Contrôle social de la privacité selon l'intégrité contextuelle dans les systèmes décentralisés

Yann Krupa, Laurent Vercouter

Contrôle social de la privacité selon l'intégrité contextuelle dans les systèmes décentralisés

Y. Krupa krupa@emse.fr
L. Vercouter vercouter@emse.fr

Département ISCOD/Institut Henri Fayol - École des Mines de Saint-Étienne, 158 cours Fauriel, 42023 Saint-Étienne

Résumé

Les approches habituelles pour la protection de la privacité s'attachent à définir un niveau de sensibilité pour chaque information. Cette information est soit publique, soit privée et sa circulation est restreinte à un groupe d'agents prédéfini. La théorie de l'intégrité contextuelle propose de redéfinir la privacité. Selon cette théorie, toute transmission peut déclencher une violation de privacité suivant le contexte dans lequel elle a lieu. Nous utilisons cette théorie afin de proposer un modèle de protection de la privacité pour les systèmes multi-agents décentralisés. Cet article décrit le modèle PrivaCIAS qui définit la notion de violation en accord avec la théorie de l'intégrité contextuelle et implémente un contrôle social. Le modèle donne le contrôle aux agents pour détecter les violations, puis punir les contrevenants en les excluant du système.

Mots-clés : *privacité, intégrité contextuelle, réseaux sociaux, systèmes multi-agent, réseaux décentralisés*

Abstract

Contextual Integrity has been proposed to define privacy in an unusual way. Most approaches take into account a sensitivity level or a "privacy circle" : the information is said to be either private or public and to be constrained to a given group of agents, e.g. "my friends", when private. In the opposite, Contextual Integrity states that any information can make a transmission a privacy violation depending on its context. We use this theory to develop a novel framework that one can use in an open and decentralized virtual community to socially enforce privacy. This paper describes the PrivaCIAS framework, in which privacy constraints are formally described to be used to detect violations according to Contextual Integrity. This PrivaCIAS framework provides social control to agents that handle the information, so that deceiving agents are excluded from the system.

Keywords: *Privacy, Contextual Integrity, Social Networks, Multiagent Systems, Open Decentralized Networks*

1 Introduction

Dans les systèmes multi-agents, l'échange d'information est une fonction capitale, car la communication permet l'interaction et la coopération entre agents. Mais dans certains de ces systèmes, l'information peut être sensible et ne doit pas être transmise à n'importe qui. Des violations de privacité [1] peuvent alors avoir lieu dans le système s'il n'y a pas de régulation ou de protection des communications. Les technologies de protection de la privacité (*Privacy Enhancing Technologies*/PET) proposent des solutions à ce problème. L'approche habituelle en informatique est d'exprimer de manière globale au système des limitations à la communication et au traitement d'information. Ceci peut être réalisé en mettant en place un contrôle au niveau du stockage de l'information (par exemple, les bases de données hippocratiques [2]), en attachant des préférences à l'information (par exemple, les politiques collantes [11]), ou attachées au service qui utilisera l'information (par exemple, les P3P [15]).

Cette approche classique n'est malheureusement pas adaptée aux système multi-agents décentralisés. En fait, les mécanismes de protection de la privacité reposent sur une architecture centralisée, souvent un logiciel ou un site web, qui doit alors être accepté comme tiers de confiance par tous les agents. Cette hypothèse est peu réaliste, car les agents ne souhaitent pas confier leurs données à un tiers qui pourrait être partial ou tenter d'exploiter leurs données à des fins commerciales. La décentralisation des réseaux sociaux a été identifiée [19] comme une étape essentielle pour la préservation de la privacité des utilisateurs. Toutefois, la décentralisation empêche l'utilisation des mécanismes classiques de protection de privacité. Il est nécessaire de proposer de nouveaux mécanismes de sécurité plus flexibles et capables de fonctionner de manière décentralisée et non intrusive.

Cet article propose un modèle pour protéger la

1. En français il n'existe pas d'équivalent au mot anglais *privacy* ou au mot espagnol *privacidad*. Nous utilisons donc dans cet article le néologisme privacité.

privacité dans les systèmes multi-agents décentralisées. Le travail suit une approche dite de « sécurité souple » dans laquelle chaque nœud du réseau doit réaliser deux tâches. Premièrement, il assiste l'utilisateur pour l'envoi de messages et l'avertit s'il s'apprête à commettre une violation de privacité. Deuxièmement, il vérifie que les messages reçus de la part des autres ne violent pas la privacité et, le cas échéant, détruit le message puis participe à l'exclusion sociale des contrevenants. Le modèle que nous proposons, qui se dénomme PrivaCIAS [2], repose sur la théorie de l'intégrité contextuelle [12] qui définit les violations de privacité par rapport au contexte de la transmission.

La section suivante présente le contexte de ce travail et explique pourquoi une approche sécuritaire souple est nécessaire. La section 3 détaille la théorie de l'intégrité contextuelle. Les éléments structurels du modèle sont donnés en section 4, la section 5 décrit la formalisation de l'intégrité contextuelle et les mécanismes de protection de la privacité. La section 6 présente l'implantation du modèle dans des agents.

2 Contexte

Le modèle défini dans cet article a pour but de protéger la privacité dans les systèmes ouverts en utilisant un contrôle social. Cette section présente la manière dont la privacité peut être protégée dans les systèmes décentralisés et décrit le contexte applicatif du modèle.

2.1 Sécurité et systèmes décentralisés

Dans les SMA, la sécurité peut être gérée de manière forte (contraintes a priori) ou souple (sanction a posteriori). La plupart des techniques utilisées pour la conservation de la privacité sont basées sur la sécurité forte nommée « contrôle d'accès ». Selon cette approche, une autorité authentifie l'utilisateur et lui donne accès aux ressources pour lesquelles un droit lui a préalablement été accordé. RBAC [9] (*Role Based Access Control*) par exemple, repose sur le contrôle d'accès : les utilisateurs se connectent et le système leur permet selon leur rôle, ou non, d'avoir accès à des informations. Il y a d'autres méthodes de contrôle d'accès plus spécifiques pour la privacité, comme PBAC [5] un contrôle d'accès dans lequel les utilisateurs doivent exprimer la raison (*Purpose Based Access Control*) pour laquelle ils ont besoin d'une information donnée. L'autorité valide ou rejette la requête.

2. Privacy as Contextual Integrity for Agent Systems

La protection de la privacité dans les systèmes ouverts et décentralisés est un champ de recherche relativement nouveau. Néanmoins, la plupart des travaux à propos de la sécurité dans les SMA sont basés sur des approches de type sécurité souple, le plus souvent en utilisant des modèles de confiance [18]. L'approche sécuritaire forte rencontre des difficultés quand de l'incertitude est présente dans les données et que le système est ouvert et décentralisé. Piolle [13] utilise dans sa thèse une solution intermédiaire afin de protéger des informations sensibles dans un SMA. Les informations sont confiées à des agents garants dont les actions sont contraintes par des systèmes de sécurité forte.

La confiance est utilisée dans les systèmes décentralisés afin de trouver des partenaires fiables pour réaliser des interactions. La confiance permet une certaine adaptabilité. Elle permet de pardonner mais aussi d'arrêter d'avoir confiance en un agent donné après quelques trahisons. Par extension, les modèles de confiance permettent la mise en place de la sécurité souple, car ils empêchent les agents d'entrer en relation avec des partenaires indignes de confiance. Ainsi, si des agents ne se comportent pas correctement, d'autres vont décider de ne plus les choisir comme partenaires et vont ruiner leur réputation. Ce type de sécurité souple tient lieu de contrôle social [6] menant à l'exclusion des agents indignes de confiance. À aucun moment les agents ne sont contraints physiquement à respecter les règles, mais s'ils se conduisent mal, alors les autres agents ruineront leur réputation. Peu d'approches utilisent la sécurité souple. Crépin [7] propose un système de gestion de la privacité utilisant la confiance. Dans ce système, les agents sont propriétaires de l'information transmise et sont donc aptes à définir les politiques relatives aux transmissions. Cette hypothèse n'est pas applicable si les agents transmettent de l'information qui ne les concerne pas, comme une photo d'autrui, ce que nous envisageons.

Toutes les solutions de sécurité forte requièrent une autorité de confiance acceptée par tous les utilisateurs. Cela n'est pas envisageable dans des systèmes ouverts et décentralisés avec un grand nombre d'agents. Premièrement, cela nécessiterait une autorité centrale. Deuxièmement « une autorité ne peut jamais être une assez bonne autorité pour tout le monde dans un système distribué de grande taille. Sa crédibilité diminue et ses recommandations augmentent en incertitude au fur et à mesure que la communauté s'agrandit » [1]. Troisièmement, si cette autorité existait, elle serait incapable d'empê-

cher physiquement les utilisateurs d'accéder à des ressources interdites, car dans les systèmes décentralisés, les agents sont incapables de se contrôler physiquement les uns les autres.

Rasmusson et Jansson ont proposé d'utiliser le contrôle social en tant que sécurité souple [14], ils affirment que la sécurité souple possède de sérieux avantages par rapport à la sécurité forte : « Une fois le système sécuritaire fort contourné, l'intrus aura accès à tout sans aucune restriction. L'approche sécuritaire souple accepte et s'attend même à des intrusion, l'idée est alors d'identifier et d'empêcher ces intrus de faire d'autres dégâts ».

Dans les systèmes ouverts et décentralisés que nous considérons, donner le contrôle aux agents nous semble être la meilleure solution car une autorité centrale ne pourrait pas être mise en place sans nuire à l'aspect décentralisé. Des techniques sécuritaires fortes seront utilisées afin de fournir des éléments sur lesquels le contrôle social pourra s'appuyer. Par exemple, des signatures électroniques permettront aux agents d'authentifier l'émetteur d'un message.

2.2 Contexte applicatif

Dans de multiples types de communautés virtuelles, les utilisateurs communiquent et partagent de l'information en utilisant des logiciels qui supportent la communauté. Ces applications soulèvent un problème difficile de préservation de la privacité. D'un côté, il est indispensable de permettre à chaque utilisateur d'échanger de l'information. D'un autre, chaque échange peut provoquer une violation de privacité.

Le but de ce travail est de spécifier un modèle dans lequel un agent peut assister son utilisateur pour préserver la privacité dans une communauté virtuelle. L'assistance est à la fois mise en place pour préserver la privacité de l'utilisateur assisté, en lui fournissant des avertissements lorsqu'une information est envoyée, et préserver la privacité des autres en détectant quand une violation a lieu et doit être punie. Cet article décrit ce travail en définissant des moyens pour détecter les violations de privacité et exclure les agents qui les effectuent.

La communauté virtuelle que nous considérons a les caractéristiques suivantes. C'est un système multi-agent décentralisé dans lequel les communications se font de pair à pair. Il est donc impossible de définir un contrôle central qui repose sur une perception globale et complète des communications comme le système est décentralisé. Les agents devront alors détecter les violations en se basant sur la théorie présentée dans la section suivante. En proposant une assistance locale aux utilisateurs, l'agent assistant peut être utilisé à la fois dans les systèmes centralisés et décentralisés et ne pose pas de restriction quant au passage à l'échelle. Le choix d'un système de communication pair à pair est assez général pour permettre de représenter d'autres types de communications. Par exemple, si nous voulons considérer un réseau social dans lequel l'information est échangée en publiant sur une page ou un « mur » lisible par les contacts de l'utilisateur, ce même échange peut être représenté par un ensemble de communications pair à pair avec chacun des contacts.

3 Intégrité contextuelle

Cette section présente la théorie de l'intégrité contextuelle de Nissenbaum, dont s'inspirent nos travaux. Afin d'avoir une description complète des fondations de la théorie, le lecteur devrait se reporter à l'article original [12].

Nissenbaum présente dans son article les 3 principes de privacité derrière les politiques de privacité et les lois aux États-Unis d'Amérique :
- limiter la surveillance des citoyens et l'utilisation de l'information les concernant par des agents du gouvernement ;
- restreindre l'accès aux informations sensibles, personnelles ou privées ;
- empêcher l'intrusion dans les espaces privés ou personnels.

Tout ce qui viole l'un de ces trois principes est une violation potentielle de privacité. L'analyse de ces principes révèle qu'en fait, ils ne sont pas absolus : par exemple, des informations sensibles et privées peuvent être accédées par un docteur pour guérir le patient, sujet de l'information. Aussi, la surveillance d'un citoyen peut être autorisée aux services de police pour sa propre protection, des perquisitions peuvent également avoir lieu dans le cadre légal de manière à entrer chez un suspect. Voici l'idée principale de l'intégrité contextuelle : savoir si une action est une violation de privacité dépend du contexte de cette action.

Un autre élément important de l'intégrité contextuelle est qu'il n'y a pas d'espace qui ne soit gouverné par des normes sur la circulation de l'information, pas de sphères pour lesquelles tout est permis. Tout ce que nous disons ou faisons se déroule dans un contexte avec ses conventions et ses attentes culturelles. L'idée d'une simple dichotomie privé/publique est de ce fait rejetée.

Notre travail se concentre uniquement sur le concept général de violation. Nissembaum prétend que savoir si une action donnée est une violation de privacité est fonction de :

1. la nature de la situation/contexte ;
2. la nature de l'information par rapport au contexte ;
3. le rôle de ceux qui reçoivent l'information ;
4. la relation entre le sujet de l'information et le destinataire ;
5. les termes de dissémination définis par le sujet.

Nissenbaum relève qu'une conséquence de sa définition est que les prescriptions relatives à la privacité, au lieu d'être prédéfinies et fixées, sont désormais dessinées par des facteurs locaux, peuvent varier selon la culture, la période historique... Des normes de transmission spécifiques à chaque contexte pourront alors être définies localement. C'est ce dernier concept que Barth *et al.* [3] formalisent dans leur article.

Nissenbaum donne, pour terminer, son avis sur les manières de contrôler l'intégrité contextuelle, parmi lesquelles se trouvent la loi et les politiques de privacité, mais aussi en dehors du cadre légal : les normes de décence, étiquette, sociabilité... C'est ce que nous décrivons plus tard dans cet article : un contrôle de l'intégrité contextuelle par des normes sociales.

4 Composants support du modèle

Cette section décrit la structure des messages et les différents composants structurels du modèle. Premièrement, la structure des messages échangés dans le système est décrite. Ensuite nous expliquons comment il est possible, pour un agent, de connaitre le rôle joué par un autre agent et les contextes associés. Ensuite, nous expliquons comment les agents peuvent spécifier des préférences lorsqu'ils transmettent de l'information.

4.1 Messages

Les agents échangent de l'**information** encapsulée dans un **message**. L'information est une donnée brute, nous ne posons pas de conditions quant au contenu ou la structuration de l'information. Un message est composé de deux parties : l'information et un ensemble de méta-informations décrites ci-après.

Les méta-informations suivantes sont adjointes à une information dans un message :

– marqueurs de contexte : faisant référence au contexte de l'information ;
– marqueurs de cible : faisant référence à la cible de l'information ;
– préférences : restreignant la diffusion de l'information ;
– chaine de transmission : gardant trace des agents ayant possédé le message et du contexte de chaque transmission.

Chacune des méta-informations doit être signée électroniquement par les agents qui l'ajoutent. Lorsqu'ils signent, les agents engagent leur responsabilité. La signature vaut certification selon l'exemple suivant : un agent qui signe le marqueur de contexte « medical » certifie que l'information appartient au contexte médical. Les méta-informations sont donc signées par signature électronique (par exemple, RSA [17]) et contiennent une référence unique à l'information qu'elles qualifient, sous forme de code de hachage (par exemple, Message Digest [16]). La chaine de transmission permet de garder une trace du parcours du message au sein du système. Tout agent doit ajouter un maillon à la chaine avec son identifiant, celui du destinataire et le contexte déclaré pour cette transmission.

4.2 Rôles et contextes

Afin d'être capables de définir un système de protection de la privacité basé sur la théorie de l'intégrité contextuelle, nous devons introduire les concepts de **contexte** et **rôle**. Le contexte décrit la situation dans laquelle l'information est échangée, par exemple : le travail de Paul, la famille de Jean, la santé de Marc. Les rôles sont définis dans un contexte donné et attachés à des utilisateurs, par exemple : le chef de Paul, le père de Jean, le médecin de Marc. Il peut y avoir plusieurs rôles dans un même contexte. Dans ce papier, nous considérons que les rôles des agents et les contextes correspondants sont fournis par des annuaires organisationnels. Pour réaliser cela, il est possible d'utiliser une infrastructure organisationnelle multi-agent décentralisée [10].

Ces concepts sont utiles afin de pouvoir exprimer des règles précises pour le maintien de l'intégrité contextuelle. Nous les utilisons dans les sections suivantes pour permettre aux agents de raisonner sur des violations de privacité.

4.3 Primitives

Afin de permettre aux agents de manipuler les concepts décrits ci-avant, comme les méta-informations, nous définissons ici un ensemble de primitives logiques.

1. Primitives de méta-information :

- `information(+M,?I)`. I est l'information contenue dans le message M [3] ;
- `contexttag(?C,+A,+M)`. C est un marqueur de contexte définit par A, contenu dans le message M ;
- `targettag(?T,+A,+M)`. T est un marqueur de cible défini par A, contenu dans le message M ;
- `policy(?P,+A,+I)`. P est une préférence définie par A pour l'information I.

2. Primitives sur les rôles de transmission :
   - `receiver(?X,+M)`. Le destinataire de M est l'agent X ;
   - `propagator(?X,+M)`. L'émetteur de M est l'agent X.
3. Primitives sur les croyances :
   - `target(?X,+I)`. L'agent croit que X est la cible de I ;
   - `policyvalid(+P,+M)`. L'agent croit que la préférence P est valide pour M ;
   - `context(?C,+I)`. L'agent croit que C est le contexte correspondant à I ;
   - `rolecontext(+A,?R,?C)`. L'agent croit que A joue le rôle R dans le contexte C ;
   - `link(+X,+Y)`. L'agent croit que X est capable de communiquer avec Y.

   Á partir de ces primitives, il est désormais possible d'exprimer des normes et des préférences.

4.4 Préférences de privacité

Le message peut contenir des préférences de privacité, spécifiées par une des cibles du message afin de restreindre sa diffusion. Ces préférences sont définies pour une information donnée par un agent donné (et sont signées par cet agent).

Bien que des langages existent pour exprimer des politiques (comme Protune [8], qui a l'avantage de permettre à un humain d'exprimer des politiques en utilisant un « langage naturel contrôlé »), la plupart de ces approches sont basées sur du contrôle d'accès, ce que nous tenons à éviter pour les raisons précédemment citées. Dans notre modèle, nous souhaitons exprimer les politiques en nous basant sur des composants sociaux (rôles, confiance) et le contexte de transmission.

3. Note : certaines primitives sont spécifiques à un message M, c'est-à-dire à une transmission donnée. D'autres à une information I, donc applicables à tous les messages contenant I.

Les préférences sont exprimées en utilisant un langage proche de Jason [4] composé des primitives précédemment définies.

Une politique (ou préférence de privacité) est composée de multiples **déclarations**. Une **déclaration** est composée par un ensemble de **primitives** et par un type de déclaration qui peut être :
- `forbidden(I):-`
  Déclare une condition qui ne doit pas se produire pour la transmission de l'information I.
- `mandatory(I):-`
  Déclare une situation qui doit obligatoirement se produire lors de la transmission de l'information I.

Une préférence donnée est satisfaite si aucune déclaration *forbidden* n'est vraie (si au moins une est vraie alors la préférence est insatisfaite) et une déclaration *mandatory* est vraie. Comme une déclaration est composée d'une conjonction de primitives, la disjonction est exprimée par la définition de multiples déclarations du même type. Autrement dit, il ne faut qu'une déclaration de type *mandatory* pour satisfaire la préférence, et une seule *forbidden* pour la rendre insatisfaite. Les préférences sont ajoutées par des agents qui sont ciblés par l'information (voir `target(?X,+I)`) afin d'exprimer leurs préférences quant à la transmission des informations.

5 PrivaCIAS

Le modèle PrivaCIAS (*Privacy as Contextual Integrity for Agent Systems*) repose sur deux ensembles de règles afin de protéger la privacité dans les systèmes ouverts et décentralisés :
- les lois d'adéquation (A-laws), qui expriment les règles de privacité en se basant sur la théorie de l'intégrité contextuelle de Nissenbaum.
- des normes de contrôle social. Comme nous avons choisi de nous reposer sur le contrôle social pour protéger la privacité, des normes sont définies pour fournir un code de conduite aux agents de manière à ce qu'ils veillent mutuellement au respect des A-laws. Ces normes sont appelées *Privacy Enforcing Norms* (PENs) car elles indiquent aux agents ce qu'ils doivent faire afin de prévenir des violations de privacité.

5.1 Lois d'adéquation (A-laws)

La théorie de l'intégrité contextuelle définit la violation comme étant fonction d'un ensemble de paramètres. Dans cette sous-section, nous proposons des règles permettant de dire si une

transmission est une violation ou non en se basant sur la définition précédente.

Nous utilisons le terme « adéquation » (*appropriateness*) pour définir l'ensemble de lois qui qualifient une transmission comme étant inadéquate lorsqu'une des lois est violée. Le terme est inspiré par l'article de Nissenbaum [12]. Elle décrit l'adéquation comme les normes qui dictent quelle information [...] il est adéquat, ou approprié, de relever dans un contexte donné.

Dans la définition suivante nous utilisons le terme « cible » au lieu du terme « sujet » utilisé par Nissenbaum *cf* section 3. Un sujet est directement désigné par l'information tandis qu'une cible peut ne pas être citée dans l'information. Par exemple, si l'information est une photo de M. Dupont avec une femme qui n'est pas Mme Dupont, les sujets sont M. Dupont et la femme, mais les cibles, ceux pour qui l'information peut être préjudiciable, sont M. Dupont, la femme, mais aussi Mme Dupont. L'utilisation du terme « cible » nous semble être plus adéquat.

En s'inspirant de la définition de la violation donnée en section 3, nous définissons une transmission comme étant adéquate si toutes les conditions suivantes sont valides :

1. Le contexte de la transmission correspond à la nature de la transmission ;
2. le destinataire a un rôle dans ce contexte ;
3. les préférences (politiques) de la cible sont respectées.

La transmission est donc inadéquate si une des conditions est invalide.

Ci-après, nous illustrons ces 3 lois d'adéquation (*Appropriateness laws ou A-laws*) avec des exemples :

1. De façon générale, le contexte d'une transmission peut être vu comme la situation où et quand la transmission a lieu. Dans le modèle, pour des raisons de simplification, le contexte d'une transmission est déclaré par le propagateur du message. La nature de l'information est inférée par le destinataire. Un contexte correspond à l'information s'il reflète la nature de l'information, par exemple : des informations médicales correspondent au contexte médical.
2. Les agents participant à la transaction doivent avoir un rôle associé à ce contexte [3]. Par exemple, un docteur a un rôle dans le contexte médical.
3. Si l'une des cibles de l'information spécifie des préférences quant à la propagation de l'information, celles-ci doivent être respectées.

Nous proposons ici une implémentation de ces lois en règles Jason [4]. En Jason, ces règles devront être soutenues par des plans qui veilleront à effectuer les actions nécessaires à leur satisfaction, comme par exemple, signer un message avant de l'envoyer. Dans le modèle, une information est générale tandis qu'un message est spécifique à une transmission donnée. Comme un message est spécifique à une transmission donnée, il est équivalent de dire « un message M est adéquat » et « une transmission M est adéquate ».

```
1 fitcontext(?C,+M):-
    information(M,I) &
3   propagator(M,P) &
    contexttag(M,C,P) &
5   context(I,C).
```

La règle `fitcontext(C,M)` déclare que si I est l'information contenue dans le message M, et l'agent P est le propagateur de M, et P déclare C comme contexte de la transmission M, alors la formule est validée si l'agent qui la vérifie croit que C est un contexte compatible avec l'information I.

```
1 fitrole(+C,+M):-
    receiver(Rc,M) &
3   rolecontext(Rc,R,C).
```

`fitrole(C,M)` est vrai si Rc est le destinataire de M, et l'agent vérifiant la formule croit que Rc joue le rôle R, et qu'il croit aussi que ce rôle R est relatif au contexte C.

```
1 fitpolicy(+M):-
    information(M,I) &
3   target(T,I) &
    policy(F,T,I) &
5   policyvalid(F,M).

7 fitpolicy(+M):-
    information(M,I) &
9   not (
      target(T,I) &
11    policy(_,T,_) &
      ).
```

Les troisième et quatrième règles déclarent que `fitpolicy(M)` est vrai si I est l'information contenue dans M, et l'agent vérifiant la formule croit que T est la cible de I, et il y a des préférences définies par T pour l'information I, et ces préférences sont valides au regard de la transmission M. La quatrième règle prend en compte le cas où aucune préférence n'est définie.

```
appropriate(+M):-
2   fitcontext(C,M) &
    fitrole(C,M) &
4   fitpolicy(M).
```

Un message M est adéquat si les trois prédicats fitcontext, fitrole, fitpolicy sont valides.

5.2 Normes de maintien de la privacité (PEN)

Cette section décrit les *Privacy Enforcing Norms* (PEN), normes pour le maintien de la privacité, qui sont définies de manière à préserver la privacité en poussant les agents à mettre en place un contrôle social.

La première norme (PEN1) que nous proposons a pour but d'empêcher les A-laws d'être violées, composant essentiel dans notre définition de la privacité.

Pour que chaque agent prenne ses responsabilités lorsqu'il effectue une transmission, la norme PEN2 impose la signature des messages. La signature permet de s'assurer de la provenance du message, ce qui est une information cruciale lorsqu'il est nécessaire d'évaluer le comportement des autres.

Trois plans d'action permettent de protéger la privacité :
– prévenir les violations : ne pas envoyer le message à des agents qui sont susceptibles d'effectuer des violations ;
– stopper les violations : supprimer lors de la réception les messages causant des violations ou provenant de violateurs ;
– punir les violations : punir socialement les violations en ruinant la réputation des contrevenants.

Les PENs préviennent les violations en indiquant aux agents qu'ils ne doivent transmettre qu'aux agents de confiance (PEN3). De plus, cette norme implémente le contrôle social car les agents avec un mauvais comportement sont exclus du système comme les autres refusent de communiquer avec eux. Le mécanisme de confiance est décrit dans la section 6.3, il permet d'évaluer le respect des PEN chez les agents afin de repérer ceux qui sont susceptibles de les violer.

Un moyen de stopper les violations est de supprimer les messages reçus de la part d'un violateur connu (PEN4), les nouvelles violations sont alors empêchées comme la retransmission de l'information inadéquatement reçue n'aura pas lieu. Cette norme implémente également le contrôle social comme les partenaires effectuant des violations sont ignorés et donc, exclus du système.

Enfin, les violations sont punies par le biais d'une réduction de la réputation des violateurs, en envoyant des commérages à d'autres pour révéler leurs agissements (PEN5).

Pour résumer, les PENs sont les suivantes :

1. Respecter les A-laws,
2. Signer la chaine de transmission avant l'envoi,
3. Ne pas envoyer d'information aux agents indignes de confiance,
4. Supprimer l'information en provenance d'agents indignes de confiance,
5. Ruiner la réputation des agents qui violent les normes précédentes.

Ainsi, les normes ne sont pas mises en place par le système mais par les agents eux-mêmes. Les agents refusant d'appliquer les normes sont punis par les autres agents. La punition consiste à ruiner la réputation du contrevenant en avertissant les autres agents de la violation constatée.

Voici la définition des PENs en Jason :

```
fitPENs(+M) :-
  respectPEN1(M) &
  respectPEN2(M) &
  respectPEN3(M) &
  respectPEN4(M).
```

Le message M satisfait les PENs si toutes les PENs sont respectées. Si une PEN n'est pas respectée, alors l'agent doit appliquer les PEN 4 et 5 (supprimer le message et punir) :

```
fitPENs(+M) :-
  enforcePEN4(M),
  enforcePEN5(M).
```

Les règles définissent dans quel cas les PENs sont satisfaites. Comme pour les lois d'adéquation, elles devront être soutenues par des plans qui s'assureront de les satisfaire.

```
respectPEN1(+M) :-
  appropriate(M).
```

La PEN1 est valide si les A-laws sont respectées.

```
respectPEN2(+M) :-
  lastLink(M,J,K) &
  propagator(M,J) &
  receiver(M,K).
```

La PEN2 est respectée si le propagateur J du message M a signé le message avant de l'envoyer. Le dernier chainon de la chaine de transmission doit être « M est envoyé de J vers K » où J est le propagateur et K le destinataire.

```
respectPEN3(+M) :-
  receiver(M,K) &
  trust(K).
```

La PEN3 est respectée si l'agent a confiance en K, destinataire du message M. Il est inutile de vérifier cette PEN en réception comme l'agent qui vérifie aura toujours confiance en lui-même.

```
respectPEN4(+M) :-
    propagator(M, J) &
    trust(J).
```

La PEN4 est respectée si J est le propagateur du message M et l'agent vérifiant la formule fait confiance à J. Pour la même raison que la PEN précédente, il n'est pas nécessaire de vérifier cette PEN à l'envoi.

La PEN5 n'est pas testée puisqu'elle consiste simplement en une contremesure à appliquer en cas de violation, comme la seconde partie de la PEN4 qui supprime un message provenant d'un agent indigne de confiance :

```
enforcePEN4(+M):-
    deleted(M).
```

```
enforcePEN5(+M):-
    propagator(M, J) &
    trust(J) &
    punish(J).
```

Si l'agent est indigne de confiance il n'est pas nécessaire de le punir comme tous les messages provenant de lui sont ignorés. Mais si l'auteur de la violation était considéré comme partenaire de confiance, alors il faut ruiner sa réputation et arrêter de lui faire confiance.

6 Privacy Enforcing Agents

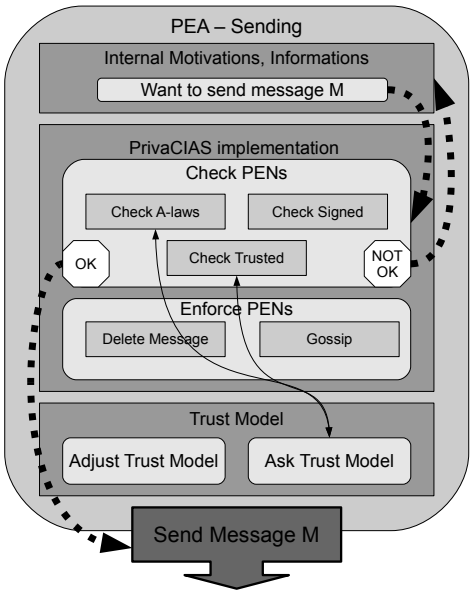

FIGURE 1 – Processus au sein d'un PEA lors de l'envoi d'un message

Le contrôle de la privacité est totalement décentralisé dans le modèle présenté, les agents doivent vérifier les PEN en émission et en réception de messages. Les agents qui respectent les PENs sont appellés PEA (*Privacy Enforcing Agents*).

La figure 1 présente l'architecture d'un PEA et le processus général pour la protection des PENs en émission. Les flèches pointillées épaisses représentent les processus, les flèches fines représentent les interdépendances entre modules. En résumé, lorsqu'un agent veut envoyer un message, il doit vérifier les PENs, certaines de celles-ci s'appuient sur le modèle de confiance. Si les PENs sont respectées, l'agent peut envoyer le message, autrement, il doit renoncer à l'envoyer.

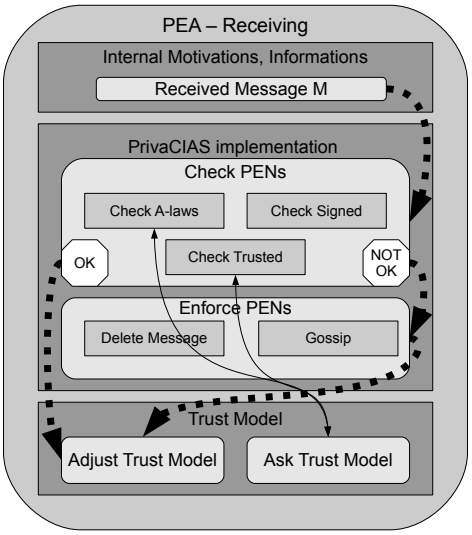

FIGURE 2 – Processus au sein d'un PEA lors de la réception d'un message

L'architecture et le processus général en réception sont présentés sur la figure 2. Quand le PEA reçoit un message, il doit vérifier les PENs, certaines des PENs s'appuient sur le modèle de confiance. Si les PENs sont respectées, alors l'agent augmente la confiance qu'il a en ce partenaire, autrement il doit supprimer le message et ruiner la réputation de celui-ci.

6.1 Réception de message

Quand l'agent reçoit un message, il doit vérifier si la transmission qui vient d'avoir lieu est une violation des PEN ou non. Premièrement, l'agent doit vérifier les A-laws afin de voir si la transmission est adéquate (PEN1). Pour ce faire, il doit inférer diverses croyances, par exemple : qui est la cible du message ? quelle est la nature de l'information ? Cela est possible en uti-

lisant les marqueurs de contexte et de cible, les croyances de l'agent, ou encore en analysant l'information directement. Comme les marqueurs de contexte sont signés, il est possible de se reposer sur ceux déposés par des agents de confiance pour déterminer le contexte de l'information. L'agent doit aussi vérifier que le message est signé (PEN2) et que l'émetteur est de confiance (PEN4).

Si l'agent détecte une violation, il marque le message pour suppression et punit l'agent en ruinant sa réputation (PEN5).

Enfin, l'agent ajuste son niveau de confiance envers le propagateur du message selon s'il a effectué une violation ou non.

6.2 Envoi de messages

Quand un agent souhaite envoyer un message à un autre, il doit tout d'abord attacher toutes les méta-informations possibles :

- Si l'agent peut identifier la cible de l'information, il ajoute un marqueur de cible et il le signe afin de certifier qu'il est l'auteur de ce marqueur.
- Si l'agent est capable de déterminer le contexte de l'information, de même il ajoute un marqueur de contexte.
- Si l'agent est lui-même la cible, il peut spécifier des restrictions quant à la diffusion de l'information, il peut donc ajouter des préférences, qu'il signe également.
- L'agent signe également la chaine de transmission (PEN2) en déclarant à qui il envoie le message et dans quel contexte.

L'agent doit ensuite vérifier les PEN par rapport au destinataire :

- L'agent violera-t-il les A-laws (PEN1) en envoyant le message au destinataire ? Un PEA n'enverra jamais un message si celui-ci causera une violation.
- L'agent fait-il confiance au destinataire ? (PEN3) S'il n'est pas digne de confiance, c'est qu'il a effectué des violations dans le passé. Comme l'agent veut protéger l'information dont il est porteur, il ne l'envoie qu'aux agents en lesquels il a confiance, et il ignore les agents indignes de confiance afin de les exclure du système.

PEN4 et PEN5 n'ont pas besoin d'être vérifiées à l'émission comme le propagateur est l'agent qui vérifie les PEN, il a confiance en lui-même (PEN4). Si une violation est détectée, l'agent ne doit alors pas envoyer le message.

6.3 Confiance et punition

Dans le modèle proposé, les agents sont incertains du comportement des autres. Ainsi, ils se reposent sur la confiance afin de décider s'ils doivent interagir ou non avec eux. De plus, la confiance permet de mettre en place un contrôle social car les agents rationnels ne veulent pas interagir avec des agents indignes de confiance. Ces derniers sont alors exclus socialement.

L'article suppose l'existence au sein de l'agent d'un modèle de confiance capable de déterminer s'il faut faire confiance ou non à un agent donné sur la base d'agrégation des expériences passées. Un de ceux présentés dans l'article de Sabater [18] peut être choisi.

Afin de partager leurs expériences, après une transmission, les agents envoient des évaluations de la transmission à leur contacts. La PEN5 demande aux agents d'envoyer un message pour punir les contrevenants, lorsqu'une violation est détectée. Le but de ce message est justement de partager l'expérience de l'agent afin d'informer les autres de la violation. Il est également possible d'envoyer un message afin de recommander un partenaire. Le message de recommandation/punition contient :

- La méta-information du message original,
- Une évaluation de la violation parmi : très mauvaise, mauvaise, convenable, bonne, très bonne.

L'envoi des méta-informations du message original est utile afin de présenter des preuves aux autres agents concernant la violation. L'avantage de n'envoyer que les méta-informations, c'est que l'agent n'a pas besoin de transmettre l'information (ce qui en soit pourrait aussi être une violation). Les agents ont le choix d'accepter telle quelle l'évaluation ou de tenter de la vérifier en utilisant les méta-informations.

En résumé, les agents se transmettent de l'information, en vérifiant à l'envoi et à la réception si une violation a eu lieu. Quand une violation est détectée, les agents envoient des messages de punition à leurs contacts afin de partager leurs expériences. Au bout d'un certain temps, les contrevenants sont exclus du système car les agents respectant les PEN ne leur transmettent plus de messages comme ils sont indignes de confiance.

7 Conclusion

La protection de la privacité est un problème intéressant et de nombreuses recherches s'en préoccupent actuellement. Néanmoins, très peu

d'approches se font sous l'angle des réseaux ouverts et décentralisés. Notre approche repose sur la théorie de l'intégrité contextuelle par Nissenbaum afin de proposer un modèle pour la protection de la privacité dans les systèmes ouverts et décentralisés.

L'idée principale du modèle est de détecter les violations du point de vue de l'agent, car il n'y a pas d'autorité centralisée pour contrôler les transmissions. L'exclusion des contrevenants est ensuite réalisée par un contrôle social.

Notre modèle est composé de 2 jeux de règles principaux :
– Les lois d'adéquation (A-laws), qui décrivent quelles transmissions sont adéquates au regard de la théorie de l'intégrité contextuelle.
– Les normes de maintien de la privacité (PEN), qui fournissent un code de conduite aux agents de manière à ce qu'ils vérifient que les transmissions soient adéquates (A-laws), et qu'ils mettent en place un contrôle social dans le système en envoyant à leurs contacts des évaluations de ces violations.

Références

[1] A. Abdul-Rahman and S. Hailes. A distributed trust model. In *Proceedings of the 1997 workshop on New security paradigms*, pages 48–60. ACM, 1998.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 143–154. VLDB Endowment, 2002.

[3] A. Barth, A. Datta, J. Mitchell, and H. Nissenbaum. Privacy and Contextual Integrity : Framework and Applications. *2006 IEEE Symposium on Security and Privacy (S&P'06)*, pages 184–198.

[4] R. Bordini, J. Hubner, and M. Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*, volume 15. Wiley-Interscience, 2007.

[5] J. Byun, E. Bertino, and N. Li. Purpose based access control of complex data for privacy protection. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 102–110. ACM, 2005.

[6] C. Castelfranchi. Engineering social order. In *Engineering societies in the agents world*, pages 1–18. Springer, 2000.

[7] L. Crépin. *Les Systèmes Multi-Agents Hippocratiques*. PhD thesis, 2009.

[8] J. De Coi, P. Kärger, D. Olmedilla, and S. Zerr. Using natural language policies for privacy control in social platforms. In *Workshop on Trust and Privacy on the Social and Semantic Web (SPOT)*. Citeseer, 2009.

[9] D. Ferraiolo, J. Cugini, and D. Kuhn. Role-based access control (RBAC) : Features and motivations. In *Proceedings of 11th Annual Computer Security Application Conference*, pages 241–48, 1995.

[10] J. F. Hübner, O. Boissier, R. Kitio, and A. Ricci. Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems*, 20(3) :369–400, 2009.

[11] M. C. Mont, S. Pearson, and P. Bramhall. Towards accountable management of identity and privacy : Sticky policies and enforceable tracing services. In *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*, pages 377–382, 2003.

[12] H. Nissenbaum. Privacy as Contextual Integrity. *Washington Law Review*, pages 101–139, 2004.

[13] G. Piolle. *Agents utilisateurs pour la protection des données personnelles : modélisation logique et outils informatiques*. PhD thesis, 2009.

[14] L. Rasmusson and S. Jansson. Simulated social control for secure Internet commerce. In *Proceedings of the 1996 workshop on New security paradigms*, pages 18–25. ACM, 1996.

[15] J. Reagle and L. F. Cranor. The platform for privacy preferences. *ACM*, 42(2) :48–55, 1999.

[16] R. L. Rivest. The MD4 message digest algorithm. *Advances in Cryptology-CRYPT0'90*, pages 303–311, 1991.

[17] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public- Key Cryptosystems. *Communications*, 21(2), 1978.

[18] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1) :33–60, 2005.

[19] C. Yeung, I. Liccardi, K. Lu, O. Seneviratne, and T. Berners-Lee. Decentralization : The future of online social networking. In *W3C Workshop on the Future of Social Networking Position Papers*, 2009.