Groupe de travail Réseau
Request for Comments : 3585
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

J. Jason Intel Corporation
L. Rafalow, IBM
E. Vyncke, Cisco Systems

août 2003

Modèle d'informations de politique de configuration IPsec

Statut du présent mémoire
Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent document présente un modèle d'informations orienté objet de la politique de sécurité IP (IPsec, *IP Security*) conçu pour faciliter l'accord sur le contenu et la sémantique de la politique IPsec, et permet de déduire des représentations spécifiques des tâches de la politique IPsec comme un schéma de mémorisation, des représentations de distribution, et des langages de spécification de politique utilisés pour configurer les points d'extrémité à capacité IPsec. Le modèle d'informations décrit dans ce document modèlise les paramètres de configuration définis par IPSec. Le modèle d'informations couvre aussi les paramètres trouvés par le protocole d'échange de clés Internet (IKE, *Internet Key Exchange*). D'autres protocoles d'échange de clés pourraient facilement être ajoutés au modèle d'informations par une simple extension. D'autres extensions peuvent aussi être facilement ajoutées du fait de la nature orientée objet du modèle.

Le présent modèle d'informations se fonde sur les classes de cœur de politique telles que définies dans le modèle d'informations de cœur de politique (PCIM, *Policy Core Information Model*) et dans les extensions du modèle d'informations de cœur de politique (PCIMe, *Policy Core Information Model Extensions*).

Table des Matières

1. Introduction

La politique de sécurité IP (IPsec, *IP security*) peut assumer diverses formes lorsque elle voyage du stockage à la distribution et aux points de décision. À chaque étape, elle doit être représentée d'une façon convenable pour la tâche courante. Par exemple, la politique pourrait, sans s'y limiter, exister comme :

- o un schéma de protocole léger d'accès à un répertoire (LDAP) [RFC2251] dans un répertoire ;
- o une représentation dans le réseau sur un protocole de transport comme le service commun de politique ouverte (COPS,

Common Open Policy Service) [RFC2748], [RFC3084] ;
- o un langage de spécification de politique fondé sur le texte convenable pour l'édition par un administrateur ;
- o un document en langage de balisage extensible (XML, *Extensible Markup Language*).

Chacune de ces représentations spécifique d'une tâche devrait être déduite d'une représentation canonique qui spécifie précisément le contenu et la sémantique de la politique IPsec. Le présent document retient ce concept et introduit une représentation canonique indépendante de la tâche pour les politiques IPsec.

Le présent document se concentre principalement sur les protocoles existants [RFC3173], [RFC2406], [RFC2402], [RFC2407], [RFC2409]. Le modèle peut facilement être étendu si nécessaire grâce à sa nature orientée objet.

Le présent document est organisé comme suit :
- o La Section 2 fournit une brève introduction aux conventions de notation graphique du langage de modélisation unifié (UML, *Unified Modeling Language*) utilisées dans le présent document.
- o La Section 3 décrit la hiérarchie d'héritage qui décrit où les classes de politique IPsec se transposent en hiérarchie de classe de politique déjà définie par le modèle d'informations de cœur de politique (PCIM, *Policy Core Information Model*) et extensions au modèle d'informations de cœur de politique (PCIMe, *Policy Core Information Model Extensions*).
- o Les sections 4 à 8 décrivent les classes qui constituent le modèle de politique IPsec.
- o La Section 9 présente les exigences de mise en œuvre des classes du modèle (c'est-à-dire, le statut DOIT/PEUT/DEVRAIT).

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans la [RFC2119].

2. Conventions UML

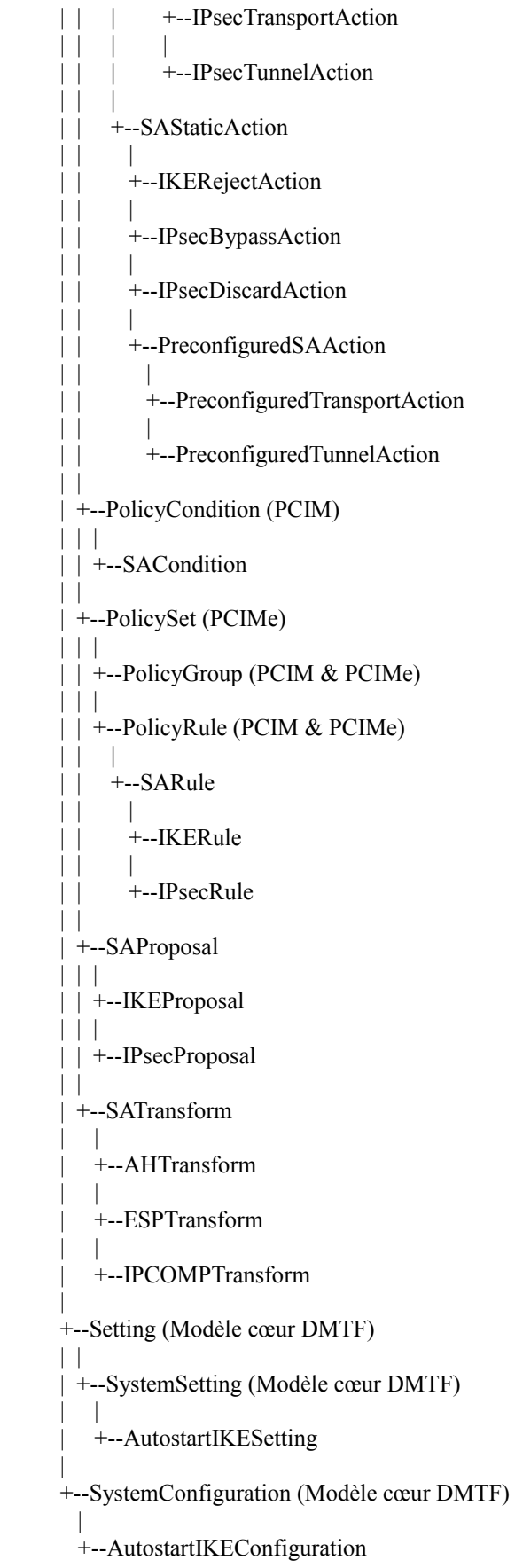
Pour le présent document, un diagramme statique UML a été choisi comme représentation canonique pour le modèle de politique IPsec, parce que UML fournit un moyen graphique, indépendant de la tâche, pour modéliser les systèmes. Il n'est pas dans l'intention du présent document de faire un traité sur la notation graphique utilisée dans UML. Cependant, sachant que les diagrammes de classe statique UML utilisent les traits ASCII, une description des conventions de notation utilisées dans ce document est donnée ici :
- o Les rectangles représentent les classes, avec le nom de la classe entre crochets ([]) représentant une classe abstraite.
- o Une ligne terminée par une flèche (<, >, ^, v) note l'héritage. La flèche pointe toujours sur la classe parente. L'héritage peut aussi être appelé généralisation ou spécialisation (selon le point de référence). Une classe de base est une généralisation d'une classe dérivée, et une classe dérivée est une spécialisation d'une classe de base.
- o Les associations sont utilisées pour modéliser une relation entre deux classes. Les classes qui partagent une association sont connectées par une ligne. Une sorte particulière d'association est aussi utilisée : l'agrégation. Une agrégation modèle une relation tout-partie entre deux classes. Les associations, et donc les agrégations, sont aussi modélisées comme des classes.
- o Une ligne qui commence par un "o" note une agrégation. L'agrégation note l'inclusion dans laquelle la classe contenue et la classe contenante ont des durées de vie indépendantes.
- o À chaque extrémité d'une ligne qui représente une association apparaît une cardinalité (c'est-à-dire que chaque association a deux cardinalités). La cardinalité indique les contraintes sur le nombre d'instances d'objet dans un ensemble de relations. La cardinalité sur une certaine extrémité d'une association indique le nombre d'instances différentes d'objet de cette classe qui peuvent être associées à une seule instance d'objet de la classe à l'autre extrémité de l'association. La cardinalité peut être :
  - une gamme de la forme "limite inférieure..limite supérieure" indiquant les nombres minimum et maximum d'objets ;
  - un nombre qui indique le nombre exact d'objets ;
  - un astérisque qui indique un nombre quelconque d'objets, incluant zéro. Un astérisque est l'abrégé de 0..n.
  - la lettre n indique de 1 à plusieurs. La lettre n est un abrégé pour de 1..n.
- o Une classe qui a une association peut avoir un "w" à côté de la ligne qui représente l'association. C'est appelé une association faible qui est discutée dans la [RFC3060].

On notera que le diagramme de classe statique UML présenté est une vue conceptuelle de la politique IPsec conçue pour faciliter la compréhension. Il n'est pas nécessairement traduisible classe pour classe dans une autre représentation. Par exemple, une mise en œuvre de LDAP peut écraser la représentation en moins de classes (les références suivantes étant inopérantes).

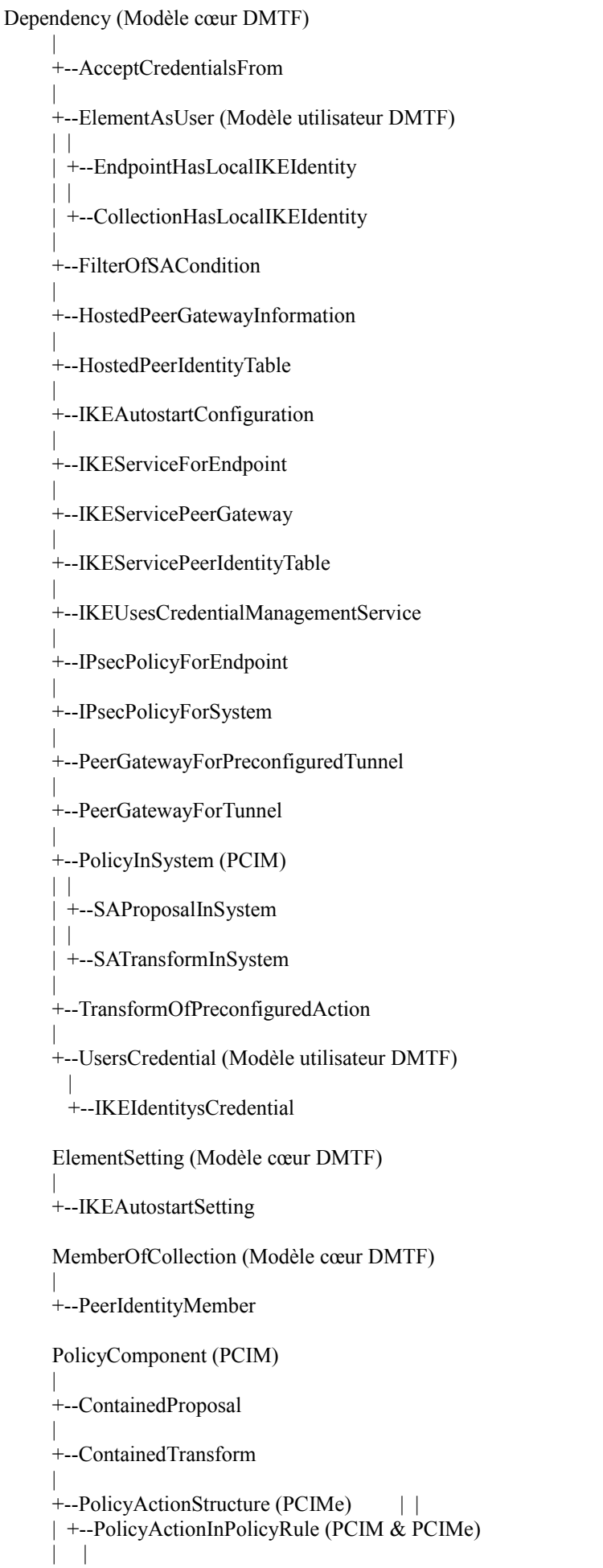
3. Hiérarchie d'héritage du modèle de politique IPsec

Comme PCIM et PCIMe, le modèle de politique de configuration IPsec dérive des classes définies dans le modèle d'informations communes (CIM, *Common Information Model*) de DMTF [DMTF] et les utilise. L'arborescence qui suit représente la hiérarchie d'héritage des classes du modèle de politique IPsec et comment elles se placent dans PCIM, PCIMe et autres modèles DMTF (voir dans les Appendices les descriptions des classes qui ne sont pas introduites au titre du modèle IPsec). Les classes CIM qui ne sont pas utilisées comme superclasses pour déduire de nouvelles classes, mais ne sont utilisées que comme références, ne sont pas incluses dans cette hiérarchie d'héritage, mais peuvent être trouvées dans le document DTMF approprié : Modèle cœur [CIMCORE], Modèle d'utilisateur [CIMUSER] ou Modèle réseau [CIMNETWORK].

```
ManagedElement (Modèle cœur DMTF)
    |
    +--Collection (Modèle cœur DMTF)
    | |
    | +--PeerIdentityTable
    |
    +--ManagedSystemElement (Modèle cœur DMTF)
    | |
    | +--LogicalElement (Modèle cœur DMTF)
    |   |
    |   +--FilterEntryBase (Modèle réseau DMTF)
    |   | |
    |   | +--CredentialFilterEntry
    |   | |
    |   | +--IPHeadersFilter (PCIMe)
    |   | |
    |   | +--IPSOFilterEntry
    |   | |
    |   | +--PeerIDPayloadFilterEntry
    |   |
    |   +--PeerGateway
    |   |
    |   +--PeerIdentityEntry
    |   |
    |   +--Service (Modèle cœur DMTF)
    |     |
    |     +--IKEService
    |
    +--OrganizationalEntity (Modèle utilisateur DMTF)
    | |
    | +--UserEntity (Modèle utilisateur DMTF)
    |   |
    |   +--UsersAccess (Modèle utilisateur DMTF)
    |     |
    |     +--IKEIdentity
    |
    +--Policy (PCIM)
    | |
    | +--PolicyAction (PCIM)
    | | |
    | | +--CompoundPolicyAction (PCIMe)
    | | |
    | | +--SAAction
    | |   |
    | |   +--SANegotiationAction
    | |   | |
    | |   | +--IKENegotiationAction
    | |   |   |
    | |   |   +--IKEAction
    | |   |   |
    | |   |   +--IPsecAction
    | |   |     |
```



L'arborescence suivante représente la hiérarchie d'héritage des classes d'association du modèle de politique IPsec et la façon dont elles se placent dans PCIM et les autres modèles DMTF (voir dans les Appendices la description des classes d'association qui ne sont pas introduites au titre du modèle IPsec).

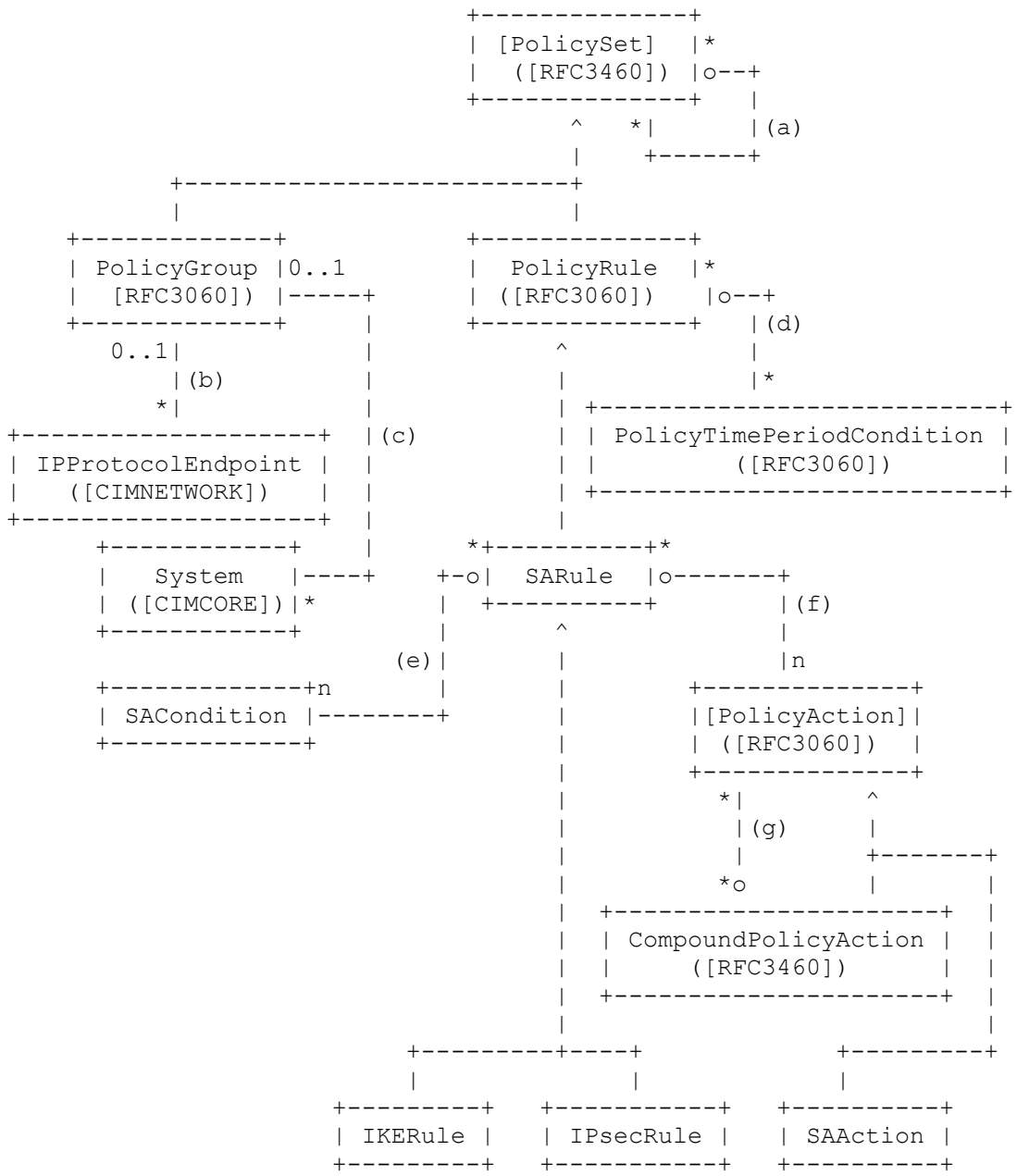

Dependency (Modèle cœur DMTF)
+--AcceptCredentialsFrom
+--ElementAsUser (Modèle utilisateur DMTF)
+--EndpointHasLocalIKEIdentity
+--CollectionHasLocalIKEIdentity
+--FilterOfSACondition
+--HostedPeerGatewayInformation
+--HostedPeerIdentityTable
+--IKEAutostartConfiguration
+--IKEServiceForEndpoint
+--IKEServicePeerGateway
+--IKEServicePeerIdentityTable
+--IKEUsesCredentialManagementService
+--IPsecPolicyForEndpoint
+--IPsecPolicyForSystem
+--PeerGatewayForPreconfiguredTunnel
+--PeerGatewayForTunnel
+--PolicyInSystem (PCIM)
+--SAProposalInSystem
+--SATransformInSystem
+--TransformOfPreconfiguredAction
+--UsersCredential (Modèle utilisateur DMTF)
+--IKEIdentitysCredential
ElementSetting (Modèle cœur DMTF)
+--IKEAutostartSetting
MemberOfCollection (Modèle cœur DMTF)
+--PeerIdentityMember
PolicyComponent (PCIM)
+--ContainedProposal
+--ContainedTransform
+--PolicyActionStructure (PCIMe)
+--PolicyActionInPolicyRule (PCIM & PCIMe)

```
|   +--PolicyActionInSARule
|
+--PolicyConditionStructure (PCIMe)
| |
| +--PolicyConditionInPolicyRule (PCIM & PCIMe)
|   |
|   +--SAConditionInRule
|
+--PolicySetComponent (PCIMe)

SystemSettingContext (Modèle cœur DMTF)
|
+--AutostartIKESettingContext
```

4. Classes de politique

Les classes de politique IPsec représentent l'ensemble des politiques qui sont contenues dans un système.



(a) PolicySetComponent ([RFC3460])

(b) IPsecPolicyForEndpoint
(c) IPsecPolicyForSystem
(d) PolicyRuleValidityPeriod ([RFC3060])
(e) SAConditionInRule
(f) PolicyActionInSARule
(g) PolicyActionInPolicyAction ([RFC3460])

Un PolicyGroup représente l'ensemble des politiques qui sont utilisées sur une interface. Ce PolicyGroup DEVRAIT être associé soit directement à l'instance de classe IPProtocolEndpoint qui représente l'interface (via l'association IPsecPolicyForEndpoint) soit indirectement (via l'association IPsecPolicyForSystem) associée au système qui héberge l'interface.

Les règles IKE et IPsec sont utilisées pour construire ou négocier la base de données d'association de sécurité IPsec (SADB, *Security Association Database*). Les règles IPsec représentent la base de données de politique de sécurité. La SADB elle-même n'est pas modélisée par le présent document.

Les règles IKE et IPsec peuvent être décrites comme (voir aussi la Section 6 sur les actions) :
- Un paquet de sortie non protégé va d'abord être vérifié à l'égard des règles IPsec. Si une correspondance est trouvée, la SADB va être consultée. Si il n'y a pas de SA IPsec correspondante dans la SADB, et si la négociation IKE est exigée par la règle IPsec, les règles IKE correspondantes seront utilisées. La SA négociée ou préconfigurée sera alors installée dans la SADB.
- Un paquet d'entrée non protégé sera d'abord vérifié par rapport aux règles IPsec. Si une correspondance est trouvée, la SADB va être consultée pour une SA IPsec correspondante. Si il n'y a pas de SA IPsec correspondante et qu'il existe une SA préconfigurée, celle-ci sera installée dans la SADB IPsec. Ce comportement ne devrait s'appliquer que dans des actions de détournement et d'élimination.
- Un paquet d'entrée protégé va d'abord être vérifié par rapport aux règles IPsec. Si une correspondance est trouvée, la SADB sera consultée sur une SA IPsec correspondante. Si il n'y a pas de SA IPsec correspondante et si existe une SA préconfigurée, celle-ci sera installée dans la SADB IPsec.
- Un paquet entrant de négociation IKE, qui ne fait pas partie d'une SA IKE existante, sera vérifié à l'égard des règles IKE. La SACondition pour la IKERule sera généralement composée d'une PeerIDPayloadFilterEntry (normalement pour une négociation IKE en mode agressif) ou un IPHeadersFilter. La SA négociée sera alors installée dans la SADB.

Il est prévu que lorsque une négociation IKE doit être initiée par une règle IPsec, l'ensemble des règles IKE sera vérifié. La vérification des règles IKE se fondera sur le paquet IKE sortant en utilisant les entrées IPHeadersFilter (normalement en utilisant la propriété HdrDstAddress).

4.1 Classe SARule

La classe SARule sert de classe de base pour IKERule et IPsecRule. Bien que la classe soit concrète, elle NE DOIT PAS être instanciée. Elle définit un point de connexion commun pour les associations aux conditions et actions pour les deux types de règles. Par sa dérivation de PolicyRule, une SARule (et donc IKERule et IPsecRule) a aussi l'association PolicyRuleValidityPeriod.

Chaque SARule dans un PolicyGroup valide DOIT avoir un numéro de priorité unique associé dans la PolicySetComponent.Priority. La définition de classe pour SARule est comme suit :

NOM : SARule
DESCRIPTION : Classe de base pour IKERule et IPsecRule.
DÉRIVÉ DE : PolicyRule (voir la [RFC3060] & [RFC3460])
ABSTRAITE : FAUX
PROPRIÉTÉS : PolicyRuleName (de PolicyRule) ; Enabled (de PolicyRule) ; ConditionListType (de PolicyRule)
RuleUsage (de PolicyRule) ; Mandatory (de PolicyRule) ; SequencedActions (de PolicyRule)
ExecutionStrategy (de PolicyRule) ; PolicyRoles (de PolicySet)
PolicyDecisionStrategy (de PolicySet) ; LimitNegotiation

4.1.1 Propriétés PolicyRuleName, Enabled, ConditionListType, RuleUsage, Mandatory, SequencedActions, PolicyRoles, et PolicyDecisionStrategy

Pour une description de ces propriétés, voir les [RFC3060] et [RFC3460].

Dans les instances de sous classe SARule :
- si la propriété Mandatory existe, elle DOIT être réglée à "vrai" ;
- si la propriété SequencedActions existe, elle DOIT être réglée à "mandatory" ;
- la propriété PolicyRoles n'est pas utilisée dans le modèle de niveau appareil ;
- si la propriété PolicyDecisionStrategy existe, elle doit être réglée à "FirstMatching".

4.1.2 Propriété ExecutionStrategy

Les propriétés ExecutionStrategy dans les sous classes PolicyRule (et dans la classe CompoundPolicyAction) déterminent le comportement des actions contenues. Elles définissent la stratégie à utiliser dans l'exécution de la séquence d'actions agrégées par une règle ou une action composée. Dans le cas d'actions au sein d'une règle, l'agrégation PolicyActionInSARule est utilisée pour collecter les actions dans un ensemble ordonné ; dans le cas d'une action composée, l'agrégation PolicyActionInPolicyAction est utilisée pour collecter les actions dans un sous ensemble ordonné.

Il y a trois stratégies d'exécution : faire jusqu'à réussite *(do until success)*, tout faire *(do all)*, et faire jusqu'à un échec *(do until failure)*.

"Faire jusqu'à réussite" cause l'exécution des actions conformément à la propriété ActionOrder dans les instances d'agrégation jusqu'à l'exécution réussie d'une seule action. Ces actions peuvent être évaluées pour déterminer si leur exécution est appropriée plutôt que d'essayer aveuglément chacune des actions jusque à ce qu'une réussisse. Pour un initiateur, elles sont essayées dans le ActionOrder jusqu'à épuisement de la liste ou qu'une s'achève avec succès. Par exemple, un initiateur IKE peut avoir plusieurs IKEActions pour la même SACondition. L'initiateur va essayer toutes les IKEActions dans l'ordre défini par ActionOrder. C'est-à-dire, il va éventuellement essayer plusieurs phases 1 de négociation avec différents modes (mode principal puis mode agressif) et/ou avec plusieurs homologues IKE. Pour un répondeur, lorsque il y a plus d'une action dans la règle avec la clause de condition "faire jusqu'à réussite", cela donne des alternatives d'actions selon les propositions reçues. Par exemple, la même IKERule peut être utilisée pour traiter les négociations en modes agressif et principal avec différentes actions. Le répondeur utilise la première action appropriée dans la liste des actions.

"Tout faire" cause l'exécution de toutes les actions dans l'ensemble agrégé conformément à leur ordre défini. L'exécution continue sans considération des échecs.

"Faire jusqu'à un échec" cause l'exécution de toutes les actions selon un ordre prédéfini jusqu'au premier échec dans l'exécution d'une instance d'action. On notera que si toutes les actions sont réussies, le résultat agrégé est alors un échec. Cette stratégie d'exécution est héritée de la [RFC3460] et n'est pas supposée être utile pour une configuration IPsec.

Par exemple, dans un cas de SA incorporées, les actions de la règle d'un initiateur pourraient être structurées comme :

```
IPsecRule.ExecutionStrategy='Do All'
  |
  +---1--- IPsecTunnelAction    // établir une SA de l'hôte à la passerelle
  |
  +---2--- IPsecTransportAction // établir une SA depuis l'hôte à travers le tunnel jusqu'à l'hôte distant
```

Un autre exemple, qui montre une règle avec des actions de repli pourrait être structurée ainsi :

```
IPsecRule.ExecutionStrategy='Do Until Success'
  |
  +---6--- IPsecTransportAction // négocier la SA avec l'homologue
  |
  +---9--- IPsecBypassAction    // mais si vous le devez, le permettre en clair
```

La classe CompoundPolicyAction (voir la [RFC3460]) peut être utilisée pour construire les actions des règles IKE et IPsec lorsque ces règles spécifient à la fois des actions multiples et des actions de repli. La propriété ExecutionStrategy dans CompoundPolicyAction est utilisée conjointement avec celles de PolicyRule.
Par exemple, pour incorporer des SA avec une passerelle de sécurité de repli, les actions d'une règle pourraient être structurées ainsi :

```
IPsecRule.ExecutionStrategy='Do All'
  |
  +---1--- CompoundPolicyAction.ExecutionStrategy='Do Until Success'
   |       |
  |        +---1--- IPsecTunnelAction  // établir la SA de l'hôte à la passerelle1
```

```
|     |
|     |
|     +---2--- IPsecTunnelAction  // ou établir la SA avec la pgsserelle2
|
+---2--- IPsecTransportAction       // puis établir la SA à partir de l'hôte jusqu'à l'hôte distant à travers le tunnel
```

Dans le cas de "tout faire", un couple d'actions peuvent être exécutées avec succès avant qu'une action suivante échoue. Dans ce cas, certaines actions IKE ou IPsec peuvent avoir résulté en la création de SA. Même si l'effet net des actions agrégées est l'échec, ces SA créées PEUVENT être conservées ou PEUVENT être supprimées.

Dans le cas de "tout faire", les sélecteurs IPsec à utiliser durant la négociation de SA IPsec sont :
- Pour la dernière IPsecAction de l'agrégation (c'est-à-dire, généralement la SA IPsec la plus interne) : c'est la combinaison de la classe IPHeadersFilter et de la propriété Granularity de l'IPsecAction.
- Pour toutes les autres IPsecActions de l'agrégation, le sélecteur est l'adresse IP de source IP qui est l'adresse IP locale, et l'adresse IP de destination est l'adresse IP PeerGateway de la IPsecAction suivante de l'agrégation "tout faire". Noter que la granularité est d'adresse IP à adresse IP.

Si le comportement ci-dessus n'est pas désirable, la solution de remplacement est de définir plusieurs SARules, une pour chaque SA IPsec à construire. Cela va permettre la définition de sélecteurs IPsec spécifiques pour toutes les IPsecActions.

4.1.3 Propriété LimitNegotiation

La propriété LimitNegotiation est utilisée au titre du traitement d'une règle IKE ou IPsec.

Avant de passer à une négociation de phase 1, cette propriété est vérifiée pour déterminer si le rôle de négociation de cette règle correspond à celui défini pour la négociation entreprise (par exemple, initiateur, répondeur, ou les deux). Si cette vérification échoue (par exemple, le rôle actuel est celui de répondeur IKE, alors que la règle spécifie initiateur IKE), la négociation IKE est alors arrêtée. Noter que ceci ne s'applique qu'aux nouvelles négociations IKE de phase 1 et n'a pas d'effet sur la renégociation ou les opérations de rafraîchissement avec des homologues pour lesquels une SA établie existe déjà.

Avant de passer à une négociation de phase 2, la propriété LimitNegotiation de la IPsecRule est d'abord vérifiée pour déterminer si le rôle de négociation indiqué pour la règle correspond à celui de la négociation en cours (initiateur, répondeur, ou l'un et l'autre). Noter que cette limite ne s'applique qu'aux nouvelles négociations de phase 2. Elle est ignorée lorsque on tente de rafraîchir une SA expirante (l'un ou l'autre côté peut initier une opération de rafraîchissement). Le système IKE peut déterminer que la négociation est une opération de rafraîchissement en vérifiant si les informations de sélecteur correspondent à celles d'une SA existante. Si LimitNegotiation ne correspond pas et si le sélecteur correspond à une nouvelle SA, la négociation est arrêtée.

La propriété est définie comme suit :

NOM : LimitNegotiation
DESCRIPTION : Limite le rôle à entreprendre durant la négociation.
SYNTAXE : Entier de 16 bits non signé
VALEUR : 1 - initiateur seul ; 2 - répondeur seul ; 3 – les deux

4.2 Classe IKERule

La classe IKERule associe des conditions et des actions pour les négociations IKE de phase 1. La définition de classe pour IKERule est la suivante :

NOM : IKERule
DESCRIPTION : Associe les conditions et les actions pour les négociations IKE de phase 1.
DÉRIVÉ DE : SARule
ABSTRAITE : FAUX
PROPRIÉTÉS : Les mêmes que SARule, plus IdentityContexts

4.2.1 Propriété IdentityContexts

Le service IKE d'un point d'extrémité de sécurité peut avoir plusieurs identités à utiliser dans différentes situations. La combinaison de l'interface (représentée par le IPProtocolEndpoint ou par une collection de IPProtocolEndpoints) le type d'identité (comme spécifié dans la IKEAction) et le IdentityContexts spécifient une identité unique.

La propriété IdentityContexts spécifie le contexte pour choisir l'identité IKE pertinente à utiliser durant la IKEAction suivante. Un contexte peut être un nom de VPN ou un autre identifiant pour choisir l'identité appropriée à utiliser sur le IPProtocolEndpoint ou la collection de IPProtocolEndpoints protégés.

IdentityContexts est un arrangement de chaînes. Les multiples valeurs dans l'arrangement sont composées ensemble avec l'opérateur logique OU dans l'évaluation des IdentityContexts. Chaque valeur dans l'arrangement peut être la composition de multiples noms de contextes. Donc, une seule valeur peut être un seul nom de contexte (par exemple, "CompagnieXVPN"), ou elle peut être une combinaison de contextes. Quand une valeur d'un arrangement est une composition, les valeurs individuelles sont ajoutées ensemble par l'opérateur logique ET pour l'évaluation et la syntaxe est :

<ContextName>[&&<ContextName>]*

où les noms des contextes individuels apparaissent en ordre alphabétique (selon la séquence de collationnement pour UCS-2). Donc, par exemple, les valeurs "CompagnieXVPN", "CompagnieYVPN&&TopSecret", "CompagnieZVPN&&Confidentiel" signifient que, pour les IPProtocolEndpoint et IdentityType appropriés, les contextes sont satisfaits si l'identité spécifie "CompagnieXVPN", "CompagnieYVPN&&TopSecret", ou "CompagnieZVPN&&Confidentiel".

La propriété est définie comme suit :

NOM : IdentityContexts
DESCRIPTION : Spécifie le contexte dans lequel choisir l'identité IKE.
SYNTAXE : Arrangement de chaînes

4.3 Classe IPsecRule

La classe IPsecRule associe les conditions et actions pour les négociations IKE de phase 2 pour le DOI IPsec. La définition de classe pour IPsecRule est la suivante :

NOM : IPsecRule
DESCRIPTION : Associe les conditions et actions pour les négociations IKE de phase 2 pour le DOI IPsec.
DÉRIVÉ DE : SARule
ABSTRAITE : FAUX
PROPRIÉTÉS : Les mêmes que SARule

4.4 Classe d'association IPsecPolicyForEndpoint

La classe IPsecPolicyForEndpoint associe un PolicyGroup à une interface réseau spécifique. Si un IPProtocolEndpoint d'un système n'a pas un PolicyGroup associé à un IPsecPolicyForEndpoint, le PolicyGroup associé au IPsecPolicyForSystem est alors utilisé pour ce point d'extrémité. La définition de classe pour IPsecPolicyForEndpoint est la suivante :

NOM : IPsecPolicyForEndpoint
DESCRIPTION : Associe un groupe de politique à une interface réseau.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref IPProtocolEndpoint[0..n]] ; Dependent[ref PolicyGroup[0..1]]

4.4.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de IPProtocolEndpoint. La cardinalité [0..n] indique qu'une instance de PolicyGroup peut être associée à zéro, une ou plusieurs instances de IPProtocolEndpoint.

4.4.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassées pour se référer à une instance de PolicyGroup. La cardinalité [0..1] indique qu'une instance de IPProtocolEndpoint peut avoir une association à au plus une instance de PolicyGroup.

4.5 Classe d'association IPsecPolicyForSystem

La classe IPsecPolicyForSystem associe un PolicyGroup à un système spécifique. Si un IPProtocolEndpoint d'un système n'a pas un PolicyGroup associé à un IPsecPolicyForEndpoint, le PolicyGroup associé au IPsecPolicyForSystem est utilisé pour ce point d'extrémité. La définition de classe pour IPsecPolicyForSystem est la suivante :

NOM : IPsecPolicyForSystem
DESCRIPTION : Groupe de politique par défaut pour un système.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent[ref System[0..n]] ; Dependent[ref PolicyGroup[0..1]]

4.5.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de système. La cardinalité [0..n] indique qu'une instance de PolicyGroup peut avoir une association à zéro, une ou plusieurs instances de système.

4.5.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de PolicyGroup. La cardinalité [0..1] indique qu'une instance de System peut avoir une association à au plus une instance de PolicyGroup.

4.6 Classe d'agrégation SAConditionInRule

La classe SAConditionInRule associe une SARule à une ou des instances de SACondition qui la déclenchent. La définition de classe pour SAConditionInRule est la suivante :

NOM : SAConditionInRule
DESCRIPTION : Associe une SARule à une ou des instances de SACondition qui la déclenchent.
DÉRIVÉ DE : PolicyConditionInPolicyRule (voir les [RFC3060] & [RFC3460])
ABSTRAITE : FAUX
PROPRIÉTÉS : GroupNumber (de PolicyConditionInPolicyRule) ; ConditionNegated (de PolicyConditionInPolicyRule) ; GroupComponent [ref SARule [0..n]] ; PartComponent [ref SACondition [1..n]]

4.6.1 Propriétés GroupNumber et ConditionNegated

Pour une description de ces propriétés, voir la [RFC3060].

4.6.2 Référence GroupComponent

La propriété GroupComponent est héritée de PolicyConditionInPolicyRule et est outrepassée pour se référer à une instance de SARule. La cardinalité [0..n] indique qu'une instance de SACondition peut être contenue dans zéro, une ou plusieurs instances de SARule.

4.6.3 Référence PartComponent

La propriété PartComponent est héritée de PolicyConditionInPolicyRule et est outrepassée pour se référer à une instance de SACondition instance. La cardinalité [1..n] indique qu'une instance de SARule DOIT contenir au moins une instance de SACondition.

4.7 Classe d'agrégation PolicyActionInSARule

La classe PolicyActionInSARule associe une SARule à une ou plusieurs instances de PolicyAction. Dans tous les cas où une SARule est utilisée, les actions contenues DOIVENT être soit des sous classes de SAAction, soit des instances de CompoundPolicyAction. Pour une IKERule, les actions contenues DOIVENT se rapporter au traitement de phase 1, c'est-à-dire, IKEAction ou IKERejectAction. De même, pour une IPsecRule, les actions contenues DOIVENT se rapporter au traitement de phase 2 ou de SA préconfigurée, par exemple, IPsecTransportAction, IPsecBypassAction, etc. La définition de classe pour PolicyActionInSARule est la suivante :

NOM : PolicyActionInSARule
DESCRIPTION : Associe une SARule à sa ou ses PolicyAction.
DÉRIVÉ DE : PolicyActionInPolicyRule (voir les [RFC3060] & [RFC3460])
ABSTRAITE : FAUX
PROPRIÉTÉS : GroupComponent [ref SARule [0..n]] ; PartComponent [ref PolicyAction [1..n]] ; ActionOrder (de PolicyActionInPolicyRule)

4.7.1 Référence GroupComponent

La propriété GroupComponent est héritée de PolicyActionInPolicyRule et est outrepassée pour se référer à une instance de SARule. La cardinalité [0..n] indique qu'une instance de SAAction peut être contenue dans zéro, une ou plusieurs instances de SARule.

4.7.2 Référence PartComponent

La propriété PartComponent est héritée de PolicyActionInPolicyRule et est outrepassée pour se référer à une instance de SAAction ou de CompoundPolicyAction. La cardinalité [1..n] indique qu'une instance de SARule DOIT contenir au moins une instance de SAAction ou de CompoundPolicyAction.

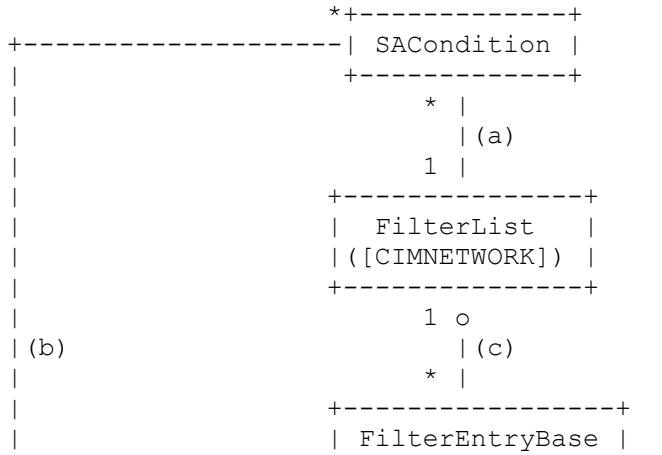
4.7.3 Propriété ActionOrder

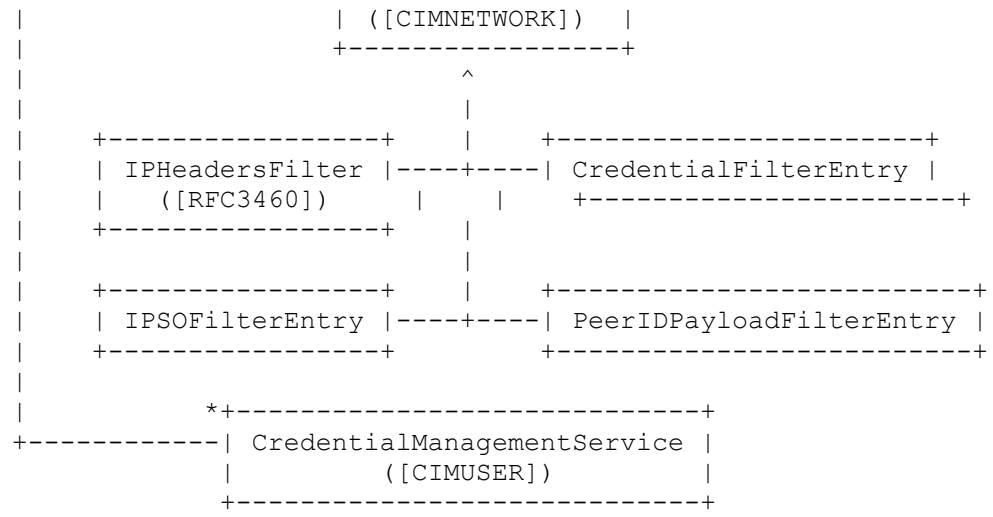
La propriété ActionOrder est héritée de la superclasse PolicyActionInPolicyRule. Elle spécifie la position relative de cette PolicyAction dans la séquence des actions associées à une PolicyRule. Le ActionOrder DOIT être unique afin de fournir un ordre déterministe. De plus, les actions dans une SARule sont exécutées comme suit. Voir au paragraphe 4.2.2, ExecutionStrategy, la discussion sur l'utilisation de la propriété ActionOrder. La propriété est définie comme suit :

NOM : ActionOrder
DESCRIPTION : Spécifie l'ordre des actions.
SYNTAXE : entier de 16 bits non signé
VALEUR : Toute valeur entre 1 et 2^16-1 inclus. Les valeurs les plus faibles ont une plus forte préséance (c'est-à-dire, 1 est la plus forte préséance). L'ordre de fusion de deux SAActions avec la même préséance est indéfini.

5. Classes de condition et de filtre

Les classes IPsec de condition et de filtre sont utilisées pour construire la partie "if" des règles IKE et IPsec.





(a) FilterOfSACondition
(b) AcceptCredentialsFrom
(c) EntriesInFilterList (voir [CIMNETWORK])

5.1 Classe SACondition

La classe SACondition définit les conditions des règles pour les négociations IKE et IPsec. Les conditions sont associées aux règles de politique via l'agrégation SAConditionInRule. Elle est utilisée comme point d'ancrage pour associer divers types de filtres aux règles de politique via l'association FilterOfSACondition. Elle définit aussi si les accréditifs peuvent être acceptés pour une certaine règle de politique via l'association AcceptCredentialsFrom.

Les objets associés représentent des composants de la condition qui peut s'appliquer ou non à une certaine évaluation de règle. Par exemple, une évaluation AcceptCredentialsFrom n'est effectuée que lorsque un accréditif est disponible pour être évalué par rapport à la liste des services de gestion d'accréditifs de confiance. De même, une PeerIDPayloadFilterEntry ne peut être évaluée que lorsque une valeur de IDPayload est disponible pour la comparer au filtre. Les composant de condition qui n'ont pas de valeurs correspondantes par rapport auxquelles on puisse les évaluer sont évaluées comme VRAI sauf si le protocole s'est achevé sans fournir les informations requises.

La définition de classe pour SACondition est la suivante :

NOM : SACondition
DESCRIPTION : Définit les préconditions pour les négociations IKE et IPsec.
DÉRIVÉ DE : PolicyCondition (voir la [RFC3060])
ABSTRAITE : FAUX
PROPRIÉTÉS : PolicyConditionName (de PolicyCondition)

5.2 Classe IPHeadersFilter

La classe IPHeadersFilter est définie dans la [RFC3460] avec la note suivante :

1) pour spécifier des quintuplets de filtres qui sont à appliquer de façon symétrique (c'est-à-dire, qui correspondent au trafic dans les deux directions des mêmes flux, ce qui est assez normal pour les entrées de SPD pour le trafic d'entrée et de sortie) la propriété Direction de la FilterList DEVRAIT être réglée à "Mirrored".

5.3 Classe CredentialFilterEntry

La classe CredentialFilterEntry définit une classe d'équivalence qui satisfait aux accréditifs des homologues IKE. Chaque CredentialFilterEntry inclut un MatchFieldName qui est interprété conformément aux CredentialManagementService associés à la SACondition (AcceptCredentialsFrom).

Ces accréditifs peuvent être des certificats X.509, des tickets Kerberos, ou d'autres types d'accréditifs obtenus durant

l'échange de phase 1.

Note : cette entrée de filtre sera probablement vérifiée durant la négociation IKE. Si la vérification échoue, la négociation IKE DOIT alors être arrêtée, et le résultat de la IKEAction qui a déclenché cette négociation sera un échec.

La définition de classe pour CredentialFilterEntry est la suivante :

NOM : CredentialFilterEntry
DESCRIPTION : Spécifie un filtre de correspondance fondé sur les accréditifs IKE.
DÉRIVÉ DE : FilterEntryBase (voir [CIMNETWORK])
ABSTRAITE : FAUX
PROPRIÉTÉS : Name (de FilterEntryBase) ; IsNegated (de FilterEntryBase) ; MatchFieldName ; MatchFieldValue ; CredentialType

5.3.1 Propriété MatchFieldName

La propriété MatchFieldName spécifie la sous partie de l'accréditif qui doit correspondre à MatchFieldValue. La propriété est définie comme suit :

NOM : MatchFieldName
DESCRIPTION : Spécifie quelle sous partie de l'accréditif doit correspondre.
SYNTAX E : chaîne
VALEUR : C'est la représentation de chaîne d'un attribut de certificat X.509, par exemple: "serialNumber", "signatureAlgorithm", "issuerName", "subjectName", "subjectAltName", ...

5.3.2 Propriété MatchFieldValue

La propriété MatchFieldValue spécifie la valeur à comparer à MatchFieldName dans un accréditif pour déterminer si l'accréditif satisfait à cette entrée de filtre. La propriété est définie comme suit :

NOM : MatchFieldValue
DESCRIPTION : Spécifie la valeur à satisfaire par le MatchFieldName.
SYNTAXE : chaîne
VALEUR : Si la CredentialFilterEntry correspond à un nom distinctif, cette valeur est représentée dans la classe CIM par une valeur de chaîne ordinaire. Cependant, une mise en œuvre doit convertir cette chaîne en DER avant de la confronter aux valeurs extraites des accréditifs au démarrage.

Un mécanisme de caractères générique peut être utilisé pour les MatchFieldNames qui contiennent des chaînes de caractères. La MatchFieldValue peut contenir un caractère générique, '*', dans la spécification de correspondance de schéma. Par exemple, si le MatchFieldName est "subjectName", alors une MatchFieldValue de "cn=*,ou=engineering,o=foo,c=be" sera confrontée avec succès à un certificat dont l'attribut sujet est "cn=Jane Doe,ou=engineering,o=foo,c=be". Le caractère générique peut être utilisé pour représenter 0, un ou plusieurs caractères comme ceux qui seraient affichés à l'utilisateur (c'est-à-dire, une confrontation de schéma générique fonctionne sur les limites des caractères affichables).

5.3.3 Propriété CredentialType

La propriété CredentialType spécifie le type particulier d'accréditif qui est confronté. La propriété est définie comme suit :

NOM : CredentialType
DESCRIPTION : Définit le type des accréditifs IKE.
SYNTAXE : entier de 16 bits non signé
VALEUR : 1 – certificat X.509
2 – ticket Kerberos

5.4 Classe IPSOFilterEntry

La classe IPSOFilterEntry est utilisée pour confronter du trafic sur la base des valeurs d'en-tête des options de sécurité IP [RFC1108] (ClassificationLevel et ProtectionAuthority) comme défini dans la RFC 1108. Ce type d'entrée de filtre est utilisé pour ajuster le niveau de chiffrement IPsec selon la classification IPSO du trafic (par exemple, secret, confidentiel, restreint, etc.) La définition de classe pour IPSOFilterEntry est la suivante :

NOM : IPSOFilterEntry
DESCRIPTION : Spécifie un filtre de correspondance sur la base des options de sécurité IP.
DÉRIVÉ DE : FilterEntryBase (voir [CIMNETWORK])
ABSTRAITE : FAUX
PROPRIÉTÉS : Name (de FilterEntryBase) ; IsNegated (de FilterEntryBase) ; MatchConditionType ; MatchConditionValue

5.4.1 Propriété MatchConditionType

La propriété MatchConditionType spécifie le champ d'en-tête IPSO qui va être confronté (par exemple, niveau de classification du trafic ou autorité de protection). La propriété est définie comme suit :

NOM : MatchConditionType
DESCRIPTION : Spécifie le champ d'en-tête IPSO à confronter.
SYNTAXE : entier de 16 bits non signé
VALEUR : 1 - ClassificationLevel
2 - ProtectionAuthority

5.4.2 Propriété MatchConditionValue

La propriété MatchConditionValue spécifie la valeur du champ d'en-tête IPSO à confronter. La propriété est définie comme suit :

NOM : MatchConditionValue
DESCRIPTION : Spécifie la valeur du champ d'en-tête IPSO à confronter.
SYNTAXE : entier de 16 bits non signé
VALEUR : Les valeurs DOIVENT être une des valeurs énumérées dans la RFC 1108 (ou tout autre document ultérieur des numéros alloués de l'IANA). Des exemples de ClassificationLevel sont :

61 - TopSecret
90 - Secret
150 - Confidentiel
171 – Non classifié

Des exemples pour ProtectionAuthority sont :

0 - GENSER
1 - SIOP-ESI
2 - SCI
3 - NSA
4 - DOE

5.5 Classe PeerIDPayloadFilterEntry

La classe PeerIDPayloadFilterEntry définit les filtres utilisés pour confronter les valeurs d'identifiant de charge utile provenant de l'échange de protocole IKE. PeerIDPayloadFilterEntry permet la spécification de certaines valeurs d'identifiant de charge utile telles que "*@example.com" ou "192.0.2.0/24".

Évidemment, ce filtre ne s'applique qu'aux IKERules agissant comme répondeur. De plus, ce filtre peut être appliqué immédiatement dans le cas d'un mode agressif mais son application sera retardée dans le cas du mode principal. La définition de classe pour PeerIDPayloadFilterEntry est la suivante :

NOM : PeerIDPayloadFilterEntry
DESCRIPTION : Spécifie un filtre de correspondance fondé sur l'identité IKE.
DÉRIVÉ DE : FilterEntryBase (voir [CIMNETWORK])
ABSTRAITE : FAUX
PROPRIÉTÉS : Name (de FilterEntryBase) ; IsNegated (de FilterEntryBase) ; MatchIdentityType ; MatchIdentityValue

5.5.1 Propriété MatchIdentityType

La propriété MatchIdentityType spécifie le type d'identité fournie par l'homologue dans l'identifiant de charge utile. La propriété est définie comme suit :

NOM : MatchIdentityType
DESCRIPTION : Spécifie le type d'ID de charge utile.
SYNTAXE : entier de 16 bits non signé
VALEUR : Consulter les valeurs valides dans la [RFC2407].

5.5.2 Propriété MatchIdentityValue

La propriété MatchIdentityValue spécifie la valeur de filtre pour la comparaison avec l'identifiant de charge utile, par exemple, "*@example.com". La propriété est définie comme suit :

NOM : MatchIdentityValue
DESCRIPTION : Spécifie la valeur de l'identifiant de charge utile.
SYNTAXE : chaîne
VALEUR : La syntaxe peut avoir besoin d'être convertie pour la comparaison. Si le type PeerIDPayloadFilterEntry est un DistinguishedName, le nom dans la propriété MatchIdentityValue est représenté par une valeur de chaîne ordinaire, mais cette valeur doit être convertie en une chaîne codée en DER avant de la confronter aux valeurs extraites de l'identifiant de charge utile IKE au démarrage. La même chose s'applique aux adresses IPv4 & IPv6.

Différents mécanismes de caractères génériques peuvent être utilisés selon l'identifiant de charge utile :
- une MatchIdentityValue de "*@example.com" va correspondre à un identifiant de charge utile de FQDN d'utilisateur de "JDOE@EXAMPLE.COM".
- une MatchIdentityValue de "*.example.com" va correspondre à un identifiant de charge utile de FQDN de "WWW.EXAMPLE.COM".
- une MatchIdentityValue de "cn=*,ou=engineering,o=company,c=us" va correspondre à un identifiant de charge utile de DN en DER de "cn=John Doe,ou=engineering,o=company,c=us".
- une MatchIdentityValue de "193.190.125.0/24" va correspondre à un identifiant de charge utile d'adresse IPv4 de 193.190.125.10.
- une MatchIdentityValue de "193.190.125.*" va aussi correspondre à un identifiant de charge utile d'adresse iPv4 de 193.190.125.10.

Les mécanismes de caractère générique ci-dessus DOIVENT être acceptés pour tous les identifiants de charge utile pris en charge par l'entité IKE locale. Le caractère '*' remplace 0, une ou plusieurs instances de tout caractère avec les restrictions du type spécifié par MatchIdentityType.

5.6 Classe d'association FilterOfSACondition

La classe FilterOfSACondition associe une SACondition aux spécifications de filtre (FilterList) qui constituent la condition. La définition de classe pour FilterOfSACondition est la suivante :

NOM : FilterOfSACondition
DESCRIPTION : Associe une condition à la liste des filtres qui constituent les éléments des conditions individuelles.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE :FAUX
PROPRIÉTÉS : Antecedent [ref FilterList[1..1]] ; Dependent [ref SACondition[0..n]]

5.6.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de FilterList. La cardinalité [1..1] indique qu'une instance de SACondition DOIT être associée à une instance de FilterList et une seule.

5.6.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de SACondition. La cardinalité [0..n] indique qu'une instance de FilterList peut être associée à zéro, une ou plusieurs instances de SACondition.

5.7 Classe d'association AcceptCredentialFrom

La classe AcceptCredentialFrom spécifie quels services de gestion d'accréditifs (par exemple, une CertificateAuthority ou un

service Kerberos) sont de confiance pour certifier les accréditifs de l'homologue. Ceci est utilisé pour s'assurer que l'accréditif confronté dans la CredentialFilterEntry est un accréditif valide qui a été fourni par un CredentialManagementService approuvé. Si un CredentialManagementService est spécifié et si une CredentialFilterEntry correspondante est utilisée, mais si l'accréditif fourni par l'homologue n'est pas certifié par ce CredentialManagementService (ou un des CredentialManagementService dans sa hiérarchie de confiance) la CredentialFilterEntry est réputée ne pas correspondre. Si un accréditif est certifié par un CredentialManagementService dans la liste AcceptCredentialsFrom des services, mais qu'il n'y a pas d'entrée CredentialFilterEntry, ceci est considéré comme équivalent à une CredentialFilterEntry qui correspond à tous les accréditifs provenant de ces services.

La définition de classe pour AcceptCredentialFrom est la suivante :

NOM : AcceptCredentialFrom
DESCRIPTION : Associe une condition aux services de gestion d'accréditifs auxquels faire confiance.
DÉRIVÉ DE ; Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref CredentialManagementService[0..n]] ; Dependent [ref SACondition[0..n]]

5.7.1 Référence Antecedent

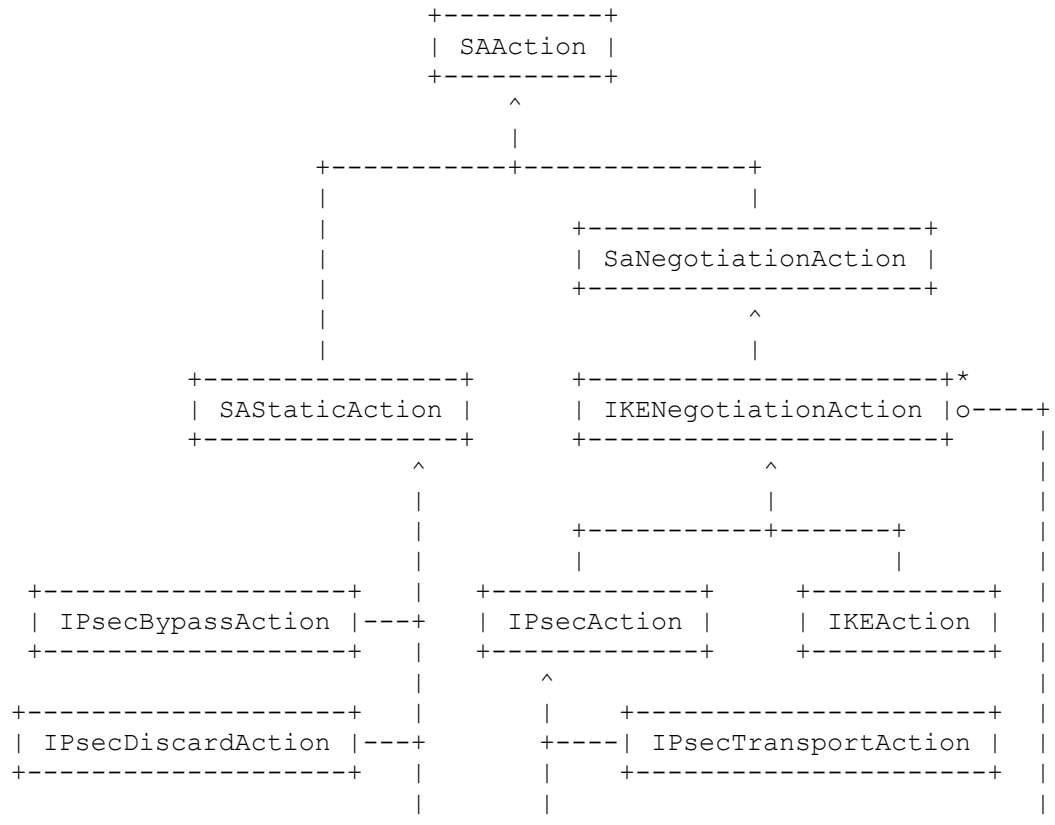
La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de CredentialManagementService. La cardinalité [0..n] indique qu'une instance de SACondition peut être associée à zéro, une ou plusieurs instances de CredentialManagementService.

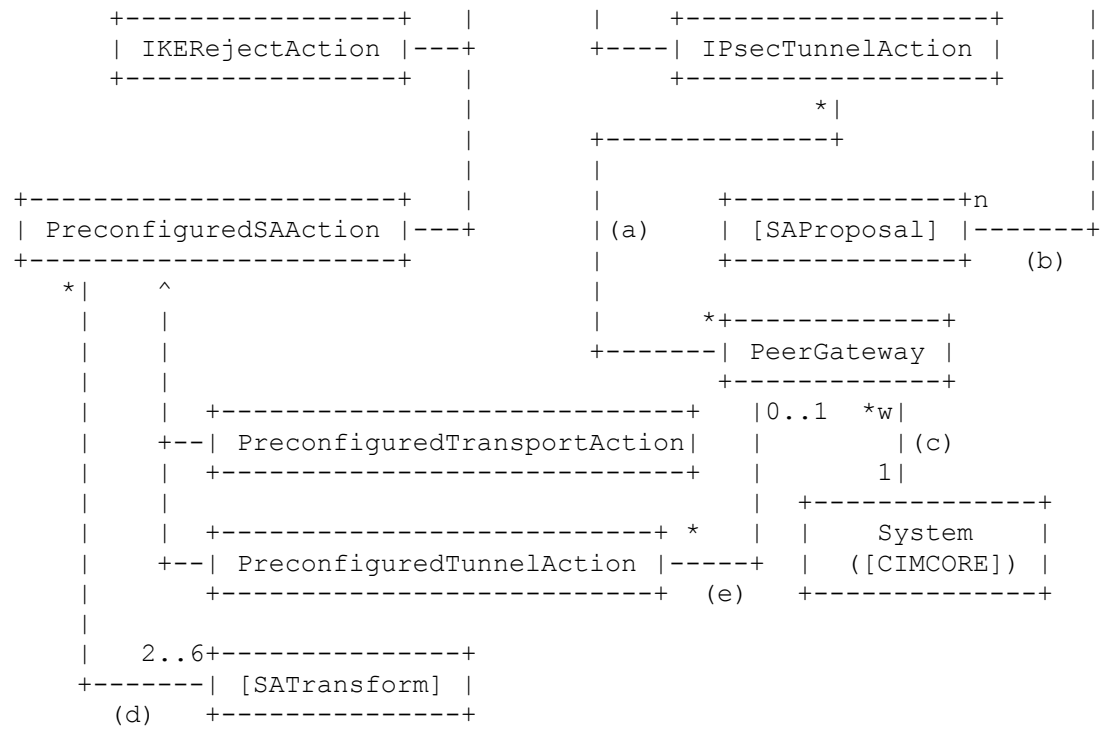
5.7.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de SACondition. La cardinalité [0..n] indique qu'une instance de CredentialManagementService peut être associée à zéro, une ou plusieurs instances de SACondition.

6. Classes d'action

Les classes d'action sont utilisées pour modéliser les différentes actions qu'un appareil IPsec peut effectuer lorsque l'évaluation de la condition associée résulte en une correspondance.





(a) PeerGatewayForTunnel
(b) ContainedProposal
(c) HostedPeerGatewayInformation
(d) TransformOfPreconfiguredAction
(e) PeerGatewayForPreconfiguredTunnel

6.1 Classe SAAction

La classe SAAction est abstraite et sert de classe de base pour les actions IKE et IPsec. Elle est utilisée pour agréger différents types d'actions aux règles IKE et IPsec. La définition de classe pour SAAction est la suivante :

NOM : SAAction
DESCRIPTION : Classe de base pour les actions IKE et IPsec.
DÉRIVÉ DE : PolicyAction (voir la [RFC3060])
ABSTRAITE : VRAI
PROPRIÉTÉS : PolicyActionName (de PolicyAction) ; DoActionLogging ; DoPacketLogging

6.1.1 Propriété DoActionLogging

La propriété DoActionLogging spécifie si un message d'enregistrement dans le journal d'événements doit être généré lorsque l'action est effectuée. Ceci s'applique aux SANegotiationAction avec la signification d'enregistrer un message lorsque on tente la négociation (avec un résultat de succès ou d'échec). Cela s'applique aussi pour SAStaticAction seulement pour PreconfiguredSAAction avec la signification d'enregistrer un message lorsque la SA préconfigurée est bien installée dans la SADB. La propriété est définie comme suit :

NOM : DoActionLogging
DESCRIPTION : Spécifie si il faut enregistrer dans le journal lorsque l'action est effectuée.
SYNTAXE : booléen
VALEUR : vrai – un message d'enregistrement est à générer lorsque l'action est effectuée.
faux – aucun message d'enregistrement n'est à générer lorsque l'action est effectuée.

6.1.2 Propriété DoPacketLogging

La propriété DoPacketLogging spécifie si un message d'enregistrement doit être généré lorsque l'association de sécurité résultante est utilisée pour traiter le paquet. Si la SANegotiationAction s'exécute avec succès et résulté en la création d'une ou plusieurs associations de sécurité, ou si la PreconfiguredSAAction s'exécute, la valeur de DoPacketLogging DEVRAIT être

propagée dans un champ facultatif de la SADB. Ce champ facultatif devrait être utilisé pour décider si un message d'enregistrement doit être généré lorsque la SA est utilisée pour traiter un paquet. Pour SAStaticAction, un message d'enregistrement doit être généré lorsque une IPsecBypassAction, IPsecDiscardAction, ou IKERejectAction est exécutée. La propriété est définie comme suit :

NOM : DoPacketLogging
DESCRIPTION : Spécifie si on doit enregistrer un événement lorsque l'association de sécurité résultante est utilisée pour traiter le paquet.
SYNTAXE : booléen
VALEUR : vrai - un message d'enregistrement est généré lorsque l'association de sécurité résultante est utilisée pour traiter le paquet.
faux - aucun message d'enregistrement n'est à générer.

6.2 Classe SAStaticAction

La classe SAStaticAction est abstraite et sert de classe de base pour les actions IKE et IPsec qui n'exigent aucune négociation. La définition de classe pour SAStaticAction est la suivante :

NOM : SAStaticAction
DESCRIPTION : Classe de base pour les actions IKE et IPsec qui n'exigent aucune négociation.
DÉRIVÉ DE : SAAction
ABSTRAITE : VRAI
PROPRIÉTÉS : LifetimeSeconds

6.2.1 Propriété LifetimeSeconds

La propriété LifetimeSeconds spécifie pendant combien de temps devrait être utilisée l'association de sécurité dérivée de cette action. La propriété est définie comme suit :

NOM : LifetimeSeconds
DESCRIPTION : Spécifie en secondes la durée pendant laquelle une association de sécurité dérivée de cette action devrait être utilisée.
SYNTAXE : entier non signé de 64 bits
VALEUR : Une valeur de zéro indique qu'il n'y a pas de durée de vie associée à cette action (c'est-à-dire, une durée de vie infinie). Une valeur différente de zéro est normalement utilisée conjointement avec d'autres SAAction effectuées lorsque il y a un échec de négociation de quelque sorte que ce soit.

Note : si l'objet SAStaticAction référencé est une PreconfiguredSAAction associée à plusieurs SATransform, la durée de vie réelle de la SA préconfigurée sera alors le plus petit de la valeur de cette propriété LifetimeSeconds et de la valeur de la propriété MaxLifetimeSeconds de la SATransform associée. Si la valeur de cette propriété LifetimeSeconds est zéro, il n'y aura alors aucune durée de vie associée à cette SA.

Note : bien que certains protocoles de négociation de SA [RFC2409] puissent négocier la durée de vie comme un champ de longueur arbitraire, les auteurs ont supposé qu'un entier de 64 bits serait suffisant.

Il est prévu que la plupart des instances de SAStaticAction auront leur propriété LifetimeSeconds réglée à zéro (signifiant qu'il n'y a pas d'expiration de la SA résultante).

6.3 Classe IPsecBypassAction

La classe IPsecBypassAction est utilisée lorsque il est permis de traiter les paquets sans leur appliquer l'encapsulation IPsec. C'est la même chose que de déclarer qu'il est permis que les paquets s'écoulent en clair. La définition de classe pour IPsecBypassAction est la suivante :

NOM : IPsecBypassAction
DESCRIPTION : Spécifie qu'il est permis de passer les paquets en clair.
DÉRIVÉ DE : SAStaticAction
ABSTRAITE : FAUX

6.4 Classe IPsecDiscardAction

La classe IPsecDiscardAction est utilisée lorsque des paquets sont à éliminer. C'est la même chose que de déclarer que les paquets sont à refuser. La définition de classe pour IPsecDiscardAction est la suivante :

NOM : IPsecDiscardAction
DESCRIPTION : Spécifie que les paquets sont à éliminer.
DÉRIVÉ DE : SAStaticAction
ABSTRAITE : FAUX

6.5 Classe IKERejectAction

La classe IKERejectAction est utilisée pour empêcher de tenter une négociation IKE avec le ou les homologues. Le principal usage de cette classe est d'empêcher des attaques de déni de service quand on agit comme répondeur IKE. Cela va au delà d'une simple élimination de paquets IKE UDP/500 parce que la SACondition peut être fondée sur une PeerIDPayloadFilterEntry spécifique (lorsque le mode agressif est utilisé). La définition de classe pour IKERejectAction est la suivante :

NOM : IKERejectAction
DESCRIPTION : Spécifie qu'une négociation IKE ne devrait pas être tentée ou continuée.
DÉRIVÉ DE : SAStaticAction
ABSTRAITE : FAUX

6.6 Classe PreconfiguredSAAction

La classe PreconfiguredSAAction sert à créer une association de sécurité en utilisant des algorithmes et des clés préconfigurés, incorporés.

Notes : le SPI pour une PreconfiguredSAAction est contenu dans l'association, TransformOfPreconfiguredAction ; la clé de session (si applicable) est contenue dans une instance de la classe SharedSecret (voir [CIMUSER]). La clé de session est mémorisée dans la propriété Secret, le protocole de propriété contient "ESP-encrypt", "ESP-auth" ou "AH", l'algorithme de propriété contient l'algorithme utilisé pour protéger le secret (ce peut être "PLAINTEXT" si l'entité IPsec n'a pas de secret mémorisé) la valeur de la propriété RemoteID est l'enchaînement de l'adresse IP de l'homologue IPsec distant en décimal séparé par des points, du caractère "/", de "IN" pour une SA entrante (respectivement "OUT" pour une SA sortante) du caractère "/", et de la représentation hexadécimale du SPI.

Bien que la classe soit concrète, elle NE DOIT PAS être instanciée. La définition de classe pour PreconfiguredSAAction est la suivante :

NOM : PreconfiguredSAAction
DESCRIPTION : Spécifie les informations préconfigurées d'algorithme et de clé pour la création d'une association de sécurité.
DÉRIVÉ DE : SAStaticAction
ABSTRAITE : VRAI
PROPRIÉTÉS : LifetimeKilobytes

6.6.1 Propriété LifetimeKilobytes

La propriété LifetimeKilobytes spécifie une limite de trafic en kilo octets qui peut être consommée avant la suppression de la SA. La propriété est définie comme suit :

NOM : LifetimeKilobytes
DESCRIPTION : Spécifie la durée de vie de la SA en kilo octets.
SYNTAXE : entier non signé de 64 bits
VALEUR : Une valeur de zéro indique qu'il n'y a pas de durée de vie associée à cette action (c'est-à-dire, une durée de vie infinie). Une valeur différente de zéro est utilisée pour indiquer qu'après que ce nombre de kilo octets a été consommé, la SA doit être supprimée de la SADB.

Note : La durée de vie réelle de la SA préconfigurée sera le plus petit de la valeur de cette propriété LifetimeKilobytes et de la valeur de la propriété MaxLifetimeSeconds de la SATransform associée. Si la valeur de cette propriété

LifetimeKilobytes est zéro, il n'y aura alors pas de durée de vie associée à cette action.

Note : Bien que certains protocoles de négociation de SA [RFC2409] puissent négocier la durée de vie comme un champ de longueur arbitraire, les auteurs ont supposé qu'un entier de 64 bits sera suffisant.

Il est prévu que la plupart des instances de PreconfiguredSAAction auront la propriété LifetimeKilobyte réglée à zéro (signifiant qu'il n'y a pas d'expiration de la SA résultante).

6.7 Classe PreconfiguredTransportAction

La classe PreconfiguredTransportAction est utilisée pour créer une association de sécurité IPsec en mode transport en utilisant les algorithmes et clés préconfigurés, incorporés. La définition de classe pour PreconfiguredTransportAction est la suivante :

NOM : PreconfiguredTransportAction
DESCRIPTION : Spécifie les informations d'algorithme et de clés préconfigurées pour la création d'une association de sécurité IPsec en mode transport.
DÉRIVÉ DE : PreconfiguredSAAction
ABSTRAITE : FAUX

6.8 Classe PreconfiguredTunnelAction

La classe PreconfiguredTunnelAction est utilisée pour créer une association de sécurité IPsec en mode tunnel en utilisant les algorithmes et clés préconfigurés, incorporés. La définition de classe pour PreconfiguredSAAction est la suivante :

NOM : PreconfiguredTunnelAction
DESCRIPTION : Spécifie les informations d'algorithme et de clés préconfigurées pour la création d'une association de sécurité IPsec en mode tunnel.
DÉRIVÉ DE : PreconfiguredSAAction
ABSTRAITE : FAUX
PROPRIÉTÉS : DFHandling

6.8.1 Propriété DFHandling

La propriété DFHandling spécifie comment le bit Ne pas fragmenter (DF, *Don't Fragment*) de l'en-tête IP interne doit être traité durant le processus IPsec. La propriété est définie comme suit :

NOM : DFHandling
DESCRIPTION : Spécifie le traitement du bit DF.
SYNTAXE : entier de 16 bits non signé
VALEUR : 1 – Copie le bit DF de l'en-tête IP interne à l'en-tête IP externe.
2 – Règle le bit DF de l'en-tête IP externe à 1.
3 – Règle le bit DF de l'en-tête IP externe à 0.

6.9 Classe SANegotiationAction

La classe SANegotiationAction spécifie une action qui demande une négociation de politique de sécurité.

C'est une classe abstraite. Actuellement, une seule action de protocole de négociation de politique de sécurité figure dans une sous classe de SANegotiationAction, la classe IKENegotiationAction. On s'attend néanmoins que d'autres protocoles de négociation de politique de sécurité existent et que les actions de négociation de ces nouveaux protocoles seront modélisées comme sous classes de SANegotiationAction.

NOM : SANegotiationAction
DESCRIPTION : Spécifie une action de négociation.
DÉRIVÉ DE : SAAction
ABSTRAITE : VRAI

6.10 Classe IKENegotiationAction

La classe IKENegotiationAction est abstraite et sert de classe de base pour les actions IKE et IPsec qui résultent en une négociation IKE. La définition de classe pour IKENegotiationAction est la suivante :

NOM : IKENegotiationAction
DESCRIPTION : Classe de base pour les actions IKE et IPsec qui spécifient les paramètres qui sont communs pour les négociations de DOI IPsec IKE phase 1 et IKE phase 2.
DÉRIVÉ DE : SANegotiationAction
ABSTRAITE : VRAI
PROPRIÉTÉS : MinLifetimeSeconds ; MinLifetimeKilobytes ; IdleDurationSeconds

6.10.1 Propriété MinLifetimeSeconds

La propriété MinLifetimeSeconds spécifie le minimum de secondes qui sera accepté par l'homologue dans une durée de vie. MinLifetimeSeconds est utilisé pour empêcher certaines attaques de déni de service où l'homologue demande une valeur arbitrairement basse de durée de vie, causant des renégociations avec de coûteuses opérations Diffie-Hellman. La propriété est définie comme suit :

NOM : MinLifetimeSeconds
DESCRIPTION : Spécifie le minimum de secondes acceptable dans une durée de vie.
SYNTAXE : entier non signé de 64 bits
VALEUR : Une valeur de zéro indique qu'il n'y a pas de valeur minimum. Une valeur différente de zéro spécifie le minimum de secondes de la durée de vie.

Note : Bien que IKE puisse négocier la durée de vie comme un champ de longueur arbitraire, les auteurs ont estimé qu'un entier de 64 bits serait suffisant.

6.10.2 Propriété MinLifetimeKilobytes

La propriété MinLifetimeKilobytes spécifie le nombre minimum de kilo octets d'une durée de vie qui sera accepté provenant de l'homologue. MinLifetimeKilobytes est utilisé pour empêcher certaines attaques de déni de service, où l'homologue demande une valeur arbitrairement basse de durée de vie, causant des renégociations avec des opérations Diffie-Hellman coûteuses. Noter qu'il y a eu des débats considérables sur l'utilité d'appliquer des durées de vie en kilo octets aux associations de sécurité IKE phase 1, de sorte qu'il est probable que cette propriété ne s'appliquera qu'à la sous-classe IPsecAction. La propriété est définie comme suit :

NOM : MinLifetimeKilobytes
DESCRIPTION : Spécifie le minimum de kilo octets acceptable dans une durée de vie.
SYNTAXE : Entier non signé de 64 bits
VALEUR : Une valeur de zéro indique qu'il n'y a pas de valeur minimum. Une valeur différente de zéro spécifie le minimum de kilo octets de la durée de vie.

Note : Bien que IKE puisse négocier la durée de vie comme un champ de longueur arbitraire, les auteurs ont estimé qu'un entier de 64 bits serait suffisant.

6.10.3 Propriété IdleDurationSeconds

La propriété IdleDurationSeconds spécifie combien de secondes une association de sécurité peut rester inactive (c'est-à-dire, pas de trafic protégé en utilisant l'association de sécurité) avant qu'elle soit supprimée. La propriété est définie comme suit :

NOM : IdleDurationSeconds
DESCRIPTION : Spécifie en secondes combien de temps une association de sécurité peut rester non utilisée avant qu'elle soit supprimée.
SYNTAXE : entier non signé de 64 bits
VALEUR : Une valeur de zéro indique que la détection d'inactivité ne devrait pas être utilisée pour l'association de sécurité (seuls les secondes et les kilo octets de durée de vie seront utilisés). Toute valeur non zéro indique le nombre de secondes pendant lequel l'association de sécurité peut rester non utilisée.

6.11 Classe IPsecAction

La classe IPsecAction sert de classe de base pour les actions IPsec de transport et de tunnel. Elle spécifie les paramètres utilisés pour une négociation de DOI IPsec IKE phase 2. La définition de classe pour IPsecAction est la suivante :

NOM : IPsecAction
DESCRIPTION : Classe de base pour les actions IPsec de transport et de tunnel qui spécifient les paramètres pour les négociations de DOI IPsec IKE phase 2.
DÉRIVÉ DE : IKENegotiationAction
ABSTRAITE : VRAI
PROPRIÉTÉS : UsePFS ; UseIKEGroup ; GroupId ; Granularity ; VendorID

6.11.1 Propriété UsePFS

La propriété UsePFS spécifie si le secret parfait de transmission devrait ou non être utilisé lors du rafraîchissement des clés. La propriété est définie comme suit :

NOM : UsePFS
DESCRIPTION : Spécifie si on utilise ou non le PFS lors du rafraîchissement de clés.
SYNTAXE : Booléen
VALEUR : Une valeur de Vrai indique que PFS devrait être utilisé. Une valeur de Faux indique que PFS ne devrait pas être utilisé.

6.11.2 Propriété UseIKEGroup

La propriété UseIKEGroup spécifie si la phase 2 devrait ou non utiliser le même groupe d'échange de clés qu'utilisé dans la phase 1. UseIKEGroup est ignoré si UsePFS est faux. La propriété est définie comme suit :

NOM : UseIKEGroup
DESCRIPTION : Spécifie si on utilise ou non le même GroupId pour la phase 2 qu'à la phase 1. Si UsePFS est faux, UseIKEGroup est alors ignoré.
SYNTAXE : booléen
VALEUR : Une valeur de vrai indique que le GroupId de phase 2 devrait être le même que dans la phase 1. Une valeur de faux indique que la propriété GroupId va contenir le groupe d'échange de clé à utiliser pour la phase 2.

6.11.3 Propriété GroupId

La propriété GroupId spécifie le groupe d'échange de clés à utiliser pour la phase 2. GroupId est ignoré si (1) la propriété UsePFS est fausse, ou (2) si la propriété UsePFS est vraie et la propriété UseIKEGroup est vraie. Si le numéro de GroupID est tiré de la gamme spécifique du fabricant (32768-65535), la propriété VendorID qualifie le numéro de groupe. La propriété est définie comme suit :

NOM : GroupId
DESCRIPTION : Spécifie le groupe d'échange de clés à utiliser pour la phase 2 lorsque la propriété UsePFS est vraie et que la propriété UseIKEGroup est fausse.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Consulter les valeurs valides dans la [RFC2409].

6.11.4 Propriété Granularity

La propriété Granularity spécifie comment le sélecteur pour l'association de sécurité devrait être déduit du trafic qui a déclenché la négociation. La propriété est définie comme suit :

NOM : Granularity
DESCRIPTION : Spécifie comment le sélecteur proposé sera créé pour l'association de sécurité.
SYNTAXE : Entier de 16 bits non signé
VALEUR : 1 – sous réseau ; les gabarits de sous réseau de source et de destination de l'entrée de filtre sont utilisés.
2 – adresse ; seules les adresses IP de source et de destination du paquet déclencheur sont utilisées.
3 – protocole ; les adresses IP de source et de destination et le protocole IP du paquet déclencheur sont utilisées.
4 – accès ; les adresses IP de source et de destination et le protocole IP et les accès de couche 4 de source et de destination du paquet déclencheur sont utilisées.

6.11.5 Propriété VendorID

La propriété VendorID est utilisée avec la propriété GroupID (lorsque elle est dans la gamme spécifique du fabricant) pour identifier le groupe d'échange de clé. VendorID est ignoré sauf si UsePFS est vrai et UseIKEGroup est faux et si GroupID est dans la gamme spécifique de fabricant (32768-65535). La propriété est définie comme suit :

NOM : VendorID
DESCRIPTION : Spécifie l'identifiant du fabricant IKE.
SYNTAXE ; Chaîne

6.12 Classe IPsecTransportAction

La classe IPsecTransportAction est une sous classe de IPsecAction qui est utilisée pour spécifier l'utilisation d'une association de sécurité IPsec en mode transport. La définition de classe pour IPsecTransportAction est la suivante :

NOM : IPsecTransportAction
DESCRIPTION : Spécifie qu'une association de sécurité IPsec en mode transport devrait être négociée.
DÉRIVÉ DE : IPsecAction
ABSTRAITE : FAUX

6.13 Classe IPsecTunnelAction

La classe IPsecTunnelAction est une sous classe de IPsecAction qui est utilisée pour spécifier l'utilisation d'une association de sécurité IPsec en mode tunnel. La définition de classe pour IPsecTunnelAction est la suivante :

NOM : IPsecTunnelAction
DESCRIPTION : Spécifie qu'une association de sécurité IPsec en mode tunnel devrait être négociée.
DÉRIVÉ DE : IPsecAction
ABSTRAITE : FAUX
PROPRIÉTÉS : DFHandling

6.13.1 Propriété DFHandling

La propriété DFHandling spécifie comment le tunnel devrait gérer le bit Ne pas fragmenter (DF). La propriété est définie comme suit :

NOM : DFHandling
DESCRIPTION : Spécifie comment traiter le bit DF.
SYNTAXE : Entier de 16 bits non signé
VALEUR : 1 – Copier le bit DF de l'en-tête interne IP sur l'en-tête IP externe.
2 – Régler le bit DF de l'en-tête IP externe à 1.
3 – Régler le bit DF de l'en-tête IP externe à 0.

6.14 Classe IKEAction

La classe IKEAction spécifie les paramètres qui sont à utiliser pour la négociation IKE de phase 1. La définition de classe pour IKEAction est la suivante :

NOM : IKEAction
DESCRIPTION : Spécifie les paramètres de négociation IKE de phase 1.
DÉRIVÉ DE : IKENegotiationAction
ABSTRAITE : FAUX
PROPRIÉTÉS : ExchangeMode ; UseIKEIdentityType ; VendorID ; AggressiveModeGroupId

6.14.1 Propriété ExchangeMode

La propriété ExchangeMode spécifie quel mode IKE devrait être utilisé pour les négociations IKE de phase 1. La propriété est

définie comme suit :

NOM : ExchangeMode
DESCRIPTION : Spécifie le mode IKE de négociation pour la phase 1.
SYNTAXE : Entier de 16 bits non signé
VALEUR : 1 – mode de base
2 – mode principal
4 – mode agressif

6.14.2 Propriété UseIKEIdentityType

La propriété UseIKEIdentityType spécifie quel type d'identité IKE devrait être utilisé lors de la négociation avec l'homologue. Ces informations sont utilisée en conjonction avec les identités IKE disponibles sur le système et les IdentityContexts de la IKERule correspondante. La propriété est définie comme suit:

NOM : UseIKEIdentityType
DESCRIPTION : Spécifie le type d'identité IKE à utiliser durant la négociation.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Consulter les valeurs valides dans la [RFC2407].

6.14.3 Propriété VendorID

La propriété VendorID spécifie la valeur à utiliser dans l'identifiant de charge utile de fabricant. La propriété est définie comme suit :

NOM : VendorID
DESCRIPTION : Identifiant de charge utile de fabricant.
SYNTAXE : Chaîne
VALEUR : Une valeur de NUL signifie que cet identifiant de charge utile de fabricant ne sera ni généré ni accepté. Une valeur non NULLE signifie qu'un identifiant de charge utile de fabricant sera généré (en agissant comme initiateur) ou est attendue (en agissant comme répondeur).

6.14.4 Propriété AggressiveModeGroupId

La propriété AggressiveModeGroupId spécifie quel identifiant de groupe est à utiliser dans les premiers paquets de la négociation de phase 1. Cette propriété est ignorée sauf si la propriété ExchangeMode est réglée à 4 (mode agressif). Si le numéro de AggressiveModeGroupID est dans la gamme spécifique de fabricant (32768-65535), la propriété VendorID qualifie le numéro de groupe. La propriété est définie comme suit :

NOM : AggressiveModeGroupId
DESCRIPTION : Spécifie l'identifiant de groupe à utiliser pour le mode agressif
SYNTAXE : Entier de 16 bits non signé

6.15 Classe PeerGateway

La classe PeerGateway spécifie la passerelle de sécurité avec laquelle les services IKE négocient. La définition de classe pour PeerGateway est la suivante :

NOM : PeerGateway
DESCRIPTION : Spécifie la passerelle de sécurité avec laquelle négocier.
DÉRIVÉ DE : LogicalElement (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Name ; PeerIdentityType ; PeerIdentity

Note : La classe PeerIdentityEntry contient plus d'informations sur l'homologue (à savoir son adresse IP).

6.15.1 Propriété Name

La propriété Name spécifie un nom facile à retenir pour cette passerelle de sécurité. La propriété est définie comme suit :

NOM : Name
DESCRIPTION : Spécifie un nom facile à retenir pour cette passerelle de sécurité.
SYNTAXE : Chaîne

6.15.2 Propriété PeerIdentityType

La propriété PeerIdentityType spécifie le type d'identité IKE de la passerelle de sécurité. La propriété est définie comme suit :

NOM : PeerIdentityType
DESCRIPTION : Spécifie le type d'identité IKE de la passerelle de sécurité.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Consulter les valeurs valides dans la [RFC2407].

6.15.3 Propriété PeerIdentity

La propriété PeerIdentity spécifie la valeur d'identité IKE de la passerelle de sécurité. Sur la base de la mémorisation choisie pour la transposition spécifique de la tâche du modèle d'informations, une conversion peut être nécessaire à partir de la représentation mémorisée de la chaîne PeerIdentity en la valeur réelle utilisée dans l'identifiant de charge utile (par exemple, l'adresse IP est à convertir d'une chaîne en décimal séparé par des points en quatre octets). La propriété est définie comme suit :

NOM : PeerIdentity
DESCRIPTION : Spécifie la valeur d'identité IKE de la passerelle de sécurité.
SYNTAXE : Chaîne

6.16 Classe d'association PeerGatewayForTunnel

La classe PeerGatewayForTunnel associe des IPsecTunnelAction à une liste ordonnée de PeerGateway. La définition de classe pour PeerGatewayForTunnel est la suivante :

NOM : PeerGatewayForTunnel
DESCRIPTION : Associe des IPsecTunnelAction à une liste ordonnée de PeerGateway.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref PeerGateway[0..n]] ; Dependent [ref IPsecTunnelAction[0..n]] ; SequenceNumber

6.16.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de PeerGateway. La cardinalité [0..n] indique qu'une instance de IPsecTunnelAction peut être associée à zéro, un ou plusieurs instances de PeerGateway.

Note : La cardinalité 0 a une signification spécifique :
- Lorsque le service IKE agit comme répondeur, cela signifie que le service IKE va accepter la négociation de phase 1 avec toute autre passerelle de sécurité;
- Lorsque le service IKE agit comme initiateur, cela signifie que le service IKE va utiliser l'adresse IP de destination (des paquets IP ont déclenché la SARule) comme adresse IP de l'entité IKE homologue.

6.16.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de IPsecTunnelAction. La cardinalité [0..n] indique qu'une instance de PeerGateway peut être associée à zéro, une ou plusieurs instances de IPsecTunnelAction.

6.16.3 Propriété SequenceNumber

La propriété SequenceNumber spécifie l'ordre à utiliser pour évaluer les instances de PeerGateway pour une certaine IPsecTunnelAction. La propriété est définie comme suit :

NOM : SequenceNumber

DESCRIPTION : Spécifie l'ordre d'évaluation pour PeerGateways.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Les valeurs inférieures sont évaluées en premier.

6.17 Classe d'agrégation ContainedProposal

La classe ContainedProposal associe une liste ordonnée de SAProposal à la IKENegotiationAction qui l'agrège. Si l'objet IKENegotiationAction référencé est une IKEAction, le ou les objets SAProposal référencées doivent alors être des IKEProposal. Si l'objet IKENegotiationAction référencé est une IPsecTransportAction ou une IPsecTunnelAction, le ou les objets SAProposal proposés doivent alors être des IPsecProposal. La définition de classe pour ContainedProposal est la suivante :

NOM : ContainedProposal
DESCRIPTION : Associe une liste ordonnée de SAProposal à une IKENegotiationAction.
DÉRIVÉ DE : PolicyComponent (voir la [RFC3060])
ABSTRAITE : FAUX
PROPRIÉTÉS : GroupComponent[ref IKENegotiationAction[0..n]] ; PartComponent[ref SAProposal[1..n]] ; SequenceNumber

6.17.1 Référence GroupComponent

La propriété GroupComponent est héritée de PolicyComponent et est outrepassée pour se référer à une instance de IKENegotiationAction. La cardinalité [0..n] indique qu'une instance de SAProposal peut être associée à zéro, une ou plusieurs instances de IKENegotiationAction.

6.17.2 Référence PartComponent

La propriété PartComponent est héritée de PolicyComponent et est outrepassée pour se référer à une instance de SAProposal. La cardinalité [1..n] indique qu'une instance de IKENegotiationAction instance DOIT être associée à au moins une instance de SAProposal.

6.17.3 Propriété SequenceNumber

La propriété SequenceNumber spécifie l'ordre de préférence pour les SAProposal. La propriété est définie comme suit :

NOM : SequenceNumber
DESCRIPTION : Spécifie l'ordre de préférence pour les SAProposal.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Les propositions de valeur inférieure sont préférées aux propositions de valeur supérieure. Pour les ContainedProposal qui font référence à la même IKENegotiationAction, les valeurs de SequenceNumber doivent être uniques.

6.18 Classe d'association HostedPeerGatewayInformation

La classe HostedPeerGatewayInformation associe faiblement une PeerGateway à un System. La définition de classe pour HostedPeerGatewayInformation est la suivante :

NOM : HostedPeerGatewayInformation
DESCRIPTION :Associe faiblement une PeerGateway à un System.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref System[1..1]] ; Dependent [ref PeerGateway[0..n] [weak]]

6.18.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de System. La cardinalité [1..1] indique qu'une instance de PeerGateway DOIT être associée à une et seulement une instance de System.

6.18.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de PeerGateway. La cardinalité [0..n] indique qu'une instance de System peut être associée à zéro, une ou plusieurs instances de PeerGateway.

6.19 Classe d'association TransformOfPreconfiguredAction

La classe TransformOfPreconfiguredAction associe une PreconfiguredSAAction à deux, quatre ou six SATransform qui vont être appliquées au trafic entrant et sortant. L'ordre d'application de la SATransform est implicitement défini dans la [RFC2401]. La définition de classe pour TransformOfPreconfiguredAction est la suivante :

NOM : TransformOfPreconfiguredAction
DESCRIPTION : Associe une PreconfiguredSAAction à de une à trois SATransform.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent[ref SATransform[2..6]] ; Dependent[ref PreconfiguredSAAction[0..n]] ; SPI ; Direction

6.19.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de SATransform. La cardinalité [2..6] indique qu'une instance de PreconfiguredSAAction peut être associée à de deux à six instances de SATransform.

6.19.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de PreconfiguredSAAction. La cardinalité [0..n] indique qu'une instance de SATransform peut être associée à zéro, une ou plusieurs instances de PreconfiguredSAAction.

6.19.3 Propriété SPI

La propriété SPI spécifie le SPI à utiliser par l'action préconfigurée pour la transformation associée. La propriété est définie comme suit :

NOM : SPI
DESCRIPTION : Spécifie le SPI à utiliser avec la SATransform.
SYNTAXE : Entier non signé de 32 bits.

6.19.4 Propriété Direction

La propriété Direction spécifie si la propriété SPI est pour le trafic entrant ou sortant. La propriété est définie comme suit :

NOM : Direction
DESCRIPTION : Spécifie si la SA est pour le trafic entrant ou sortant.
SYNTAXE : Entier non signé de 8 bits.
VALEUR : 1 - cette SA est pour le trafic entrant
2 - cette SA est pour le trafic sortant

6.20 Classe d'association PeerGatewayForPreconfiguredTunnel

La classe PeerGatewayForPreconfiguredTunnel associe zéro ou une PeerGateway à plusieurs PreconfiguredTunnelAction. La définition de classe pour PeerGatewayForPreconfiguredTunnel est la suivante :

NOM : PeerGatewayForPreconfiguredTunnel
DESCRIPTION : Associe une PeerGateway à plusieurs PreconfiguredTunnelAction.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent[ref PeerGateway[0..1]] ; Dependent[ref PreconfiguredTunnelAction[0..n]]

6.20.1 Référence Antecedent

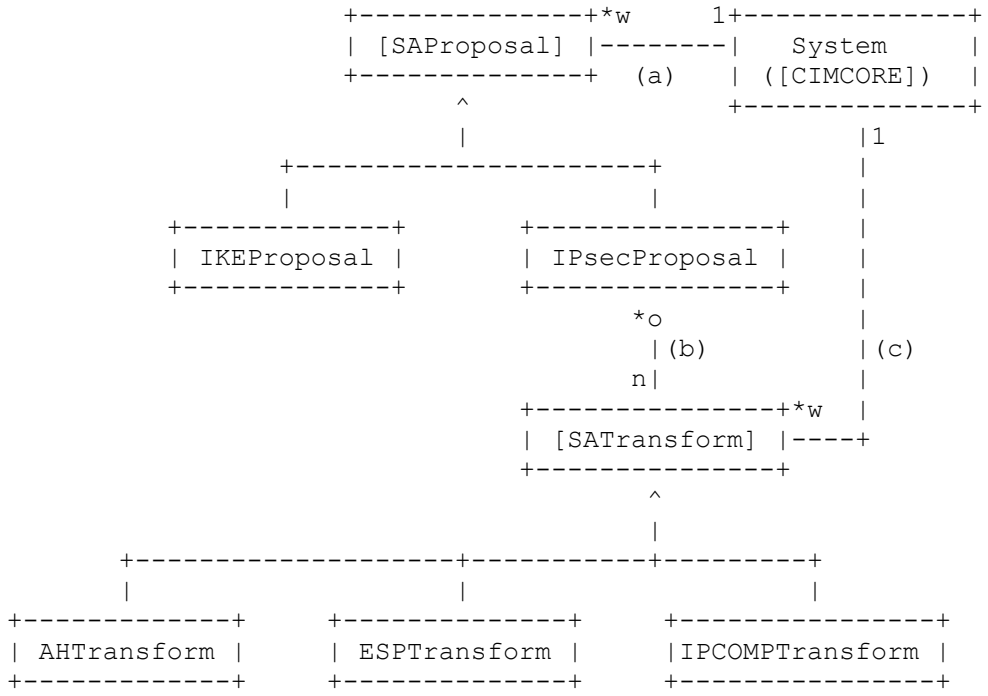
La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de PeerGateway. La cardinalité [0..1] indique qu'une instance de PreconfiguredTunnelAction peut être associée à une instance de PeerGteway.

6.20.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de PreconfiguredTunnelAction. La cardinalité [0..n] indique qu'une instance de PeerGateway peut être associée à zéro, une ou plusieurs instances de PreconfiguredSAAction.

7. Classes de proposition et de transformation

Les classes de proposition et de transformations modélisent les réglages de propositions qu'un appareil IPsec va utiliser durant les négociations IKE de phase 1 et 2.



(a) SAProposalInSystem
(b) ContainedTransform
(c) SATransformInSystem

7.1 Classe abstraite SAProposal

La classe abstraite SAProposal sert de classe de base pour les classes de propositions IKE et IPsec. Elle spécifie les paramètres qui sont communs aux deux types de propositions. La définition de classe pour SAProposal est la suivante :

NOM : SAProposal
DESCRIPTION : Spécifie les paramètres de proposition communs pour la négociation d'association de sécurité IKE et IPsec.
DÉRIVÉ DE : Policy ([RFC3060])
ABSTRAITE : VRAI
PROPRIÉTÉS : Name

7.1.1 Propriété Name

La propriété Name spécifie un nom facile à mémoriser pour SAProposal. La propriété est définie comme suit :

NOM : Name

DESCRIPTION : Spécifie un nom facile à mémoriser pour cette proposition.
SYNTAXE : Chaîne

7.2 Classe IKEProposal

La classe IKEProposal spécifie les paramètres de propositions nécessaires pour conduire une négociation d'association de sécurité IKE. La définition de classe pour IKEProposal est la suivante :

NOM : IKEProposal
DESCRIPTION : Spécifie les paramètres de propositions pour la négociation d'association de sécurité IKE.
DÉRIVÉ DE : SAProposal
ABSTRAITE : FAUX
PROPRIÉTÉS : CipherAlgorithm ; HashAlgorithm ; PRFAlgorithm ; GroupId ; AuthenticationMethod ; MaxLifetimeSeconds ; MaxLifetimeKilobytes ; VendorID

7.2.1 Propriété CipherAlgorithm

La propriété CipherAlgorithm spécifie l'algorithme de chiffrement proposé pour l'association de sécurité de phase 1. La propriété est définie comme suit :
NOM : CipherAlgorithm
DESCRIPTION : Spécifie l'algorithme de chiffrement proposé pour l'association de sécurité de phase 1.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Consulter les valeurs valides dans la [RFC2409].

7.2.2 Propriété HashAlgorithm

La propriété HashAlgorithm spécifie l'algorithme de hachage proposé pour l'association de sécurité de phase 1. La propriété est définie comme suit :

NOM : HashAlgorithm
DESCRIPTION : Spécifie l'algorithme de hachage proposé pour l'association de sécurité de phase 1.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Consulter les valeurs valides dans la [RFC2409].

7.2.3 Propriété PRFAlgorithm

La propriété PRFAlgorithm spécifie la fonction pseudo aléatoire proposée pour l'association de sécurité de phase 1. La propriété est définie comme suit :

NOM ;: PRFAlgorithm
DESCRIPTION : Spécifie la fonction pseudo aléatoire proposée pour l'association de sécurité de phase 1.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Aucune n'est actuellement définie dans la [RFC2409], si les [RFC2407] et [RFC2409] sont étendues, alors les valeurs des [RFC2407], [RFC2409] seront utilisées comme valeurs de PRFAlgorithm.

7.2.4 Propriété GroupId

La propriété GroupId spécifie le groupe d'échange de clés proposé pour l'association de sécurité de phase 1. Cette propriété est ignorée pour tous les échanges de mode agressif. Si le numéro de GroupID est dans la gamme spécifique de fabricant (32768-65535), la propriété VendorID qualifie le numéro de groupe. La propriété est définie comme suit :

NOM : GroupId
DESCRIPTION : Spécifie le groupe d'échange de clés proposé pour l'association de sécurité de phase 1.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Consulter les valeurs valides dans la [RFC2409].

Note : La valeur de cette propriété est à ignorer dans le mode agressif.

7.2.5 Propriété AuthenticationMethod

La propriété AuthenticationMethod spécifie la méthode proposée d'authentification de phase 1. La propriété est définie comme

suit :

NOM : AuthenticationMethod
DESCRIPTION : Spécifie la méthode proposée d'authentification pour la phase 1 d'association de sécurité.
SYNTAXE : Entier de 16 bits non signé
VALEUR : 0 est une valeur spéciale qui indique que cette proposition devrait être répétée une fois pour chaque méthode d'authentification qui correspond aux accréditifs installés sur la machine. Par exemple, si le système a une clé pré partagée et un certificat, une liste de propositions pourrait être construite incluant une proposition qui spécifierait une clé pré partagée et des propositions pour toutes les méthodes d'authentification de clé publique. Consulter les valeurs valides dans la [RFC2409].

7.2.6 Propriété MaxLifetimeSeconds

La propriété MaxLifetimeSeconds spécifie le temps maximum proposé, en secondes, pendant lequel une association de sécurité va rester valide après sa création. La propriété est définie comme suit :

NOM : MaxLifetimeSeconds
DESCRIPTION : Spécifie le temps maximum proposé pendant lequel une association de sécurité va rester valide.
SYNTAXE : Entier non signé de 64 bits
VALEUR : Une valeur de zéro indique l'utilisation de la valeur par défaut de 8 heures. Une valeur différente de zéro indique la durée de vie maximum en secondes.

Note : Bien que IKE puisse négocier la durée de vie comme un champ de longueur arbitraire, les auteurs ont estimé qu'un entier de 64 bits serait suffisant.

7.2.7 Propriété MaxLifetimeKilobytes

La propriété MaxLifetimeKilobytes spécifie la durée de vie maximum proposée en kilo octets pendant laquelle une association de sécurité va rester valide après sa création. La propriété est définie comme suit :

NOM : MaxLifetimeKilobytes
DESCRIPTION : Spécifie la durée de vie proposée maximum en kilo octets pendant laquelle une association de sécurité va rester valide.
SYNTAXE : Entier non signé de 64 bits
VALEUR : Une valeur de zéro indique qu'il ne devrait pas y avoir de durée de vie maximum en kilo octets. Une valeur non à zéro spécifie la durée de vie désirée en kilo octets.

Note : Bien que IKE puisse négocier la durée de vie comme un champ de longueur arbitraire, les auteurs ont estimé qu'un entier de 64 bits serait suffisant.

7.2.8 Propriété VendorID

La propriété VendorID qualifie mieux le groupe d'échange de clés. La propriété est ignorée sauf si l'échange n'est pas en mode agressif et si la propriété GroupID est dans la gamme spécifique du fabricant. La propriété est définie comme suit :

NOM : VendorID
DESCRIPTION : Spécifie l'identifiant de fabricant pour mieux qualifier le groupe d'échange de clé.
SYNTAXE : Chaîne

7.3 Classe IPsecProposal

La classe IPsecProposal n'ajoute pas de nouvelle propriété, mais hérite des propriétés de proposition de SAProposal et agrège les transformations d'association de sécurité nécessaires pour construire une proposition IPsec (voir la classe d'agrégation ContainedTransform). La définition de classe pour IPsecProposal est la suivante :

NOM : IPsecProposal
DESCRIPTION : Spécifie les paramètres proposés pour la négociation d'association de sécurité IPsec.
DÉRIVÉ DE : SAProposal
ABSTRAITE : FAUX

7.4 Classe abstraite SATransform

La classe abstraite SATransform sert de classe de base pour les transformations IPsec qui peuvent être utilisées pour composer une proposition IPsec ou comme action pré configurée. La définition de classe pour SATransform est la suivante :

NOM : SATransform
DESCRIPTION : Classe de base pour les différentes transformations IPsec.
ABSTRAITE : VRAI
PROPRIÉTÉS : CommonName (de Policy) ; VendorID ; MaxLifetimeSeconds ; MaxLifetimeKilobytes

7.4.1 Propriété CommonName

La propriété CommonName est héritée de Policy [RFC3060] et spécifie un nom facile à mémoriser pour SATransform. La propriété est définie comme suit :

NOM : CommonName
DESCRIPTION : Spécifie un nom facile à retenir pour cet objet en rapport avec la politique.
SYNTAXE : Chaîne

7.4.2 Propriété VendorID

La propriété VendorID spécifie l'identifiant de fabricant pour les transformations définies par le fabricant. La propriété est définie comme suit :

NOM : VendorID
DESCRIPTION : Spécifie l'identifiant de fabricant pour les transformations définies par le fabricant.
SYNTAXE : Chaîne
VALEUR : Une chaîne VendorID vide indique que la transformation est standard.

7.4.3 Propriété MaxLifetimeSeconds

La propriété MaxLifetimeSeconds spécifie la durée maximum proposée, en secondes, pendant laquelle une association de sécurité va rester valide après sa création. La propriété est définie comme suit :

NOM : MaxLifetimeSeconds
DESCRIPTION : Spécifie la durée maximum proposée pendant laquelle une association de sécurité va rester valide.
SYNTAXE : Entier non signé de 64 bits
VALEUR :Une valeur de zéro indique que par défaut on utilise 8 heures. Une valeur non à zéro indique la durée de vie maximum en secondes.

Note : Bien que IKE puisse négocier la durée de vie comme un champ de longueur arbitraire, les auteurs ont estimé qu'un entier de 64 bits serait suffisant.

7.4.4 Propriété MaxLifetimeKilobytes

La propriété MaxLifetimeKilobytes spécifie la durée de vie proposée maximum en kilo octets pendant laquelle une association de sécurité va rester valide après sa création. La propriété est définie comme suit :

NOM : MaxLifetimeKilobytes
DESCRIPTION : Spécifie la durée de vie maximum proposée en kilo octets pendant laquelle une association de sécurité va rester valide.
SYNTAXE : Entier non signé de 64 bits
VALEUR : Une valeur de zéro indique qu'il ne devrait pas y avoir de durée de vie maximum en kilo octets. Une valeur non à zéro spécifie la durée de vie désirée en kilo octets.

Note : Bien que IKE puisse négocier la durée de vie comme un champ de longueur arbitraire, les auteurs ont estimé qu'un entier de 64 bits serait suffisant.

7.5 Classe AHTransform

La classe AHTransform spécifie l'algorithme AH à proposer durant la négociation IPsec d'association de sécurité. La définition de classe pour AHTransform est la suivante :

NOM : AHTransform
DESCRIPTION : Spécifie l'algorithme AH proposé.
ABSTRAITE : FAUX
PROPRIÉTÉS : AHTransformId ; UseReplayPrevention ; ReplayPreventionWindowSize

7.5.1 Propriété AHTransformId

La propriété AHTransformId spécifie l'identifiant de transformation de l'algorithme AH. La propriété est définie comme suit :

NOM : AHTransformId
DESCRIPTION : Spécifie l'identifiant de transformation de l'algorithme AH.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Consulter les valeurs valides dans la [RFC2407].

7.5.2 Propriété UseReplayPrevention

La propriété UseReplayPrevention spécifie si on doit utiliser la détection de prévention de répétition. La propriété est définie comme suit :

NOM : UseReplayPrevention
DESCRIPTION : Spécifie si il faut activer la détection de prévention de répétition.
SYNTAXE : Booléen
VALEUR : vrai – la détection de prévention de répétition est activée.
faux – la détection de prévention de répétition est désactivée.

7.5.3 Propriété ReplayPreventionWindowSize

La propriété ReplayPreventionWindowSize spécifie, en bits, la longueur de la fenêtre glissante utilisée par le mécanisme de détection de prévention de la répétition. La valeur de cette propriété n'a pas de signification si UseReplayPrevention est faux. On suppose que la taille de la fenêtre sera une puissance de 2. La propriété est définie comme suit :

NOM : ReplayPreventionWindowSize
DESCRIPTION : Spécifie la longueur de la fenêtre utilisée par le mécanisme de détection de prévention de la répétition.
SYNTAXE : Entier non signé de 32 bits

7.6 Classe ESPTransform

La classe ESPTransform spécifie les algorithmes ESP à proposer durant la négociation d'association de sécurité IPsec. La définition de classe pour ESPTransform est la suivante :

NOM : ESPTransform
DESCRIPTION : Spécifie les algorithmes ESP proposés.
ABSTRAITE : FAUX
PROPRIÉTÉS : IntegrityTransformId ; CipherTransformId ; CipherKeyLength ; CipherKeyRounds ; UseReplayPrevention ; ReplayPreventionWindowSize

7.6.1 Propriété IntegrityTransformId

La propriété IntegrityTransformId spécifie l'identifiant de transformation de l'algorithme d'intégrité ESP. La propriété est définie comme suit :

NOM : IntegrityTransformId
DESCRIPTION : Spécifie l'identifiant de transformation de l'algorithme d'intégrité ESP.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Consulter les valeurs valides dans la [RFC2407].

7.6.2 Propriété CipherTransformId

La propriété CipherTransformId spécifie l'identifiant de transformation de l'algorithme de chiffrement ESP. La propriété est définie comme suit :

NOM : CipherTransformId
DESCRIPTION : Spécifie l'identifiant de transformation de l'algorithme de chiffrement ESP.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Consulter les valeurs valides dans la [RFC2407].

7.6.3 Propriété CipherKeyLength

La propriété CipherKeyLength spécifie, en bits, la longueur de clé pour l'algorithme ESP de chiffrement. Pour les algorithmes de chiffrement qui utilisent des clés de longueur fixe, cette valeur est ignorée. La propriété est définie comme suit :

NOM : CipherKeyLength
DESCRIPTION : Spécifie la longueur de la clé de chiffrement ESP en bits.
SYNTAXE : Entier de 16 bits non signé

7.6.4 Propriété CipherKeyRounds

La propriété CipherKeyRounds spécifie le nombre de tours de clés pour l'algorithme de chiffrement ESP. Pour les algorithmes de chiffrement qui utilisent un nombre fixe de tours de clés, cette valeur est ignorée. La propriété est définie comme suit :

NOM : CipherKeyRounds
DESCRIPTION : Spécifie le nombre de tours de clés pour l'algorithme de chiffrement ESP.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Actuellement aucun tour de clé n'est défini pour des algorithmes de chiffrement ESP.

7.6.5 Propriété UseReplayPrevention

La propriété UseReplayPrevention spécifie si la détection de prévention de la répétition est à utiliser. La propriété est définie comme suit :

NOM : UseReplayPrevention
DESCRIPTION : Spécifie si on active la détection de prévention de répétition
SYNTAXE : booléen
VALEUR : vrai – la détection de prévention de répétition est activée.
faux – la détection de prévention de répétition est désactivée.

7.6.6 Propriété ReplayPreventionWindowSize

La propriété ReplayPreventionWindowSize spécifie, en bits, la longueur de la fenêtre glissante utilisée par le mécanisme de détection de prévention de répétition. La valeur de cette propriété n'a pas de signification si UseReplayPrevention est faux. On suppose que la taille de fenêtre sera une puissance de 2. La propriété est définie comme suit :

NOM : ReplayPreventionWindowSize
DESCRIPTION : Spécifie la longueur de la fenêtre utilisée par le mécanisme de détection de la prévention de répétition.
SYNTAXE : Entier non signé de 32 bits

7.7 Classe IPCOMPTransform

La classe IPCOMPTransform spécifie l'algorithme de compression IP (IPCOMP) à proposer durant la négociation d'association de sécurité IPsec. La définition de classe pour IPCOMPTransform est la suivante :

NOM : IPCOMPTransform
DESCRIPTION : Spécifie l'algorithme IPCOMP proposé .
ABSTRAITE : FAUX
PROPRIÉTÉS : Algorithm ; DictionarySize ; PrivateAlgorithm

7.7.1 Propriété Algorithm

La propriété Algorithm spécifie l'identifiant de transformation de l'algorithme de compression IPCOMP. La propriété est définie comme suit :

NOM : Algorithm
DESCRIPTION : Spécifie l'identifiant de transformation de l'algorithme de compression IPCOMP.
SYNTAXE : Entier de 16 bits non signé
VALEUR : 1 – OUI, un algorithme spécifique du fabricant est utilisé et spécifié dans la propriété PrivateAlgorithm. Consulter les autres valeurs valides dans la [RFC2407].

7.7.2 Propriété DictionarySize

La propriété DictionarySize spécifie le log2 de la taille maximum du dictionnaire pour l'algorithme de compression. Pour les algorithmes de compression qui ont une taille de dictionnaire prédéfinie, cette valeur est ignorée. La propriété est définie comme suit :

NOM : DictionarySize
DESCRIPTION : Spécifie le log2 de la taille maximum du dictionnaire.
SYNTAXE : Entier de 16 bits non signé

7.7.3 Propriété PrivateAlgorithm

La propriété PrivateAlgorithm spécifie un algorithme de compression spécifique de fabricant privé. Cette valeur n'est utilisée que lorsque la propriété Algorithm est 1 (OUI). La propriété est définie comme suit :

NOM : PrivateAlgorithm
DESCRIPTION : Spécifie un algorithme de compression spécifique de fabricant privé.
SYNTAXE : Entier non signé de 32 bits

7.8 Classe d'association SAProposalInSystem

La classe SAProposalInSystem associe faiblement les SAProposal à un System. La définition de classe pour SAProposalInSystem est la suivante :

NOM : SAProposalInSystem
DESCRIPTION : Associe faiblement les SAProposal à un System.
DÉRIVÉ DE : PolicyInSystem (voir [RFC3060])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent[ref System [1..1]] ; Dependent[ref SAProposal[0..n] [weak]]

7.8.1 Référence Antecedent

La propriété Antecedent est héritée de PolicyInSystem et est outrepassée pour se référer à une instance de System. La cardinalité [1..1] indique qu'une instance de SAProposal DOIT être associée à une instance de System et une seule.

7.8.2 Référence Dependent

La propriété Dependent est héritée de PolicyInSystem et est outrepassée pour se référer à une instance de SAProposal. La cardinalité [0..n] indique qu'une instance de System peut être associée à zéro, une ou plusieurs instances de SAProposal.

7.9 Classe d'agrégation ContainedTransform

La classe ContainedTransform associe une IPsecProposal à l'ensemble des SATransform qui constitue la proposition. Si plusieurs transformations du même type sont dans une proposition, elles doivent alors être composées avec l'opérateur logique OU et l'ordre de préférence est dicté par la propriété SequenceNumber. Les ensembles de transformations de types différents sont composées avec l'opérateur logique ET. Par exemple, si la liste ordonnée des propositions est :

ESP = { (HMAC-MD5, 3DES), (HMAC-MD5, DES) }
AH = { MD5, SHA-1 }

celui qui envoie la proposition va vouloir que l'autre côté prenne une des transformations de la liste ESP (de préférence (HMAC-MD5, 3DES)) ET une de la liste de transformations AH (de préférence MD5).

La définition de classe pour ContainedTransform est la suivante :

NOM : ContainedTransform
DESCRIPTION : Associe une IPsecProposal à l'ensemble des SATransform qui constitue la proposition.
DÉRIVÉ DE : PolicyComponent (voir [RFC3060])
ABSTRAITE : FAUX
PROPRIÉTÉS : GroupComponent[ref IPsecProposal[0..n]] ; PartComponent[ref SATransform[1..n]] ; SequenceNumber

7.9.1 Référence GroupComponent

La propriété GroupComponent est héritée de PolicyComponent et est outrepassée pour se référer à une instance de IPsecProposal. La cardinalité [0..n] indique qu'une instance de SATransform peut être associée à zéro, une ou plusieurs instances de IPsecProposal.

7.9.2 Référence PartComponent

La propriété PartComponent est héritée de PolicyComponent et est outrepassée pour se référer à une instance de SATransform. La cardinalité [1..n] indique qu'une instance de IPsecProposal DOIT être associée à au moins une instance SATransform.

7.9.3 Propriété SequenceNumber

La propriété SequenceNumber spécifie l'ordre de préférence pour les SATransform du même type. La propriété est définie comme suit :

NOM : SequenceNumber
DESCRIPTION : Spécifie l'ordre de préférence pour les SATransform du même type.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Les transformations de valeur inférieure sont préférées aux transformations du même type avec des valeurs supérieures. Pour les ContainedTransform qui se réfèrent à la même IPsecProposal, les valeurs de SequenceNumber doivent être uniques.

7.10 Classe d'association SATransformInSystem

La classe SATransformInSystem associe faiblement les SATransform à un System. La définition de classe pour SATransformInSystem est la suivante :

NOM : SATransformInSystem
DESCRIPTION : Associe faiblement les SATransforms à un System.
DÉRIVÉ DE : PolicyInSystem (voir [RFC3060])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent[ref System[1..1]] ; Dependent[ref SATransform[0..n] [weak]]

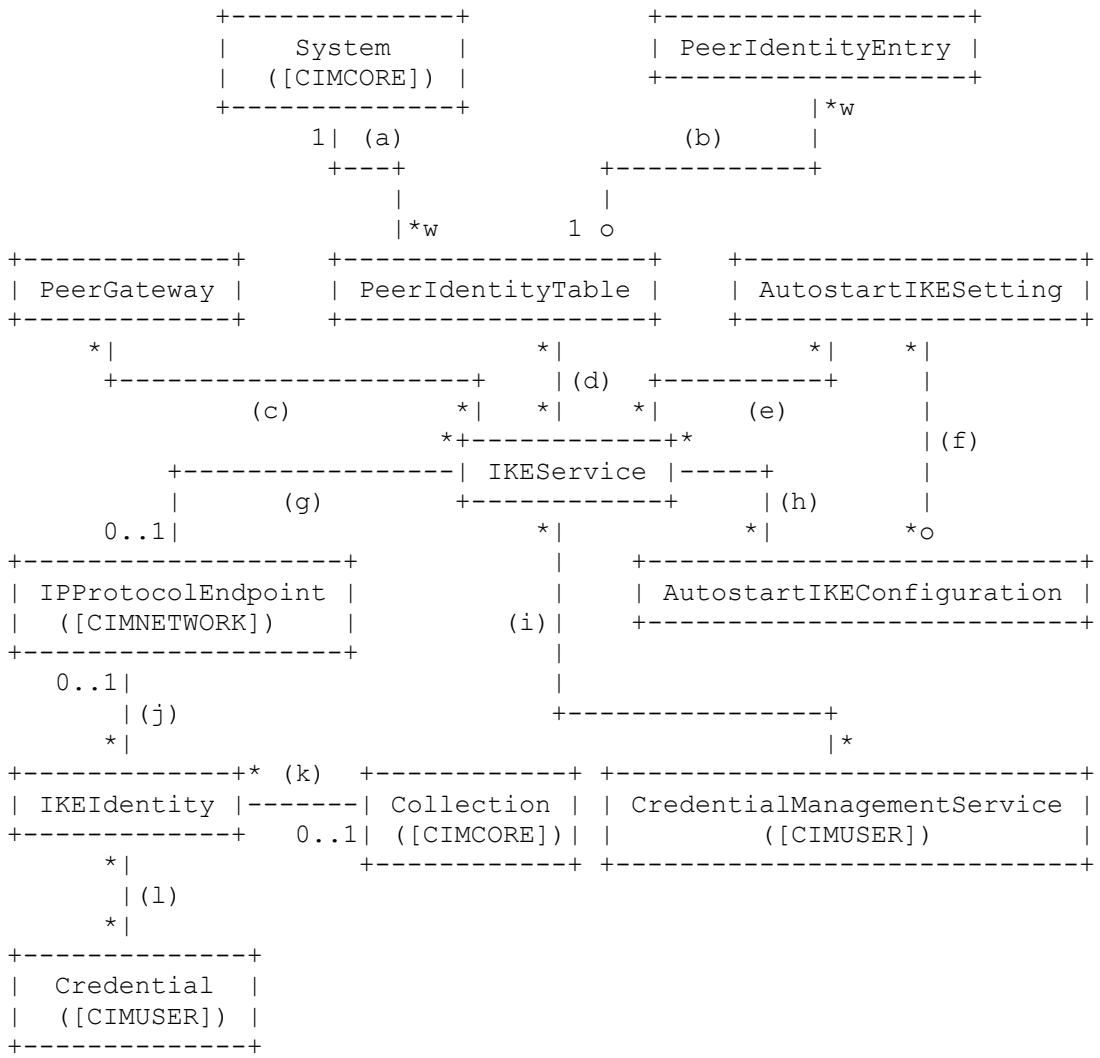
7.10.1 Référence Antecedent

La propriété Antecedent est héritée de PolicyInSystem et est outrepassée pour se référer à une instance de System. La cardinalité [1..1] indique qu'une instance de SATransform DOIT être associée à une seule instance de System.

7.10.2 Référence Dependent

La propriété Dependent est héritée de PolicyInSystem et est outrepassée pour se référer à une instance de SATransform. La cardinalité [0..n] indique qu'une instance de System peut être associée à zéro, une ou plusieurs instances de SATransform.

8. Classes de service et d'identité IKE



(a) HostedPeerIdentityTable
(b) PeerIdentityMember
(c) IKEServicePeerGateway
(d) IKEServicePeerIdentityTable
(e) IKEAutostartSetting
(f) AutostartIKESettingContext
(g) IKEServiceForEndpoint
(h) IKEAutostartConfiguration
(i) IKEUsesCredentialManagementService
(j) EndpointHasLocalIKEIdentity
(k) CollectionHasLocalIKEIdentity
(l) IKEIdentitysCredential

Cette portion du modèle contient des informations supplémentaires qui sont utiles pour appliquer la politique. La classe IKEService PEUT être utilisée pour représenter la fonction de négociation IKE dans un système. Le IKEService utilise les divers tableaux qui contiennent les informations sur les homologues IKE ainsi que la configuration pour spécifier les associations de sécurité qui sont démarrées automatiquement. Les informations de PeerGateway, PeerIdentityTable et des classes qui s'y rapportent sont nécessaires pour spécifier complètement les politiques.

Une interface (représentée par un IPProtocolEndpoint) a un IKEService qui fournit les services de négociation pour cette interface. Ce service PEUT aussi avoir une liste d'associations de sécurité démarrées automatiquement au moment de l'initialisation du service IKE.

IKEService a aussi un ensemble d'identités qu'il peut utiliser dans les négociations avec ses homologues. Ces identités sont associées aux interfaces (ou collections d'interfaces).

8.1 Classe IKEService

La classe IKEService représente la fonction de négociation IKE. Une instance de ce service peut fournir ce service de négociation pour une ou plusieurs interfaces (représentées par la classe IPProtocolEndpoint) d'un System. Il peut y avoir plusieurs instances de services IKE sur un système mais seulement une par interface. La définition de classe pour IKEService est la suivante :

NOM : IKEService
DESCRIPTION : IKEService est utilisé pour représenter la fonction de négociation IKE.
DÉRIVÉ DE : Service (voir [CIMCORE])
ABSTRAITE : FAUX

8.2 Classe PeerIdentityTable

La classe PeerIdentityTable agrège les entrées du tableau qui fournissent les transpositions entre les identités et leurs adresses. La définition de classe pour PeerIdentityTable est la suivante :

NOM : PeerIdentityTable
DESCRIPTION : PeerIdentityTable agrège les instances de PeerIdentityEntry pour fournir un tableau des transpositions identité-adresse.
DÉRIVÉ DE : Collection (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Name

8.2.1 Propriété Name

La propriété Name identifie le tableau de façon univoque. La propriété est définie comme suit :

NOM : Name
DESCRIPTION : Name identifie le tableau de façon univoque.
SYNTAXE : Chaîne

8.3 Classe PeerIdentityEntry

La classe PeerIdentityEntry spécifie la transposition entre l'identité de l'homologue et son adresse IP. La définition de classe pour PeerIdentityEntry est la suivante :

NOM : PeerIdentityEntry
DESCRIPTION : PeerIdentityEntry donne la transposition entre l'identité d'un homologue et son adresse.
DÉRIVÉ DE : LogicalElement (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : PeerIdentity ; PeerIdentityType ; PeerAddress ; PeerAddressType

La clé pré-partagée à utiliser avec cet homologue (si applicable) est contenue dans une instance de la classe SharedSecret (voir [CIMUSER]). La clé pré-partagée est mémorisée dans la propriété Secret, le protocole de la propriété contient "IKE", l'algorithme de la propriété contient l'algorithme utilisé pour protéger le secret (il peut être "PLAINTEXT" si l'entité IPsec n'a pas mémorisé de secret) la valeur de la propriété RemoteID doit correspondre à la propriété PeerIdentity de l'instance de PeerIdentityEntry qui décrit l'homologue IKE.

8.3.1 Propriété PeerIdentity

La propriété PeerIdentity contient une chaîne codant la charge utile Identity pour l'homologue IKE. La propriété est définie comme suit :

NOM : PeerIdentity
DESCRIPTION : PeerIdentity est l'identifiant de charge utile d'un homologue.
SYNTAXE : Chaîne

8.3.2 Propriété PeerIdentityType

La propriété PeerIdentityType est une énumération qui spécifie le type de PeerIdentity. La propriété est définie comme suit :

NOM : PeerIdentityType
DESCRIPTION : PeerIdentityType est le type de l'identifiant de charge utile d'un homologue.
SYNTAXE : Entier de 16 bits non signé
VALEUR : Les valeurs de l'énumération sont spécifiées dans la [RFC2407] paragraphe 4.6.2.1.

8.3.3 Propriété PeerAddress

La propriété PeerAddress spécifie la représentation de chaîne de l'adresse IP de l'homologue formatée selon la convention appropriée définie dans la propriété PeerAddressType (par exemple, notation en décimal séparé par des points). La propriété est définie comme suit :

NOM : PeerAddress
DESCRIPTION : PeerAddress est l'adresse de l'homologue avec l'identifiant de charge utile.
SYNTAXE : Chaîne
VALEUR : Représentation de chaîne d'une adresse IPv4 ou IPv6.

8.3.4 Propriété PeerAddressType

La propriété PeerAddressType spécifie le format de la valeur de la propriété PeerAddress. La propriété est définie comme suit :

NOM : PeerAddressType
DESCRIPTION : PeerAddressType est le type de l'adresse dans PeerAddress.
SYNTAXE : Entier de 16 bits non signé
VALEUR : 0 - Inconnu
1 - IPv4
2 - IPv6

8.4 Classe AutostartIKEConfiguration

La classe AutostartIKEConfiguration groupe des instances de AutostartIKESetting en ensembles de configuration. Lorsque appliqués, les réglages causent le démarrage automatique par un service IKE (négocié ou établi de façon statique comme approprié) des associations de sécurité. La définition de classe pour AutostartIKEConfiguration est la suivante :

NOM : AutostartIKEConfiguration
DESCRIPTION : Un ensemble de configuration d'instances de AutostartIKESetting à lancer automatiquement par le service IKE.
DÉRIVÉ DE : SystemConfiguration (voir [CIMCORE])
ABSTRAITE : FAUX

8.5 Classe AutostartIKESetting

La classe AutostartIKESetting est utilisée pour initier automatiquement les négociations IKE avec les homologues (ou créer statiquement une SA) comme spécifié dans les propriétés AutostartIKESetting. Les actions appropriées sont initiées conformément à la politique qui correspond aux paramètres du réglage. La définition de classe pour AutostartIKESetting est la suivante :

NOM : AutostartIKESetting

DESCRIPTION : AutostartIKESetting est utilisé pour initier automatiquement les négociations IKE avec les homologues ou créer statiquement une SA.
DÉRIVÉ DE : SystemSetting (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Phase1Only ; AddressType ; SourceAddress ; SourcePort ; DestinationAddress ; DestinationPort ; Protocol

8.5.1 Propriété Phase1Only

La propriété Phase1Only est utilisée pour limiter la négociation IKE à un établissement de SA de phase 1. Lorsque elle est réglé à Faux, les deux SA de phase 1 et de phase 2 sont négociées. La propriété est définie comme suit :

NOM : Phase1Only
DESCRIPTION : Utilisé pour indiquer si l'établissement d'une association de sécurité seulement de phase 1 ou si les deux phases 1 et 2 devraient être tentés.
SYNTAXE : Booléen
VALEUR : vrai – tenter d'établir une association de sécurité de phase 1
faux – tenter d'établir des associations de sécurité de phase 1 et phase 2

8.5.2 Propriété AddressType

La propriété AddressType spécifie un type des adresses dans les propriétés SourceAddress et DestinationAddress. La propriété est définie comme suit :

NOM : AddressType
DESCRIPTION : AddressType est le type d'adresse dans les propriétés SourceAddress et DestinationAddress.
SYNTAXE : Entier de 16 bits non signé
VALEUR : 0 - Inconnu
1 - IPv4
2 - IPv6

8.5.3 Propriété SourceAddress

La propriété SourceAddress spécifie l'adresse IP formatée en décimal séparé par des points ou par des deux-points utilisée comme adresse de source pour comparer aux entrées de filtre de politique et utilisée dans toutes négociations de phase 2. La propriété est définie comme suit :

NOM : SourceAddress
DESCRIPTION : Adresse de source à comparer aux filtres pour déterminer la règle de politique approprié.
SYNTAX E : Chaîne
VALEUR : Adresse IP formatée en décimal séparé par des points ou par des deux-points.

8.5.4 Propriété SourcePort

La propriété SourcePort spécifie le numéro d'accès utilisé comme accès de source pour comparer les entrées de filtre de politique et est utilisée dans toutes les négociations de phase 2. La propriété est définie comme suit :

NOM : SourcePort
DESCRIPTION : Accès de source à comparer aux filtres pour déterminer la règle de politique appropriée.
SYNTAXE : Entier de 16 bits non signé

8.5.5 Propriété DestinationAddress

La propriété DestinationAddress spécifie l'adresse IP formatée en décimal séparé par des points ou par des deux-points utilisée comme adresse de destination dans la comparaison des entrées de filtre de politique et est utilisée dans toutes les négociations de phase 2. La propriété est définie comme suit :

NOM : DestinationAddress
DESCRIPTION : Adresse de destination à comparer aux filtres pour déterminer la règle de politique appropriée.
SYNTAXE : Chaîne
VALEUR : Adresse IP formatée en décimal séparé par des points ou par des deux-points.

8.5.6 Propriété DestinationPort

La propriété DestinationPort spécifie le numéro d'accès utilisé comme accès de destination pour comparer les entrées de filtre de politique et est utilisé dans toutes les négociations de phase 2. La propriété est définie comme suit :

NOM : DestinationPort
DESCRIPTION : Accès de destination à comparer aux filtres pour déterminer la règle de politique appropriée.
SYNTAXE : Entier de 16 bits non signé.

8.5.7 Propriété Protocol

La propriété Protocol spécifie le numéro de protocole utilisé pour la comparaison aux entrées de filtre de politique et est utilisée dans toutes les négociations de phase 2. La propriété est définie comme suit :

NOM : Protocol
DESCRIPTION : Numéro de protocole utilisé pour la comparaison avec les entrées de filtre de politique.
SYNTAXE : Entier non signé de 8 bits.

8.6 Classe IKEIdentity

La classe IKEIdentity est utilisée pour représenter les identités qui peuvent être utilisées pour un IPProtocolEndpoint (ou une collection de IPProtocolEndpoint) pour identifier le service IKE dans les négociations IKE de phase 1. La politique IKEAction.UseIKEIdentityType spécifie quel type des identités disponibles utiliser dans un échange de négociation et la IKERule.IdentityContexts spécifie les valeurs de correspondance à utiliser, ainsi que l'adresse locale, pour choisir l'identité appropriée pour une négociation. La valeur de la propriété ElementID (définie dans une classe parente, UsersAccess) devrait être soit IPProtocolEndpoint, soit Collection de points d'extrémité, comme approprié. La définition de classe pour IKEIdentity est la suivante :

NOM : IKEIdentity
DESCRIPTION : IKEIdentity est utilisé pour représenter les identités qui peuvent être utilisées pour un IPProtocolEndpoint (ou collection de IPProtocolEndpoint) pour identifier le service IKE dans les négociations IKE de phase 1.
DÉRIVÉ DE : UsersAccess (voir [CIMUSER])
ABSTRAITE : FAUX
PROPRIÉTÉS : IdentityType ; IdentityValue ; IdentityContexts

8.6.1 Propriété IdentityType

La propriété IdentityType est une énumération qui spécifie le type de IdentityValue. La propriété est définie comme suit :

NOM : IdentityType
DESCRIPTION : IdentityType est le type de IdentityValue.
SYNTAXE : Entier de 16 bits non signé
VALEUR : L'énumération des valeurs est spécifiée dans la [RFC2407] paragraphe 4.6.2.1.

8.6.2 Propriété IdentityValue

La propriété IdentityValue contient une chaîne codant la charge utile de l'identité. Pour les instances de IKEIdentity qui sont des types d'adresses (c'est-à-dire, des adresses IPv4 ou IPv6) la valeur de la chaîne IdentityValue PEUT être omise ; le IPProtocolEndpoint associé (ou le membre approprié de la collection de points d'extrémité) est alors utilisé comme la valeur de l'identité. La propriété est définie comme suit :

NOM : IdentityValue
DESCRIPTION : IdentityValue contient une chaîne codant la charge utile de Identity.
SYNTAXE : Chaîne

8.6.3 Propriété IdentityContexts

La propriété IdentityContexts est utilisée pour contraindre l'utilisation des instances de IKEIdentity pour correspondre à celles spécifiées dans les IKERule.IdentityContexts. Les IdentityContexts sont formatés comme des rôles de politique et des combinaisons de rôles [RFC3060] & [RFC3460]. Chaque valeur représente un contexte ou une combinaison de contextes. Comme c'est une propriété multi-valeurs, plus d'un contexte ou combinaisons de contextes peuvent être associés à une seule

IKEIdentity. Chaque valeur est une chaîne de la forme :

<ContextName>[&&<ContextName>]*

où les noms des contextes individuels apparaissent en ordre alphabétique (selon la séquence de collationnement pour UCS-2). Si une ou plusieurs valeurs dans la matrice des IKERule.IdentityContexts correspond à un ou plusieurs IKEIdentity.IdentityContexts, le contexte de l'identité correspond. (C'est-à-dire, chaque valeur de la matrice de IdentityContext est une condition OUixée.) En combinaison avec l'adresse du IPProtocolEndpoint et du IKEAction.UseIKEIdentityType, il DEVRAIT y avoir exactement une IKEIdentity. La propriété est définie comme suit :

NOM : IdentityContexts
DESCRIPTION : Le service IKE d'un point d'extrémité de sécurité peut avoir plusieurs identités à utiliser dans différentes situations. La combinaison de l'interface (représentée par le IPProtocolEndpoint) le type d'identité (comme spécifié dans la IKEAction) et le IdentityContexts sélectionne une identité unique.
SYNTAXE : Disposition de chaîne
VALEUR : Chaîne de la forme <ContextName>[&&<ContextName>]*

8.7 Classe d'association HostedPeerIdentityTable

La classe HostedPeerIdentityTable fournit les relations de portée de noms pour les entrées de PeerIdentityTable dans un System. La PeerIdentityTable est faible pour le System. La définition de classe pour HostedPeerIdentityTable est la suivante :

NOM : HostedPeerIdentityTable
DESCRIPTION : Les instances de PeerIdentityTable sont faibles (la portée du nom est fournie par le système propriétaire).
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref System[1..1]] ; Dependent [ref PeerIdentityTable[0..n] [weak]]

8.7.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de System. La cardinalité [1..1] indique qu'une instance de PeerIdentityTable DOIT être associée dans une relation faible avec une seule instance de System.

8.7.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de PeerIdentityTable. La cardinalité [0..n] indique qu'une instance de System peut être associée à zéro, une ou plusieurs instances de PeerIdentityTable.

8.8 Classe d'agrégation PeerIdentityMember

La classe PeerIdentityMember agrège les instances de PeerIdentityEntry dans un PeerIdentityTable. C'est une agrégation faible. La définition de classe pour PeerIdentityMember est la suivante :

NOM : PeerIdentityMember
DESCRIPTION : PeerIdentityMember agrège les instances de PeerIdentityEntry dans un PeerIdentityTable.
DÉRIVÉ DE : MemberOfCollection (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Collection [ref PeerIdentityTable[1..1]] ; Member [ref PeerIdentityEntry [0..n] [weak]]

8.8.1 Référence Collection

La propriété Collection est héritée de MemberOfCollection et est outrepassée pour se référer à une instance de PeerIdentityTable. La cardinalité [1..1] indique qu'une instance de PeerIdentityEntry DOIT être associée à une seule instance de PeerIdentityTable (c'est-à-dire, les instances de PeerIdentityEntry ne sont pas partagées entre les PeerIdentityTables).

8.8.2 Référence Member

La propriété Member est héritée de MemberOfCollection et est outrepassée pour se référer à une instance de PeerIdentityEntry.

La cardinalité [0..n] indique qu'une instance de PeerIdentityTable peut être associée à zéro, une ou plusieurs instances de PeerIdentityEntry.

8.9 Classe d'association IKEServicePeerGateway

La classe IKEServicePeerGateway fournit l'association entre un IKEService et la liste des instances de PeerGateway qu'elle utilise dans la négociation avec les passerelles de sécurité. La définition de classe pour IKEServicePeerGateway est la suivante :

NOM : IKEServicePeerGateway
DESCRIPTION : Associe un IKEService et la liste des instances de PeerGateway qu'elle utilise pour négocier avec les passerelles de sécurité.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref PeerGateway[0..n]] ; Dependent [ref IKEService[0..n]]

8.9.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de PeerGateway. La cardinalité [0..n] indique qu'une instance de IKEService peut être associée à zéro, une ou plusieurs instances de PeerGateway.

8.9.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de IKEService. La cardinalité [0..n] indique qu'une instance de PeerGateway peut être associée à zéro, une ou plusieurs instances de IKEService.

8.10 Classe d'association IKEServicePeerIdentityTable

La classe IKEServicePeerIdentityTable fournit la relation entre un IKEService et un PeerIdentityTable qu'elle utilise pour transposer comme nécessaire entre adresses et identités. La définition de classe pour IKEServicePeerIdentityTable est la suivante :

NOM : IKEServicePeerIdentityTable
DESCRIPTION : IKEServicePeerIdentityTable fournit la relation entre un IKEService et un PeerIdentityTable qu'il utilise.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref PeerIdentityTable[0..n]] ; Dependent [ref IKEService[0..n]]

8.10.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de PeerIdentityTable. La cardinalité [0..n] indique qu'une instance de IKEService peut être associée à zéro, une ou plusieurs instances de PeerIdentityTable.

8.10.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de IKEService. La cardinalité [0..n] indique qu'une instance de PeerIdentityTable peut être associée à zéro, une ou plusieurs instances de IKEService.

8.11 Classe d'association IKEAutostartSetting

La classe IKEAutostartSetting associe un AutostartIKESetting à un IKEService qui peut l'utiliser pour démarrer automatiquement une négociation IKE ou créer une SA statique. La définition de classe pour IKEAutostartSetting est la suivante :

NOM : IKEAutostartSetting

DESCRIPTION : Associe un AutostartIKESetting à un IKEService.
DÉRIVÉ DE : ElementSetting (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Element [ref IKEService[0..n]] ; Setting [ref AutostartIKESetting[0..n]]

8.11.1 Référence Element

La propriété Element est héritée de ElementSetting et est outrepassée pour se référer à une instance de IKEService. La cardinalité [0..n] indique qu'une instance de AutostartIKESetting peut être associée à zéro, une ou plusieurs instances de IKEService.

8.11.2 Référence Setting

La propriété Setting est héritée de ElementSetting et est outrepassée pour se référer à une instance de AutostartIKESetting. La cardinalité [0..n] indique qu'une instance de IKEService peut être associée à zéro, une ou plusieurs instances de AutostartIKESetting.

8.12 Classe d'agrégation AutostartIKESettingContext

La classe AutostartIKESettingContext agrège les réglages utilisés pour démarrer automatiquement les négociations ou créer une SA statique dans un ensemble de configuration. La définition de classe pour AutostartIKESettingContext est la suivante :

NOM : AutostartIKESettingContext
DESCRIPTION : AutostartIKESettingContext agrège les instances de AutostartIKESetting dans un ensemble de configuration.
DÉRIVÉ DE : SystemSettingContext (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Context [ref AutostartIKEConfiguration [0..n]] ; Setting [ref AutostartIKESetting [0..n]] ; SequenceNumber

8.12.1 Référence Context

La propriété Context est héritée de SystemSettingContext et est outrepassée pour se référer à une instance de AutostartIKEConfiguration. La cardinalité [0..n] indique qu'une instance de AutostartIKESetting peut être associée à zéro, une ou plusieurs instances de AutostartIKEConfiguration (c'est-à-dire, un réglage peut être dans plusieurs ensembles de configuration).

8.12.2 Référence Setting

La propriété Setting est héritée de SystemSettingContext et est outrepassée pour se référer à une instance de AutostartIKESetting. La cardinalité [0..n] indique qu'une instance de AutostartIKEConfiguration peut être associée à zéro, une ou plusieurs instances de AutostartIKESetting.

8.12.3 Propriété SequenceNumber

La propriété SequenceNumber spécifie l'ordre à utiliser pour débuter les négociations ou créer une SA statique. Une valeur de zéro indique que l'ordre n'est pas significatif et que les réglages peuvent être appliqués en parallèle avec d'autres réglages. Tous les autres réglages dans la configuration sont exécuté en séquence des valeurs inférieures aux supérieures. Les numéros de séquence n'ont pas besoin d'être uniques dans une AutostartIKEConfiguration et l'ordre n'est pas significatif pour les réglages qui ont le même numéro de séquence. La propriété est définie comme suit :

NOM : SequenceNumber
DESCRIPTION : La séquence dans laquelle les réglages sont appliqués au sein d'un ensemble de configuration.
SYNTAXE : Entier de 16 bits non signé

8.13 Classe d'association IKEServiceForEndpoint

La classe IKEServiceForEndpoint fournit l'association qui montre quel service IKE, si il en est, fournit les services de négociation IKE pour quelles interfaces réseau. La définition de classe pour IKEServiceForEndpoint est la suivante :

NOM : IKEServiceForEndpoint
DESCRIPTION : Associe un IPProtocolEndpoint à un IKEService qui fournit les services de négociation pour le point d'extrémité.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref IKEService[0..1]] ; Dependent [ref IPProtocolEndpoint[0..n]]

8.13.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de IKEService. La cardinalité [0..1] indique qu'une instance de IPProtocolEndpoint DOIT être associée au plus à une instance de IKEService.

8.13.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de IPProtocolEndpoint qui est associée au plus à un IKEService. La cardinalité [0..n] indique qu'une instance de IKEService peut être associée à zéro, une ou plusieurs instances de IPProtocolEndpoint.

8.14 Classe d'association IKEAutostartConfiguration

La classe IKEAutostartConfiguration fournit la relation entre un IKEService et un ensemble de configuration qu'il utilise pour démarrer automatiquement un ensemble de SA. La définition de classe pour IKEAutostartConfiguration est la suivante :

NOM : IKEAutostartConfiguration
DESCRIPTION : IKEAutostartConfiguration fournit la relation entre un IKEService et une AutostartIKEConfiguration qu'il utilise pour démarrer automatiquement un ensemble de SA.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref AutostartIKEConfiguration [0..n]] ; Dependent [ref IKEService [0..n]] ; Active

8.14.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de AutostartIKEConfiguration. La cardinalité [0..n] indique qu'une instance de IKEService peut être associée à zéro, une ou plusieurs instances de AutostartIKEConfiguration.

8.14.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de IKEService. La cardinalité [0..n] indique qu'une instance de AutostartIKEConfiguration peut être associée à zéro, une ou plusieurs instances de IKEService.

8.14.3 Propriété Active

La propriété Active indique si l'ensemble AutostartIKEConfiguration est actuellement actif pour le IKEService associé. C'est-à-dire que au démarrage, la configuration active est utilisée pour commencer automatiquement les négociations IKE et créer les SA statiques. La propriété est définie comme suit :

NOM : Active
DESCRIPTION : Active indique si l'ensemble AutostartIKEConfiguration est actuellement actif pour le IKEService associé.
SYNTAXE ; Booléen
VALEUR : vrai - AutostartIKEConfiguration est actuellement actif pour le IKEService associé.
faux - AutostartIKEConfiguration est actuellement inactif pour le IKEService associé

8.15 Classe d'association IKEUsesCredentialManagementService

La classe IKEUsesCredentialManagementService définit l'ensemble des CredentialManagementService qui sont des sources de confiance des accréditifs pour les négociation IKE de phase 1. La définition de classe pour IKEUsesCredentialManagementService est la suivante :

NOM : IKEUsesCredentialManagementService
DESCRIPTION : Associe l'ensemble des CredentialManagementService qui sont de confiance pour le IKEService comme sources des accréditifs utilisés dans les négociations IKE de phase 1.
DÉRIVÉ DE : Dependency (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref CredentialManagementService [0..n]] ; Dependent [ref IKEService [0..n]]

8.15.1 Référence Antecedent

La propriété Antecedent est héritée de Dependency et est outrepassée pour se référer à une instance de CredentialManagementService. La cardinalité [0..n] indique qu'une instance de IKEService peut être associée à zéro, une ou plusieurs instances de CredentialManagementService.

8.15.2 Référence Dependent

La propriété Dependent est héritée de Dependency et est outrepassée pour se référer à une instance de IKEService. La cardinalité [0..n] indique qu'une instance de CredentialManagementService peut être associée à zéro, une ou plusieurs instances de IKEService.

8.16 Classe d'association EndpointHasLocalIKEIdentity

La classe EndpointHasLocalIKEIdentity associe un IPProtocolEndpoint à un ensemble d'instances de IKEIdentity qui peuvent être utilisées dans la négociation des associations de sécurité sur le point d'extrémité. Une IKEIdentity DOIT être associée à un IPProtocolEndpoint qui utilise cette association ou à une collection d'instances de IKEIdentity qui utilisent l'association CollectionHasLocalIKEIdentity. La définition de classe pour EndpointHasLocalIKEIdentity est la suivante :

NOM : EndpointHasLocalIKEIdentity
DESCRIPTION : EndpointHasLocalIKEIdentity associe un IPProtocolEndpoint à un ensemble d'instances de IKEIdentity.
DÉRIVÉ DE :ElementAsUser (voir [CIMUSER])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref IPProtocolEndpoint [0..1]] ; Dependent [ref IKEIdentity [0..n]]

8.16.1 Référence Antecedent

La propriété Antecedent est héritée de ElementAsUser et est outrepassée pour se référer à une instance de IPProtocolEndpoint. La cardinalité [0..1] indique qu'une instance de IKEIdentity DOIT être associée à au plus une instance de IPProtocolEndpoint.

8.16.2 Référence Dependent

La propriété Dependent est héritée de ElementAsUser et est outrepassée pour se référer à une instance de IKEIdentity. La cardinalité [0..n] indique qu'une instance de IPProtocolEndpoint peut être associée à zéro, une ou plusieurs instances de IKEIdentity.

8.17 Classe d'association CollectionHasLocalIKEIdentity

La classe CollectionHasLocalIKEIdentity associe une collection d'instances de IPProtocolEndpoint à un ensemble d'instances de IKEIdentity qui peuvent être utilisées pour négocier des SA pour les points d'extrémité dans la collection. Une IKEIdentity DOIT être associée à un IPProtocolEndpoint qui utilise l'association EndpointHasLocalIKEIdentity ou à une collection d'instances de IKEIdentity qui utilisent cette association. La définition de classe pour CollectionHasLocalIKEIdentity est la suivante :

NOM : CollectionHasLocalIKEIdentity
DESCRIPTION : CollectionHasLocalIKEIdentity associe une collection d'instances de IPProtocolEndpoint à un ensemble

d'instances de IKEIdentity.
DÉRIVÉ DE : ElementAsUser (voir [CIMUSER])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref Collection [0..1]] ; Dependent [ref IKEIdentity [0..n]]

8.17.1 Référence Antecedent

La propriété Antecedent est héritée de ElementAsUser et est outrepassée pour se référer à une instance de Collection instance. La cardinalité [0..1] indique qu'une instance de IKEIdentity DOIT être associée à au plus une instance de Collection.

8.17.2 Référence Dependent

La propriété Dependent est héritée de ElementAsUser et est outrepassée pour se référer à une instance de IKEIdentity. La cardinalité [0..n] indique qu'une instance de Collection peut être associée à zéro, une ou plusieurs instances de IKEIdentity.

8.18 Classe d'association IKEIdentitysCredential

La classe IKEIdentitysCredential est une association qui met en rapport un ensemble d'accréditifs avec leur identité IKE locale correspondante. La définition de classe pour IKEIdentitysCredential est la suivante :

NOM : IKEIdentitysCredential
DESCRIPTION : IKEIdentitysCredential associe un ensemble d'accréditifs à leur IKEIdentity locale correspondante.
DÉRIVÉ DE : UsersCredential (voir [CIMCORE])
ABSTRAITE : FAUX
PROPRIÉTÉS : Antecedent [ref Credential [0..n]] ; Dependent [ref IKEIdentity [0..n]]

8.18.1 Référence Antecedent

La propriété Antecedent est héritée de UsersCredential et est outrepassée pour se référer à une instance de Credential. La cardinalité [0..n] indique que l'instance de IKEIdentity peut être associée à zéro, une ou plusieurs instances de Credential.

8.18.2 Référence Dependent

La propriété Dependent est héritée de UsersCredential et est outrepassée pour se référer à une instance de IKEIdentity. La cardinalité [0..n] indique qu'une instance de Credential peut être associée à zéro, une ou plusieurs instances de IKEIdentity.

9. Exigences de mise en œuvre

Le tableau qui suit spécifie quelles classes, propriétés, associations et agrégations DOIVENT ou DEVRAIENT ou PEUVENT être mises en œuvre.

4. Classes de politique	
4.1. Classe SARule.	DOIT
4.1.1. Propriété PolicyRuleName	PEUT
4.1.1. Propriété Enabled.	DOIT
4.1.1. Propriété ConditionListType	DOIT
4.1.1. Propriété RuleUsage	PEUT
4.1.1. Propriété Mandatory	PEUT
4.1.1. Propriété SequencedActions	DOIT
4.1.1. Propriété PolicyRoles	PEUT
4.1.1. Propriété PolicyDecisionStrategy	PEUT
4.1.2 Propriété ExecutionStrategy	DOIT
4.1.3 Propriété LimitNegotiation	PEUT
4.2. Classe IKERule.....	DOIT
4.2.1. Propriété IdentityContexts	PEUT
4.3. Classe IPsecRule.	DOIT
4.4. Classe d'association IpsecPolicyForEndpoint	PEUT
4.4.1. Référence Antecedent.	DOIT
4.4.2. Référence Dependent..	DOIT

4.5. Classe d'association IpsecPolicyForSystem	PEUT
4.5.1. Référence Antecedent.	DOIT
4.5.2. Référence Dependent.	DOIT
4.6. Classe d'agrégation SAConditionInRule	DOIT
4.6.1. Propriété GroupNumber.	DEVRAIT
4.6.1. Propriété ConditionNegated	DEVRAIT
4.6.2. Référence GroupComponent.	DOIT
4.6.3. Référence PartComponent.....	DOIT
4.7. Classe d'agrégation PolicyActionInSARule	DOIT
4.7.1. Référence GroupComponent	DOIT
4.7.2. Référence PartComponent...	DOIT
4.7.3. Propriété ActionOrder.....	DEVRAIT
5. Classes de condition et de filtre	
5.1. Classe SACondition.	DOIT
5.2. Classe IPHeadersFilter	DEVRAIT
5.3. Classe CredentialFilterEntry	PEUT
5.3.1. Propriété MatchFieldName	DOIT
5.3.2. Propriété MatchFieldValue	DOIT
5.3.3. Propriété CredentialType	DOIT
5.4. Classe IPSOFilterEntry	PEUT
5.4.1. Propriété MatchConditionType	DOIT
5.4.2. Propriété MatchConditionValue	DOIT
5.5. Classe PeerIDPayloadFilterEntry.	PEUT
5.5.1. Propriété MatchIdentityType	DOIT
5.5.2. Propriété MatchIdentityValue	DOIT
5.6. Classe d'association FilterOfSACondition	DEVRAIT
5.6.1. Référence Antecedent	DOIT
5.6.2. Référence Dependent.	DOIT
5.7. Classe d'association AcceptCredentialFrom	PEUT
5.7.1. Référence Antecedent	DOIT
5.7.2. Référence Dependent	DOIT
6. Classes d'action	
6.1. Classe SAAction	DOIT
6.1.1. Propriété DoActionLogging	PEUT
6.1.2. Propriété DoPacketLogging	PEUT
6.2. Classe SAStaticAction.	DOIT
6.2.1. Propriété LifetimeSeconds	DOIT
6.3. Classe IPsecBypassAction.	DEVRAIT
6.4. Classe IpsecDiscardAction	DEVRAIT
6.5. Classe IKERejectAction.	PEUT
6.6. Classe PreconfiguredSAAction	DOIT
6.6.1. Propriété LifetimeKilobytes	DOIT
6.7. Classe PreconfiguredTransportAction	DOIT
6.8. Classe PreconfiguredTunnelAction	DOIT
6.8.1. Propriété DFHandling	DOIT
6.9. Classe SANegotiationAction	DOIT
6.10. Classe IKENegotiationAction	DOIT
6.10.1. Propriété MinLifetimeSeconds	PEUT
6.10.2. Propriété MinLifetimeKilobytes	PEUT
6.10.3. Propriété IdleDurationSeconds	PEUT
6.11. Classe IPsecAction.	DOIT
6.11.1. Propriété UsePFS	DOIT
6.11.2. Propriété UseIKEGroup	PEUT
6.11.3. Propriété GroupId	DOIT
6.11.4. Propriété Granularity	DEVRAIT
6.11.5. Propriété VendorID	PEUT
6.12. Classe IpsecTransportAction	DOIT
6.13. Classe IpsecTunnelAction	DOIT
6.13.1. Propriété DFHandling	DOIT
6.14. Classe IKEAction	DOIT
6.14.1. Propriété ExchangeMode	DOIT
6.14.2. Propriété UseIKEIdentityType	DOIT
6.14.3. Propriété VendorID	PEUT

6.14.4. Propriété AggressiveModeGroupId	PEUT
6.15. Classe PeerGateway	DOIT
6.15.1. Propriété Name	DEVRAIT
6.15.2. Propriété PeerIdentityType	DOIT
6.15.3. Propriété PeerIdentity	DOIT
6.16. Classe d'association PeerGatewayForTunnel	DOIT
6.16.1. Référence Antecedent	DOIT
6.16.2. Référence Dependent	DOIT
6.16.3. Propriété SequenceNumber	DEVRAIT
6.17. Classe d'agrégation ContainedProposal	DOIT
6.17.1. Référence GroupComponent	DOIT
6.17.2. Référence PartComponent	DOIT
6.17.3. Propriété SequenceNumber	DOIT
6.18. Classe d'association HostedPeerGatewayInformation	PEUT
6.18.1. Référence Antecedent	DOIT
6.18.2. Référence Dependent	DOIT
6.19. Classe d'association TransformOfPreconfiguredAction	DOIT
6.19.1. Référence Antecedent	DOIT
6.19.2. Référence Dependent	DOIT
6.19.3. Propriété SPI	DOIT
6.19.4. Propriété Direction	DOIT
6.20. Classe d'association PeerGatewayForPreconfiguredTunnel	DOIT
6.20.1. Référence AntecedentDOIT	
6.20.2. Référence Dependent.	DOIT
7. Classes de proposition et de transformation	
7.1. Classe abstraite SAProposal	DOIT
7.1.1. Propriété Name	DEVRAIT
7.2 Classe IKEProposal	DOIT
7.2.1. Propriété CipherAlgorithm	DOIT
7.2.2. Propriété HashAlgorithm	DOIT
7.2.3. Propriété PRFAlgorithm	PEUT
7.2.4. Propriété GroupId	DOIT
7.2.5. Propriété AuthenticationMethod	DOIT
7.2.6. Propriété MaxLifetimeSeconds	DOIT
7.2.7. Propriété MaxLifetimeKilobytes	DOIT
7.2.8. Propriété VendorID	PEUT
7.3. Classe IpsecProposal	DOIT
7.4. Classe abstraite SATransform	DOIT
7.4.1. Propriété TransformName	DEVRAIT
7.4.2. Propriété VendorID	PEUT
7.4.3. Propriété MaxLifetimeSeconds	DOIT
7.4.4. Propriété MaxLifetimeKilobytes	DOIT
7.5. Classe AHTransform	DOIT
7.5.1. Propriété AHTransformId	DOIT
7.5.2. Propriété UseReplayPrevention	PEUT
7.5.3. Propriété ReplayPreventionWindowSize	PEUT
7.6. Classe ESPTransform	DOIT
7.6.1. Propriété IntegrityTransformId	DOIT
7.6.2. Propriété CipherTransformId	DOIT
7.6.3. Propriété CipherKeyLength	PEUT
7.6.4. Propriété CipherKeyRounds	PEUT
7.6.5. Propriété UseReplayPrevention	PEUT
7.6.6. Propriété ReplayPreventionWindowSize	PEUT
7.7. Classe IPCOMPTransform	PEUT
7.7.1. Propriété Algorithm	DOIT
7.7.2. Propriété DictionarySize	PEUT
7.7.3. Propriété PrivateAlgorithm	PEUT
7.8. Classe d'association SAProposalInSystem	PEUT
7.8.1. Référence Antecedent	DOIT
7.8.2. Référence Dependent.	DOIT
7.9. Classe d'agrégation ContainedTransform	DOIT
7.9.1. Référence GroupComponent	DOIT
7.9.2. Référence PartComponent	DOIT

7.9.3. Propriété SequenceNumber	DOIT
7.10. Classe d'association SATransformInSystem	PEUT
7.10.1. Référence Antecedent	DOIT
7.10.2. Référence Dependent	DOIT
8. Classes de service et d'identité IKE	
8.1. Classe IKEService	PEUT
8.2. Classe PeerIdentityTable	PEUT
8.3.1. Propriété Name	DEVRAIT
8.3. Classe PeerIdentityEntry	PEUT
8.3.1. Propriété PeerIdentity	DEVRAIT
8.3.2. Propriété PeerIdentityType	DEVRAIT
8.3.3. Propriété PeerAddress.	DEVRAIT
8.3.4. Propriété PeerAddressType	DEVRAIT
8.4. Classe AutostartIKEConfiguration	PEUT
8.5. Classe AutostartIKESetting	PEUT
8.5.1. Propriété Phase1Only	PEUT
8.5.2. Propriété AddressType	DEVRAIT
8.5.3. Propriété SourceAddress	DOIT
8.5.4. Propriété SourcePort.	DOIT
8.5.5. Propriété DestinationAddress	DOIT
8.5.6. Propriété DestinationPort	DOIT
8.5.7. Propriété Protocol.	DOIT
8.6. Classe IKEIdentity	PEUT
8.6.1. Propriété IdentityType	DOIT
8.6.2. Propriété IdentityValue	DOIT
8.6.3. Propriété IdentityContexts	PEUT
8.7. Classe d'association HostedPeerIdentityTable	PEUT
8.7.1. Référence Antecedent	DOIT
8.7.2. Référence Dependent	DOIT
8.8. Classe d'agrégation PeerIdentityMember	PEUT
8.8.1. Référence Collection	DOIT
8.8.2. Référence Member	DOIT
8.9. Classe d'association IKEServicePeerGateway	PEUT
8.9.1. Référence Antecedent	DOIT
8.9.2. Référence Dependent	DOIT
8.10. Classe d'association IKEServicePeerIdentityTable	PEUT
8.10.1. Référence Antecedent	DOIT
8.10.2. Référence Dependent	DOIT
8.11. Classe d'association IKEAutostartSetting	PEUT
8.11.1. Référence Element	DOIT
8.11.2. Référence Setting	DOIT
8.12. Classe d'agrégation AutostartIKESettingContext	PEUT
8.12.1. Référence Context	DOIT
8.12.2. Référence Setting	DOIT
8.12.3. Propriété SequenceNumber	DEVRAIT
8.13. Classe d'association IKEServiceForEndpoint	PEUT
8.13.1. Référence Antecedent	DOIT
8.13.2. Référence Dependent	DOIT
8.14. Classe d'association IKEAutostartConfiguration	PEUT
8.14.1. Référence Antecedent	DOIT
8.14.2. Référence Dependent	DOIT
8.14.3. Propriété Active	DEVRAIT
8.15. Classe d'association IKEUsesCredentialManagementService	PEUT
8.15.1. Référence Antecedent	DOIT
8.15.2. Référence Dependent	DOIT
8.16. Classe d'association EndpointHasLocalIKEIdentity	PEUT
8.16.1. Référence Antecedent	DOIT
8.16.2. Référence Dependent	DOIT
8.17. Classe d'association CollectionHasLocalIKEIdentity	PEUT
8.17.1. Référence Antecedent	DOIT
8.17.2. Référence Dependent	DOIT
8.18. Classe d'association IKEIdentitysCredential	PEUT
8.18.1. Référence Antecedent	DOIT

8.18.2. Référence Dependent DOIT

10. Considérations sur la sécurité

Le présent document décrit seulement un modèle d'information pour la politique IPsec. Il ne détaille pas les exigences de sécurité pour la mémorisation ou la livraison des dites informations.

Les modèles physiques déduits de ce modèle d'information DOIVENT mettre en œuvre la sécurité pertinente pour la mémorisation et la livraison. La plupart des classes (par exemple, IpHeadersFilter, SAAction,...) DOIVENT au moins fournir le service d'intégrité ; d'autres éléments d'information DOIVENT aussi recevoir le service de confidentialité (par exemple, SharedSecret comme décrit dans les classes PeerIdentityEntry et PreconfiguredSAAction).

11. Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourrait être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à http://www.ietf.org/ipr.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.

12. Références

12.1 Références normatives

[CIMCORE] "DMTF Common Information Model - Core Model v2.5" disponible à http://www.dmtf.org/standards/CIM_Schema25/CIM_Core25.mof

[CIMNETWORK] "DMTF Common Information Model - Network Model v2.5" disponible à http://www.dmtf.org/standards/CIM_Schema25/CIM_Network25.mof

[CIMUSER] "DMTF Common Information Model - User-Security Model v2.5" disponible à http://www.dmtf.org/standards/CIM_Schema25/CIM_User25.mof

[RFC1108] S. Kent, "Options de sécurité du Ministère US de la défense pour le protocole Internet", novembre 1991. *(Hist.)*

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC2401] S. Kent et R. Atkinson, "Architecture de sécurité pour le protocole Internet", novembre 1998. (*Obsolète, voir* RFC4301)

[RFC2402] S. Kent et R. Atkinson, "En-tête d'authentification IP", novembre 1998. (*Obsolète, voir* RFC4302, 4305)

[RFC2406] S. Kent et R. Atkinson, "Encapsulation de charge utile de sécurité IP (ESP)", novembre 1998. *(Obsolète, voir* RFC4303)

[RFC2407] D. Piper, "Le domaine d'interprétation de sécurité IP de l'Internet pour ISAKMP", novembre 1998. *(Obsolète, voir* 4306)

[RFC2409] D. Harkins et D. Carrel, "L'échange de clés Internet (IKE)", novembre 1998. *(Obsolète, voir la* RFC4306)

[RFC3060] B. Moore et autres, "Spécification du modèle d'information de cœur de politique -- version 1", février 2001. (*MàJ par* RFC3460) *(P.S.)*

[RFC3173] A. Shacham et autres, "Protocole de compression de charge utile IP (IPComp)", septembre 2001. *(P.S.)*

[RFC3460] B. Moore, éd., "Extensions au modèle d'information de cœur de politique (PCIM)", janvier 2003. *(P.S.)*

12.2 Références pour information

[DMTF] Distributed Management Task Force, http://www.dmtf.org/

[RFC2251] M. Wahl, T. Howes et S. Kille, "Protocole léger d'accès à un répertoire (v3)", décembre 1997.

[RFC2748] D. Durham et autres, "Protocole COPS (Service commun de politique ouverte)", janvier 2000. (*MàJ par* RFC4261) *(P.S.)*

[RFC3084] K. Chan et autres, "Utilisation de COPS pour l'approvisionnement de politique (COPS-PR)", mars 20010 *(P.S.)*

13. Déclinatoire de responsabilité

Les idées de la présente spécification sont celles des auteurs et ne sont pas nécessairement celles de leur employeur. Les auteurs et leur employeur dénient spécifiquement toute responsabilité pour tout problème découlant de la mise en œuvre ou de l'utilisation correcte ou incorrecte de la présente spécification.

14. Remerciements

Les auteurs tiennet à remercier Mike Jeronimo, Ylian Saint-Hilaire, Vic Lortz, William Dixon, Man Li, Wes Hardaker et Ricky Charlet de leurs contributions à ce modèle de politique IPsec.

De plus, le présent document n'aurait pas été possible sans les documents précédents de schéma IPsec. Pour cela nos remerciement s'adressent à Rob Adams, Partha Bhattacharya, William Dixon, Roy Pereira, et Raju Rajan.

15. Adresse des auteurs

Jamie Jason
Intel Corporation
MS JF3-206
2111 NE 25th Ave.
Hillsboro, OR 97124
mél : jamie.jason@intel.com

Lee Rafalow
IBM Corporation, BRQA/502
4205 So. Miami Blvd.
Research Triangle Park, NC 27709
mél : rafalow@watson.ibm.com

Eric Vyncke
Cisco Systems
7 De Kleetlaan
B-1831 Diegem
Belgium
mél : evyncke@cisco.com

16. Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.