A Policy-Aware Epistemic Framework for Social Networks*

Zahra Moezkarimi[1][0000−0001−5495−9098], Fatemeh Ghassemi[1,2,**][0000−0002−9677−3854], and Mohammad Reza Mousavi[3][0000−0002−4869−6794]

[1] School of Electrical and Computer Engineering, University College of Engineering, University of Tehran, North Kargar st., Tehran, Iran {zmoezkarimi,fghassemi}@ut.ac.ir

[2] School of Computer Science, Institute for Reseach in Fundamental Sciences, PO. Box 19395-5746, Tehran, Iran

[3] Department of Informatics, Kings College London, London, UK, mohammad.mousavi@kcl.ac.uk

Abstract. We provide a semantic framework to specify information propagation in social networks; our semantic framework features both the operational description of information propagation and the epistemic aspects in social networks. In our framework, based on annotated labelled transition systems, actions are decorated with function views to specify different types of announcements. Our function views enforce various common types of local privacy policies, i.e., those policies concerning a single action. Furthermore, we specify global privacy policies, those concerning multiple actions, using a combination of modal μ-calculus and epistemic logic. To illustrate the applicability of our framework, we apply it to the specification of a real-world case study. As a fundamental property for the epistemic aspect of our semantic model, we prove that its indistinguishability relations are equivalence relations, namely, they are reflexive, symmetric, and transitive. We also study the complexity bounds for the model-checking problem concerning a subset of our logic, and show that model checking is PSPACE-complete for the studied subset.

Keywords: Social networks · Privacy · Epistemic logic · Operational semantics · Dynamic epistemic logic.

1 Introduction

Motivation. Online social networks are an indispensable part of the modern life and people regularly use them to communicate and socialise, share information, and even do business. Some of the most popular social networks, such as Facebook and WhatsApp, have more than one billion users each [38]. Information and influence propagation [14] is a challenging problem in social networks, which has been considered from different perspectives [19, 1, 20]. Analysing propagation is useful for a variety of applications such as fake news- and fashion diffusion, disease epidemics, and viral marketing.

An important aspect of information propagation is privacy, which has received significant interest from the scientific community in the recent years [35, 48]. The concept of privacy was initially introduced as "the right to be left alone" [45] and is recognised as a fundamental human right by many international norms and regulations [44]. Privacy is further elaborated by Alan Westin as "the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated" [46]; this is currently known as the informational self-determination right and used as a means to limit the abuse of personal data [44]. Privacy is one of the main concerns in the digital age. However, because of the pervasive use of social networks, most people tend to reveal their personal information there despite their privacy concerns. Different stakeholders try to address this conflict between privacy concerns and the pervasive use of social networks differently; such stakeholders include digital rights activists, social network designers and engineers, and data privacy specialists. Despite their different interests

* Mohammad Reza Mousavi has been partially supported by the UKRI Trustworthy Autonomous Systems Node in Verifiability, Grant Award Reference EP/V026801/1.

** Corresponding author

in this conflict and in any attempt to reach a solution, these stakeholders will eventually benefit from more rigorous concepts, definitions and specifications. In this context, laying a rigorous mathematical foundation for analysing privacy in social networks is a significant and relatively understudied subject. Privacy is an inherently knowledge-related property and thus, epistemic frameworks, such as epistemic logic [18], provide a natural means to specify and reason about privacy. An important advantage of using epistemic knowledge in such contexts is that it provides an effective means to not only reason about local knowledge of agents, but also about their inferred knowledge, i.e., what they can learn about the behaviour of other agents and their privacy policies based on what they observe locally.

There have been some earlier approaches in the literature to incorporate the use of epistemic logic in the specification and reasoning frameworks for social networks [6, 43, 42, 39]. We provide an overview of these approaches in the related work section in this paper. However, to the best of our knowledge, none of these pieces of work consider privacy policies. The only notable exception is the work of Pardo, Balliu, and Schneider [39], in which privacy policies have been formalised using a combination of different formalisms. However, the epistemic aspects have been assumed to be provided as a plug-in component to their theory; the present paper builds a rigorous foundation for those aspects that are underspecified in the aforementioned approach. To summarise, our paper bridges a major gap in the landscape of formal semantics of social networks; namely, it provides an epistemic semantic model for social networks and as such, enables formal verification of privacy in social networks using epistemic approaches. By a semantic model, we mean providing a model that can be used to define the semantics of different formalisms. From a privacy perspective, our main focus is on the ability to define, enforce, and verify privacy policies, which can be used by different stakeholders with different perspectives. For example, social network designers can define templates for local privacy policies and enforce them using the notion of decorated actions; users can choose their policies from a number of possible options; and all stakeholders can check the desired properties pertaining to the global privacy policies over the network.

Contributions. In this paper, we propose a semantic framework for social networks. Our first step is to define a formal semantic model, called SNSM, merging the operational and epistemic aspects of social networks. Our formal semantic model builds upon the well-known operational model of labelled transition systems representing system state changes and actions causing them; actions, representing communication, are decorated with a function view (along the lines of [15, 30]) to allow for their epistemic interpretation indicating different types of announcements and local privacy policies of social network users. To make the semantic model applicable to the analysis of social networks, we also include a multi-layer graph-based model of relationships in social networks. Using function views and graph models of the social network, we construct an indistinguishability relation that enables epistemic reasoning, among others, about privacy aspects. A combination of modal μ-calculus and epistemic logic is used to specify global privacy of social networks (and any other desired property). We prove some knowledge-related properties of our semantic framework and analyse a real-world case study. A discussion on the complexity of our model-checking problem for a subset of our logic is provided as well.

Running example. We illustrate our approach using a running example concerning the threshold model in social networks. Agents can adopt a behaviour (e.g., choose a product, select a style, or vote for candidate) in online social networks. This adoption can be influenced by others who are connected to those agents in the network and can be described by Threshold Models [20]. In Threshold Models, based on the graph of social relationships, a threshold value is defined to quantify the influence of agents on each other. We use the Linear Threshold Model [26] in our running example, but depending on the context, other models for explaining propagation phenomena such as Independent Cascade- and Epidemic models [14, 27] can be adopted as well. Our approach is also applicable to other realistic and larger examples, which are discussed towards the end of the paper. Our running example is informally specified below.

Example 1 (Simple Social Network (SSN).) *Consider a simple social network (SSN), having six agents u_1 to u_6 in the social network graph in Fig. 1. An online event is advertised to happen* *soon and each agent announces join when she decides to participate in that event. (Throughout the rest of the paper, we use pronouns "she" and "they" to refer to agents, regardless of their gender.) We distinguish between two types of agents: determined and undetermined. Determined agents (in our case: u_1 and u_2), represented by double circles, join the event based on their own initiative. However, undetermined agents (u_3 to u_6) only join the event under certain circumstances, which have to do with a "participation threshold"; namely, they join only if they know that two agents have already decided to join the event. SSN announcements are organised in such a way that when agents make a join announcement, only their neighbours in the social network graph can notice their action.*

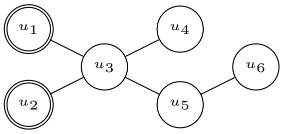

Fig. 1. Topology of the Simple Social Network SSN.

In Example 1 the threshold for undetermined agents is fixed at two for all undetermined agents. For the sake of brevity, in this example, we assume that agents will not leave the event once they decide to join and therefore the model is monotonic (in the set of joined agents) and hence, will reach a fixed point from the social network perspective. However, our framework allows for generalisations of this example that involve more sophisticated action and threshold models. We formalise and elaborate on various aspects of this running example throughout the rest of the paper.

Structure of the paper. The rest of this paper is structured as follows. In Section 2, we provide some preliminary definitions. We introduce our approach for specifying local privacy policies using decorated actions in Section 3. In Section 4, we formally define our semantic model SNSM. We use a logic based on modal μ-calculus and epistemic logic in Section 5 to specify the global privacy policies in social networks. To illustrate the applicability of our approach, we investigate a case study from a real-world social network in Section 6. We discuss the complexity of verification of global policies in Section 7. A comprehensive overview of related work is provided in Section 8. Finally, we conclude the paper by summarising the main contributions, and discuss the avenues of future work in Section 9.

2 Preliminaries

In this section, we define the basic concepts that are used to define our framework throughout the rest of the paper. To start with, we define the notion of *social graph*; social networks are usually multi-layer, where each layer represents a specific relation among agents, such as friend, follower, and blocked, depending on the application. The following notion of *social graph* formalises this intuition.

Definition 2 (Social Graph) *A social graph is a pair $(Id, \{r_1, \ldots, r_n\})$ where Id is the set of agent identifiers and each $r_i \subseteq Id \times Id$ is a set of pairs of identifiers, called edges. We use SG to denote the set of social graphs.*

In our application domain, it often comes handy to project a social graph on a particular relation, e.g., to analyse the propagation of a specific type of information. This concept is formalised below.

Definition 3 (Projection Function) *Consider a social graph $sg = (Id, \{r_1, \ldots, r_n\})$; its projection on relation r_i, denoted by $sg \downarrow_{r_i} : Id \to \mathcal{P}(Id)$, is the function satisfying that for all $x, y \in Id$, $y \in sg \downarrow_{r_i} (x)$ if and only if $(x, y) \in r_i$.*

To illustrate these concepts, consider an Instagram application with two relations among agents: follower and blocked. Its social graph can be modelled by $\mathit{Instagram} = (\mathit{Id}, \{\mathit{Followers}, \mathit{Blocked}\})$, where the relation *Followers* defines which agent is following which, while *Blocked* denotes the agents blocked by another agent. The followers of the agent $\mathit{Bob} \in \mathit{Id}$ are hence denoted by $\mathit{Instagram} \downarrow_{\mathit{Followers}} (\mathit{Bob})$.

We model the operational behaviour of systems in terms of actions. These actions provide an abstract model for the evolution of social networks and communications among agents. The actions are decorated with function views denoting for each agent the type and amount of information released to her.

Definition 4 (Decorated Action [15, 37]) *Consider two sets A and C of actions and contexts, respectively; a decorated action $d \in \mathit{DAct}$ is a pair (a, f) in which $a \in A$ is an atomic action and $f : A \times C \to A$ is a renaming function that maps each pair of action and context to an action. Action a is used to model a communication action and f is used to denote how it is perceived under different contexts. A particular action $\tau \in A$ is designated as an unobservable action; it is used to model perfect information hiding in our semantics.*

This definition of context provides us with a flexible way of representing how different communications are perceived by different agents in a social network. For instance, a straightforward choice of context is to let $C = \mathit{Id}$ [37] (the set of agent identifiers, introduced in Definition 2); this way, one can model what every agent in *Id* observes regarding each and every action. By studying several case studies in the domain of social networks, we settled upon the following definition for the contexts in this domain: $C = \mathit{Id} \times \mathit{SG}$, where *Id* is the set of agent identifiers in the social network, and *SG* denotes the set of social graphs. We henceforth use *DAct* to denote the set of decorated actions in the context of social networks, as specified above. For the sake of readability, we remove the internal brackets and write $f : \mathit{DAct} \times \mathit{Id} \times \mathit{SG} \to \mathit{DAct}$, where we mean $f : \mathit{DAct} \times (\mathit{Id} \times \mathit{SG}) \to \mathit{DAct}$. We demonstrate the usefulness of our chosen definition of context by providing several instances of this function for different types of local privacy policies in various social networks in the remainder of this paper.

For example, suppose we would like to define a policy in the *Instagram* social network, by which only the followers of *Bob* are allowed to see the action *story* and others do not even notice that an action has happened. In this case, we define $f(\mathit{story}, i, \mathit{Instagram}) = \mathit{story}$, for all $i \in \mathit{Instagram} \downarrow_{\mathit{Followers}} (\mathit{Bob})$ and $f(\mathit{story}, j, \mathit{Instagram}) = \tau$, for all other $j \notin \mathit{Instagram} \downarrow_{\mathit{Followers}} (\mathit{Bob})$. The idea is that after performing *story*, it appears to any agent *Alice*, who is not a follower of *Bob*, as if no action has taken place at all.

The notion of context could be adopted for and adapted to other specific domains, such as healthcare and education, in order to define domain-specific policies. The appearance of actions will be justified according to the nature of the contexts in these domains and their specific relationships and structures [8].

Our proposed semantic model is based on an extension of labelled transition systems, called annotated labelled transition system (ALTS) [15], defined below.

Definition 5 (Annotated Labelled Transition System) *An annotated labelled transition system (ALTS) is a quadruple $\langle St, \to, Ind, s_0 \rangle$, where St is the set of operational states, $\to \subseteq St \times D \times St$ is the transition relation where D denotes the set of decorated actions, $Ind \subseteq St \times Id \times St$ is the set of indistinguishability relations from the point view of agents in which Id is the set of agent identifiers, and s_0 is the initial state.*

In the above-given definition, indistinguishability relations are used to denote the uncertainty of agents regarding the past actions. They are constructed formally in the remainder of this paper, using the renaming functions available in the decorated actions.

Example 6 (Annotated Labelled Transition System for SSN) *Two states of an ALTS for the SSN in Example 1 have been depicted in Fig.2; this figure specifies how an SSN evolves when the agent u_1 joins an event, modelled by the action join_1. We define the appearance of join_1*

to be the unobservable action τ for all u_i, where $i \in \{2,4,5,6\}$; this intuitively means that no other agent except u_3 explicitly takes note of agent u_1 joining the event. This is reflected by the indistinguishability relations for these agents between the source and target states of the transition, meaning that from the point view of each u_i, these two states are the same. Note that although these two states are indistinguishable, future events in the social network may reveal this fact (u_1 joining the event) to others due to some known consequences (e.g., publicly posting a photo from the event). This (knowledge by inference) constitutes a fundamental difference between explicit- and inferred knowledge [22].

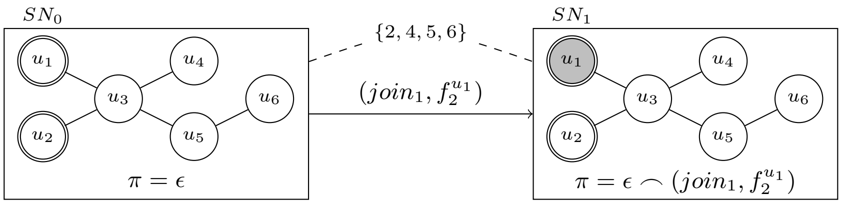

Fig. 2. A portion of semantic model SNSM for the diffusion process in SSN

Indistinguishability relations formalise the knowledge and the epistemic uncertainty of agents. A set of properties is used to capture the fundamental properties of knowledge (e.g., positive and negative introspection and truth); these properties, given below, sometimes are collectively called $S5$ properties, for historical reasons [23]. Note that the properties of knowledge vary in different applications. In this paper, we use the properties of $S5$ modal logic [23], which is sufficient to model truthful knowledge. Note that our use of $S5$ modal logic still allows for hiding and delimiting the communicated facts, due to privacy policies in our operational model. Elsewhere [37] a subset of authors have investigated a similar framework with the possibilities of lies and hence, untruthful belief. Introducing this possibility in our framework remains an area of future work.

Definition 7 (Knowledge Properties of $S5$ modal logic) *For a set S, the relation $R \subseteq S \times S$ is an equivalence relation and satisfies the knowledge properties of $S5$ modal logic [23], when it is:*

- *Reflexive. For all $i \in S$, we have that $(i,i) \in R$.*
- *Symmetric. For all $i,j \in S$, if $(i,j) \in R$ then $(j,i) \in R$.*
- *Transitive. For all $i,j,k \in S$, if $(i,j) \in R$ and $(j,k) \in R$, then $(i,k) \in R$.*

Intuitively, the reflexive property means that if an agent knows that ϕ (in state s), then ϕ is true (in the state s). The symmetric property means that if ϕ holds in a state, the agent knows that she does not know $\neg\phi$ (regardless of whether she knows or does not know ϕ). Transitivity means that if an agent knows that ϕ, she knows that she knows that ϕ holds. For a more formal treatment of $S5$ logics, we refer to Chapter 3 of the standard text by Fagin, Halpern, Moses, and Vardi [23].

3 Local Privacy Policies

In this section, we explain how local privacy policies are formally specified. Subsequently, we introduce a set of common templates for local privacy policies inspired by our survey of the common policies in the domain.

3.1 Formal Specification of local Privacy Policies

Online social networks provide various means for agents to specify their desired local privacy policies. For example in WhatsApp, agents can choose among three options regarding those who can see their "*last seen*" status: 1) Every one, 2) My contacts, and 3) Nobody[4]. In Instagram, one can block (or unblock) agents by first visiting their profile and then clicking on Block (or Unblock). When you block an agent, she is not notified about this. [5] The point here is that while the blocking action has some consequences, it is not directly visible to other agents, i.e., in our formalism, its appearance is τ for all other agents. We show in the remainder of this section, that the decorated actions in Definition 4 provide a very expressive means for specifying local privacy policies.

Definition 8 (Formal Specification of Local Privacy Policy) *A local privacy policy on an action a is specified by the renaming function $f : A \times Id \times SG \to A$. Upon occurrence of the action (a, f) in the social network with the graph sg, the agent id observes $f(a, id, sg)$.*

For instance in our running example, one could specify her local privacy policy regarding the action *join*; this will involve specifying what other agents will perceive if *join* is performed. A set of local privacy policy options for the action *join* is formally specified in the following example.

Example 9 *There are three options for local privacy policies P_1, P_2, and P_3 for each agent u_i that owns the privacy policy, and their corresponding renaming functions $f_1^{u_i}$ to $f_3^{u_i}$ for the action join in the SSN of Example 1. Assume that the social graph is specified as $SSN = (\{u_1, \dots, u_6\}, \{Friend\})$. Renaming functions $f_1^{u_i}$ to $f_3^{u_i}$ capture these local privacy policies as follows:*

- *P_1: Nobody can see u_i's join action*

$$f_1^{u_i}(join, id, SSN) = \begin{cases} join & id = u_i \\ \tau & otherwise \end{cases}$$

- *P_2: Only u_i's friends can see u_i's join action.*

$$f_2^{u_i}(join, id, SSN) = \begin{cases} join & id \in (SSN \downarrow_{Friend} (u_i)) \\ \tau & otherwise \end{cases}$$

- *P_3: All agents can see u_i's join action.*

$$f_3^{u_i}(join, id, SSN) = \begin{cases} join & id \in \{u_1, \dots, u_6\} \\ \tau & otherwise \end{cases}$$

There are different kinds of announcements (message exchanges) among agents in a network. In Example 9, policy P_1 turns the action *join* to something only visible for the agent performing it. Policy P_2 specifies that the action is a private announcement for the friends of the agent u_i and P_3 specifies a public announcement policy. One can refine these policies further; for example, consider the following policy P_4, specified by the renaming function $f_4^{u_3}$ where action *join* is only visible to the friends of u_3, apart from u_2:

$$f_4^{u_3}(join, id, SSN) = \begin{cases} join & id \in (SSN \downarrow_{Friend} (u_3)) \setminus \{u_2\} \\ \tau & \text{otherwise} \end{cases}$$

Thus, u_2 will not notice the occurrence of $(join, f_4^{u_3})$ as $f_4^{u_3}(join, u_2, SSN) = \tau$.

In commonly-used social networks, the choices of local privacy policies are limited; below we show how most common local privacy policies can be captured using some generic templates for our renaming function. In particular, we show how the above-specified policies are examples of such templates.

[4] `https://www.whatsapp.com/privacy` (Accessed: 6 August 2021).

[5] `https://help.instagram.com/454180787965921` (Accessed: 6 August 2021).

3.2 Local Privacy Policy Templates

Most local privacy policies in social network follow generic templates. In Table 1, we provide an inventory of these templates inspired by those in real-world social networks. One can certainly extend this table as needed and such an extension will not affect our semantic model, presented in the next section.

Table 1. Templates for local privacy policies in social networks

Template	*Renaming Function*	*Description*
Graph-based with exception	$GB^u_{r,e}(a, id, sg) = \begin{cases} a & id \in (sg \downarrow_r (u)) \setminus e \\ \tau & \text{otherwise} \end{cases}$	Agents who have a relationship type r (e.g. friendship) with the agent u in the social graph except those in e can see the action.
Private communication	$Pr^u_V(a, id\ sg) = \begin{cases} a & id \in \{u\} \cup V \\ \tau & \text{otherwise} \end{cases}$	Only the sender and the agents in V can see the action.
Fully private	$FP^u(a, id, sg) = \begin{cases} a & id = u \\ \tau & \text{otherwise} \end{cases}$	Nobody can see the action except the agent itself.
Public announcement with exception	$PA^u_{Id,e}(a, id, sg) = \begin{cases} a & id \in Id \setminus e \\ \tau & \text{otherwise} \end{cases}$	Everybody, except for those in e can see the action .

Template *graph-based with exception* ($GB_{r,e}$), specified below, is parametrised with relation r and allows for an additional exception set $e \subset Id$:

$$GB^u_{r,e}(a, id, sg) = \begin{cases} a & id \in (sg \downarrow_r (u)) \setminus e \\ \tau & \text{otherwise} \end{cases}$$

For instance, policy P_2 in Example 9 is an instance of this template on relation *Friend*, i.e., $GB^{u_i}_{Friend,\emptyset} = f^{u_i}_2$. Policy P_4 specified by $f^{u_2}_4$ is another instance of this by instantiating $e = \{u_3\}$, i.e., $GB^{u_2}_{Friend,\{u_3\}} = f^{u_2}_4$. The exception relation can be computed from the relations in the given social graph. As an example in Instagram, agent u can allow her action *story* to be viewed by all of her followers except for the blocked one. To enforce this policy, we adjust the parameter e to $Instagram \downarrow_{Blocked} (u)$, resulting in the renaming function $GB^u_{Followers,Instagram\downarrow_{Blocked}(u)}$.

Template *private communication* (Pr_V), parametrised by $V \subset Id$, denotes a policy in which only agent u (adopting the policy) and the agents in V can see the action. This template is helpful when an agent wants to communicate privately with a group of agents.

$$Pr^u_V(a, id\ sg) = \begin{cases} a & id \in \{u\} \cup V \\ \tau & \text{otherwise} \end{cases}$$

Template *Fully private* (FP) specifies that nobody except the agent itself can see the action. The policy P_1 can be easily specified by this template, i.e., $FP^{u_i} = f^{u_i}_1$. Note that we can consider the template *Fully private* as an special instance of the template *private* Pr^u_V by setting the set V to empty.

$$FP^u(a, id, sg) = \begin{cases} a & id = u \\ \tau & \text{otherwise} \end{cases}$$

Finally, the template *public announcement with exception* ($PA_{Id,e}$) means that everybody, identified by the set Id, except for those in $e \subseteq Id$ can see the action. Therefore, the policy P_3 can be specified by this template, i.e., $PA^{u_i}_{\{u_1,\ldots,u_6\},\emptyset} = f^{u_i}_3$.

$$PA^u_{Id,e}(a, id, sg) = \begin{cases} a & id \in Id \setminus e \\ \tau & \text{otherwise} \end{cases}$$

To show how these templates can be applied to capture local privacy policies in common social networks, we specify the available local privacy policies of WhatsApp in Table 2, according to the WhatsApp official website. [6] In this table, we specify how each local privacy policy can be specified using our templates from Table 1.

Table 2. Local privacy policy templates for some selected actions in WhatsApp

Action	Description	Privacy options	Specification
Last seen	see when I last opened my WhatsApp application	Everyone	$PA_{Id,\, WhatsApp\downarrow_{Blocked}}$
		My contact	$GB_{Contacts,\, WhatsApp\downarrow_{Blocked}}$
		Nobody	FP
Groups	add me to groups	Everyone	$PA_{Id,\, WhatsApp\downarrow_{Blocked}}$
		My contacts	$GB_{Contacts,\, WhatsApp\downarrow_{Blocked}}$
		My contact except e	$GB_{Contacts,\, WhatsApp\downarrow_{Blocked} \cup e}$
Profile Photo	see my profile photo	Everyone	$PA_{Id,\, WhatsApp\downarrow_{Blocked}}$
		My contacts	$GB_{Contacts,\, WhatsApp\downarrow_{Blocked}}$
		Nobody	FP
Status	see my status updates	My contacts	$GB_{Contacts,\, WhatsApp\downarrow_{Blocked}}$
		My contacts except e	$GB_{Contacts,\, WhatsApp\downarrow_{Blocked} \cup e}$
		Only Share with V	Pr_V

3.3 Partial Order on Local Privacy Policies

We define a partial order relation on local privacy policies to compare them with respect to their information leakage. We measure information leakage for each action based on its intended audience, as specified below.

Definition 10 (Partial Order on Local Privacy Policies) *For two local privacy policies f and f', f' leaks at least as much information as f, if and only if for any action $a \in A$ and agent identifier $i \in Id$ such that the appearance of action a for i by applying f is non-τ, the appearance of a is the same by applying f' for that agent:*

$$f \sqsubseteq f' \quad iff \quad \forall i \in Id, \forall a \in A, \forall sg \in SG. f(a, i, sg) \neq \tau \implies f(a, i, sg) = f'(a, i, sg).$$

We use this partial order in the remainder of this paper in order to compare local privacy policies when reasoning about global privacy policies. For instance, it holds that $FP^u \sqsubseteq Pr^u_V$; intuitively, this means that Pr^u does not leak more information than Pr^u_V and moreover, Pr^u agrees with Pr^u_V whenever it does reveal some information. Formally, we have that for each action a, the appearance of a is τ for all agents except for u according to FP^u and only for u, $FP^u(a, u, sg) = a$; while according to Pr^u_V, we have that $Pr^u_V(a, id, sg) = a$ for $id \in \{u\} \cup V$ and it is τ otherwise. This means that a has the same appearance for u in both policies (and is visible to more agents according to Pr^u_V).

4 Social Network Semantic Model (SNSM)

Our proposed semantic model (SNSM) is an instance of ALTS, specified in Definition 5, addressing temporal and epistemic aspects of social networks in tandem. Intuitively, it can be seen as a Kripke structure [23] that is equipped with transitions among possible worlds and indistinguishability relations which are built based on the history of actions.

In [39], the dynamic behaviour of the network is specified through operational semantics. The rules can be epistemic, topological, policy-related, or hybrid. Therefore, for each action, one has to specify a set of SOS (Structured Operational Semantics [41]) rules indicating the effects of that action. By having transitions and indistinguishability relations in SNSM, we generalise their approach.

Common to some of the earlier approaches [6, 39], the states of our ALTS semantics (henceforth called *configurations*, also to distinguish them from the local state of agents) include the

[6] `https://www.whatsapp.com/privacy` (Accessed: 7 August 2021).

social graph. Transitions among configurations are caused by (communication) actions in the social network which models the dynamic behaviour of the network. To be able to reason about the epistemic aspects of the social network, configurations also comprise a path π encoding the actions that have been performed up to the current configuration. Therefore a configuration is a combination of the social graph and the corresponding path to that point. To establish indistinguishability relations among configurations, we only consider those paths that are built from the initial configuration. From a social network perspective, a configuration is composed of the social graph, the local states of the agents, and the history of what has happened on the social network.

Definition 11 (Social Network Semantic Model (SNSM)) *A social network semantic model SNSM is a quadruple of the form $\langle \mathit{Conf}, \rightarrow, \mathit{Ind}, \mathit{conf}_0 \rangle$ in which:*

- *$\mathit{Conf} \subseteq SG \times \mathcal{S} \times \mathit{DAct}^*$ is the set of configurations (sg, s, π) each comprising:*
  - *$sg \in SG$ is the social graph,*
  - *$s \in \mathcal{S} = \prod_{n \leq N} S^n$ is the vector comprising the local states of all agents, where S is the set of local states of each agent and N is the maximum number of agents in the specification (possibly different in different configurations), and*
  - *$\pi \subseteq \mathit{DAct}^*$ is the history of decorated actions;*
- *$\rightarrow \subseteq \mathit{Conf} \times \mathit{DAct} \times \mathit{Conf}$ is the transition relation;*
- *$\mathit{Ind} \subseteq \mathit{Conf} \times \mathit{Id} \times \mathit{Conf}$ is the indistinguishability relation; and*
- *conf_0 is the initial configuration.*

Upon the occurrence of an enabled action from a configuration, not only the relations among agents in the current social graph may change, but also their local states and their knowledge may be updated. (The latter aspect will be demonstrated next when we present the indistinguishability relation among configurations.) Therefore, our proposed semantics supports dynamicity in topology, local state of agents, and epistemic aspects.

Example 12 (Configurations of the SSN.) *A portion of the SNSM for the social network in Example 1 is depicted in Fig. 2. In each configuration, SN_0 and SN_1, the social graph and a history of actions are maintained. The local state of each agent can be represented by a Boolean value indicating whether the agent will participate in the advertised event or not (i.e., has announced join or not). We illustrate the local states of agents by colouring the nodes in the social graph (i.e., white versus gray, where gray denotes a true value). The initial configuration is SN_0 and the corresponding path is ϵ. We assume that the local privacy policy P_2 from Example 9 has been adopted by all the agents in SSN; as a result, only their friends can see their actions and other agents should not notice anything. Action $join_1$ is an announcement by agent u_1 meaning that she will participate in the advertised event. This action is decorated by the renaming function $f_2^{u_1}$ to enforce P_2. The appearance of the message $join_1$ is τ for all agents except u_3 (the only friend of u_1). Since non-friend agents do not notice anything, configurations SN_0 and SN_1 are indistinguishable for u_2 and u_4 to u_6; hence, there are indistinguishability relations with labels $\{2, 4, 5, 6\}$ between them. The formal definitions leading to these indistinguisability relations are presented in the remainder of the section.*

Indistinguishability relations of the semantics are constructed mechanically based on the path part of the configurations using the rules defined below. We lift the relation among the paths to the configuration by the rule given in Definition 15. For more readability, we use $\cdots$ for indistinguishability relations between paths and ---, defined further on, for indistinguishability relations between configurations.

Definition 13 (Indistinguishability Relations) *The indistinguishability relation $\cdots \subseteq \mathit{DAct}^* \times \mathit{Id} \times \mathit{DAct}^*$ is the smallest relation defined by the following deduction rules. We let $\pi \overset{id}{\cdots} \pi'$ denote*

$(\pi, id, \pi') \in \cdots$, *where* $id \in Id$.

$$(\epsilon)\frac{}{\epsilon \overset{id}{\cdots} \epsilon} \qquad (\mathbf{match})\frac{\pi \overset{id}{\cdots} \pi' \quad f(a, id, sg) = f'(b, id, sg)}{\pi \frown (a, f) \overset{id}{\cdots} \pi' \frown (b, f')}$$

$$(\mathbf{hid_0})\frac{\pi \overset{id}{\cdots} \pi' \quad f(a, id, sg) = \tau}{\pi \frown (a, f) \overset{id}{\cdots} \pi'} \qquad (\mathbf{hid_1})\frac{\pi \overset{id}{\cdots} \pi' \quad f(a, id, sg) = \tau}{\pi \overset{id}{\cdots} \pi' \frown (a, f)}$$

Deduction rule (ϵ) is self-explanatory. Deduction rule (**match**) specifies that if an agent observes $f(a, id, sg)$, she is uncertain whether she may be in another run of the social network, if the corresponding run has been indistinguishable from the run in the source state and the run ends with the same observable action. This has an implicit assumption that the observing agent has the set of all plausible local privacy policies in her local knowledge. This gives us the possibility of modelling various possible scenarios: if the local privacy policies of agents are common knowledge (an assumption made in some prior work [39]), then we only model outgoing transitions with the actual renaming functions, reflecting the local privacy policies known to all. If they are private, we model each action with all possible local privacy policies (renaming function templates) to model that the observing agent does not know which local privacy policy is applied. One can certainly model other sorts of assumptions about the knowledge of local privacy policies by adopting a modelling strategy that falls in between the two extremes sketched above. In addition, the rule (**match**) preserves ignorance, i.e., if an agent u does not know whether she is in states s_1 or s_2, and an α-transition is possible from both s_1 and s_2 leading to states s'_1 and s'_2, respectively; then the uncertainty will be kept between the two new states s'_1 and s'_2 for the agent u, as well. Deduction rules ($\mathbf{hid_0}$) and ($\mathbf{hid_1}$) state that if an action a is hidden from an agent, i.e., its appearance is τ, then she should not notice anything at all when a happens.

In most practical cases, deduction rule (**match**) is not used: in practice, τ is used whenever agents want to apply a local privacy policy, i.e., to hide an action from an unintended audience. However, in some experimental social networks (e.g., an extension of Diaspora for photo sharing [40]), agents may need to strip off some pieces of information from a message to have delimited information leakage. In such cases, illustrated by the following examples, deduction rule (**match**) comes in handy.

Example 14 (Controlled Information Leakage in Local Privacy Policies.) *Consider Alice's friend Sara who does not know at the moment whether Alice is in Germany or in Turkey. Alice hides her location and posts a message post("eating Baklava")! Consider the local privacy policy hl that replaces any action post("eating x") with post("eating ..."), where x is any specific type of food that may reveal the agent's location. In this particular case, it holds that hl(post("eating Baklava"), id, sg)*
= post("eating ..."). Adopting this local privacy policy and according to deduction rule (**match**)*, the run after eating Baklava is indistinguisable for Sara from the run after which Alice performs post("eating Blutwurst"), where in both cases she will see "eating ..." on her screen.*

The indistinguishability relation is lifted from paths to configurations in the following definition.

Definition 15 (Operational Semantics of a Process) *Given the initial social graph* sg_0 *and the initial states of agents* $s_0 = (s_1, \ldots, s_n)$*, the indistinguishability relation* $\dashrightarrow \subseteq Conf \times Id \times Conf$ *for* $conf_0 = (sg_0, s_0, \epsilon) \in Conf$ *is the smallest relation satisfying the following deduction rule:*

$$(\mathbf{ind})\frac{conf_0 \rightarrow^* (sg, s, \pi) \quad conf_0 \rightarrow^* (sg', s', \pi') \quad \pi \overset{id}{\cdots} \pi'}{(sg, s, \pi) \overset{id}{\dashrightarrow} (sg', s', \pi')}$$

where $\rightarrow^*$ *is the reflexive and transitive closure* $\bigcup_{d \in |DAct|} \xrightarrow{d}$.

The semantics of a configuration $conf_0 \in Conf$ *is defined as an ALTS with* $conf_0$ *as the initial state,* $(\bigcup_{d \in |DAct|} \overset{d}{\rightarrow})$ *as its transition relations and* $(\bigcup_{id \in Id} \overset{id}{\dashrightarrow})$ *as its indistinguishability relations.*

In order to analyse global privacy policies in our framework, it is essential that we consider in Definition 15 only those paths that are built from the initial configuration. Otherwise the indistinguishability relation may lead to unreachable configurations and ruin the analysis. We also assumed that for each agent, only visible actions modify the locally visible part of the social graph and hence, in our lifting from paths to configurations, we left the social graphs unconstrained, as longs the paths leading to them are indistinguishable.

Example 16 (Semantic Model of the SSN.) *The semantic model of SSN in Example 1 is illustrated in Fig. 3. For the sake of readability, the contents of each configuration and self-loops for indistinguishability relations have been omitted. In this model, the set of configurations Conf is* $\{SN_1, SN_2, SN_3, SN_4, SN_5, SN_6\}$*. If an agent* u_i *announces* $join_i$*, we have a transition labelled by* $(join_i, f_2^{u_i})$ *since the adopted policy by all the agents is* P_2*. The protocol terminates when there is no other agent to be influenced by her friends (in order to join the event). The indistinguishability relations on configurations semantics are established using the deductions rules in Definitions 13 and 15.*

The initial configuration is SN_0*. Since the path of this configuration is* ϵ*, based on the rule* (ϵ)*, there is a self-loop on* SN_0*. As only* u_1 *and* u_2 *are determined agents, initially only two transitions* $(join_i, f_2^{u_i})$ *to* SN_i*, where* $i \in \{1, 2\}$*, are possible. (Just to recall, determined agents join the event based on their own initiative and undetermined agents only join the event under certain circumstances, which have to do with a "participation threshold". In the SSN, the "participation threshold" means that undetermined agents, i.e.,* u_3 *to* u_6 *in the Example 1, join only if they know that two agents have already decided to join the event.) We have previously explained the indistinguishability relation between the two configurations* SN_0 *and* SN_1 *in Fig. 2. Upon occurrence of* $join_2$*, only* u_3 *will see the actual message in* SN_2 *as she is the only friend of* u_2*. In configurations* SN_1 *and* SN_2*, only* $join_2$ *and* $join_1$ *are possible, respectively, leading to configurations* SN_3 *and* SN_4*, respectively. In these two configurations* SN_3 *and* SN_4*, the threshold conditions, explained in Example 1, are met for the undetermined agent* u_3 *and therefore, she will adopt the decision and decide to participate in the event by performing* $join_3$*. As* u_6 *cannot see this action in these configurations,* SN_3 *and* SN_4 *are indistinguishable from the resulting configurations* SN_5 *and* SN_6 *for* u_6*. In* SN_5 *and* SN_6 *the threshold conditions are not met for any other agents and therefore the propagation will stop.*

For indistinguishability relations defined on paths created by the rules from Definition 13, we prove that they are equivalence relations in Lemma 17. Then, we prove the equivalence properties for the relations resulted from Definition 15.

Lemma 17 (Equivalence of Indistinguishability on Paths) *The indistinguishability relations from Definition 13 are equivalence relations.*

Proof. The proof is divided into the three sections of the reflexive property, symmetric property, and transitivity:

- Reflexive property. We need to prove, for each $\pi \in DAct^*$, $\pi \overset{i}{\cdots} \pi$. This can be done by induction on the length of the path π.
  - Base Case: $n = 0$. When $\pi = \epsilon$, then $n = 0$ and due to the rule (ϵ) we have $\epsilon \overset{i}{\cdots} \epsilon$.
  - Induction Hypothesis: We assume that $|\pi| = n$, and $\pi \overset{i}{\cdots} \pi$.
  - Induction Step: Using the induction hypothesis $\pi \overset{i}{\cdots} \pi$, and $\pi' = \pi \frown (a, f)$, by applying the rule (**match**) on $\pi \overset{i}{\cdots} \pi$, since both sides are expanded with an action with the same appearances, we have that $\pi' \overset{i}{\cdots} \pi'$ holds.

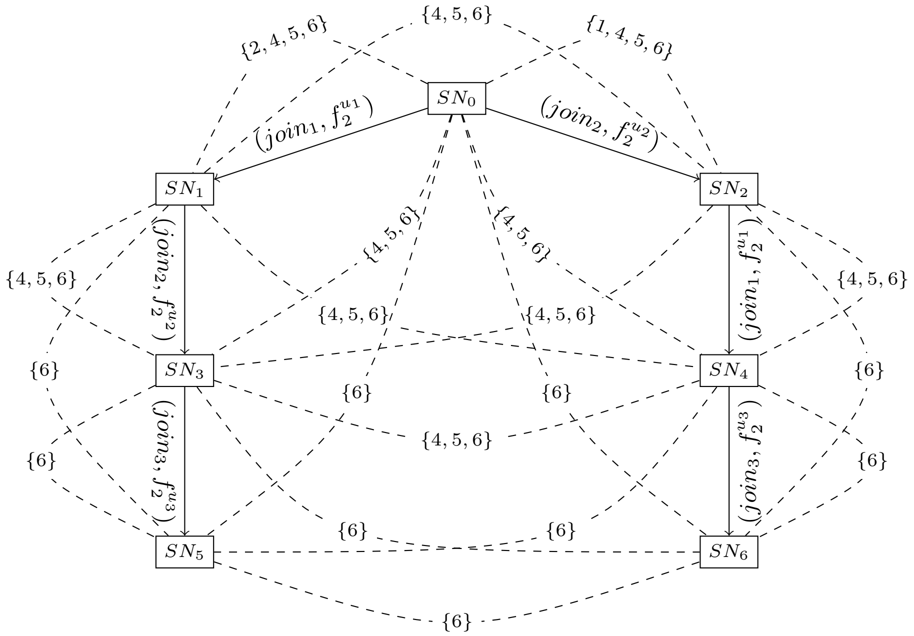

Fig. 3. Semantic model of Simple Social Network SSN.

- Symmetric property. We need to prove that for each $\pi, \pi' \in DAct^*$, if $\pi \overset{i}{\cdots} \pi'$, then $\pi' \overset{i}{\cdots} \pi$. We prove this by induction on the depth of the proof leading to $\pi \overset{i}{\cdots} \pi'$ and $\pi' \overset{i}{\cdots} \pi$. We distinguish the following cases based on the last deduction rule that has been applied for $\pi \overset{i}{\cdots} \pi'$ from Definition 13.
  - (ϵ). Then $\pi = \epsilon$ and $\pi' = \epsilon$. By this rule, we have that $\pi' \overset{i}{\cdots} \pi$.
  - (**match**). Then $\pi = \pi_0 \frown (a, f)$ and $\pi' = \pi'_0 \frown (b, f')$ for some π_0 and π'_0 such that $\pi_0 \overset{i}{\cdots} \pi'_0$ and $f(a, id, sg) = f'(b, id, sg)$. Using $\pi_0 \overset{i}{\cdots} \pi'_0$ and the induction hypothesis, we have that $\pi'_0 \overset{i}{\cdots} \pi_0$. By applying the deduction rule (**match**), we will obtain $\pi'_0 \frown (b, f') \overset{i}{\cdots} \pi_0 \frown (a, f) \implies \pi' \overset{i}{\cdots} \pi$.
  - ($\mathbf{hid_0}$). Then, $\pi = \pi_0 \frown (a, f)$ for some π_0 such that $f(a, id, sg) = \tau$ and $\pi_0 \overset{i}{\cdots} \pi'$. By the induction hypothesis we have that $\pi' \overset{i}{\cdots} \pi_0$. Having $\pi' \overset{i}{\cdots} \pi_0$, and $f(a, id, sg) = \tau$, by applying the deduction rule ($\mathbf{hid_1}$), we obtain $\pi' \overset{i}{\cdots} \pi_0 \frown (a, f) \implies \pi' \overset{i}{\cdots} \pi$.
  - ($\mathbf{hid_1}$). Then, $\pi' = \pi'_0 \frown (b, f')$ for some π'_0 such that $f'(b, id, sg) = \tau$ and $\pi \overset{i}{\cdots} \pi'_0$. Having $\pi'_0 \overset{i}{\cdots} \pi$ by the induction hypothesis, and $f'(b, id, sg) = \tau$, by applying the deduction rule ($\mathbf{hid_0}$), we can obtain $\pi'_0 \frown (b, f') \overset{i}{\cdots} \pi \implies \pi' \overset{i}{\cdots} \pi$.
- Transitivity. We have to prove, for each $\pi, \pi', \pi'' \in DAct^*$, that if $\pi \overset{i}{\cdots} \pi'$, and $\pi' \overset{i}{\cdots} \pi''$, then $\pi \overset{i}{\cdots} \pi''$. We prove this by induction on the depth of the proofs leading to $\pi \overset{i}{\cdots} \pi'$, and $\pi' \overset{i}{\cdots} \pi''$. We distinguish the following cases based on the last deduction rule used to derive these relations in Definition 13. The first level of items (represented by solid circles) are the last deduction rules that have been applied to obtain $\pi \overset{i}{\cdots} \pi'$. The second level of items (represented by dash) are the last deduction rules that have been applied to obtain $\pi' \overset{i}{\cdots} \pi''$.
  - (ϵ) Then, $\pi = \pi' = \epsilon$. $\pi' \overset{i}{\cdots} \pi''$ can be due to one of the following deduction rules (since the path is not ϵ in the left hand side of the rules ($\mathbf{hid_0}$) and (**match**)):

  - (ϵ): We have that $\pi'' = \epsilon$ and hence, it follows from (ϵ) that $\pi \overset{i}{\cdots} \pi''$.
  - $(\mathbf{hid_1})$. Then, $\pi'' = \pi''_0 \frown (c, f'')$ for some π''_0 such that $\pi'_0 \overset{i}{\cdots} \pi''_0$ and $f''(c, id, sg) = \tau$. From $\pi_0 \overset{i}{\cdots} \pi'_0$, $\pi'_0 \overset{i}{\cdots} \pi''_0$, and the induction hypothesis, it follows that $\pi_0 \overset{i}{\cdots} \pi''_0$. From the latter statement and $f''(c, id, sg) = \tau$, by applying $(\mathbf{hid_1})$, we will obtain $\pi_0 \overset{i}{\cdots} \pi''_0 \frown (c, f'') \implies \pi_0 \overset{i}{\cdots} \pi'' \frown \tau \implies \pi \overset{i}{\cdots} \pi''$.
- $(\mathbf{match})$: Then, $\pi = \pi_0 \frown (a, f)$ and $\pi' = \pi'_0 \frown (b, f')$ for some π_0, π'_0 such that $\pi_0 \overset{i}{\cdots} \pi'_0$ and $f(a, id, sg) = f'(b, id, sg)$. We distinguish the following cases based on the last deduction rule used in the proof of $\pi' \overset{i}{\cdots} \pi''$ (since $\pi' = \pi'_0 \frown (b, f')$ the deduction rules (ϵ) and $(\mathbf{hid_1})$ can not be the cases):
  - $(\mathbf{match})$: Then, $\pi'' = \pi''_0 \frown (c, f'')$ for some π''_0 such that $\pi'_0 \overset{i}{\cdots} \pi''_0$ and $f'(b, id, sg) = f''(c, id, sg)$. From $\pi_0 \overset{i}{\cdots} \pi'_0$, $\pi'_0 \overset{i}{\cdots} \pi''_0$, and the induction hypothesis, it follows that $\pi_0 \overset{i}{\cdots} \pi''_0$. Since $f(a, id, sg) = f'(b, id, sg)$ and $f'(b, id, sg) = f''(c, id, sg)$, therefore $f(a, id, sg) = f''(c, id, sg)$. From the latter statement and $\pi_0 \overset{i}{\cdots} \pi''_0$, by applying $(\mathbf{match})$, we will obtain $\pi_0 \frown (a, f) \overset{i}{\cdots} \pi''_0 \frown (c, f'') \implies \pi \overset{i}{\cdots} \pi''$.
  - $(\mathbf{hid_0})$. Then, $f'(b, id, sg)$ and $\pi'_0 \overset{i}{\cdots} \pi''$. From $\pi_0 \overset{i}{\cdots} \pi'_0$, $\pi'_0 \overset{i}{\cdots} \pi''$, and the induction hypothesis, it follows that $\pi_0 \overset{i}{\cdots} \pi''$. Since $f(a, id, sg) = f'(b, id, sg)$ and $f'(b, id, sg) = \tau$, therefore $f(a, id, sg) = \tau$. From the latter statement and $\pi_0 \overset{i}{\cdots} \pi''$, by applying $(\mathbf{hid_0})$, we will obtain $\pi_0 \frown (a, f) \overset{i}{\cdots} \pi'' \implies \pi \overset{i}{\cdots} \pi''$.
- $(\mathbf{hid_0})$. Then, $\pi = \pi_0 \frown (a, f)$ for some π_0 such that $\pi_0 \overset{i}{\cdots} \pi'$ and $f(a, id, sg) = \tau$. By assumption, we have that $\pi' \overset{i}{\cdots} \pi''$. Having $\pi_0 \overset{i}{\cdots} \pi'$ and $\pi' \overset{i}{\cdots} \pi''$, using the induction hypothesis we obtain $\pi_0 \overset{i}{\cdots} \pi''$. By the latter, $f(a, id, sg) = \tau$ and applying the deduction rule $(\mathbf{hid_0})$, we have $\pi \overset{i}{\cdots} \pi''$.
- $(\mathbf{hid_1})$. Then $\pi' = \pi'_0 \frown (b, f')$ for some π'_0 such that $\pi \overset{i}{\cdots} \pi'_0$ and $f'(b, id, sg) = \tau$. We distinguish the following cases based on the last deduction rule used in the proof of $\pi' \overset{i}{\cdots} \pi''$: (since $\pi' = \pi'_0 \frown (b, f')$, the deduction rule (ϵ) can not be the case)
  - $(\mathbf{match})$: Then, $\pi'' = \pi''_0 \frown (c, f'')$ for some π''_0 such that $\pi'_0 \overset{i}{\cdots} \pi''_0$ and $f'(b, id, sg) = f''(c, id, sg)$. From $\pi_0 \overset{i}{\cdots} \pi'_0$, $\pi'_0 \overset{i}{\cdots} \pi''_0$, and the induction hypothesis, it follows that $\pi_0 \overset{i}{\cdots} \pi''_0$. Since $f'(b, id, sg) = f''(c, id, sg)$ and $f'(b, id, sg) = \tau$, therefore $f''(c, id, sg) = \tau$. From the latter statement and $\pi \overset{i}{\cdots} \pi''_0$, by applying the deduction rule $(\mathbf{hid_1})$, we will obtain $\pi \overset{i}{\cdots} \pi''_0 \frown (c, f'') \implies \pi \overset{i}{\cdots} \pi''$.
  - $(\mathbf{hid_0})$. Then, $\pi'_0 \overset{i}{\cdots} \pi''$. From $\pi \overset{i}{\cdots} \pi'_0$, $\pi'_0 \overset{i}{\cdots} \pi''$, and the induction hypothesis, it follows that $\pi \overset{i}{\cdots} \pi''$.
  - $(\mathbf{hid_1})$: Then $\pi'' = \pi''_0 \frown (c, f'')$ for some π''_0 such that $\pi' \overset{i}{\cdots} \pi''_0$ and $f''(c, id, sg) = \tau$. By assumption we have that $\pi \overset{i}{\cdots} \pi'$. By the latter, $\pi' \overset{i}{\cdots} \pi''_0$, and induction hypothesis, we have $\pi \overset{i}{\cdots} \pi''_0$. Using $f''(c, id, sg) = \tau$, we can now apply the deduction rule $(\mathbf{hid_1})$, and obtain $\pi \overset{i}{\cdots} \pi''$.

■

In Theorem 18, we show that indistinguishability relations --- between configurations are equivalence relations.

Theorem 18 (Equivalence of Indistinguishability on Configurations) *The indistinguishability relations in Definition 15 are equivalence relations.*

Proof. Using Lemma 17, we know that the indistinguishability relation $\cdots$ defined on paths in Definition 13 is an equivalence relation, i.e., it is reflexive, symmetric, and transitive. Therefore, by application of the rule (**ind**), the result is extended to configurations. ■

This theorem implies that the agents only believe in facts (that hold) and they are conscious about what they know and what they do not know. This also paves the way for relating our semantic framework to those used in model checking results for Dynamic Epistemic Logic with S5 models, as we will show in section 7.

Next, we can check our desired knowledge properties related to Example 1 on the model given in Fig. 3.

Example 19 *Global Properties of the SSN. Consider the property that "all agents will eventually know whether the event will be held". It can be shown that this property does not hold in the SNSM semantic model of SSN depicted above, because there are indistinguishability relations labeled by* 6 *between all configurations, i.e., all of them are indistinguishable from the perspective of agent u_6. Hence, agent u_6 will always hold it plausible that nothing has happened at all and hence, cannot form any knowledge about the event.*

We formalise global privacy policies in the next section, which enable us to specify and verify the informal observations made in the example given above.

5 Global Privacy Policies

Privacy is not always guaranteed through hiding individual actions; often agents can infer implicit or explicit knowledge by considering the sequence of observed actions. We use the term local privacy (and local privacy policies) to refer to the former aspect of privacy, i.e., those aspects that can be ensured by defining the visibility and information leakage in singular actions. The latter aspect of privacy that concerns the interaction of multiple actions is called global privacy [32, 47]. Global privacy and its corresponding policies constitute the subject matter of this section.

As an example of a global privacy violation, we relate the following real scenario in a version of Instagram: when the non-blocked agent *Alice* replays back a story of the agent *Bob*, a temporarily blocked agent for that story that follows both *Alice* and *Bob* will know about the story. A desired global privacy policy should ensure that if an agent is blocked from seeing a special story, she should never get to know about that story. Although in this scenario none of the local privacy policies of the agents have been violated, the inferred information violates the above-mentioned global privacy policy.

Global privacy policies are formulated and verified on the semantic models of social networks. In this section, we first introduce a logical formalism to specify global privacy policies. This formalism enables us to specify such policies as properties on knowledge of agents and the occurrences of actions in the past. So, we can verify that the effects of the actions do not lead to any leakage. Then, we define the semantics of our logic. Finally, similar to local privacy policies, we provide typical templates for global privacy policies. This sets the scene for model checking global privacy policies on our case study. We discuss the complexity of verification of global properties on our models in the next sections.

5.1 Syntax of Logic

Our logical framework is a combination of the modal μ-calculus with past, on the one hand, and epistemic logic, on the other hand. We use the modal μ-calculus due to its expressiveness, which allows us to specify various patterns of temporal and epistemic and combinations thereof [11]. The grammar of logical formulae is given below.

$$\phi ::= \top \mid Y \mid \bigwedge_{j \in J} \phi_j \mid \neg\phi \mid \langle\langle a, f\rangle\rangle\phi \mid \langle\overline{\langle a, f\rangle}\rangle\phi \mid K_i\phi \mid \nu Y.\phi(Y)$$

$$\textit{(if } Y \textit{ occurs only positively in } \phi\textit{)},$$

In the above-given grammar, logical connectives have their traditional meanings. We note that $\bigwedge_{j\in J}\phi_j$ enables us to have any number of conjuncts. In fact, it is a class of operators indexed by J. This is a convenient notation leading to concise specifications when the number of conjuncts is not fixed (e.g., when ranging over an arbitrary number of policies). In this paper, since our specifications are finitely branching and we have finitely many agents, actions and policies, we only need a finite number of conjunctions and assume J to be finite henceforth. However, one can use this construct for infinitely branching systems as well. In such a case, the forthcoming complexity analysis needs to be revisited. Y stands for the class of recursive variables ranged over by $x, y, x_0, \ldots$; $\langle(a,f)\rangle\phi$, means that it is possible to perform a transition labelled with decorated action (a,f), after which the formula ϕ holds; $\langle\overline{(a,f)}\rangle\phi$, means that in the past an (a,f)-transition has been made, before which the formula ϕ held; $K_i\phi$ means that agent i knows that ϕ and $\nu Y.\phi(Y)$ denotes the maximal fixed point of the equation $Y = \phi(Y)$.

In order to facilitate the specification of logical properties, we define and use some abbreviations and enrich our syntax with regular expressions which will be described in the following.

Abbreviations For notational convenience, we define the following abbreviations for commonly used logical formulae:

$$\begin{array}{ll}[(a,f)]\phi & \doteq \neg\langle(a,f)\rangle\neg\phi\\ \bigvee_{j\in J}\phi_j & \doteq \neg\bigwedge_{j\in J}\neg\phi_j\\ \mu Y.\phi & \doteq \neg\nu Y.\neg\phi \text{ with } \neg Y = Y\end{array}$$

To explain briefly, $[(a,f)]\phi$ means that after all (a,f)-transitions ϕ holds (this formula holds vacuously if no (a,f) is enabled at the current configuration), disjunction is known from propositional logic, $\mu Y.\phi$ stands for the minimal fixed point (often used to indicate holding a formula after/before a finite path). Here, $\neg Y = Y$ is a standard definition meaning that when we are applying a negation operator, we stop when we reach the recursive variable [11]. Other abbreviations, following the temporal logic tradition, can be defined as follows:

$$\begin{array}{ll}\langle(a,-)\rangle\phi & \doteq \bigvee_{i=1}^{k}\langle(a,f_i)\rangle\phi\\ AG\ \phi & \doteq \nu Y.\phi\wedge(\bigwedge_{a\in A}[(a,-)]Y)\\ EF\ \phi & \doteq \neg AG\neg\phi\\ \phi^{\Lsh} & \doteq \mu Y.\phi\vee(\bigvee_{a\in A}\langle\overline{(a,-)}\rangle Y)\\ \langle(a,f)^c\rangle\phi & \doteq \bigvee\limits_{b\in A, b\neq a\vee f\neq f'}\langle(b,f')\rangle\phi\end{array}$$

Assuming that action a has k possible policy adoptions $f_1,\ldots,f_k$, we use $\langle(a,-)\rangle\phi$ for $\bigvee\limits_{i=1}^{k}\langle(a,f_i)\rangle\phi$, which means that it is possible to perform a transition labelled with the action a decorated with any local policy (no matter which policy has been adopted), after which the formula ϕ holds. Similarly, $[(a,-)]$ can be defined using conjunction. $AG\ \phi$ means that ϕ holds everywhere, $EF\ \phi$ means that there is a finite run after which ϕ will eventually hold, and $\phi^{\Lsh}$ means that somewhere in the past ϕ held. Finally, $\langle(a,f)^c\rangle\phi$ means that it is possible to perform a transition labelled with any decorated action except for (a,f) (or any (a,f') such that $f = f'$), after which the formula ϕ holds. The notion of identity that is used in $\langle(a,f)^c\rangle\phi$ is defined as follows (inequality between local privacy policies is defined dually):

$$f = f' \quad \mathit{iff} \quad \forall a\in A, \forall sg\in SG, i\in Id\cdot f(a,sg,i) = f'(a,sg,i).$$

Actions and Regular formulas The next set of abbreviations are dedicated to using regular expressions in modal formulae. This notion was first introduced in Propositional Dynamic Logic (PDL) [24] and extended by Logics of communication and change (LCC) [10] in the form of DEL with factual change. For the sake of readability, we extend the modalities below with regular expressions for sets and sequences of decorated actions:

$$\begin{aligned} R &::= \epsilon \mid \alpha \mid R.R \mid R + R \mid R^* \mid R^+ \\ \alpha &::= \top \mid (a, f) \mid \alpha \cap \alpha \mid \alpha \cup \alpha \mid \alpha^c \end{aligned}$$

The action formula $\top$ denotes the set of all actions A and α^c is the complement of α, i.e., $\{(b, f') \mid \forall (a, f) \in \alpha : \ (a \neq b \ \vee \ f \neq f')\}$. A regular expression R could be either empty, or a decorated action, or a composition of regular expressions. By some abuse of notation, we consider (a, f) as the singleton $\{(a, f)\}$. $R.R$ and $R + R$ stand for the concatenation and choice of regular expressions, respectively. R^* stands for zero or more occurrences of the R while R^+ stands for one or more occurrences of the R. The syntactical abbreviations for commonly used regular expressions are given below:

$$\begin{aligned}
\langle \alpha \rangle \phi &\doteq \textstyle\bigvee_{(a,f)\in\alpha} \langle (a, f) \rangle \phi \\
[\ \alpha\ ]\phi &\doteq \textstyle\bigwedge_{(a,f)\in\alpha} [\ (a, f)\ ]\phi \\
\langle \epsilon \rangle \phi &\doteq \ \ [\epsilon]\phi \ = \ \ \phi \\
\langle R_1 + R_2 \rangle \phi &\doteq \langle R_1 \rangle \phi \ \vee \ \langle R_2 \rangle \phi \\
[\ R_1 + R_2\ ]\phi &\doteq [R_1]\phi \ \wedge \ [R_2]\phi \\
\langle R_1.R_2 \rangle \phi &\doteq \langle R_1 \rangle \langle R_2 \rangle \phi \\
[\ R_1.R_2\ ]\phi &\doteq [R_1][R_2]\phi \\
\langle\ R^*\ \rangle \phi &\doteq \mu Y.(\phi \vee \langle R \rangle Y) \\
\langle\ R^+\ \rangle \phi &\doteq \langle\ R.R^*\ \rangle \phi
\end{aligned}$$

For example $\langle R_1 + R_2 \rangle \phi$ means that it is possible to perform R_1 or R_2 and end up in a state where formula ϕ holds.

5.2 Semantics of Logic

The semantics of our epistemic logic follows the common existing semantics and has no peculiarities. Namely, the semantics is verified with respect to the L which is a social network semantic model defined by Definition 11 and a given current configuration $c \in \mathit{Conf}$. The semantic definitions for various constructs of our logic are given below.

$$\begin{array}{lll}
\mathrm{L}, c \models \top & \text{iff} & \text{always} \\
\mathrm{L}, c \models \bigwedge_{j\in J} \phi_j & \text{iff} & \mathrm{L}, c \models \phi_j \text{ for each } j \in J \\
\mathrm{L}, c \models \neg\phi & \text{iff} & \mathrm{L}, c \models \phi \text{ is not true} \\
\mathrm{L}, c \models \langle (a, f) \rangle \phi & \text{iff} & \text{there is an } c' \in \mathit{Conf},\ c \xrightarrow{(a,f)} c' \text{ and } \mathrm{L}, c' \models \phi \\
\mathrm{L}, c \models \langle \overline{(a, f)} \rangle \phi & \text{iff} & \text{there is an } c' \in \mathit{Conf},\ c' \xrightarrow{(a,f)} c \text{ and } \mathrm{L}, c' \models \phi \\
\mathrm{L}, c \models K_i\ \phi & \text{iff} & \text{for all } c' \in \mathit{Conf} \text{ such that} \\
 & & c \overset{i}{\text{---}} c' \text{ it holds that } \mathrm{L}, c' \models \phi \\
\mathrm{L}, c \models \nu X.\phi(X) & \text{iff} & c \in \bigcup\{C' \subseteq \mathit{Conf} \mid \forall c' \in C', \mathrm{L}, s' \models \phi(X := C')\}
\end{array}$$

The semantics of $\top$, conjunction, negation, and maximal fixed point are standard. The semantics of $\langle (a, f) \rangle \phi$ specifies that from the current configuration c, a configuration c' satisfying ϕ can be reached by an (a, f)-labelled transition Dually, $\langle \overline{(a, f)} \rangle \phi$ specifies that the current configuration c can be reached by an (a, f)-labelled transition from a configuration c' satisfying ϕ. The semantics of the knowledge operator $K_i\ \phi$ specifies that for all indistinguishable configurations c' (that are reachable from the current configuration), ϕ should hold.

5.3 Global Policy Templates

In this section, we specify some generic examples of global privacy policies that concern personal data (such as location and birthday), policy (such as friendships and blocking), or protocol(such as retaining or erasing certain information), detailed below:

1. Data: the agents' inferred knowledge about privacy-sensitive information,
2. Policy: the agents' knowledge about occurrences of actions that can reveal local privacy policies; note that as opposed to enforcing local privacy policies, in this section, we check whether local privacy policies are themselves kept private despite agents' inference, and
3. Protocol: assurances about occurrences of actions that concern storage and manipulations of privacy-sensitive data in protocols.

Data. In social networks, actions may carry explicit or implicit personal data. Our first set of global privacy policies templates specify that the occurrence of data-carrying actions should not be known to a specific group of agents.

Example 20 *Consider Example 1, which we extend below. Assume that agent u decides to attend event e, of which the location is publicly known . The participation of agent u in the event e is represented by action $attend(u, e)$. If somebody knows that this action has taken place, then the location of u is also revealed. Assuming that the global policy informally specifies that the location of agents should not be revealed without their explicit consent, its formalisation in our framework should specify that unless u announces her participation, the occurrence of $attend(u, e)$ in the past should never be inferred by any agent.*

Let f_{NotLoc} be the local privacy policy that hides the action $attend(u, e)$ from all non-friend agents and A_{Loc} be the set of actions that, if executed, will reveal the user's location. We define γ^c to be the set of decorated actions that have an action with a policy f' such that $f' \sqsubseteq f_{NotLoc}$ (such as Fully private (FP) or private communication Pr^u_V where V is a subset of u's friends). The global policy is formalised as follows (we provide two equivalent formulations of this policy using regular expressions and fixed points):

$$(RE) \quad [(\gamma^c)^*.(attend(u,e), f_{NotLoc}).(\gamma^c)^*] \bigwedge_{i \in nInt} \neg K_i(((\langle\overline{attend(u,e), -}\rangle \top)^{\Lsh})$$

where $nInt = \{i \mid i \in Id \cdot f(attend(u,e), i, sg) \neq attend(u,e)\}$.

$$(FP) \quad \nu X.[\gamma^c]X \wedge [(attend(u,e), f_{NotLoc})]\big(\nu Y.\big(\bigwedge_{i \in nInt} \neg K_i(\langle\overline{attend(u,e), -}\rangle \top^{\Lsh})\big) \wedge [\gamma^c]Y\big)$$

The above-given equivalent formulae express that whenever $(attend(u, e), f_{NotLoc})$ occurs, $\neg K_i(\langle\overline{attend(u, e), -}\rangle \top^{\Lsh})$ must hold as long as no other action in A_{Loc} explicitly revealing u's location occurs.

The formulae in Example 20 only consider the data-carrying action $attend(u, e)$. Below we generalise these formulae into a template for a set δ of data-carrying actions and an f^* local privacy policy that reveals these actions to a desired set of agents (e.g., f_{NotLoc}). We define the set of decorated actions revealing information to non-intended audiences as a function of δ and f^*, denoted by $\Gamma_{(\delta,f^*)}$; based on these building blocks, our global privacy policy is defined in terms of the following fixed point formula:

$$\Gamma_{(\delta,f^*)} = \{(a, f) \mid a \in \delta \wedge f^* \sqsubseteq f\}$$

The action-based global policy template for the set δ and local policy f^* is defined as:

$$\nu Y. \bigwedge_{a \in \delta, f^*(a,i) \neq a} \neg K_i(\langle\overline{a, -}\rangle \top^{\Lsh}) \wedge [\Gamma^c_{(\delta,f^*)}]Y$$

This formula specifies that neither at the current moment, nor after any action not in $\Gamma_{(\delta,f^*)}$, the occurrence of actions in δ will ever be known.

Policy. In some scenarios for preventing information leakage, adopted local privacy policies by the agents should be kept hidden from an unintended audience. The following example illustrates how these types of global privacy policies can be specified in our framework.

Example 21 *For example, if the agent u adopts a local privacy policy that excludes u' from receiving certain data, a global privacy policy may specify that u' should never learn about this (change of) policy by u. In other words, agent u' cannot find out that any action has happened that was hidden from u':*

$$AG \bigwedge_{a\in A, f: f(a,u')=\tau} \neg K_{u'}(\langle\overline{(a,f)}\rangle\top^{\Lsh})$$

Protocol. The third class of global privacy policies specify when certain actions, pertaining to storage or removal of data, should always / never occur in certain scenarios. The following example illustrates how a template for such properties can be constructed using our logic.

Example 22 *Consider a GDPR-like policy, which enforces that upon the occurrence of a request action req by a user, an erasure action er should happen immediately by the system and no one among the agents in soc working for the social network company should be able to recall the occurrence of a certain data-carrying action a. This template is specified as follows:*

$$AG[(req,-)](\langle(er,-)\rangle\top \wedge [(er,-)^c]\bot \wedge AG \bigwedge_{i\in soc} \neg K_i(\langle\overline{(a,-)}\rangle\top^{\Lsh}))$$

In the above example, we assume that the set of *soc* is fixed. If there was a change in *soc* -due to hiring or firing employees- then we have to define the set *soc* for each state, modify the formula and verify it for that state.

6 Case Study

In this section, we specify a model of WhatsApp and perform a formal analysis of the model to illustrate the applicability of our framework. To this aim, we first specify the semantic model of a scenario in WhatsApp and then show how by verifying an epistemic property corresponding to a global privacy policy, a realistic case of information leakage can be identified.

6.1 Information leakage in WhatsApp

Despite the attempts to preserve local privacy policies, there may still be some leakage of epistemic knowledge in social networks. Such leakages are difficult to identify and reason about as they are not about explicit flow of information, as demonstrated below.

WhatsApp uses different types of checkmarks next to messages to indicate the different kinds of status for read receipts. In particular, there are three kinds of status:

1. one checkmark means that the message was successfully sent by the sender,
2. two gray checkmarks mean that the message was successfully received at the recipient's side, and
3. two blue checkmarks mean that the recipient has read the message.

For the first status, if the sender does not see one checkmark, it means that the message has not yet been successfully sent from the sender side. For the second status, if the sender does not see two (gray) checkmarks, it means that the message is not successfully received at the recipient's side. This could happen for two reasons:

- the recipient might have blocked the sender, or
- the recipient's phone might be off or she may have poor connectivity.

For the third status, it is stated that if the sender does not see two blue checkmarks next to the sent message, there could be several reasons such as having disabled read receipts in the privacy settings, connectivity issues, having blocked the sender or having turned off the (receiving) phone. [7]

We consider a message exchange scenario between two agents u_1 and u_2 in WhatsApp. Agent u_1 wants to send two messages m_1 and m_2 to agent u_2. We distinguish three main scenarios for these message communications:

1. a normal message communication: first, message m_1 is successfully sent and received and then m_2 is successfully sent and received, and
2. a scenario in which the mobile phone of the agent u_2 as the receiver is off at the moment of sending the messages or has poor connectivity, and
3. a scenario in which agent u_2 has blocked agent u_1, before u_1 sends m_1 but u_2 unblocks u_1 before u_1 sends m_2.

Other scenarios for these message exchanges can also be considered.

We define the social graph of *WhatsApp* with the relation *Blocked*. The SNSM model for the above-specified behaviour is illustrated in Fig. 4. For the sake of readability, we have not shown the social graphs, the local states, and paths in the configurations; moreover, self-loops are also omitted. Since the states with one checkmark and two blue checkmarks statuses for read receipt have no contribution in our example and to fit the ALTS here, we have abstracted them away in our model by omitting their corresponding actions. Action $\mathit{Send}(m_i)$ means that agent u_1 sends a message m_i to u_2. Action $\mathit{Block}(u_2, u_1)$ ($\mathit{UnBlock}(u_2, u_1)$) means that u_2 has blocked (unblocked) u_1. As the consequence of $\mathit{Block}(u_2, u_1)$ ($\mathit{UnBlock}(u_2, u_1)$), the social graph is updated such that u_1 is added to (removed from) $\mathit{WhatsApp} \downarrow_{\mathit{Blocked}} (u_2)$. Note that other actions have no effect on the social graph. Action $\mathit{DCh}(m_i)$ means that u_2 has received the message m_i so u_1 has seen two (gray) checkmarks next to the message m_i. We modify the private communication template to consider the social relation r of the receiving agents in the policy:

$$Pr^{u}_{V,r}(a, id, sg) = \begin{cases} a & id \in (\{u\} \cup V) \wedge u \notin (sg \downarrow_r (id)) \\ \tau & \text{otherwise} \end{cases}$$

We define the local policy $f^{\diamondsuit} = Pr^{u_1}_{\{u_2\},\mathit{Blocked}}$ by which u_1 sends its messages. We assume that both agents have adopted the policy Fully private (FP) for the actions $\mathit{Block}(u_2, u_1)$, $\mathit{UnBlock}(u_2, u_1)$, and $\mathit{DCh}(m_i)$.

In the initial configuration, no blocked relation has been formed among the agents, i.e., $\mathit{WhatsApp} \downarrow_{\mathit{Blocked}} (u_2) = \emptyset$. The left-most branch models the normal scenario, the middle branch addresses the second scenario in which the mobile phone of the agent u_2 is turned off, while the right-most branch specifies the third, i.e., blocking, scenario. At the left-most branch, after agent u_1 sends the message m_i, $i \in \{1, 2\}$, to u_2, she will receive a double checkmarks since the message is immediately delivered to u_2. As both agents u_1 and u_2 can see the action $\mathit{Send}(m_i)$ due to the policy $f^{\diamondsuit}$, there is no indistinguishability relation between the source and target of $\mathit{Send}(m_i)$-transitions (e.g. SN_0 and SN_1). In the right-most branch, agent u_2 has decided to block agent u_1. However, agent u_1 does not know that agent u_2 has blocked her and sends message m_1 to u_2. To see this, note that action $\mathit{Block}(u_2, u_1)$ will not be visible to agent u_1 and therefore by applying ($\mathbf{hid_1}$), configurations SN_0 and SN_2 become indistinguishable for u_1. We remark that the social graph of SN_2 is achieved by adding the blocked relation from u_2 to u_1, i.e., $\mathit{WhatsApp} \downarrow_{\mathit{Blocked}} (u_2) = u_1$, to the social graph of SN_0. Also, by applying (**match**), configurations SN_1 and SN_5 become indistinguishable for u_1.

After sending m_1, action $\mathit{UnBlock}(u_2, u_1)$ will not be visible to agent u_1 and therefore by applying ($\mathbf{hid_1}$), configurations SN_1 and SN_8 become indistinguishable for u_1. Similarly, by ($\mathbf{hid_1}$), configurations SN_5 and SN_8 become indistinguishable for u_1. Note that the social graph of SN_8

[7] `https://faq.whatsapp.com/android/security-and-privacy/how-to-check-read-receipts/?lang=en`(Accessed: 19 August 2021).

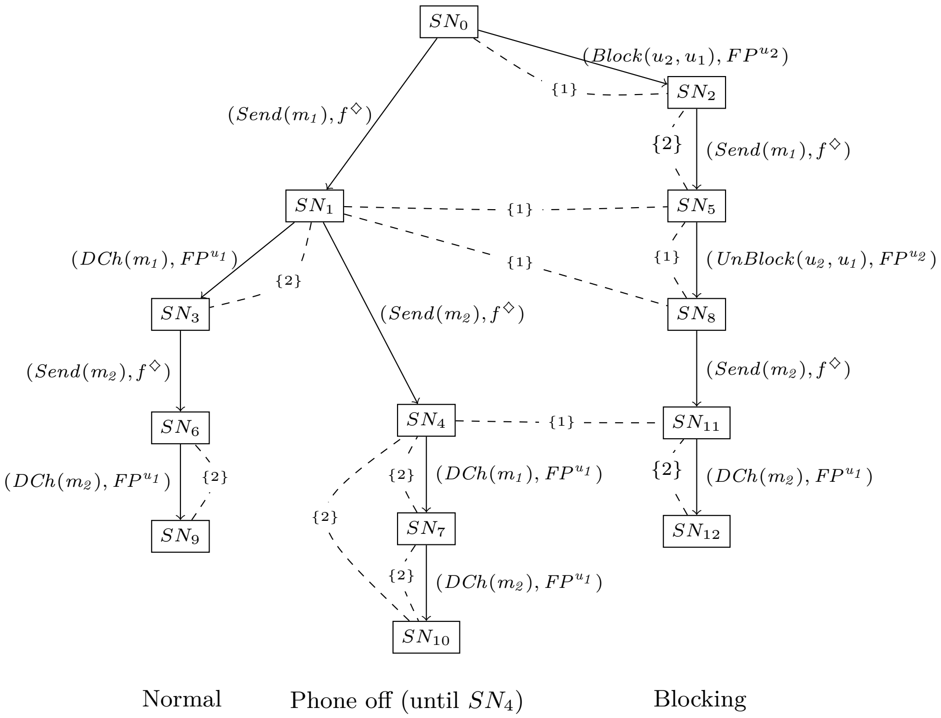

Fig. 4. A partial SNSM semantic model for WhatsApp.

is achieved by removing the blocked relation from u_2 to u_1 from the social graph of SN_5, i.e., $WhatsApp \downarrow_{Blocked} (u_2) = \emptyset$. At configuration SN_1, when agent u_1 does not receive $DCh(m_1)$, she realises that there could be a problem and sends m_2. At configuration SN_4, by receiving $DCh(m_1)$, u_1 realises that there was a connection problem. However, by receiving $DCh(m_2)$ while waiting for the $DCh(m_1)$ at SN_{12}, u_1 infers that u_2 had blocked her for a while. Therefore, in WhatsApp there is an epistemic leakage in the third scenario.

In the second scenario, e.g., when the recipient's phone is initially off, once the phone is turned on, the sent message will be received at the recipient's side and the sender will see a double checkmarks for the sent message. If the agent sends another message, she will see again double checkmarks. However, if she does not receive double checkmarks for the first message, the sender will infer the possibility of having been block.

As we demonstrate next, these policy violations can be checked as an epistemic property based on our proposed semantic model and our global privacy policy templates.

6.2 Reasoning about Global Privacy Policies

Consider the following epistemic property to detect information leakage regarding the block/unblock actions in WhatsApp:

property: *"In all runs of the protocol, for all configurations, it never happens that an agent knows that she had been previously blocked by another agent".*

For this specific scenario, we would like to verify that u_1 never knows that it was previously blocked by u_2. The property can be formulated using our templates for global privacy policies as follows:

$$AG\ \neg K_1(\langle\overline{(Block(u_2,u_1), FP^{u_2})}\rangle\top^{\Lsh})$$

It turns out that the specified property does not hold in the configurations SN_{12} of the semantic model in Fig. 4; below we illustrate the reasoning leading to this conclusion. In order to verify the original formula, we should check that $K_1(\langle\overline{(Block(u_2, u_1), FP^{u_2})}\rangle\top^{\Lsh})$ does not hold in any reachable configuration; we do this using a search on the ALTS model. In configuration SN_{11}, everything is still fine and since this configuration is indistinguishable from SN_4, u_1 thinks that the second scenario (network problem or the phone is off) is the case. Upon hitting configuration SN_{12} through transition $(DCh(m_2), FP^{u_1})$, this configuration is not indistinguishable from SN_7 for u_1 (see Fig. 4). In configuration SN_{12}, $\langle\overline{(Block(u_2, u_1), FP^{u_2})}\rangle\top^{\Lsh}$ holds because the configuration SN_{12} is reachable from SN_2 in which $\langle(Block(u_2, u_1), FP^{u_2})\rangle$ holds. By the Semantics of our logic, a knowledge formula holds in a configuration if it holds in all indistinguishable configurations from that configuration. As SN_{12} is only indistinguishable from itself by u_1, it holds that agent u_1 knows that it was blocked before, i.e., $K_1(\langle\overline{(Block(u_2, u_1), FP^{u_2})}\rangle\top^{\Lsh})$. We conclude that $\neg K_1(\langle\overline{(Block(u_2, u_1), FP^{u_2})}\rangle\top^{\Lsh})$ does not hold in the configuration SN_{12}.

It is worth noting that the purpose of our framework is to analyse privacy violations. Repairing such violations through changes in the social network protocol requires further analysis, e.g., through a causal analysis of the detected violation, and remains an interesting avenue for future research.

7 Complexity of Model Checking

The model-checking problem of our logic for a given configuration *conf* is defined as follows:

- **Input:** a social network semantic model *SNSM* of the form $\langle Conf, \rightarrow, Ind, conf_0\rangle$ and a formula ϕ in our logic,
- **Output:** yes iff $SNSM, conf \models \phi$; no otherwise (note that for full modal μ-calculus, the notion of counter-example is challenging to define)

It is known that the model-checking problem is *PSPACE-Complete* for the Dynamic Epistemic Logic language (DEL) based on the event model semantics [4] (even if one adds the possibility of reasoning about common knowledge [13]). This result holds when the models are $KD45$. The result also carries over to $S5$ models, as well [13]. To obtain this set of results, an efficient algorithm with *PSPACE* upper bound is provided for DEL model-checking problem. For the lower bound, a polynomial reduction is provided from the *quantified Boolean formula satisfiability* problem to the model-checking problem of DEL [4]. A more recent and specific analysis is also provided by de Haan et al. for the complexity of DEL with S5 models and S5 event models [28]. In this work, three elements of the framework of DEL are considered to categorise the computational complexity of the model checking problem of DEL with $S5$ models. These elements include the number of agents, whether the models are single-pointed or multi-pointed, and whether updates have post-conditions. They show that by having one agent, a single-pointed model, and no post condition, the model checking problem can be solved in P. Having more than one agent or multi-pointed models leads to a $PSPACE-complete$ computational complexity. Finally, one agent, a single-pointed model with post condition, has a model checking problem of the order $\Delta_2^p - hard$. Since we have more than one agent and we do not need multi-pointed models, our model-checking problem is expected to fall into the $PSPACE-complete$ complexity class.

To rigorously examine the complexity of our model-checking problem, we provide a one-to-one mapping from our semantic model to the semantics of DEL and back. As a result, the complexity of our model checking problem for logical formulas with no fixed point and no past operators is shown to be *PSPACE-Complete*. In the remainder of this section, we first briefly introduce DEL. Then, we give a mapping from our semantic model to the semantics of DEL. Finally, we prove Theorem 30 which is related to the complexity of our model checking problem, by showing that the mapping of our social network semantic model to DEL semantics is sound and complete.

7.1 Event Model, DEL and Product Update

Let *Atm* be a countable set of atomic propositions. An *epistemic model* is a tuple of the form $M = (W, R, V)$, where W is a non-empty set of possible worlds, R is a set of indistinguishability relations of the form $R_i \subseteq W \times W$ between possible worlds for each $i \in Id$ where Id is the set of agent identifiers, and V is a valuation function assigning a proposition to a set of worlds, i.e., $Atm \to 2^W$. The pair (M, w) is a pointed epistemic model, where $w \in M$. The language of epistemic logic (L^{EL}) is defined as follows [4]:

$$L^{EL} : \phi \ ::= \ p \mid \phi \wedge \phi \mid \neg\phi \mid B_i\phi$$

Event Model. By introducing the notion of *event models* [7], epistemic logic has been extended with dynamic operators of the form $[M', w']\phi$ which means that ϕ holds after the occurrence of the event $[M', w']$. The event (M', w') is a pointed epistemic model of the form $M' = (W', R', Pre)$ where W' and R' have the same meaning as in the epistemic model and Pre is a function of the form $W' \to L^{EL}$ mapping each event to a precondition specified by L^{EL}. Intuitively, (M', w') means that how the actual event w' is perceived by agents. The language of DEL is defined as follows:

$$L^{DEL} : \phi \ ::= \ p \mid \phi \wedge \phi \mid \neg\phi \mid B_i\phi \mid [M', w']\phi$$

An extension of DEL is provided in which union of event models is allowed in preconditions as well. Therefore, the language of DEL is redefined as follows on which the proof of complexity is provided in [4].

$$L^{DEL} : \phi \ :: \ = p \mid \phi \wedge \phi \mid \neg\phi \mid B_i\phi \mid [\pi]\phi$$
$$\pi \ ::= \ M', w' \mid (\pi \cup \pi)$$

Product Update. To create dynamism, when an event (M', w') occurs, a pointed epistemic model (M, w), is converted to a new epistemic model $M'' = (W'', R'', V'')$. M'' is obtained using the notion of *product update* of the form $M \otimes M', (w, w')$ which is defined as follows:

$$\begin{aligned} W'' &= \{(w, w') \in W \times W' \mid M, w \models Pre(w')\} \\ R''_i &= \{\langle (w, w'), (v, v') \rangle \in W'' \times W'' \mid wR_iv \wedge w'R_iv'\} \\ V''(p) &= \{(w, w') \in W'' \mid w \in V(p)\} \end{aligned}$$

There is nothing special in the semantic definitions for various constructs of DEL. Given an epistemic model $M = (W, R, V)$ and a formula $\phi \in L^{DEL}$, we have that:

$$\begin{array}{lll} M, w \models p & \text{iff} & w \in V(p) \\ M, w \models \phi_1 \wedge \phi_2 & \text{iff} & M, w \models \phi_1 \wedge M, w \models \phi_2 \\ M, w \models \neg\phi & \text{iff} & M, w \not\models \phi \\ M, w \models B_i\phi & \text{iff} & \forall v \in R_i(w) \cdot (M, v \models \phi) \\ M, w \models [M', w']\phi & \text{iff} & M, w \models Pre(w') \implies M \otimes M', (w, w') \models \phi \\ M, w \models [\pi \cup \gamma]\phi & \text{iff} & M, w \models [\pi]\phi \wedge M, w \models [\gamma]\phi \end{array}$$

If $M, w \models Pre(w')$, we say that (M', w') is executable in (M, w).

7.2 Mapping *SNSM* to *DEL* semantics

Translation of *SNSM* is achieved by a one-to-one mapping from configurations and indistiguishability relations among them to an epistemic model and from transitions to event models of DEL. In this section, we first provide a translation of the configurations of a given *SNSM* to the states of the corresponding epistemic model, while respecting the indistinguishability relations between them. Viewing the actions of our semantic model as events, we translate transitions of *SNSM* into an event model.

Definition 23 (Translation of Configurations) *Given an SNSM model $\mathcal{M} = \langle Conf, \to, Ind, conf_0 \rangle$, we define the corresponding epistemic model $M = (W, R, V)$ of $\mathcal{M}$'s configurations and the indistinguishability relations between them as follows.*

- *$W = \{map(\pi) \mid (sg, s, \pi) \in Conf\}$, where $map(\pi)$ is defined as:*

$$\begin{cases} map(\epsilon) = \epsilon \\ map(\pi' \frown \alpha) = (map(\pi'), \alpha) \end{cases}$$

- *$R = \{w_i R_{id} w_j \mid w_i = map(\pi_i), w_j = map(\pi_j), \pi_i \overset{id}{\cdots} \pi_j\}$, i.e., all indistinguishability relations between configurations are preserved by their corresponding states, and*
- *V: assuming the set of atomic propositions $Atm = Conf$, then for each $(sg, s, \pi) \in Conf$, $V(sg, s, \pi) = \{w \mid w = map(\pi)\}$.*

Definition 24 (Translation of Transitions) *Given an SNSM model $\mathcal{M} = \langle Conf, \rightarrow, Ind, conf_0 \rangle$, we define the corresponding event model of the transitions as the event model $M' = (W', R', Pre)$ where:*

$$\begin{aligned} W' &= \{\alpha \mid c \overset{\alpha}{\rightarrow} c'\} \cup \{\epsilon\}, \\ Pre(\alpha) &= \bigvee\nolimits_{\{c \in C\}} c, \text{ where } C = \{c \mid c \overset{\alpha}{\rightarrow} c'\} \\ Pre(\epsilon) &= \bigvee\nolimits_{\{c \in C\}} c, \text{ where } C\{c \mid c \xrightarrow{(a,f)} c' \wedge f(a, id, sg) = \tau\} \\ R'_{id} &= \{((a, f), (b, f')) \mid f(a, id, sg) = f'(b, id, sg) : id \in Id, a, b \in A, sg \in SG\} \cup \\ R'_{id} &= \{((a, f), \epsilon) \mid f(a, id, sg) = f'(\tau, id, sg) = \tau : id \in Id, a \in A, sg \in SG\} \end{aligned}$$

Note that by Definition 24, the label of states in the event model would be the decorated actions of *SNSM*. We consider the special state ϵ for those actions with τ exploration to some agent. The precondition of each state in the event model would be disjunction of the source configuration of transitions carrying such label in the social network semantic model $\mathcal{M}$. Note that for the precondition of the state ϵ, we also added the source configuration of transitions with τ as their appearance (include τ itself).

Due to the product update, some states may be introduced in the resulting epistemic model by multiplying the states corresponding configurations with outgoing transitions having appearance τ, and the state ϵ from the event model. We define the structural equivalence relation $(w, \epsilon) = w$ over the states of the epistemic model and merge those states to obtain the resulting epistemic model. The valuation function assigning to each state in product update is identical to Definition 23, i.e. $V(sg, s, \pi) = \{w \mid w = map(\pi)\}$.

Example 25 *An example for mapping an SNSM model $\mathcal{M} = \langle Conf, \rightarrow, Ind, conf_0 \rangle$ to its corresponding DEL semantics is provided in Fig. 5 in which $f(a, 2, sg) = \tau$ and $f'(b, 2, sg) = f''(c, 2, sg)$. The models M and M' illustrates the corresponding epistemic model and constructed event model for transitions, respectively. Self-loops have been omitted for brevity.*

7.3 Translation to DEL and Complexity of Model Checking

To prove the correctness of our translation, we show that the corresponding constructed DEL semantic model for a configuration of a given *SNSM* model $\langle Conf, \rightarrow, Ind, conf_0 \rangle$ preserve the epistemic properties of the original model. Please note that unlike the original setting, we consider true knowledge, rather than belief in our framework; however, since the complexity result of DEL also holds for $S5$ models [13], the two notions coincide on $S5$ models and and hence, we use K_i freely in the remainder of this section. To prove Theorems 28, we use Lemma 26 expressing the correspondence between a transition in SNSM and updating an epistemic model by the event of that transition.

Lemma 26 *For the given SNSM model $\mathcal{M} = \langle Conf, \rightarrow, Ind, conf_0 \rangle$, assume that $M = (W, R, V)$ and $M' = (W', R', Pre)$ denote its corresponding epistemic and event model, respectively. Let $c \xrightarrow{(a,f)} c^*$ where $c = (sg, s, \pi)$, $c^* = (sg^*, s^*, \pi^*)$, and $\pi^* = \pi \frown (a, f)$. Assume that $w = map(\pi)$ and $w^* = map(\pi^*)$. It holds that $(M, w) \otimes (M', (a, f)) = (M, w^*)$.*

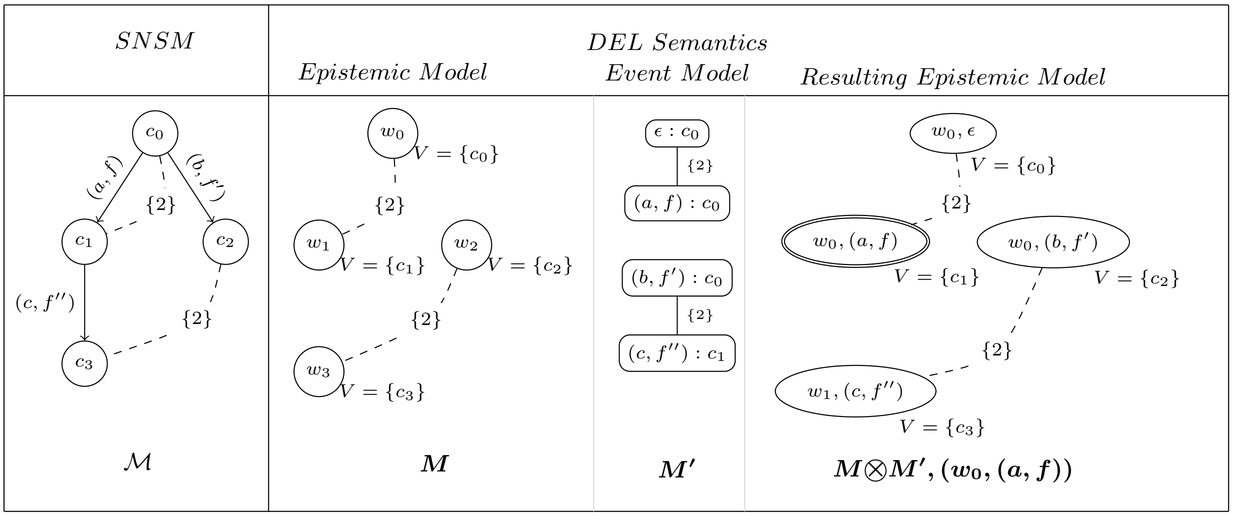

Fig. 5. Mapping an SNSM to its corresponding DEL semantic model.

Proof. By the definition of product update, the resulting epistemic model consists of the following states:

$$W'' = \{(w, w') \mid M, w \models pre(w')\}.$$

Consider an arbitrary state $w_1 = map(\pi_1)$ such that there exists a configuration $c_1 = (sg_1, s_1, \pi_1) \in Conf$. For any transition of c_1 like $c_1 \xrightarrow{(a,f)} c_2$, where $c_2 = (sg_2, s_2, \pi_2)$ and $\pi_2 = \pi_1 \frown (a, f)$, the state (a, f) in the event model has the disjunction c_1 by Definition 24 and $w_1 \in V(c_1)$ by Definition 23. Thus, it holds that $w_1 \models Pre(a, f)$ and hence, $(w_1, (a, f)) \in W''$. However, $map(\pi_2) = (map(\pi_1), (a, f))$ and thus $w_2 = (w_1, (a, f))$, and $W \subset W''$ (result †).

Without loss of generality, we assume that (a, f) has a τ exploration for some agent. Thus, the state ϵ in the event model has a disjunct c_1 by Definition 23. It hence follows that $(w_1, \epsilon) \in W''$ (result ‡). By the results † and ‡, we conclude that

$$W'' = W \cup \{(w_i, \epsilon) \mid c_i = (sg_i, s_i, \pi_i), w_i = map(\pi_i), Pre(\epsilon) = c_1 \vee \cdots \vee c_n\}.$$

As $(w, \epsilon) = w$, W'' is equal to W modulo structural equivalence.

We prove that $R = R''$. Consider those $(w, (a, f))$ and $(w', (b, f'))$ in W such that $(w, (a, f))R_i(w', (b, f'))$, where $w = map(\pi)$, and $w' = map(\pi')$. Three cases can be distinguished regarding the last step in the proof of indistinguishability relation $\pi \frown (a, f) \overset{i}{\cdots} \pi' \frown (b, f')$:

- it resulted from the application of rule (**match**). So, $(a, f)R'_i(b, f')$ in the event model M' and wR_iw' in the epistemic model M. Hence, $(w, (a, f))R''_i(w', (b, f'))$.
- it resulted from the application of rule ($\mathbf{hid_0}$). So, $(a, f)R'_i\epsilon$ in the event model M' and $wR_i(w', (b, f'))$ in the epistemic model M. Hence, $((w, (a, f)), \epsilon)R''_i(w', (b, f'))$. Concluding that W'' modulo structural equivalence has $(w, (a, f))R''_i(w', (b, f'))$.
- it resulted from the application of rule ($\mathbf{hid_1}$). This case follows from a similar reasoning as in the previous case.

As $W'' = W$, it is trivial that $V'' = V$. ■

Example 27 *The result of product $(M, w_0) \otimes (M', (a, f))$ of example 25 is shown in Figure 5. In the SNSM model $\mathcal{M}$, we have that $c_0 \xrightarrow{(a,f)} c_1$ where $c_0 = (sg_0, s_0, \pi_0)$, $c_1 = (sg_1, s_1, \pi_1)$, and $\pi_1 = \pi_0 \frown (a, f)$. By Lemma 26, it holds that $(M, w_0) \otimes (M', (a, f)) = M \otimes M', (w_0, (a, f))$. As $w_0 = map(\pi_0)$, by Definition 23, we have that $(map(\pi_0), (a, f)) = map(\pi_0 \frown (a, f)) = map(\pi_1) =$*

w_1. Therefore, by applying an event corresponding to a transition (a, f), we obtain a pointed epistemic model with an initial state corresponding to the target configuration of the transition.

Theorem 28 (Soundness) *For any given SNSM model $\mathcal{M} = \langle Conf, \rightarrow, Ind, conf_0\rangle$, and any logical SNSM formula ϕ, if $\mathcal{M}, c \models \phi$, then $M, w \models map(\phi)$ where M is the corresponding epistemic model of $\mathcal{M}$, $c = (sg, s, \pi)$ and $w = map(\pi)$ and ϕ is restricted to*

$$\phi ::= \top \mid \phi \wedge \phi \mid \neg\phi \mid [(a, f)]\phi \mid K_i\phi$$

and its mapping to a DEL formula is defined as

$$\begin{aligned} map(\top) &= \textstyle\bigvee_{c \in Conf} c \\ map(\phi_1 \wedge \phi_2) &= map(\phi_1) \wedge map(\phi_2) \\ map(\neg\phi) &= \neg map(\phi) \\ map([(a, f)]\phi) &= [M', (a, f)]map(\phi) \\ map(K_i\phi) &= K_i(map(\phi)) \end{aligned}$$

where M' is the corresponding event model of $\mathcal{M}$.

Proof. We prove by induction on the structure of ϕ.

- $\phi \equiv \top$: by Definition 23, $w \in V(c)$, than trivially $M, w \models \bigvee_{c \in Conf} c$.
- $\phi \equiv \phi_1 \wedge \phi_2$: by definition, $\mathcal{M}, c \models \phi$ implies that $\mathcal{M}, c \models \phi_1$ and $\mathcal{M}, c \models \phi_2$: by induction, $M, w \models map(\phi_1)$ and $M, w \models map(\phi_2)$ hold. Consequently, $M, w \models map(\phi)$.

- $\phi \equiv \neg\phi'$: by definition $\mathcal{M}, c \models \phi$ implies that ϕ' does not hold. By induction, $M, w \models map(\phi')$ does not hold, and so $M, w \models map(\phi)$

- $\phi \equiv [(a, f)]\phi'$: by definition $\mathcal{M}, c \models \phi$ implies that for all configurations c' such that $c \xrightarrow{(a,f)} c^*$, $\mathcal{M}, c^* \models \phi'$ holds. By induction, $M, w^* \models map(\phi')$, where $w^* = map(\pi^*)$, $c^* = (sg^*, s^*, \pi^*)$, and $\pi = \pi^* \frown (a, f)$. By Lemma 26, $(M, w) \otimes (M', (a, f)) = (M, w^*)$. So, it holds that $M, w \models map(\phi)$.
- $\phi \equiv K_i\phi'$: by definition for all c' such that $c \overset{i}{\cdots} c^*$, it holds that $\mathcal{M}, c^* \models \phi'$. By induction, $M, w^* \models map(\phi')$, where $w^* = map(\pi^*)$ and $c^* = (sg^*, s^*, \pi^*)$. By Definition 23, $c \overset{i}{\cdots} c^*$ implies that wR_iw^*. Hence, it trivially holds that $M, w \models map(\phi)$.

■

Theorem 29 (Completeness) *For any given SNSM model $\mathcal{M} = \langle Conf, \rightarrow, Ind, conf_0\rangle$ and any logical DEL formula ϕ, if $M, w \models \phi$ then $\mathcal{M}, c \models map^{-1}(\phi)$ where M is the corresponding epistemic model of $\mathcal{M}$, $c = (sg, s, \pi) \in Conf$ and $w = map(\pi)$ and ϕ is restricted to*

$$\phi ::= c \mid \phi \wedge \phi \mid \neg\phi \mid [M', (a, f)]\phi \mid K_i\phi$$

where M' is the corresponding event model of $\mathcal{M}$, and its mapping to our logic is defined as

$$\begin{aligned} map^{-1}(c) &= \top \\ map^{-1}(\phi_1 \wedge \phi_2) &= map^{-1}(\phi_1) \wedge map^{-1}(\phi_2) \\ map^{-1}(\neg\phi) &= \neg map^{-1}(\phi) \\ map^{-1}([M', (a, f)]\phi) &= [(a, f)]map^{-1}(\phi) \\ map^{-1}(K_i\phi) &= K_i(map^{-1}(\phi)). \end{aligned}$$

Proof. We prove by induction on the structure of ϕ.

- $\phi \equiv c$: Since $map^{-1}(c) = \top$, it trivially holds that $\mathcal{M}, c \models \top$.

- $\phi \equiv \phi_1 \wedge \phi_2$: by definition $M, w \models \phi$ implies that $M, w \models \phi_1$ and $M, w \models \phi_2$. by induction, $\mathcal{M}, c \models map^{-1}(\phi_1)$ and $\mathcal{M}, c \models map^{-1}(\phi_2)$ hold. Consequently, we have that $\mathcal{M}, c \models map^{-1}(\phi)$.

- $\phi \equiv \neg\phi'$: by definition $M, w \models \phi$ implies that ϕ' does not hold. By induction, $\mathcal{M}, c \models map^{-1}(\phi')$ does not hold, and so $\mathcal{M}, c \models map^{-1}(\phi)$

- $\phi \equiv [M', (a, f)]\phi'$: by definition $M, w \models [M', (a, f)]\phi'$ implies that if $M, w \models Pre(a, f)$, then $M \otimes M', (w, (a, f)) \models \phi'$. The assumption $M, w \models Pre(a, f)$ implies that $w \in V(c)$ and c is an arbitrary disjunction of $Pre(a, f)$. Regarding Definition 24, having c in $Pre(a, f)$ implies that c has some (a, f) transitions leading to configuration c^* as an example. By Lemma 26, $M \otimes M', (w, (a, f)) = M, w^*$ where $w^* = map(\pi^*)$ and $c^* = (sg^*, s^*, \pi^*)$ (please note that the path part of next configurations of all (a, f)-transitions of c are the same and so mapped to w^*). So, $M, w^* \models \phi'$ and by induction, $\mathcal{M}, c^* \models map^{-1}(\phi')$. Hence, we conclude that $\mathcal{M}, c \models map^{-1}(\phi)$.
- $\phi \equiv K_i\phi'$: by definition, if $M, w \models K_i\phi'$, then for all w^* such that $w^* R_i w$, it holds that $M, w^*, \models \phi'$. By Definition 23, $w^* R_i w$ implies that $c \overset{i}{\cdots} c^*$ where $c = (sg, s, \pi)$, $c^* = (sg^*, s^*, \pi^*)$, and $w^* = map(\pi^*)$. By induction, $\mathcal{M}, c^* \models map^{-1}(\phi')$, and consequently $\mathcal{M}, c \models map^{-1}(\phi)$.

■

In this section, we show that our model-checking is equivalent to the model checking problem of DEL, as stated in the following theorem.

Theorem 30 (Verification Complexity) *Consider a pointed social network semantic model $(SNSM, conf)$ where $conf \in Conf$. The model-checking problem for a property ϕ on SNSM $((SNSM, conf) \models \phi)$, where ϕ contains no infinite conjunctions, no fixed-point and past operators, is PSPACE-Complete.*

Proof. Theorem 30 is an immediate consequence of Theorems 28 and 29: our translations both to DEL and from DEL are linear in time and the size of event models and epistemic models. i.e., each transition exactly corresponds to a state in the event model, and each configuration to a state in the resulting epistemic model and vice versa (subsection 7.2). As we provide a linear time reduction to and from the model-checking problem of DEL, it follows that the model checking of *SNSM* is *PSPACE-Complete* for the mentioned subset of ϕ.

■

These translations lead to the insight that our semantic model is a domain specific variant of DEL where events are actions with local privacy policies and indistinguishability relations are derived automatically from those privacy policies. We leave the influence of fixed-point and past operator on the complexity to future work. We expect that this follows from the traditional results in modal μ-calculus. Model checking problem is PSPACE-complete for LTL and LTL+past and μ-calculus+ past [31].

8 Related Work

This paper bridges three otherwise mostly separate areas namely, information propagation in social networks, privacy policies, and epistemic logic. Modelling information propagation is an important and well-studied research in the field of social networks [14, 27]. Regarding security and privacy policies, some basic approaches are proposed for expressing and reasoning about epistemic properties [29, 5]. There is a considerable body of work on modelling different aspects of social networks, among which we review a few of the most relevant pieces of work. Alvim et al. [2] provide probabilistic models for social networks and quantify the beliefs of agents to analyse dynamics of the network towards polarisation. There are other probabilistic approaches that formalise information diffusion and address information leakage such as [16, 17]. There are some more applied approaches that focus on modelling the interactions on specific social platforms such as Twitlang for Twitter [36].

A comparison of the closest work to ours based on a set of selected features related to privacy policy-aware propagation and epistemic aspects of the social network has been summarised in Table 3. The features include the year of (the latest) publication, and the approaches that support epistemic reasoning, whether the approach supports modelling and reasoning about dynamic topological (social-graph-related) and privacy-related semantic aspects. We also note whether it is possible to define different kinds of relationships among the agents in the social network and whether privacy policies and permissions can be explicitly specified. In addition to this brief comparison, we explain each piece of related work in more details below. We refer to each approach by the name of its model or logic.

Table 3. Comparison of the most related work

Metrics	SEM	DEFL	SNM	ETM	SNSM
Year	2011	2013	2017	2018	2020
Semantics	Kripke + topology	Kripke + friendship	social graph+ knowledge-base	Kripke + social graph	ALTS+ social graph
Knowledge Dyn.	✓	✓	✓	✓	✓
Topology Dyn.	✓	✓	✓	×	✓
Privacy Dyn.	×	×	✓	×	✓
Multi-relational	×	×	✓	✓	✓
Privacy policies	×	×	✓	×	✓
References	[42]	[43]	[39]	[6]	-

The Social Epistemic Model (SEM) [42] focuses on modelling knowledge flow in social networks. SEM is a bipartite model containing a graph of social relations on one part and an epistemic model with the agents on the other part. Using a dynamic epistemic semantics, it can specify social actions, represent communication channels and model propagation in the network. It does not consider privacy policies and assumes that the network structure is common knowledge.

The Dynamic Epistemic Friendship Logic (DEFL) [43] is a modal logic for modelling dynamics of knowledge and friendship on Facebook. Using DEFL, one can specify sender, receiver, and the message which enables reasoning about communications and their epistemic consequences. The structure of the network is considered as part of a possible world and social relations are symmetric.

Epistemic Threshold Model (ETM) [6] uses logic for reasoning about the dynamics of threshold models (TM) and the effects of knowledge of agents on the model. Both network structure and the agent's behaviour can be modelled in the proposed logic. Although this approach supports modelling different types of social relationships, it does not consider dynamicity in the structure of the network and privacy policies.

Pardo et al. have proposed an epistemic framework for specifying and reasoning about privacy policies in social networks [39]. They have used the social network model, SNM, containing network structure and a knowledge base for each agent, and a denotational approach for defining local/global privacy policies. Based on operational semantic rules, they can describe the behavior of social networks from structural, epistemic and privacy aspects. It is not possible to specify policies in the protocol, directly, i.e. whenever the policy rules are changed, the semantic rules have to be revised accordingly. These policies are epistemic formulae that must hold along any possible execution of the system (similar to our global policies), but they do not specify how policies must be enforced, in other words, what actions must be executed in order for the policies to not be violated.

To be more precise, although they also have a formal framework, the differences between our work and their approach are that first, they have to initiate agents knowledge-base and update them using an engine while we provide a formal approach to automatically generate and update the knowledge of agents; secondly, we have sequences of actions in the form of paths in our framework and can reason about the sequences while in [39], principally the actions are considered separately. The fourth difference is that we can enforce local privacy policies using decorated actions determining what can be seen by whom. In contrast in [39], one can specify the protocol and fine-grained privacy policies and upon receipt of any event, it is checked whether the state reached violates some privacy policies (similar to our global privacy policies). Finally, the semantics of the approach proposed in [39] is an interpreted system, based on traces and therefore verification of properties with branching structures is not possible.

Our work is mainly inspired by those previous approaches that have combined operational and epistemic worlds to be able to reason about epistemic aspects of the protocols [3, 15, 37]. Our proposed semantics SNSM allows to handle the agents' epistemic knowledge which is very relevant for real-world social networks. In our semantic model, we can consider propagation issues and social influences alongside with privacy policies. Besides, it is possible to specify multiple relationships and their dynamism; transitions encode the effect of actions on the underlying social graph and local states of agents while indistingushibility relations express the privacy-aware epistemic consequences. Our approach is more flexible in comparison with SNM in specifying privacy policies, as there is no need to specify privacy policies directly in our semantics. Furthermore, our ALTS-based semantics allows verification of both temporal and epistemic properties in a single semantic model.

9 Discussion and Future Work

In this paper, we proposed a formal framework for social networks comprising a social network semantic model (SNSM). Our semantic model addresses both the operational and the epistemic aspects of social networks, particularly concerning privacy policies. In this framework, we formalised local privacy policies using decorated actions and specified a set of common local policy templates for the domain of social networks. Our indistinguishability relation based on the history of actions in the semantic configurations allows for reasoning about epistemic consequences of sequences of actions as well. We distinguish between privacy concerns in terms of function views that decorate actions, from operational aspects of communications and dynamic changes in the topology of the network that are modelled as the effect of these actions. Using a combination of modal μ-calculus and epistemic logic, we specify global privacies to check that the indirect effects of actions do not lead to privacy breaches (even under correct local privacy policy configurations). We integrated all of these in a semantic framework to specify policy-aware information propagation in social networks. We analyse and prove the formal properties of the semantic model and the complexity of model checking for a subset of logic.

We applied our framework to a real-world scenario and showed how privacy breaches can be identified through verification of a global privacy policy on the constructed semantic model.

The process of verification can be mechanised in the future by using automated model checking tools for the standard semantics of epistemic logic. DEMO-S5 [21][8] is a tool optimised for equivalence relations. Some other model checkers such as MCK [25][9] and MCMAS [34, 33][10] can also be considered in our future research. However, the use of latter tools will require further research, since they use temporal logic on interpreted systems. It thus remains an avenue of future research to establish a formal translation from our framework to theirs. Two other DEL model checking approaches are succinct models [12] and symbolic model checking. For the latter, an approach is provided in [9] for DEL models and the notion of knowledge transformers is introduced for action models. They implement a symbolic model checker for Dynamic Epistemic Logic called SMCDEL [11], which will also be further investigated in our future research.

We also aim to provide a high-level modelling language based on our semantic model to formally specify the behaviour of social networks in a modular way. Another direction for future work is state space reduction, e.g., by reducing τ appearances of actions and/or considering an abstract description of a social network. Besides, detecting and resolving conflicts among policies are other interesting future research directions.

[8] `https://staff.fnwi.uva.nl/d.j.n.vaneijck2/software/demo_s5/` (Accessed: 16 September 2021).

[9] `http://cgi.cse.unsw.edu.au/~mck/pmck/` (Accessed: 6 August 2021).

[10] `https://vas.doc.ic.ac.uk/software/mcmas/` (Accessed: 6 August 2021).

[11] `https://w4eg.de/malvin/illc/smcdelweb/index.html` (Accessed: 10 December 2021).

References

1. Albert, R., Barabási, A.: Statistical Mechanics of Complex Networks. CoRR **cond-mat/0106096** (2001), http://arxiv.org/abs/cond-mat/0106096
2. Alvim, M.S., Knight, S., Valencia, F.D.: Toward a Formal Model for Group Polarization in Social Networks. In: Alvim, M.S., Chatzikokolakis, K., Olarte, C., Valencia, F. (eds.) The Art of Modelling Computational Systems: A Journey from Logic and Concurrency to Security and Privacy - Essays Dedicated to Catuscia Palamidessi on the Occasion of Her 60th Birthday, Lecture Notes in Computer Science, vol. 11760, pp. 419–441. Springer (Nov 2019). https://doi.org/10.1007/978-3-030-31175-9_24
3. Apt, K.R., Grossi, D., van der Hoek, W.: Epistemic Protocols for Distributed Gossiping. Electronic Proceedings in Theoretical Computer Science **215**, 5166 (Jun 2016). https://doi.org/10.4204/eptcs.215.5
4. Aucher, G., Schwarzentruber, F.: On the Complexity of Dynamic Epistemic Logic. CoRR **abs/1310.6406** (2013), http://arxiv.org/abs/1310.6406
5. Balliu, M.: A Logic for Information Flow Analysis of Distributed Programs. In: Riis Nielson, H., Gollmann, D. (eds.) Secure IT Systems. pp. 84–99. Springer Berlin Heidelberg, Berlin, Heidelber (2013), https://doi.org/10.1007/978-3-642-41488-6_6
6. Baltag, A., Christoff, Z., Rendsvig, R.K., Smets, S.: Dynamic Epistemic Logics of Diffusion and Prediction in Social Networks. Studia Logica **107**, 489–531 (2019). https://doi.org/10.1007/s11225-018-9804-x
7. Baltag, A., Moss, L.S., Solecki, S.: The Logic of Public Announcements, Common Knowledge, and Private Suspicions. In: Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge. p. 4356. TARK '98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998), https://link.springer.com/chapter/10.1007%2F978-3-319-20451-2_38
8. Barth, A., Datta, A., Mitchell, J., Nissenbaum, H.: Privacy and Contextual Integrity: Framework and Applications. In: 2006 IEEE Symposium on Security and Privacy (S P'06). pp. 15 pp.–198 (2006). https://doi.org/10.1109/SP.2006.32
9. van Benthem, J., van Eijck, J., Gattinger, M., Su, K.: Symbolic Model Checking for Dynamic Epistemic Logic - S5 and Beyond. Journal of Logic and Computation **28**(2), 367–402 (11 2017). https://doi.org/10.1093/logcom/exx038
10. van Benthem, J., van Eijck, J., Kooi, B.: Logics of Communication and Change. Information and Computation **204**(11), 1620–1662 (2006). https://doi.org/10.1016/j.ic.2006.04.006
11. Bradfield, J., Stirling, C.: Modal mu-Calculi. In: Handbook of Modal Logic. pp. 721–756. Elsevier, Netherlands (2007). https://doi.org/10.1.1.143.9834
12. Charrier, T., Schwarzentruber, F.: A Succinct Language for Dynamic Epistemic Logic. In: Larson, K., Winikoff, M., Das, S., Durfee, E.H. (eds.) Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017. pp. 123–131. ACM (2017), http://dl.acm.org/citation.cfm?id=3091148
13. Charrier, T., Schwarzentruber, F.: Complexity of Dynamic Epistemic Logic with Common Knowledge. In: Bezhanishvili, G., D'Agostino, G., Metcalfe, G., Studer, T. (eds.) Advances in Modal Logic 12, proceedings of the 12th conference on "Advances in Modal Logic," held in Bern, Switzerland, August 27-31, 2018. pp. 103–122. College Publications (2018), http://www.aiml.net/volumes/volume12/Charrier-Schwarzentruber.pdf
14. Chen, W., Lakshmanan, L., Castillo, C.: Information and Influence Propagation in Social Networks. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2013), https://doi.org/10.2200/S00527ED1V01Y201308DTM037
15. Dechesne, F., Mousavi, M.R., Orzan, S.: Operational and Epistemic Approaches to Protocol Analysis: Bridging the Gap. In: Dershowitz, N., Voronkov, A. (eds.) Proceedings of the 14th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2007) 15-19 October 2007, Yerevan, Armenia. pp. 226–241. Lecture Notes in Computer Science, Springer, Germany (2007), https://link.springer.com/chapter/10.1007%2F978-3-540-75560-9_18
16. Dennis, L.A., Slavkovik, M.: Model-Checking Information Diffusion in Social Networks with PRISM. In: Bassiliades, N., Chalkiadakis, G., de Jonge, D. (eds.) Multi-Agent Systems and Agreement Technologies. pp. 475–492. Springer International Publishing, Cham (2020), https://link.springer.com/chapter/10.1007%2F978-3-030-66412-1_30
17. Dennis, L.A., Slavkovik, M., Fisher, M.: "How Did They Know?" - Model-Checking for Analysis of Information Leakage in Social Networks. In: Cranefield, S., Mahmoud, S., Padget, J.A., Rocha, A.P. (eds.) Coordination, Organizations, Institutions, and Norms in Agent Systems XII - COIN 2016 International Workshops, COIN@AAMAS, Singapore, Singapore, May 9, 2016, COIN@ECAI, The

Hague, The Netherlands, August 30, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10315, pp. 42–59. Springer (2016). https://doi.org/10.1007/978-3-319-66595-5_3

18. Ditmarsch, H.v., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer Publishing Company, Incorporated, 1st edn. (2007), https://link.springer.com/book/10.1007/978-1-4020-5839-4
19. Dorri, A., Kanhere, S.S., Jurdak, R.: Multi-Agent Systems: A Survey. IEEE Access **6**, 28573–28593 (2018). https://doi.org/10.1109/ACCESS.2018.2831228
20. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, USA (2010). https://doi.org/10.1017/CBO9780511761942
21. van Eijck, J.: DEMO-S5. Tech. rep., Tech. rep., CWI (2014), https://staff.fnwi.uva.nl/d.j.n.vaneijck2/software/demo_s5/DEMO_S5.pdf
22. Fagin, R., Halpern, J.Y.: Belief, Awareness, and Limited Reasoning. Artificial Intelligence **34**(1), 39–76 (1987). https://doi.org/10.1016/0004-3702(87)90003-8
23. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning About Knowledge. MIT Press, Cambridge, MA, USA (2003), https://dl.acm.org/doi/10.5555/995831
24. Fischer, M.J., Ladner, R.E.: Propositional Dynamic Logic of Regular Programs. Journal of Computer and System Sciences **18**(2), 194–211 (1979). https://doi.org/10.1016/0022-0000(79)90046-1
25. Gammie, P., van der Meyden, R.: MCK: Model Checking the Logic of Knowledge. In: Alur, R., Peled, D.A. (eds.) Computer Aided Verification. pp. 479–483. Springer Berlin Heidelberg, Berlin, Heidelberg (2004), https://link.springer.com/chapter/10.1007/978-3-540-27813-9_41
26. Granovetter, M.: Threshold Models of Collective Behavior. American Journal of Sociology **83**(6), 1420 (1978), https://doi.org/10.1007/978-3-658-21742-6_54
27. Guille, A., Hacid, H., Favre, C., Zighed, D.A.: Information Diffusion in Online Social Networks: A Survey. SIGMOD Rec. **42**(2), 1728 (Jul 2013). https://doi.org/10.1145/2503792.2503797
28. de Haan, R., van de Pol, I.: On the Computational Complexity of Model Checking for Dynamic Epistemic Logic with S5 Models. FLAP **8**(3), 621–658 (2021), https://collegepublications.co.uk/ifcolog/?00045
29. Halpern, J.Y., O'Neill, K.R.: Secrecy in Multiagent Systems. ACM Trans. Inf. Syst. Secur. **12**(1) (Oct 2008). https://doi.org/10.1145/1410234.1410239
30. Hughes, D., Shmatikov, V.: Information Hiding, Anonymity and Privacy: A Modular Approach. Journal of Computer Security **12** (02 2003). https://doi.org/10.3233/JCS-2004-12102
31. Laroussinie, F., Markey, N., Schnoebelen, P.: Temporal Logic with Forgettable Past. In: Proceedings 17th Annual IEEE Symposium on Logic in Computer Science. pp. 383–392 (2002). https://doi.org/10.1109/LICS.2002.1029846
32. Li, Y., Li, Y., Yan, Q., Deng, R.H.: Privacy Leakage Analysis in Online Social Networks. Computers and Security **49**, 239–254 (2015). https://doi.org/10.1016/j.cose.2014.10.012
33. Lomuscio, A., Qu, H., Raimondi, F.: MCMAS: An Open-Source Model Checker for the Verification of Multi-Agent Systems. Int. J. Softw. Tools Technol. Transf. **19**(1), 930 (Feb 2017), https://doi.org/10.1007/s10009-015-0378-x
34. Lomuscio, A., Raimondi, F.: MCMAS: A Model Checker for Multi-agent Systems. In: Hermanns, H., Palsberg, J. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 450–454. Springer Berlin Heidelberg, Berlin, Heidelberg (2006), https://link.springer.com/chapter/10.1007%2F11691372_31
35. Madejski, M., Johnson, M., Bellovin, S.M.: A Study of Privacy Settings Errors in an Online Social Network. In: 2012 IEEE International Conference on Pervasive Computing and Communications Workshops. pp. 340–345 (2012). https://doi.org/10.1109/PerComW.2012.6197507
36. Maggi, A., Petrocchi, M., Spognardi, A., Tiezzi, F.: A language-based approach to modelling and analysis of Twitter interactions. Journal of Logical and Algebraic Methods in Programming **87**, 67–91 (2017). https://doi.org/10.1016/j.jlamp.2016.11.003
37. Mousavi, M.R., Varshosaz, M.: Telling Lies in Process Algebra. In: Pang, J., Zhang, C., He, J., Weng, J. (eds.) 2018 International Symposium on Theoretical Aspects of Software Engineering, TASE 2018, Guangzhou, China, August 29-31, 2018. pp. 116–123. IEEE Computer Society (2018). https://doi.org/10.1109/TASE.2018.00023
38. Ortiz-Ospina, E.: The Rise of Social Media. https://ourworldindata.org/rise-of-social-media (2019), accessed: 2020-01-29
39. Pardo, R., Balliu, M., Schneider, G.: Formalising Privacy Policies in Social Networks. Journal of Logical and Algebraic Methods in Programming **90** (03 2017). https://doi.org/10.1016/j.jlamp.2017.02.008
40. Picazo-Sanchez, P., Pardo, R., Schneider, G.: Secure Photo Sharing in Social Networks. In: De Capitani di Vimercati, S., Martinelli, F. (eds.) ICT Systems Security and Privacy Protection. pp. 79–92. Springer International Publishing, Cham (2017), https://doi.org/10.1007/978-3-319-58469-0_6

41. Plotkin, G.: A Structural Approach to Operational Semantics. The Journal of Logic and Algebraic Programming **60-61**, 17–139 (2004). https://doi.org/10.1016/j.jlap.2004.05.001
42. Ruan, J., Thielscher, M.: A Logic for Knowledge Flow in Social Networks. In: Wang, D., Reynolds, M. (eds.) AI 2011: Advances in Artificial Intelligence. pp. 511–520. Springer Berlin Heidelberg, Berlin, Heidelberg (2011), https://link.springer.com/chapter/10.1007%2F978-3-642-25832-9_52
43. Seligman, J., Liu, F., Girard, P.: Facebook and the Epistemic Logic of Friendship. CoRR **abs/1310.6440** (2013), http://arxiv.org/abs/1310.6440
44. Such, J.M., Espinosa, A., García-Fornes, A.: A survey of Privacy in Multi-Agent Systems. The Knowledge Engineering Review **29**(3), 314344 (2014). https://doi.org/10.1017/S0269888913000180
45. Warren, S.D., Brandeis, L.D.: The Right to Privacy. Harvard Law Review **4**(5), 193–220 (1890), http://www.jstor.org/stable/1321160
46. Westin, A.F.: Privacy and Freedom. Washington and Lee Law Review **25**(1) (1968), https://scholarlycommons.law.wlu.edu/wlulr/vol25/iss1/20/
47. Yu, L., Motipalli, S.M., Lee, D., Liu, P., Xu, H., Liu, Q., Tan, J., Luo, B.: My Friend Leaks My Privacy: Modeling and Analyzing Privacy in Social Networks. In: Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies. p. 93104. SACMAT '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3205977.3205981
48. Zheleva, E., Getoor, L.: Privacy in Social Networks: A Survey. In: Aggarwal, C.C. (ed.) Social Network Data Analytics. pp. 277–306. Springer US, Boston, MA (2011), https://link.springer.com/chapter/10.1007/978-1-4419-8462-3_10