Secrecy Preserving BDI Agents Based on Answerset Programming

Patrick Krümpelmann and Gabriele Kern-Isberner

Technische Universität Dortmund, Information Engineering Group

Abstract. We consider secrecy from the point of view of an autonomous knowledge-based and resource-bound agent with incomplete and uncertain information, situated in a multi agent system. We investigate properties of secrecy and the preservation thereof in this setting and formulate desirable properties. Based on these ideas we develop a flexible BDI-based agent model and define an instance widely based on answerset programming. We show that and how our model and instance satisfy the proposed properties. We implemented our developed extendable framework for secrecy-preserving agents based on JAVA and answerset programming.

1 Introduction

On the topic of secrecy a large body of work exists and diverse definitions of secrecy in various settings with different properties have been developed. For multiagent systems the main research focus herein lies on strong notions of secrecy of a whole (multiagent) system, for an overview see [4,11]. Secrecy is generally imposed by some global definition of secret information from a global, complete view of the entire system. While substantial work on the definition of secrecy exists mechanisms for secrecy preservation in multiagent systems are lacking.

In this work we consider secrecy and secrecy preservation from the point of view of an autonomous knowledge-based agent with incomplete and uncertain information, situated in a multiagent system. Agents reason under uncertainty about the state of the environment, the reasoning of other agents and possible courses of action. They pursue their goals by performing actions in the environment including the communication with other agents. On the one hand, the exchange of information with other agents is often essential for an agent in order to achieve its goals; especially if the agent is part of a coalition. On the other hand the agent is interested, or obliged, not to reveal certain information, its secrets. Restriction of communication leads to a loss of performance and utility of the individual agent, coalitions and the whole multiagent system. A good solution of the implied conflict between the agent's goal to preserve secrecy and its other goals is one that restricts communication as little as necessary in order to preserve secrecy. Secrecy of information and in particular the inference problem depend on the representation of information and the appropriate modeling of background information and of the reasoning capabilities of the agents.

M. Klusch, M. Thimm, and M. Paprzycki (Eds.): MATES 2013, LNAI 8076, pp. 124–137, 2013.
© Springer-Verlag Berlin Heidelberg 2013

Our contributions lay in several aspects. We investigate, motivate and formalize novel and general properties of an agent model with respect to secrecy and secrecy preservation from a subjective perspective of an agent with incomplete information. We develop an epistemic agent model for secrecy preservation, which is based on the abstract model presented in [7]. We show that besides the pure declaration of secrets, the properties of the belief change, the attacker modeling and the means-end reasoning components of the agent are essential for secrecy declaration and preservation and define the properties of each of the three components in detail. Moreover we define answerset programming (ASP) [3] based concrete instances to illustrate how the properties can be satisfied. We implemented the general framework as well as the ASP instance presented in this work using JAVA and available ASP solvers.

The remainder of this paper is structured as follows. First we give a very brief introduction to ASP in Section 2. Then, in Section 3, we motivate and informally develop desiderata of a secrecy preserving agent based on the belief change, the attacker modeling and the means-end-reasoning component of an agent. Based on these ideas we formalize our notion of an epistemic agent in Section 4. In Section 5 we elaborate the first, the belief change, component of an agent with respect to secrecy. In Section 6 we elaborate the second component by presenting a formalization and an approach to attacker modeling and its relevance for secrecy preservation. In Section 7 we consider the third component and develop properties and for means-end-reasoning and how to satisfy them in our instance. In Section 8 we sum up, discuss the relation to other approaches and give an outlook.

2 Answerset Programming Basics

We give a brief introduction to answerset programming [3]. Let At be the set of all atoms and Lit the set of all literals $Lit = At \cup \{\neg A \mid A \in At\}$. A rule r is written as $H(r) \leftarrow \mathcal{B}^+(r), \mathcal{B}^-(r).$, the head of the rule $H(r)$ is either empty or consists of a single literal, the body consists of $\mathcal{B}^+ = \{L_0, \ldots, L_m\}$ and $\mathcal{B}^- = \{not\ L_{m+1}, \ldots, not\ L_n\}$ with $L_0, \ldots, L_n \in Lit$. The language of rules constructed over the set of atoms At is referred to as $\mathcal{L}_{At}^{asp}$. A finite set of sentences from $\mathcal{L}_{At}^{asp}$ is called an extended logic program $P \subseteq \mathcal{L}_{At}^{asp}$. A state S is a set of literals that does not contain complementary literals L and $\neg L$ is called. A state S is a model of a program P if for all $r \in P$ if $\mathcal{B}(r)^+ \subseteq S$ and $\mathcal{B}(r)^- \cap S = \emptyset$ then $H(r) \cap S \neq \emptyset$. The reduct P^S of a program P relative to a set S of literals is defined as $P^S = \{H(r) \leftarrow \mathcal{B}^+(r) \mid r \in P, \mathcal{B}^-(r) \cap S = \emptyset\}$. An answerset of a program P is a state S that is a minimal model of P^S. The set of all answersets of P is denoted by $AS(P)$. Rule schemas can use variables which we denote by x, y, z and _ for the anonymous variable [3].

3 Properties of Secrecy and Secrecy Preservation

In this section we argue that the definition of secrecy is complex and dependent on various aspects which influence the actually obtained secrecy and the

restriction of information flow. Furthermore, we elaborate the key ideas and properties of secrecy preserving agents. In the following we give the introduction to our running example and then statements about secrecy followed by examples.

Example 1. Consider an employee, *emp*, working in a company for his boss *boss*. He wants to attend a strike committee meeting (*scm*) next week and has to ask his boss for a day off in order to attend. It is general knowledge that the agent *boss* puts every agent who attends the *scm* on her blacklist of employees to be fired next.

Secrets are not uniform in their content as an agent has different secrets with respect to different agents.

Example 2. In our example, *emp* wants to keep his attendance to the *scm* secret from *boss* but not from other employees that also want to attend the *scm*.

Secrets are also not uniform with respect to their strength. That is, an agent wants to keep some information more secret than other. These differences in strength of secrets arise naturally from the value of the secret information. The value of secret information depends on the severeness of the negative effects, or the cost, for the agent resulting from disclosure of the secret information. These costs can differ widely and consequently the agent is interested in not revealing secret information to different degrees.

Example 3. *emp* does not even want his *boss* to be suspicious about him attending the *scm* (secrecy with respect to a credulous reasoner). He also does not want other employees that are against the strike to know that he attends the *scm*. However, with respect to the latter he considers it sufficient that they do not know for sure that he attends (secrecy with respect to a skeptical reasoner).

Secrets are also not static, they arise, change and disappear during runtime of an agent such that it has to be able to handle these changes adequately.

Example 4. If *emp* realizes that *boss* overheard his phone call with the strike committee he should give up his corresponding secret.

These considerations lead to the following formulation of properties of secrets: *(S1)* secrets can be held with respect to specific agents, *(S2)* secrets can vary in strength, *(S3)* secrets can change over time.

Now we want to formulate properties of a secrecy preserving agent and begin with an informal formulation. We assume a multiagent system with a set of agents $\mathfrak{A}$. We use the agent identifier $\mathcal{X}$ to denote an arbitrary agent. For the representation of the secrecy scenario it is convenient to focus on the communication between two agents, the modeled agent $\mathcal{D}$ which wants to defend its secrets from a potentially attacking agent $\mathcal{A}$. Defining secrets does not define the preservation of secrecy and its properties. The intuitive formulation of our notion of secrecy preservation can be formulated as: *An agent $\mathcal{D}$ preserves secrecy if, from its point of view, none of its secrets Φ that it wants to hide from agent $\mathcal{A}$ is, from $\mathcal{D}$'s perspective, believed by $\mathcal{A}$ after any of $\mathcal{D}$'s actions (given that $\mathcal{A}$ does not believe Φ already).*

The actual quality of secrecy preservation is highly dependent on the accuracy of the view of $\mathcal{D}$ on the agent $\mathcal{A}$ and its supposed reasoning capabilities as well as on $\mathcal{D}$'s information processing and adaptation of its beliefs and view on $\mathcal{A}$ in the dynamic scenario. To make the importance clear, a completely ignorant agent would never subjectively violate secrecy as it would ignore its violation of secrecy. Likewise underestimating as well as overestimating the capabilities of an $\mathcal{A}$ can lead to a violation of secrecy. In particular a secrecy preserving agent should satisfy the following properties: (P1) The agent is aware of the information communicated to other agents and the meta-information conveyed by its actions, (P2) The agent simulates the reasoning of other agents, (P3) The agent considers possible meta-inferences from conspicuous behavior such as (a) selfcontradiction, (b) refusal, (P4) For all possible states and perceptions the agent does not perform any action that leads to secrecy violation, (P5) The agent only weakens secrets if it is unavoidable due to information coming from third parties and only as much as necessary.

As we shall see, the properties (P1) and (P5) are related to the belief change component of $\mathcal{D}$, (P2) and (P3) to the way $\mathcal{D}$ models $\mathcal{A}$ and (P4) to the means-end reasoning behavior of $\mathcal{D}$. In the following we elaborate on all properties, formalize them and develop corresponding agent components and show which formalized properties are satisfied.

4 Formal Framework

We present an epistemic model of agency which stresses the knowledge representation and reasoning under uncertainty and incorporates secrets, and views of an agent on the information available to other agents. The reasoning under uncertainty is formalized by belief operators which can be more or less credulous. We then use these notions to define secrets and secrecy preservation.

The general framework as presented in [7] generalizes a variety of agent models. Here, we use a more concrete model loosely based the well known beliefs, desires, intentions (BDI) architecture [10]. Note that the BDI model just serves as an example agent model and that all properties and operators developed here are independent of it and are applicable to virtually all agent models. In our epistemic view of agency, the agent's epistemic state contains a representation of its current desires and intentions which guides its behavior. The functional component of a BDI agent consists of a change operation of the epistemic state and an action function, executing the next action as determined by the current epistemic state. Our agent model is illustrated in Figure 4.

Definition 1 (Epistemic BDI Agent). *An* agent $\mathcal{D}$ *is a tuple* $(\mathcal{K}_\mathcal{D}, \xi_\mathcal{D})$ *comprising an epistemic state* $\mathcal{K}_\mathcal{D}$ *and a functional component* $\xi_\mathcal{D}$*. A* BDI-Epistemic-State *is a tuple* $\mathcal{K}_\mathcal{D} = \langle\langle V_{\mathcal{D},W}, \mathcal{V}_\mathcal{D}, \mathcal{S}_\mathcal{D}\rangle, \Delta_\mathcal{D}, \mathcal{I}_\mathcal{D}\rangle$*. It consists of a world view* $V_{\mathcal{D},W}$*, a set of agent views* $\mathcal{V}_\mathcal{D} = \{V_{\mathcal{D},\mathcal{X}} \mid \mathcal{X} \in \mathfrak{A} \setminus \{\mathcal{D}\}\}$*, a set of secrets* $\mathcal{S}_\mathcal{D}$*, a set of desires* $\Delta_\mathcal{D}$*, and a set of intentions* $\mathcal{I}_\mathcal{D}$*. We refer to the first component as the agent's beliefs* $\mathcal{B}(\mathcal{K}_\mathcal{D}) = \mathcal{B}_\mathcal{D}$*. We set* $V_W(\mathcal{B}) = V_W(\mathcal{K}_\mathcal{D}) = V_{\mathcal{D},W} \subseteq \mathcal{L}^{asp}_{At}$*,* $V_\mathcal{X}(\mathcal{B}_\mathcal{D}) = V_\mathcal{X}(\mathcal{K}_\mathcal{D}) = V_{\mathcal{D},\mathcal{X}} \subseteq \mathcal{L}^{asp}_{At}$ *and* $\mathcal{S}(\mathcal{K}_\mathcal{D}) = \mathcal{S}(\mathcal{B}_\mathcal{D}) = \mathcal{S}_\mathcal{D}$*. The functional*

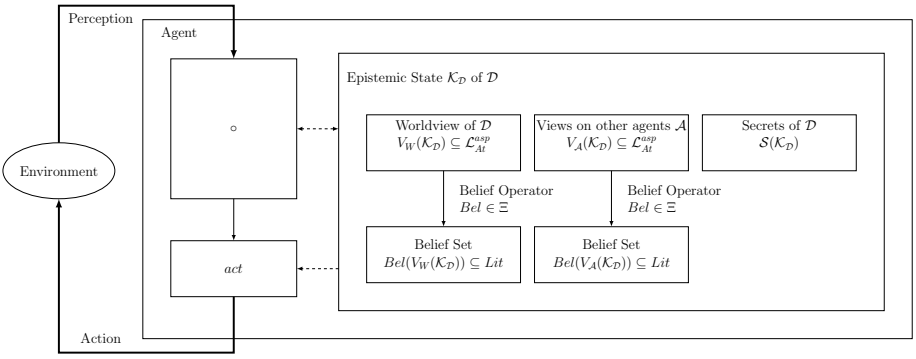

Fig. 1. Epistemic Agent Model

component $\xi_\mathcal{D} = (\circ_\mathcal{D}, \mathsf{act}_\mathcal{D})$ *consists of an change operator* $\circ_\mathcal{D}$ *and an action operator* $\mathsf{act}_\mathcal{D}$.

A belief operator determines the currently held beliefs of the agent given a view. In the ASP setting beliefs are represented by an answerset, i. e. a set of literals, and a view by an extended logic program. An agent with incomplete and uncertain information might employ different belief operators which are more or less credulous. A belief operator is *more credulous* than another one if for all views the belief set of the latter is a subset of the belief set of the former.

Definition 2 (Belief Operators). *A* belief operator *is a function* $Bel : \mathcal{L}_{At}^{asp} \rightarrow Lit$. Ξ *is a finite* family of belief operators Ξ *plus the ignorant operator* $Bel_\emptyset(V) = \emptyset$. *We assume a credulity order* $\prec$ *on* Ξ *such that if* $Bel < Bel'$ *for some* $Bel, Bel' \in \Xi$ *then for all* $V \in \mathcal{L}_{At}^{asp}$ $Bel(V) \subseteq Bel'(V)$. *The* ASP belief operator family *is given by* $\Xi^{asp} = \{Bel_{skep}^{asp}, Bel_{cred}^{asp}, Bel_\emptyset\}$, $Bel_{cred}^{asp}(P) = \cap AS(P)$ *and* $Bel_{skep}^{asp}(P) = \cup AS(P)$ *and* $Bel_{cred}^{asp} \succ Bel_{skep}^{asp} \succ Bel_\emptyset$.

The definition of a family of belief operators abstracts form the underlying formalism and inference mechanism. Thereby it captures a wide range of formalisms from purely qualitative ones to plausibilistic ones. The ASP instance considered here is just one example, used for the illustration of the approach in this paper. To define secrets, the information to be kept secret has to be defined. Also, the agent from which the information shall be kept secret has to be defined and lastly the strength of the secret has to be expressed. We make use of the belief operators to express the strength of a secret.

Definition 3 (Secrets). *A* secret *is a tuple* $(\Phi, Bel, \mathcal{A})$ *which consists of a formula* $\Phi \in Lit$, *a belief operator* $Bel \in \Xi$ *and an agent identifier* $\mathcal{A} \in \mathfrak{A}$. *The set of secrets of agent* $\mathcal{D}$ *is denoted by* $\mathcal{S}(\mathcal{K}_\mathcal{D})$.

Assigning a more credulous belief operator to a secret leads to a stronger protection of secret information, as illustrated in Example 3. That is, if $\mathcal{D}$ reveals some

information, a credulous attacker might infer some secret information while a skeptical one with the same revealed information might not. In the former case the defender should not have revealed the information. Formally, considering two secrets $(\Phi, Bel, \mathcal{A})$ and $(\Phi, Bel', \mathcal{A})$, the former is stronger than the latter iff $Bel \succ Bel'$.

Observation 1. *The definition of secrets in Definition 3 satisfies (S1), (S2) and (P2).*

Example 5. We model the *scm* scenario from Example 2 and in particular the initial epistemic state of the employee *emp*, $\mathcal{K}_{emp} = \langle \{V_{emp,W}, \mathcal{V}_{emp}, \mathcal{S}_{emp}\}, \Delta_{emp}, \mathcal{I}_{emp} \rangle$, with $\mathcal{V}_{emp} = \{V_{emp,boss}\}$. We assume that *emp* and *boss* share the same background knowldge, such that $V_{emp,W} = V_{emp,boss} = P_{view}$ with:

$$
\begin{aligned}
P_{view} = \{\ & r_1 : \neg attend\_work && \leftarrow\ excused.\\
& r_2 : excused && \leftarrow\ attend\_scm.\\
& r_3 : excused && \leftarrow\ medical\_appointment.\\
& r_4 : attend\_scm && \leftarrow\ not\ medical\_appointment, asked\_for\_excuse.\\
& r_5 : medical\_appointment && \leftarrow\ not\ attend\_scm, asked\_for\_excuse.\\
& r_6 : blacklist && \leftarrow\ not\ excused, \neg attend\_work.\\
& r_7 : blacklist && \leftarrow\ attend\_scm.\\
& r_8 : attend\_work && \leftarrow\ not\ \neg attend\_work.\}
\end{aligned}
$$

The program encodes that *emp* has to be excused in order to not go to work (r_1). He is excused if the attends the scm or if he has a medical appointment (r_1–r_2). If he asks to be excused these two possible explanations exist (r_4–r_5). If he is absent without being excused he will be blacklisted (r_6). If he attends the scm, and is thus excused, he will still be blacklisted (r_7). He normally goes to work (r_8). The set of answer sets is $AS(P_{view}) = \{\{attend\_work\}\}$. The secret of the employee is $\mathcal{S}_{emp} = \{(attend\_scm, Bel^{asp}_{skep}, boss)\}$. The initial set of desires of the employee is $\Delta_{emp} = \{attend\_scm\}$.

For secrecy preservation the dynamics of the epistemic state induced by actions and perceptions have to be considered. We assume a set of possible actions *actions* and a set of possible perceptions *percepts*, including the empty ones. To make the formalism more comprehensible and to illustrate a concrete instance we consider communicating agents here. Note that other types of actions and perceptions, such as manipulations in some environment are also captured by the general framework. For the illustration here, we assume that actions as well as perceptions τ are speech acts from a set of speech acts $\langle A_s, \{A_{r_1}, \ldots, A_{r_n}\}, type, \Phi \rangle$ specifying the source $A_s \in \mathfrak{A}$, the receivers $A_{r_1} \in \mathfrak{A}$ to $A_{r_n} \in \mathfrak{A}$, the type *type* and the informational content $\Phi \in Lit$. The main difference between perceptions and actions is that perceptions represent actions performed by other agents while actions represent the actions the agent under consideration has performed. We differentiate between requesting speech acts $\Psi_R = \{\mathsf{query}, \mathsf{justify}\}$ and informative speech acts $\Psi_I = \{\mathsf{inform}, \mathsf{answer}, \mathsf{justification}\}$, so $type \in \Psi_R \cup \Psi_I$. The set of all possible speech acts is denoted by $\Gamma = percepts = actions$. For each perception $p \in percepts$ an agent cycle results in a new epistemic state determined by $\mathcal{K}_{\mathcal{D}} \circ_{\mathcal{D}} p \circ_{\mathcal{D}} \mathsf{act}_{\mathcal{D}}(\mathcal{K}_{\mathcal{D}} \circ_{\mathcal{D}} p)$. The set of all possible successive epistemic states of

agent $\mathcal{D}$ is determined by the set of initial epistemic states $\Lambda^0_\mathcal{D}$ and all respective successor states for all possible perceptions and corresponding actions of $\mathcal{D}$. i. e. $\Omega_{\mathsf{act}_\mathcal{D},\circ_\mathcal{D}}(\Lambda^0_\mathcal{D}, percepts) = \{\mathcal{K} \mid \mathcal{K} = \mathcal{K}_0 \circ_\mathcal{D} p_0 \circ_\mathcal{D} \mathsf{act}_\mathcal{D}(\mathcal{K}_0 \circ_\mathcal{D} p_0) \circ_\mathcal{D} \ldots, p_0, \ldots, p_i \subseteq percepts, i \in \mathbb{N}_0, \mathcal{K}_0 \in \Lambda^0_\mathcal{D}\}$. Our intuitive idea of secrecy preservation as given in Section 3 expresses that we want to assure that the secrecy preserving agent always maintains an epistemic state in which it believes that no other agent believes in something that it wants to keep secret. More exactly, it also distinguishes between secrets towards different agents and what it means to it that the information is kept secret. The term "*always maintains*" means that for all possible scenarios of communication the agent acts such that a safe epistemic state is maintained.

Definition 4 (Secrecy-preserving Agent). *Let $\mathcal{D} = (\mathcal{K}_\mathcal{D}, (\circ_\mathcal{D}, \mathsf{act}_\mathcal{D}))$ be an agent and percepts a set of perceptions. An epistemic state $\mathcal{K}_D$ is* safe *iff $\Phi \notin Bel(V_\mathcal{A}(\mathcal{K}_\mathcal{D}))$ for all $(\Phi, Bel, \mathcal{A}) \in \mathcal{S}(\mathcal{K}_\mathcal{D})$.*

Let $\Lambda^0_\mathcal{D}$ be a set of initial safe epistemic states. We call $\mathcal{D}$ secrecy preserving *with respect to $\Lambda^0_\mathcal{D}$ and percepts if and only if for all $\mathcal{K}_\mathcal{D} \in \Omega_{\mathsf{act},\circ}(\Lambda^0_\mathcal{D}, percepts)$ it holds that $\mathcal{K}_\mathcal{D}$ is safe.*

Example 6. We continue the previous example and check whether the initial $\mathcal{K}_{emp}$ is safe. The set of answersets of P_{view} is $AS(P_{view}) = \{\{attend\_work\}\}$. Consequently $attend\_scm \notin Bel^{asp}_{skep}(P_{view})$ and $\mathcal{K}_{emp}$ is safe.

Observation 2. *The definition of a secrecy preserving agent in Definition 4 satisfies (P4).*

We just defined the notion of a secrecy preserving agent. However, as discussed in Section 3 the actual resulting properties of secrecy preservation result from the properties of the change operation $\circ$ and the attacker modeling. Moreover, the actual preservation of secrecy is realized by the means-end-reasoning of the agent. We elaborate these aspects in the next sections.

5 Belief Change and Secrecy

We decompose the belief change of an epistemic state into sub-operations on its components. Based on these we motivate and define properties with respect to secrecy preservation which formalize and concretize the ideas given in (P1) and (P5). Finally we give concrete instances of such operators for the ASP instance and show that they satisfy the defined propertires.

5.1 Structure and Properties of the Change Operator

The change operator updates epistemic state of an agent upon incoming perceptions and actions. Formally, $\mathcal{K} \circ \tau = \mathcal{K}' = \langle \mathcal{B}', \Delta', \mathcal{I}' \rangle$. The change operator can be structured into several sub-operations for the different components of the epistemic state. Hereby the belief component is the only one being directly influenced by the new information, then the change of the desires is only dependent

on the changed beliefs and the update of the intentions on the changed beliefs and desires. Formally the sub-operations are $\circ_{\mathcal{B}} : \mathcal{B} \times \Gamma \to \mathcal{B}$, $\circ_{\Delta} : \Delta \times \mathcal{B} \to \Delta$ and $\circ_{\mathcal{I}} : \mathcal{I} \times \mathcal{B} \times \Delta \to \mathcal{I}$. The update operations can then be represented as

$$\langle \mathcal{B}, \Delta, \mathcal{I} \rangle \circ \tau = \langle \mathcal{B} \circ_{\mathcal{B}} \tau, \Delta \circ_{\Delta} (\mathcal{B} \circ_{\mathcal{B}} \tau), \circ_{\mathcal{I}} (\mathcal{I}, \mathcal{B} \circ_{\mathcal{B}} \tau, \Delta \circ_{\Delta} (\mathcal{B} \circ_{\mathcal{B}} \tau)) \rangle.$$

In this section we focus on the belief change operation and its relevance to secrecy and elaborate the desire and intention change in the context of means-end-reasoning later on. The input speech act τ for the $\circ_{\mathcal{B}}$ operation can be either a perception or an action. In both cases it might be an informative or a requesting speech acts. All four cases have different semantics and lead to different changes. That is, the input has to be interpreted and represented in the language of the respective belief component we introduce translation operators $t_W : \Gamma \to \mathcal{L}_{At}^{asp}$ for the world view and $t_V : \mathfrak{A} \times \Gamma \to \mathcal{L}_{At}^{asp}$ for agent views. The result of the translation is then used to update the respective component by use of an *inner revision operator* $* : \mathcal{L}_{At}^{asp} \to \mathcal{L}_{At}^{asp}$. Secrets are updated on the basis of the agent's updated beliefs and views such that the change operator for secrets $*_{\mathcal{S}}$ is dependent on these as well as on the incoming information, i. e. $*_{\mathcal{S}} : 2^{\mathcal{L}_{\mathcal{S}}} \times \mathcal{L}_{At}^{asp} \times 2^{\mathcal{L}_{At}^{asp}} \times \Gamma \to \mathcal{L}_{\mathcal{S}}$. We define the changes of the $\circ_{\mathcal{B}}$ operator to the components by suboperations, such that

$$(V_W, \mathcal{V}, \mathcal{S}) \circ_{\mathcal{B}} \tau = \langle V_W * t_W(\tau),\ \mathcal{V} * t_V(\mathcal{A}, \tau),\ *_{\mathcal{S}}(\mathcal{S}, V_W * t_W(\tau), \mathcal{V} * t_V(\mathcal{A}, \tau), \tau) \rangle \ (*)$$

We define a set of properties on the just defined operations which formalize the properties (S3), (P1) and (P5). For secrecy preservation it is necessary that the agent does not give up any secrets upon reflecting its own actions since it would be able to perform arbitrary actions without violating secrecy by abandoning its secrets. Thus, the agent must not be able to preserve a safe epistemic state by modifying its secrets.

Secrets-Invariance$_{\circ_{\mathcal{B}}}$ If $\tau \in$ *actions* then $\mathcal{S}(\mathcal{B} \circ_{\mathcal{B}} \tau) = \mathcal{S}(\mathcal{B})$

The *Secrets-Invariance* property is restricted to inputs that are actions. These actions are those of the agent itself and perceptions reflect changes in the environment or actions of other agents. For the latter the postulate should not hold. That is, an agent should not be able to ignore the fact that a secret has been revealed due to changes in the environment or actions of other agents. This is expressed in the following property.

Acknowledgment$_{\circ_{\mathcal{B}}}$ If $\tau \in$ *percepts* then $\mathcal{B} \circ_{\mathcal{B}} \tau$ is safe.

The changes to the set of secrets in order to achieve a safe epistemic state should be minimal. That is, a secret should not be weakened without a reason, i. e. it is violated, it should not be strengthened and it should be weakened minimally.

Min-Secrecy-Weakening$_{\circ_{\mathcal{B}}}$ If $(\Phi, Bel, \mathcal{A}) \in \mathcal{S}(\mathcal{B})$ and $(\Phi, Bel', \mathcal{A}) \in \mathcal{S}(\mathcal{B} \circ_{\mathcal{B}} \tau)$ with $Bel \neq Bel'$ then $\Phi \in Bel(V_{\mathcal{A}}(\mathcal{K}_{\mathcal{D}} \circ_{\mathcal{B}} \tau))$ and there is no Bel'' such that $\Phi \notin Bel''(V_{\mathcal{A}}(\mathcal{K}_{\mathcal{D}} \circ_{\mathcal{B}} \tau))$ with $s(Bel'') > s(Bel')$.

Another secrecy relevant property of belief change arises from the changes to views of other agents. An agent should not be able to preserve secrecy by ignoring the effects of its own actions on the beliefs of potentially attacking agents. In particular the information for some agent $\mathcal{A}$ contained in an action of $\mathcal{D}$ should be incorporated into $\mathcal{D}$'s view on $\mathcal{A}$. This is formulated by the next property.

Awareness$_{\circ_\mathcal{B}}$ If $\tau \in \mathit{actions}$ then $t_W(\tau) \in V_W(\mathcal{B} \circ_\mathcal{B} \tau)$ and for each $\mathcal{A} \in \mathfrak{A}$ $t_V(\mathcal{A}, \tau) \in V_\mathcal{A}(\mathcal{B} \circ_\mathcal{B} \tau)$.

There might very well be actions which are not visible to all agents and therefore should also not affect the view on all agents.

Example 7. If agent $\mathcal{D}$ is communicating privately with some agent $\mathcal{A}$ it should change its view on $\mathcal{A}$ but not its view on other agents.

This is achieved by use of appropriate translation operators which select the relevant information for each agent. For agents that are not affected by the information the transformation function returns the empty set.

Observation 3. *The satisfaction of* Secrets-Invariance$_\circ$, Acknowledgment$_\circ$, Min-Secrecy-Weakening$_\circ$ *and* Awareness$_\circ$ *of the belief change operator of an agent corresponds to the satisfaction of (P1), (P5) and (S3).*

We consider all of the properties defined in this section essential for a secrecy preserving agent, hence the goal is to define appropriate operators.

5.2 Concrete Revision Operations

In the following we define an instance of the translation operators t_V, t_W, the inner revision operator $*$ and the revision of secrets operator $*_\mathcal{S}$.

The translation operator, in accordance with (P1), has to consider information on two levels, on the one hand the actual information, that is the informational content of the speech act. On the other hand the meta-information about the speech act that has been performed which includes especially the information about the sender and the type of speech act and information revealed by these parameters. Both aspects have to be represented in the language of a logic program. We introduce an auxiliary logic program to support the representation of information on both levels for the ASP translation operators and to enable reasoning possibilities on the meta-information which are used for attacker modeling. To this end we reify literals, for each atom $A \in At$ introduce three constant symbols $C(A) = \{a, na, \lambda a\}$ the first two to represent the literals $A, \neg A$; λa can stand for both occurrences a and na. We define $const(L) = a$ if $L = A$ and $const(L) = na$ if $L = \neg A$. And $var(L) = \lambda a$ if $L = A$ or $L = \neg A$. We assume that the time is represented by a simple counter t and a literal $time(t)$, counting the agent cycles. The program P_{aux} consists of the following set of rules:

For all $A \in At$:

$$
\begin{array}{lll}
A & \leftarrow holds(a). \\
\neg A & \leftarrow holds(na). \\
related(\lambda a, a). & related(\lambda a, na). \\
at(t) & \leftarrow time(t), not\ time(s), s > t.
\end{array}
$$

The predicate $related(x, y)$ expresses that x is semantically related to y. We make use of the auxiliary construction to represent the informational content of a speech act and formulate the translation function as follows.

Definition 5 (Translation Function). *Let be* $\mathcal{D} \in \mathfrak{A}, \tau = \langle A_s, \{A_{r_1}, \ldots, A_{r_n}\}, type, L\rangle$ *and* $counter = t$. *The translation functions of* $\mathcal{D}$ *are defined as:*

$$t_V(\tau) = \begin{cases} \{type(A_s, const(L), t)., L.\} & \text{if } \tau \in \Psi_I \\ \{type(A_s, var(L), t).\} & \text{if } \tau \in \Psi_R \end{cases}$$

$$t_W(\tau) = \begin{cases} \{type(A_s, const(L), t)., L.\} & \text{if } A_s \neq \mathcal{D} \text{ and } \tau \in \Psi_I \\ \{type(A_s, var(L), t).\} & \text{else} \end{cases}$$

In general the information of a speech act is represented by the predicate $type(A_s, const(L), t)$ with the semantics that a speech act of type *type* has been performed by agent A_S with logical content $const(L)$ at time t. For requesting speech acts $var(L)$ is used to represent that information related to L has been requested. Besides this representation of the information about the speech act the logical content has to be represented. For informational speech acts this is the actual literal L of which the agent has been informed. Hence L is added to the input set for the inner revision operator for informational speech acts, unless $\mathcal{D}$ is updating its world view by its own action. The result of the translation operator is the input for the inner revision operator. As intended we revise a a logic program by another one and face a standard belief revision problem and can make use of operators for it. For our ASP instance any operator satisfying the basic set of properties can be used. In particular *Success:* $Q \subseteq P * Q$, *Inclusion:* $P * Q \subseteq P \cup Q$ and *Vacuity:* If $P \cup Q$ is consistent, then $P \cup Q \subseteq P * Q$ are satisfied. For details refer to, e. g., [6].

As specified by the properties given above secrets are updated only by information about actions of other agent. In this case the secrets shall be modified minimally in order to preserve secrecy. To this end, we determine the strongest wrt. secrecy, that is the belief operator by which some information Φ is preserved in the current view $\mathcal{V}$. Formally: $curr(\Xi, \mathcal{V}, \Phi) = \arg\max s(Bel), Bel \in \{Bel \in \Xi \mid \Phi \notin Bel(\mathcal{V})\}$. Then we can define the change operator for secrets $*_{\mathcal{S}}$ as

$$\mathcal{S}(\mathcal{K}) *_{\mathcal{S}} (V'_W, \mathcal{V}', \tau) = \begin{cases} \mathcal{S}(\mathcal{K}) & \text{if } A_S = \mathcal{D} \\ \omega(\mathcal{S}(\mathcal{K}), \mathcal{V}') & \text{else} \end{cases}$$

with $\omega(\mathcal{S}(\mathcal{K}), \mathcal{V}') = \{(\Phi, Bel', \mathcal{A}) \mid (\Phi, Bel, \mathcal{A}) \in \mathcal{S}(\mathcal{K})$ and

$$Bel' = \begin{cases} curr(\Xi, \mathcal{V}_{\mathcal{A}}, \Phi) & \text{if } curr(\Xi, \mathcal{V}_{\mathcal{A}}, \Phi) < Bel \\ Bel & \text{else.} \end{cases} \}$$

If the updating information is an action of $\mathcal{D}$, no changes are performed. Otherwise the belief operator of any secret whose assigned operator is stronger than the currently strongest one preserving secrecy is replaced by the latter. This means that only those secrets are modified which would be violated otherwise. We can show that this specification $\circ_{\mathcal{B}}$ satisfies the properties postulated previously.

Proposition 1. *Let* $\mathcal{D}$ *be an agent,* t_V, t_W *and* $*_{\mathcal{S}}$ *be operators as defined in this section and* $P_{aux} \subseteq V_{\mathcal{A}}(\mathcal{K}_{\mathcal{D}})$ *and* $P_{aux} \subseteq V_W(\mathcal{K}_{\mathcal{D}})$. *Let* $*$ *be an ASP base-revision*

operator. The $\circ_{\mathcal{B}}$ *operator of* $\mathcal{D}$ *defined by as in (*) satisfies* Secrets-Invariance$_\circ$, Acknowledgment$_\circ$, Min-Secrecy-Weakening$_\circ$ *and* Awareness$_\circ$.

Proof. Sketch: The satisfaction of Secrets-Invariance$_\circ$, Acknowledgment$_\circ$ and Min-Secrecy-Weakening$_\circ$ *follow from the definition of* $*_{\mathcal{S}}$*, the satisfaction of* Awareness$_\circ$ *follows from Definition 5 and the satisfaction of the* Success *postulate by* $*$.

6 Attacker Modelling

The principles P2 (simulation) and P3 (meta-inferences) of secrecy preservation laid out in Section 3 raise the need for adequate modeling of the background information and reasoning methods and capabilities of the attacker. Both over- and underestimating the capabilities of an attacker can lead to violation of secrecy. Hence modeling these is essential for realistic preservation of secrecy. In particular information about the declaration of secrets and meta-inference from $\mathcal{D}$'s behavior have to be considered. Which behavior is conspicuous and will lead to a violation of secrecy is heavily dependent on the reasoning capabilities and properties of the attacker. We define three properties of an attacker for secrecy preservation which $\mathcal{D}$ might take into consideration.

Secret Aware $\mathcal{A}$ knows which information $\mathcal{D}$ does not want to reveal to $\mathcal{A}$ if $\mathcal{D}$ would belief it to be true.

Contradiction Sensitive $\mathcal{A}$ considers self-contradictions of $\mathcal{D}$ with respect to information it wants to keep secret as reason to infer the secret.

Refusal Sensitive $\mathcal{A}$ considers the $\mathcal{D}$'s refusal to answer with respect to information it wants to keep secret as reason to infer the secret.

We present a simple version of an ASP approach to realize views of attackers satisfying the properties. To this end we consider the following set of rules which is then used to define programs which represent a specific property and can be modularly added to a view on an $\mathcal{A}$.

(1) For each $(L, Bel, \mathcal{A}) \in \mathcal{S}(\mathcal{K}_{\mathcal{D}})$: $has\_secret(D, const(L))$.
(2) For all $A \in At$:
$contradiction(D, \lambda a) \leftarrow inform(D, a), inform(D, na)$.
(3) $holds(x) \leftarrow has\_secret(D, x), contradiction(D, y), related(y, x)$.
(4) $refused(D, x) \leftarrow request(\_, D, x, t_1), not\ answer(D, y, t_2), at(t_2), related(x, y), t_2 = t_1 + 1$.
(5) $holds(x) \leftarrow has\_secret(D, x), refused(D, y), related(y, x)$.

Line (1) represents the information about the secrets $\mathcal{D}$ has with respect to $\mathcal{A}$. In (2) it is expressed that $\mathcal{A}$ infers that $\mathcal{D}$ contradicted itself with respect to an atom A if it said both A and $\neg A$. If $\mathcal{D}$ contradicted itself with respect to some secret information, then $\mathcal{A}$ infers that the secret holds (3). Line (4) represents that $\mathcal{A}$ infers that $\mathcal{D}$ refused to answer about x if it was requested to do so and did not inform after the request. According to (5) $\mathcal{A}$ infers that a secret holds if $\mathcal{D}$ refused to answer with respect to it. We can define programs from the defined rules and formalize the properties given above in the ASP setting.

Definition 6. *Let* $P^{meta}_{\mathcal{S}-aware} = (1)$, $P^{meta}_{C-sensitive} = P^{meta}_{\mathcal{S}-aware} \cup (2) \cup (3)$ *and* $P^{meta}_{R-sensitive} = P^{meta}_{\mathcal{S}-aware} \cup (4) \cup (5)$. *An attacker modeling* $V_{\mathcal{A}}(\mathcal{K}_{\mathcal{D}})$ *is* secrecy aware *if* $P^{meta}_{\mathcal{S}-aware} \subseteq V_{\mathcal{A}}(\mathcal{K}_{\mathcal{D}})$, *it is* contradiction sensitive *if* $P^{meta}_{\mathcal{C}-sensitive} \subseteq V_{\mathcal{A}}(\mathcal{K}_{\mathcal{D}})$ *and it is* refusal sensitive *if* $P^{meta}_{\mathcal{R}-sensitive} \subseteq V_{\mathcal{A}}(\mathcal{K}_{\mathcal{D}})$.

Observation 4. *The satisfaction of the properties* contradiction sensitive *and* refusal sensitive *corresponds to the properties (P3) (a) and (b), respectively.*

The determination of contradictions inflicted by $\mathcal{D}$ and the one of refusals can and should be more elaborate and can easily be extended and formulated by more complex logic programs, but are outside of the scope here.

7 Means-End-Reasoning and Secrecy Preservation

We equipped the agent with the abilities to be aware of its secrets and to detect violation of secrecy. The question now is what are the necessary properties on the desire and intention change operators for secrecy preservation.

In any BDI system desire and intention change operators are implemented in one way or the other which is. Here we give a general model of intention change and show the relevant properties for secrecy preservation. Any agent has to determine how it can satisfy its intentions, high-level intentions are resolved down to atomic intentions $\mathsf{AtInt} \subseteq \mathfrak{I}$ which can be satisfied by a single action $\alpha(I)$. In any case at some point the options to satisfy some intention have to be evaluated and one of the options has to be chosen. The options for a given intention are determined by the options function $\mathsf{options} : \mathcal{I} \to 2^{\mathcal{I}'}$ and the evaluation results in a preference relation $<$ on the possible options. Here we assume that all intentions can directly be resolved to an atomic intention and set $\mathcal{I}' = \mathsf{AtInt}$. Then the set of maximally preferred options from which one is selected by the agent is set to $pref(\mathsf{options}(I)) = max_<(\mathsf{options}(I))$.

To show that an agent is secrecy preserving, as given by Definition 4, we have to show that it prefers secrecy preserving actions over non-secrecy preserving ones and that it always has a secrecy preserving option.

We generalize the change operator $\circ$ to intentions as input. In general this allows to determine the effects of the satisfaction of arbitrary intentions. For our presentation here we can set $\mathcal{K} \circ I = \mathcal{K} \circ \alpha(I)$. Given an epistemic state $\mathcal{K}$ and an intention I, an option $o \in \mathsf{options}(I)$ is *safe* iff $\mathcal{K} \circ o$ is safe. Based on this definition we define a secrecy relevant property on the preference relation on options.

Confidentiality-preference For all $I \in \mathcal{I}$, for $o, o' \in \mathsf{options}(I)$ if o is safe and o' is not then $o > o'$.

In combination with an $\mathsf{options}$ function we can show the agent choses secrecy preserving options, if they exist.

Lemma 1. *If* $<$ *satisfies* confidentiality-preference*, then if there is some safe option* $o \in \mathsf{options}(I)$*, then for all* $o' \in pref(\mathsf{options}(I))$, o' *is safe.*

Proof. The lemma follows directly from the definitions of $pref$ *and* confidentiality-preference.

A preference relation satisfying *confidentiality-preference* alone is not sufficient to guarantee secrecy preservation since it is dependent on the existence of a safe option. Hence we define the following property of an options function.

Existence For all $I \in \mathcal{I}$ and safe $\mathcal{K}$ there exists $o \in \mathsf{options}(I) : \mathcal{K} \circ o$ is safe.

We can now formulate the dependency of the notion of a secrecy preserving agent, Definition 4, and the properties of option selection as defined in this section.

Proposition 2. *Let* $>$ *be a preference relation on* AtInt *and* options *an options function of an agent* $\mathcal{D}$*. If* $>$ *satisfies Confidentiality-preference and* options *satisfies Existence, then* $\mathcal{D}$ *is secrecy preserving.*

Proof. Sketch: The proposition follows from Lemma 1 and the definitions of Confidentiality-preference, Existence, safe options and safe epistemic state.

8 Related Work and Conclusion

We presented an theoretical, conceptional and practical account of secrecy from the subjective view of an autonomous epistemic agent. We formulated properties of secrecy and secrecy preservation and developed a framework for and ASP-based instance satisfying them. We have shown in [7] that other many aspects of notions of secrecy such as [1] and [4] can be captured by our underlying model.

To the best of our knowledge no subjective account of agent based secrecy nor a concrete model or implementation of a secrecy preserving agent system has been presented so far. The closest to this is the preliminary account of integrating techniques for controlled query evaluation for databases [1] into an agent system, as presented in [2]. The database techniques are naturally limited to a fixed client-server architecture and to query-answer scenarios. In [2] it is proposed to use a *censor* to check the agents actions prior to execution and modifying them if necessary similar to the approaches from database theory for a negotiation scenario. The actual realization of this approach is left open. We argue that instead of adding an controlling instance secrecy has to be integrated into the agents' reasoning, deliberation and means-end reasoning processes to achieve autonomous secrecy preserving agents apt to perform well in a dynamic setting. However, secrecy is a very special epistemic goal which calls an appropriate epistemic model, operators and actions as presented in this work and can hardly be captured by the few existing approaches for maintenance goals, e. g. [5].

We see our model and implementation as a good basis for the further theoretical investigation as well as the implementation of secrecy preserving agents. It opens a plethora of possibilities for further investigation. In current work we run empirical evaluations and integrate advanced deliberation [9] and means-end reasoning techniques [8] in our model and implementation, and investigate further properties of secrecy in this model and the relation to other approaches.

Acknowledgements. This work has been supported by the DFG, Collaborative Research Center SFB876, Project A5. (`http://sfb876.tu-dortmund.de`)

References

1. Biskup, J.: Usability confinement of server reactions: Maintaining inference-proof client views by controlled interaction execution. In: Kikuchi, S., Sachdeva, S., Bhalla, S. (eds.) DNIS 2010. LNCS, vol. 5999, pp. 80–106. Springer, Heidelberg (2010)
2. Biskup, J., Kern-Isberner, G., Thimm, M.: Towards enforcement of confidentiality in agent interactions. In: Proc. of the 12th Intl. Workshop on Non-Monotonic Reasoning (NMR 2008), pp. 104–112 (2008)
3. Gelfond, M., Leone, N.: Logic programming and knowledge representation: the A-Prolog perspective. Artificial Intelligence 138 (2002)
4. Halpern, J.Y., O'Neill, K.R.: Secrecy in multiagent systems. ACM Transactions on Information and System Security 12, 5:1–5:47 (2008)
5. Hindriks, K.V., van Riemsdijk, M.B.: Satisfying maintenance goals. In: Baldoni, M., Son, T.C., van Riemsdijk, M.B., Winikoff, M. (eds.) DALT 2007. LNCS (LNAI), vol. 4897, pp. 86–103. Springer, Heidelberg (2008)
6. Krümpelmann, P., Kern-Isberner, G.: Belief base change operations for answer set programming. In: del Cerro, L.F., Herzig, A., Mengin, J. (eds.) JELIA 2012. LNCS, vol. 7519, pp. 294–306. Springer, Heidelberg (2012)
7. Krümpelmann, P., Kern-Isberner, G.: On agent-based epistemic secrecy. In: Proc. of the 14th Int'l Workshop on Non-Monotonic Reasoning, NMR 2012 (2012)
8. Krümpelmann, P., Thimm, M.: A logic programming framework for reasoning about know-how. In: Proc. of the 13th Int'l Workshop on Non-Monotonic Reasoning (NMR 2010) (2010)
9. Krümpelmann, P., Thimm, M., Kern-Isberner, G., Fritsch, R.: Motivating agents in unreliable environments: A computational model. In: Klügl, F., Ossowski, S. (eds.) MATES 2011. LNCS, vol. 6973, pp. 65–76. Springer, Heidelberg (2011)
10. Rao, A.S., Georgeff, M.P.: BDI-agents: from theory to practice. In: Proc. of the 1st Int'l Conference on Multiagent Systems (ICMAS 2005) (1995)
11. van der Torre, L.: Logics for Security and Privacy. In: Cuppens-Boulahia, N., Cuppens, F., Garcia-Alfaro, J. (eds.) DBSec 2012. LNCS, vol. 7371, pp. 1–7. Springer, Heidelberg (2012)