Furnell, M. Papadaki, and P. Haskell-Dowland. A toolkit approach to information security awareness and education. *Journal of Cybersecurity Education, Research and Practice*, 2(5), 2017.
35. E. Kritzinger and S. von Solms. Cyber security for home users: A new way of protection through awareness enforcement. *Computers & Security*, 29(8):840–847, 2010.
36. H. Kruger and W. Kearney. A prototype for assessing information security awareness. *Computers & Security*, 25(4):289–296, 2006.
37. W. A. Labuschagne, I. Burke, N. Veerasamy, and M. M. Eloff. Design of cyber security awareness game utilizing a social media framework. *Information Security for South Africa*, pages 1–9, 2011. doi: 10.1109/ISSA.2011.6027538.
38. D. Lacey. *Managing the human factor in Information Security*. John Wiley & Sons Inc, 2009.
39. K. Lewin. Aktionsforschung und minderheitenprobleme. In K. Lewin, editor, *Die Lösung sozialer Konflikte*, pages 278–298. Christian-Verlag, Bad-Neuheim, 1948.
40. A. C. Myers and B. Liskov. Complete, safe information flow with decentralized labels. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, 1999.
41. J. R. C. Nurse, O. Buckley, P. A. Legg, M. Goldsmith, S. Creese, G. R. T. Wright, and M. Whitty. Understanding Insider Threat: A Framework for Characterising Attacks. In *IEEE Security and Privacy Workshops (SPW)*. IEEE, 2014.

42. L. C. Paulson. Proving properties of security protocols by induction. In *CSFW*, pages 70–83. IEEE Computer Society, 1997.
43. J. Scott. Rational choice theory. In *UNDERSTANDING CONTEMPORARY SOCIETY: THEORIES OF THE PRESENT*, pages 126–138. SAGE, 2000.
44. C. I. T. Team. Unintentional insider threats: A foundational study. Technical Report CMU/SEI-2013-TN-022, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2013.
45. M. Weber. Conceptual exposition. In *Economy and Society*. Bedminster Press, 1968.
46. M. Weber. *Wirtschaft und Gesellschaft. Grundriss der verstehenden Soziologie*. Tübingen, 1972. 5. Auflage.