**République Algérienne Démocratique et Populaire**
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**


**Université A/Mira de Béjaia**
**Faculté des Sciences Exactes**
**Département d'Informatique**

# *MÉMOIRE DE MASTER RECHERCHE*

**En**
**Informatique**

**Option**
***Réseaux et Systèmes Distribués***

# Thème

# Confidentialité Des Utilisateurs Dans Le Cloud Computing

**Présenté par :**

**MERABET Fares**
**MAKHLOUF Karima**

**Soutenu le 28 Juin 2017 devant le jury composé de :**

| | | | |
|---|---|---|---|
| **Présidente** | **M$^{me}$ OUYAHIA Samira** | **M.C.B** | **U. A/Mira Béjaia.** |
| **Promotrice** | **M$^{me}$ YAICI Malika** | **M.C.B** | **U. A/Mira Béjaia.** |
| **Examinatrice** | **M$^{me}$ KHALED Hayette** | **M.A.A** | **U. A/Mira Béjaia.** |

Béjaia, Juin 2017.

# ⁕ *Remerciements* ⁕

Nous tenons à remercier Dieu tout puissant de nous avoir donné le courage et la patience jusqu'à l'achèvement de ce modeste travail.

Nous tenons à exprimer nos vifs remerciements pour notre respectueuse promotrice M$^{me}$ YAICI Malika, d'avoir accepté de nous encadré, ainsi pour son soutien et ses remarques pertinentes, ainsi notre vive reconnaissance pour M$^{lle}$ DJELLABIA Amina pour ses précieux conseils et ses orientations.

Nous tenons à remercier aussi les membres du jury de nous avoir honorés en acceptant de juger notre modeste travail. Veuillez trouver ici le témoignage de notre respect le plus profond.

Nos remerciements vont aussi à nos parents, nos familles, amis et toutes les personnes qui nous ont soutenus jusqu'au bout, et qui n'ont pas cessé de nous donner des conseils et le courage pour réaliser ce modeste travail.

# ⁂ *Dédicaces* ⁂

Nous tenons à dédier cet humble travail comme preuve

de respect,de gratitude et de reconnaissance.

À nos chers parents, nos familles et amis.

À tous ceux que nous aimons.

Pour leurs encouragements, soutiens, patiences et prières.

# Table des matières

# Table des figures

# Liste des tableaux

# Liste des algorithmes

# Notations et symboles

| | |
|---|---|
| **AOL** | **America OnLine** |
| **API** | **Application Programming Interface** |
| **AT** | **Access Ticket** |
| **CNIL** | **Commission Nationale de l'Informatique et des Libertés** |
| **CSP** | **Cloud Service Provider** |
| **DDM** | **Data Dissemination Models** |
| **FHE** | **Fully Homomorphic Encryption** |
| **GAPP** | **the Generally Accepted and Privacy Principles** |
| **IaaS** | **Infrastructure as a Service** |
| **IdM** | **Identity Mmanagment** |
| **IdP** | **Identity Provider** |
| **IP** | **Internet Protocol address** |
| **ISO/IEC** | **International Standardization Organization / International Electrotechnical Commission** |
| **MACA** | **Multi Factor Cloud Authentication** |
| **MFA** | **Multi Factor Authentication** |
| **MIP** | **Modified Internet Protocole** |
| **NIST** | **National over Institute of Standards and Technology** |
| **NSP** | **National Service Pseudonymisation** |
| **OECD** | **Organisation for Economic Co-operation and Development** |
| **OIDC** | **OpenIDware Connect** |
| **ORUTA** | **One Ring to Rule Them All** |
| **OTIP** | **Owned Translated IP address** |

| | |
|---|---|
| **PaaS** | **Platform as a Service** |
| **PAP** | **Profile Acquisition Program** |
| **PCCP** | **Preserving Cloud Computing Privacy** |
| **PII** | **Personally Identifiable Information** |
| **PIdMM** | **Privacy Identity Managment Model** |
| **PIICMM** | **Personally Identifiable Information Confidential Management Method** |
| **RSA** | **Ronald Rivest Sdi Shamir Aeonard Adleman** |
| **R-SA-I** | **Request Service Access Interface** |
| **RTM** | **Registration and Ticket Manager** |
| **SaaS** | **Software as a Service** |
| **SLA** | **Service Level Agreement** |
| **SM** | **Service Manager** |
| **SP** | **Service Provider** |
| **TPA** | **Third Party Auditor** |
| **TR** | **Ticket Registration** |
| **TTP** | **The Trusted Third Party** |
| **TOT** | **Ticket Of Ticket** |
| **UPDB** | **User Profile DataBase** |
| **URL** | **Uniform Resource Locator** |
| **USID** | **User Service Identity Dependent** |
| **VM** | **Virtual Machine** |
| **XRI** | **Extensible Resource Identifier** |

# Introduction générale

Indéniablement, la technologie de l'internet se développe de manière exponentielle depuis sa création. Actuellement, une nouvelle tendance a fait son apparition dans le monde des technologies de l'information et de la communication, il s'agit du "cloud computing".

Le cloud computing est un paradigme informatique dans lequel les entreprises peuvent stocker leurs données et accéder aux applications à distance. C'est ainsi que le cloud a prouvé un grand succès au cours de ces dernières années. Toutefois, l'émergence de ce paradigme a fait apparaitre de nouveaux problèmes en termes de sécurité des informations sensibles, en particulier la confidentialité des informations personnelles identifiables (PII) des utilisateurs pour lesquels les solutions et les mécanismes existants sont inadéquats d'assurer leurs protections. Il est donc nécessaire que ces informations restent incompréhensibles par le fournisseur cloud, afin d'assurer une confidentialité maximale des utilisateurs et préserver leur vie privée.

L'objectif assigné à notre travail consiste à proposer une approche qui prend en charge la protection des PII des utilisateurs. Le principe est de fournir une authentification anonyme aux utilisateurs et garantir leur intraçabilité en masquant l'adresse IP d'origine.

Ce mémoire est structuré en quatre chapitres :

- Dans le premier chapitre, nous définissons le cloud computing, ses modèles de services ainsi que ses avantages et inconvénients.
- Dans le deuxième chapitre, nous allons présenter les problèmes de sécurité dans le cloud computing, ainsi que les définitions des deux concepts "vie privée" et "confidentialité". Ensuite, nous décrirons la problématique qui se pose dans le cadre de notre travail.
- Dans le troisième chapitre, nous avons effectué un état de l'art sur la confidentialité dans le cloud, et nous avons établi comme synthèse un tableau comparatif des différentes approches étudiées.
- Dans le quatrième chapitre, nous présenterons en détail les différentes phases par lesquelles notre proposition passe pour assurer la confidentialité des PII.

Enfin, notre mémoire s'achève par une conclusion générale résumant les grands points qui ont été abordés, ainsi que des perspectives que l'ont souhaite accomplir prochainement avec une annexe pour présenter l'outil de vérification Scyther.

Chapitre 1

# Généralités sur le cloud computing

## 1.1 Introduction

Afin de comprendre notre mémoire qui porte sur la confidentialité dans le cloud computing, nous allons fournir les concepts généraux qui permettront aux lecteurs de saisir fermement l'idée de ce qu'est le cloud computing, ses avantages et de fournir une meilleure appréhension du sujet traité.

## 1.2 Définitions de cloud computing

Il existe diverses définitions et interprétations de cloud computing. Nous allons essayer de donner un ensemble représentatif de ces définitions.

Le NIST américain (National over Institute of Standards and Technology) [33] a élaboré une quinzaine de versions de sa définition pour finalement ne retenir que la suivante : *"Le cloud computing est un modèle qui permet d'offrir, à la demande, un accès réseau commode à un ensemble de ressources informatiques configurables partagées (par exemple : des réseaux, des serveurs, des systèmes de stockage, des applications et des services) qui peuvent être rapidement mises à disposition et libérer avec un effort minimal."*

Jeffery et Neidecker [23] ont défini le cloud computing comme la plate-forme ou l'infrastructure dans laquelle les ressources dynamiquement évolutives (élastiques) sont fournies comme un service via internet, permettant aux utilisateurs de traiter

les données à l'extérieur des frontières de l'entreprise, fournissant ainsi des avantages économiques par l'infrastructure virtualisée et partagée sans le besoin d'expertise, ni connaissance sur la technologie sous-jacente.

Les deux définitions décrivent un paradigme dans lequel les utilisateurs peuvent demander des services par internet (serveurs, applications, infrastructures, plates-formes de développement) chaque fois qu'ils en ont besoin, comme une marchandise. Juste pour fournir un facile aperçu afin de comprendre la comparaison, le cloud computing est l'eau ou le gaz de l'informatique. À la maison, vous ne disposez généralement pas d'une pompe à eau, ni d'un générateur de gaz, mais votre maison est reliée à un ensemble de tuyaux où l'eau et le gaz arrivent, et généralement vous payez ce que vous consommez. Plus votre consommation en gaz ou en eau est élevée, plus le montant de la facture à régler sera élevé. L'idée de cloud computing est à peu près la même, on remplace l'eau et le gaz par les services et les infrastructures par des tuyaux pour une connexion à internet [18].

## 1.3 Les caractéristiques du cloud

Le cloud computing est défini avec un ensemble de caractéristiques correspondant aux définitions suivantes [18] :

- Mutualisée (ressources partagées) : l'un des avantages du cloud computing est qu'il est basé sur un modèle d'affaires où les ressources sont partagées entre plusieurs utilisateurs en même temps. Cela est généralement atteint par la virtualisation.
- Élasticité : les utilisateurs peuvent augmenter et/ou diminuer leurs ressources informatiques en fonction de leurs besoins.
- Pay-as-you-go : les utilisateurs paient pour les ressources qu'ils utilisent et seulement pour le temps d'occupation.
- Accès réseau ubiquitaire : les clients peuvent accéder à leurs services demandés partout là où ils en ont besoin, à partir de leurs navigateurs web.

## 1.4 Les trois modèles de services de cloud computing

Trois modèles de services peuvent être offerts sur le cloud : Software as a Service (SaaS), Platform as a Service (PaaS) et Infrastructure as a Service (IaaS). Ces trois modèles de service doivent être déployés sur des infrastructures qui possèdent les cinq caractéristiques essentielles citées plus haut pour être considérées comme du cloud computing.

### 1.4.1 Software as a Service

C'est un modèle de déploiement d'application dans lequel un fournisseur loue une application clé en main à ses clients en tant que service à la demande au lieu de leur facturer des licences. De cette façon, l'utilisateur final n'a plus besoin d'installer tous les logiciels existants sur sa machine de travail. Cela réduit également la maintenance en supprimant le besoin de mettre à jour les applications (tâche toujours ardue dans une entreprise). Ce modèle transforme les budgets logiciels en dépenses variables au lieu quelles soient fixes supprimant ainsi la nécessité d'acquérir une version du logiciel pour chaque personne au sein de l'entreprise [28].

### 1.4.2 Plateform as a Service

Le PaaS est une excroissance du modèle de déploiement SaaS. Une architecture PaaS est un modèle composé de tous les éléments nécessaires pour soutenir la construction, la livraison, le déploiement et le cycle de vie complet des applications et des services exclusivement disponibles à partir d'Internet [28]. L'avantage est que ces environnements sont hébergés par un prestataire basé à l'extérieur de l'entreprise, ce qui permet de ne disposer d'aucune infrastructure et de personnel de maintenance et donc de pouvoir se consacrer au développement [42].

### 1.4.3 Infrastructure as a Service

Il s'agit de la mise à disposition, à la demande, de ressources d'infrastructures dont la plus grande partie est localisée à distance dans des centres de données. L'IaaS permet l'accès aux serveurs et à leurs configurations pour les administrateurs de l'entreprise. Le client a la possibilité de louer des clusters, de la mémoire ou du stockage de données. Le coût est directement lié au taux d'occupation. Une analogie peut être faite avec le mode d'utilisation des industries des commodités (électricité, eau, gaz) ou des télécommunications [42].

La figure1.1 présente une vue d'ensemble des zones de contrôle entre le client et le fournisseur en fonction du service offert sur le cloud.

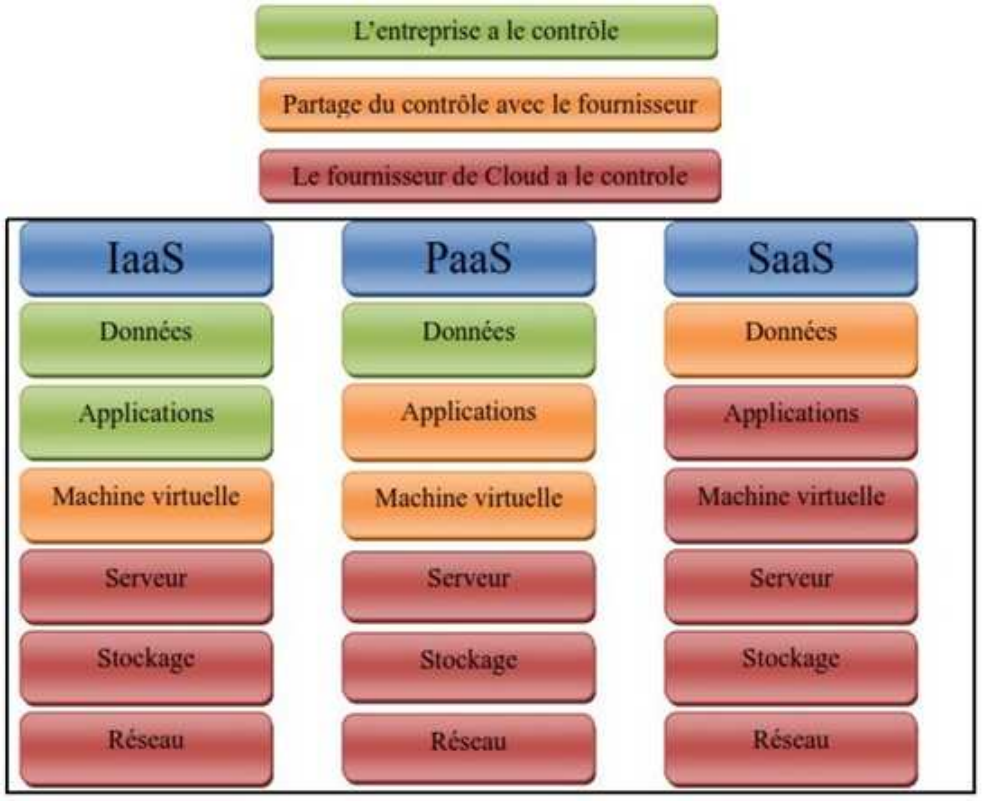


FIGURE 1.1 – Répartition des responsabilités
[26].

## 1.5 Les principaux acteurs dans le cloud computing

Le tableau 1.1 répertorie les acteurs définis dans l'architecture de référence de NIST. Chaque acteur est une entité (une personne ou une organisation) qui participe à une transaction ou à un processus et/ou effectue des tâches dans le cloud computing.

| Acteur | Définition |
|---|---|
| consommateur cloud | Une personne ou une organisation qui entretient une relation d'affaires et/ou utilise le service de fournisseur cloud. |
| fournisseur cloud | Une personne, une organisation ou une entité responsable de services à la disposition des parties intéressées. |
| vérificateur cloud | Une partie qui peut procéder à une évaluation indépendante des services de cloud computing, au fonctionnement du système d'information, à la performance et à la sécurité de la mise en œuvre du cloud. |
| courtier cloud | Une entité qui gère l'utilisation, la performance et la prestation des services de cloud computing et négocie les relations entre les fournisseurs clouds et les consommateurs clouds. |
| transporteur cloud | Un intermédiaire qui fournit la connectivité et le transport des services de cloud computing depuis le fournisseur cloud jusqu'au consommateur cloud. |

TABLE 1.1 – Les acteurs dans le cloud computing selon NIST [35].

## 1.6 Modèles de déploiement dans le cloud computing

Le NIST a défini quatre modèles de déploiement du cloud computing : le cloud privé, le cloud communautaire, le cloud public et le cloud hybride.

### 1.6.1 Le cloud privé

L'infrastructure du cloud est réservée à une entreprise. Elle peut être gérée par l'entreprise ou par un tiers et peut se trouver dans les locaux de l'entreprise ou ailleurs. Les deux caractéristiques du cloud privé sont la délimitation d'un cloud pour l'utilisation d'une seule organisation ainsi qu'un degré plus élevé de sécurité du réseau [26].

### 1.6.2 Le cloud communautaire

L'infrastructure du cloud est partagée par plusieurs entreprises et est destinée à une communauté précise aux préoccupations communes (par exemple une mission, des exigences de sécurité, une stratégie ou des questions de conformité). Elle peut être gérée par des entreprises ou un tiers, et peut se trouver dans leurs locaux ou ailleurs [26].

### 1.6.3 Le cloud public

L'infrastructure du cloud est rendue disponible au grand public ou à un grand groupe industriel et elle appartient à une entreprise qui vend des services dans le cloud. C'est le modèle le plus connu vis-à-vis des utilisateurs cloud. Les services sont fournis dans un environnement virtualisé, construit en utilisant des ressources physiques partagées et accessibles via un réseau public (Internet) [26].

### 1.6.4 Le cloud hybride

L'infrastructure du cloud est constituée de deux clouds ou plus (privés, communautaires ou publics) qui restent des entités indépendantes, mais sont reliés par une technologie standardisée ou propriétaire afin d'autoriser une portabilité des données et des applications (par exemple le "cloud bursting" pour la répartition de charge entre les différents clouds) [26].

## 1.7 Les composantes du cloud computing

Cloud computing ne fait pas référence à une technologie spécifique, mais à une combinaison de technologies préexistantes et des protocoles qui ont rendu ce paradigme possible. Dans cette section, nous allons décrire les composantes technologiques et non technologiques les plus pertinentes.

### 1.7.1 Composantes technologiques

#### 1.7.1.1 Les centres de données et fermes de serveurs

Un centre de données (data center) est un site physique sur lequel se regroupe des équipements constituants un système d'information de l'entreprise (mainframes, serveurs, baies de stockage, équipements réseaux, etc.). Il peut être interne ou externe à l'entreprise [47].

#### 1.7.1.2 La virtualisation

La virtualisation est la principale technologie dans le cloud, c'est une manière pour partitionner une ressource physique en plusieurs ressources virtuelles, par exemple : un serveur, un espace de stockage ou un réseau lors de la création des machines virtuelles. Elle permet d'intégrer différents serveurs de façons plus flexibles pour faciliter l'utilisation [47].

#### 1.7.1.3 Interfaces de programmation d'applications

API (Application Programming Interface) offrent des fonctionnalités aux clients, comme l'auto-approvisionnement et le contrôle des services et des ressources. Ils permettent également la communication entre les applications étrangères et le service sur le cloud. Le type d'API dépend du modèle de déploiement [18].

#### 1.7.1.4 Le cryptage

Le cryptage est le processus de codage des messages ou des informations d'une manière que seuls les partis autorisés puissent lire cette information, empêchant ainsi des parties indésirables à les intercepter. Cela se fait habituellement à l'aide d'une clé de cryptage qui spécifie la façon dont le message est codé. Ensuite, la partie autorisée utilise une clé de décryptage secrète pour déchiffrer le message et le lire. Le cryptage est la technique couramment utilisée pour protéger la confidentialité des messages [18].

## 1.7.2 Composantes non technologiques

### 1.7.2.1 Accords de niveau de service

Le contrat mutuel entre les fournisseurs et les utilisateurs, appelé généralement SLA (Service Level Agreement ), offre des garanties sur la qualité du service en définissant ce qui est à attendre du service offert, et définit les schémas de compensation dans le cas où le fournisseur ne respecte pas ce contrat. Quelques exemples de ces accords pourraient être de 99% de la disponibilité, la résolution des incidents dans une période de temps spécifiée, la sécurité des données, etc [18].

### 1.7.2.2 Les politiques de confidentialité

Une politique est une déclaration d'intentions, mise en œuvre grâce à des protocoles ou des procédures, afin de guider les décisions et d'atteindre un but désiré. Lorsqu'elles sont appliquées sous le thème de la vie privée, ces politiques représentent des documents juridiques qui expliquent et décrivent la façon dont une partie recueille, utilise, divulgue et gère les données du client (par exemple le nom, l'adresse, les antécédents médicaux, les dossiers financiers, les transactions commerciales, etc.). Elle informe également le client à propos des informations recueillies et si elles sont gardées confidentielles, partagées avec des partenaires ou vendues à d'autres entreprises. Ces aspects sont généralement recueillis dans un document écrit qui énonce les règles, fournit les principes qui guident les actions, définit les rôles et les responsabilités, reflète les valeurs et les croyances et indique un protocole d'actions. Dans ce sens, il existe des différences dans la façon dont ces lois sur la protection sont mises en œuvre et appliquées dans différents pays. À titre d'exemple, l'Union européenne exige le respect des lois de protection des données par toute exploitation d'entreprise ou transfert de renseignements personnels sur tout citoyen de l'UE ou encore les affaires. Toutefois, les lois sur la vie privée dans les États-Unis sont appliquées uniquement sur le secteur public sans être appliquées sur le secteur privé [18].

## 1.8 Avantages et inconvénients du cloud computing

### 1.8.1 Avantages

Nous citons quelques avantages du cloud computing [2] :

- Accessibilité garantie : avec le cloud, les services et les applications sont accessibles à tout moment et depuis n'importe quel ordinateur, téléphone portable ou tablette.
- Coût optimisé : le cloud computing est avant tout économique, il représente un gain de coût non négligeable.
- Flexibilité et partage : les services sont flexibles et peuvent être ajustés à tout moment en fonction des besoins et de l'activité de l'entreprise. celle-ci peut diminuer ou augmenter les ressources disponibles, payant ainsi seulement ce qu'elle consomme. Ces ressources peuvent être partagées permettant aux employés de travailler à plusieurs sur un même document, et ce en temps réel.
- Mises à jour automatiques : en plus de la maintenance, le fournisseur cloud se charge de toutes les mises à jour du service, ce qui permet à l'entreprise et à ses employés de se concentrer plus efficacement sur leurs missions et par la même occasion, d'optimiser leur productivité.

### 1.8.2 Inconvénients

Le cloud computing présente également des inconvénients, parmi lesquels nous pouvons citer :

- Connexion Internet obligatoire : sans celle-ci, nous ne pouvons pas accéder aux ressources stockées dans le cloud computing [1].
- Les performances des applications peuvent être amoindries : un cloud public n'améliorera définitivement pas les performances des applications par rapport à un cloud privé [42].
- Une sécurité qui peut être fictive : peu d'entreprises et de particuliers savent qu'en faisant appel à des entreprises américaines, elles s'exposent au Patriot

Act. Cette loi antiterroriste qui a été mise en place aux USA autorise les services de sécurité américains à accéder aux données informatiques des particuliers et entreprises qui utilisent les services des entreprises américaines [6].

- La sécurité des locaux : sont-ils inaccessibles pour des personnes malintentionnées ? [5].

## 1.9 Conclusion

Les logiciels, les applications et les données nous suivent partout grâce à cette nouvelle technologie de l'information qui a révolutionné le monde. La mise en œuvre de ces nouveaux services suppose toutefois une révision complète des procédures et des mesures de sécurité à mettre en place pour garantir la réussite de ce passage vers le cloud. Les questions de sécurité et de confidentialité restent ouvertes, d'où le nombre de recherches qui se font dans ce sens.

Dans ce chapitre, nous avons décrit le principe du cloud computing, ses trois modèles de services, ses quatre modèles de déploiement ainsi que ses avantages et ses inconvénients. Dans ce qui suit, nous allons traiter en détail ces questions de sécurité et de confidentialité des données hébergées chez le fournisseur du cloud.

Chapitre 2

# La sécurité dans le cloud computing

## 2.1 Introduction

Malgré de nombreux avantages qu'offre le cloud, la question : "pourquoi tout de même garder ses données sur un ordinateur personnel ?" reste toujours posé par les futurs utilisateurs de cloud. La réponse à cette question tourne beaucoup autour de la confiance et les nombreux problèmes de sécurité et les défis dont le cloud computing doit faire face.

Dans ce chapitre, nous allons traiter cette question autour de la confiance. Ensuite, nous exposerons quelques solutions existantes pour remédier à ces défis. Enfin, nous présenterons les deux concepts "confidentialité" et "vie privée" comme cités dans la littérature pour établir la relation entre eux.

## 2.2 Puis-je faire confiance aux acteurs du cloud ?

Pour évoquer la sécurité du cloud, on peut utiliser la métaphore de la banque : notre compte chèque est dématérialisé, nous ne savons pas où sont nos données bancaires, et nous faisons confiance à notre banque pour les gérer correctement. C'est comparable au cloud. De même que nous ne stockons pas nos billets de banque sous notre matelas par peur des voleurs, nous pouvons considérer qu'il vaut mieux ne pas conserver nos données chez nous, car un ordinateur peut être volé ou cassé. Les acteurs du cloud sont en effet beaucoup plus compétents que nous pour assurer l'intégrité de nos données : ce sont des professionnels des données informatiques [19].

Pour nuancer ce discours, il faut noter que la différence entre banque et cloud se situe au niveau de la maturité : les acteurs du cloud sont plus récents, et on sait que la confiance s'installe dans le temps [19].

## 2.3 Le point de vue juridique

Les lois sur la vie privée sont différentes selon les pays. Les pays de l'Union européenne ont des institutions qui protègent la vie privée (par exemple : la CNIL en France), et la loi est assez homogène au sein de l'Europe. En revanche, ce n'est pas le cas des États-Unis ou du Japon. Or, la plupart des acteurs du cloud sont d'origine américaine. Ce qui, pour le client de cloud, pose des questions complexes sur la protection de nos données. Et ce d'autant plus qu'une loi crée par Georges Bush après le 11 septembre 2011, le " Patriot Act ", vient encore compliquer les choses. Le Patriot Act dit qu'une entreprise dont le siège est aux États-Unis doit ouvrir ses centres de données sur demande du gouvernement américain, même s'ils sont situés dans un autre pays [19].

Les choses se compliquent encore lorsque l'on parle de la localisation des données. Sur tout que les acteurs du cloud ne communiquent pas toujours sur l'emplacement de nos données : si le client de cloud ignore le pays de localisation, il ne saura donc pas quel est le droit applicable à ses données. Pour prendre un exemple caricatural, si un acteur cloud américain stockait nos données en Chine, elles seraient soumises aux règles d'un régime non démocratique, en plus du Patriot Act. Si les lois sur la protection de la vie privée et la propriété de données étaient les mêmes dans tous les pays, cela faciliterait l'usage du cloud en toute confiance [19].

## 2.4 La question de l'identité réelle

Depuis la création du web, il a toujours été simple de créer des identifiants sur des sites sans lien avec son identité. Ainsi, vous pouvez décider de vous appeler Kaci Mohand ou Steve Jobs lors de la création de votre compte. Cette possibilité garantit

l'anonymat sur le web. Elle est particulièrement utile aux personnes qui se battent contre des régimes totalitaires [19].

Certains acteurs du cloud souhaitent faire disparaître cette possibilité d'anonymat : c'est le cas de Facebook et de Google avec son service Google+. Leurs conditions générales d'utilisation stipulent que l'on doit s'inscrire avec son patronyme réel sous peine de voir son compte désactivé. Leur motivation est la suivante : comme ils gagnent de l'argent en analysant notre comportement, ils ont tout à fait intérêt à nous connaître le plus possible. Par ailleurs, ils souhaitent tous deux devenir le porteur de notre identité en ligne et le service de référence qui présente notre personnalité. On pourrait presque dire qu'ils tendent à ce que notre identité numérique leur appartienne. La conséquence de cette obligation d'identité réelle est la suivante : lorsqu'on utilise son identité Google ou Facebook pour accéder à leurs services, mais aussi à des services tiers, on est perpétuellement identifié par une sorte de syndrome du " Big Brother " [1]. Aujourd'hui, la méfiance vis-à-vis de Facebook atteint un paroxysme chez beaucoup d'utilisateurs. Google en rassure certains avec sa devise " don't be evil " (ne soyez pas malveillant) [19].

## 2.5 Problèmes de sécurité dans le cloud computing

Les problèmes peuvent être catégorisés et considérés comme ci-dessous :

### 2.5.1 Confidentialité

La confidentialité reste l'une des plus grandes préoccupations concernant le cloud computing. Elle implique que les données d'un client et les tâches de calcul doivent être confidentielles vis-à-vis de fournisseur de cloud ainsi qu'aux autres clients. Cela est dû au fait que les clients externalisent leurs tâches de calcul et les données sur les serveurs cloud, qui sont contrôlés et gérés par des fournisseurs de cloud potentielle-

1. L'expression "Big Brother" est utilisée pour qualifier toutes les institutions ou pratiques portant atteinte aux libertés fondamentales et à la vie privée des populations ou des individus.

ment non fiables [45]. Une perte de confidentialité est une divulgation non autorisée des informations [13].

### 2.5.2 Intégrité

L'intégrité dans le cloud devient vulnérable, car les clients ne contrôlent pas physiquement leurs données et leurs logiciels. L'intégrité est la propriété d'une information de ne pas être altérée [32]. Dans le cloud, les applications offrent un stockage en tant que service. Néanmoins, les serveurs cloud peuvent être moins sécurisés, cela signifie que les données peuvent être perdues ou modifiées malicieusement ou accidentellement [45].

### 2.5.3 Disponibilité

La disponibilité est cruciale puisque la fonction de base de cloud computing est de fournir un service à la demande de différents niveaux [45]. Elle garantit un accès opportun et fiable aux informations et à leur utilisation. Une perte de disponibilité est une interruption de l'accès à l'information ou à son utilisation ou au système d'information [13]. À titre d'exemple : les plateformes de jeux vidéo en ligne de Sony (PlayStation) et Microsoft (Xbox) ont été plusieurs fois la cible d'attaques de déni de service, rendant inaccessibles ces plateformes et violant ainsi la propriété de disponibilité des données [32].

### 2.5.4 Contrôle d'accès aux données

L'accès illégal peut se faire en raison des failles de sécurité dans le système d'accès aux données secrètes ou privées [8].

## 2.6 Comment sécuriser le cloud ?

Il existe de nombreux problèmes de sécurité dans le cloud computing, car elle englobe de nombreuses technologies, y compris les réseaux, bases de données, systèmes

d'exploitation, la virtualisation, la planification des ressources, la gestion des transactions, l'équilibrage de charge, le contrôle et la gestion de la concurrence de la mémoire. Par conséquent, il y a six domaines spécifiques dans l'environnement de cloud computing où les équipements et logiciels nécessitent une sécurité maximale. Comme le montre la figure 2.1, ces six domaines sont les suivants : (1) la sécurité des données au repos, (2) la sécurité des données en transit, (3) l'authentification des utilisateurs / applications / processus, (4) la séparation solide entre les données appartenant à des clients différents, (5) les questions juridiques et réglementaires dans le cloud et (6) la réponse aux incidents.

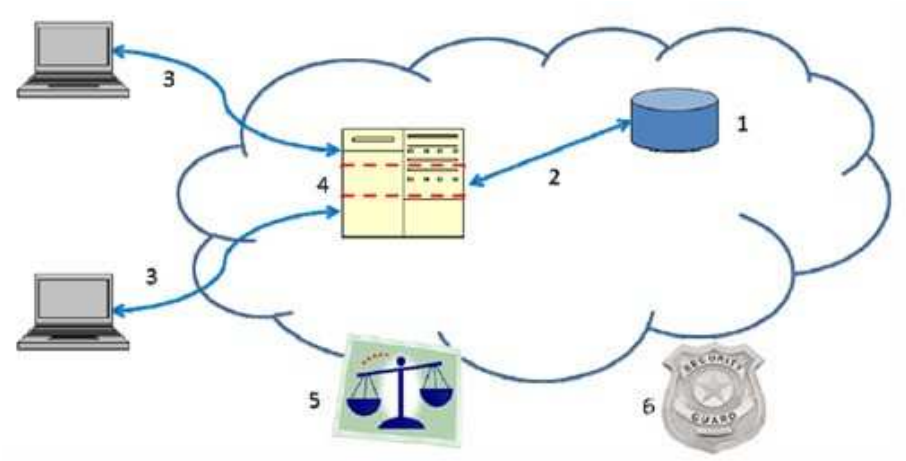


FIGURE 2.1 – Domaines de problèmes de sécurité dans le cloud computing [39].

Pour sécuriser les données au repos, les mécanismes de chiffrement sont certainement les meilleures options. Les fabricants de disques durs sont maintenant en train d'autoriser des programmes de chiffrement automatique. Bien que le cryptage logiciel peut également être utilisé pour la protection des données, il rend le processus plus lent [39].

Le cryptage est la meilleure option pour sécuriser les données en transit. En outre, les mécanismes d'authentification et de protection d'intégrité garantissent que

les données de client soient acheminées vers le cloud et ne seront pas modifiés en transit [39].

L'authentification forte est une exigence obligatoire pour tout type de cloud. L'authentification des utilisateurs est la base principale pour le contrôle d'accès [39].

L'une des plus évidentes préoccupations des clouds est la séparation entre les utilisateurs d'un fournisseur de cloud (qui peuvent être des entreprises, des utilisateurs lambda, des utilisateurs malveillants) pour éviter un accès intentionnel à des informations sensibles, en général, un fournisseur de cloud utiliserait des machines virtuelles (VM) et un hyperviseur à des clients distincts [39].

Les questions juridiques et réglementaires sont extrêmement importantes dans le cloud computing, car elles auront des implications directes sur la sécurité. Pour vérifier qu'un fournisseur de cloud a des politiques et des pratiques solides qui traitent les questions juridiques et réglementaires, chaque client doit avoir ses experts juridiques pour inspecter les politiques et les pratiques des fournisseurs de cloud. Les questions à examiner comprennent la sécurité des données et à l'exportation, la conformité, l'audit, la conservation des données et la destruction, et la découverte juridique [39].

Dans le cadre de prévoir l'imprévisible, les clients doivent prévoir la possibilité de failles de sécurité du fournisseur de cloud ou la malveillance des autres utilisateurs. Une réponse automatique est la meilleure solution à cet effet. Cela est réalisé en intégrant des systèmes de sécurité qui fournissent des notifications en temps réel d'incidents et de mauvaise conduite de l'utilisateur [39].

## 2.7 Qu'est - ce que la vie privée ?

La vie privée est un vaste concept qui varie selon les pays, les cultures et les juridictions. Donner une définition précise est difficile, et cette question, en soi, pose un problème en essayant d'établir un consensus.
Dans la littérature, nous avons trouvé quelques documents pertinents liés aux définitions au sujet de la vie privée dans le cloud computing et des informations

générales à ce sujet.

Les deux Mather et al [41] et X. Ma [46] ont effectué des etudes sur la vie privée dans le cloud computing. Pearson et Charlesworth [40] examinent également certains problèmes de confidentialité. Ci-dessous nous listons plusieurs définitions de la vie privée qui ont été identifiées dans la littérature :

**Définition 2.7.1.** La définition adoptée par l'Organisation de Coopération et de Développement Economiques (OCDE) [2] : *"c'est le statut accordé aux données qui ont été convenues entre la personne ou l'organisation fournissant les données et l'organisation de réception, et qui décrit le degré de protection qui sera fourni "* [41].

**Définition 2.7.2.** La définition donnée par the Generally Accepted Privacy Principles (GAPP) : *"Les droits et obligations des individus et des organisations en ce qui concerne la collecte, l' utilisation, la conservation et la divulgation de renseignements personnels."* [41] [16].

**Définition 2.7.3.** La définition fournie par le dictionnaire d'Oxford : *" la vie privée est définie comme un état dans lequel n'est pas observé ou perturbé par d'autres personnes et un état d'être libre de l' attention du public. Plus précisément, les droits ou obligations de la protection de la vie privée sont liés à la collecte, l' utilisation, la divulgation, le stockage, et la destruction des données personnelles ou d'informations personnelles identifiables."* [46].

**Définition 2.7.4.** *La vie privée est un droit humain fondamental qui englobe le droit d'être laissé seul, elle implique la protection et l'utilisation appropriée des renseignements personnels des clients* [40].

Certaines différences peuvent être notées dans les définitions. Bien que l'OCDE définit la vie privée comme un lien direct entre une personne et ses données connexes. GAPP, le dictionnaire d'Oxford et [40] sont allés plus loin, en incluant à la définition

2. L'Organisation de coopération et de développement économiques (OCDE, en anglais Organisation for Economic Co-operation and Development, OECD) est une organisation internationale d'études économiques, dont les pays membres - des pays développés pour la plupart - ont en commun un système de gouvernement démocratique et une économie de marché [3].

le droit de gestion de cette information. En tenant compte des préoccupations soulevées sur la vie privée, une définition profonde et plus complexe est nécessaire pour la définir correctement, mais l'idée générale est que la vie privée est liée à la collecte, l'utilisation, la divulgation, le stockage et la destruction des données personnelles. En outre, la définition et la conception de la vie privée sont différentes dans les législations de différents pays. Selon Mather et al. [41] dans l'Union européenne la vie privée est un droit fondamental, alors qu'aux États-Unis elle est plus centrée sur la prévention des dommages. Cependant, les grands principes et les définitions décrites ci-dessus seraient applicables à la plupart des pays.

## 2.8 Qu'est - ce que la confidentialité ?

La confidentialité est un terme assez flou dans la littérature. Très souvent, les auteurs se réfèrent à la vie privée comme un concept global lorsqu'ils proposent leurs solutions.

Dans la littérature, quelques papiers définissent le terme de la confidentialité :

**Définition 2.8.1.** Selon la norme ISO / IEC 27000 : *" la confidentialité est la propriété que les informations ne soient pas rendues disponibles ou divulguées aux individus non autorisés."* [29].

**Définition 2.8.2.** *" Garder le secret des données des utilisateurs dans les systèmes de Cloud "* [27].

**Définition 2.8.3.** *"La confidentialité implique que les données d'un client et les tâches de calcul doivent être confidentielles à la fois du fournisseur de cloud ainsi qu'aux autres clients"* [45].

Ces définitions entrent en contraste avec celles fournies au sujet de la vie privée. Comme on peut le noter, la confidentialité est illustrée plus spécifiquement au domaine lié de "garder le secret des données" alors que la vie privée est un concept plus global, qui englobe les droits, les obligations et la gestion sur ces données.

## 2.9 La vie privée et la confidentialité

S. Pearson [31] décrit trois types d'informations sensibles de la vie privée :

- Les informations personnelles identifiables (PII) : toute information pouvant être utilisée pour identifier ou localiser un individu (par exemple : nom, adresse) ou des informations qui peuvent être corrélées avec d'autres informations à identifier un individu (par exemple : numéro de carte de crédit, code postal, adresse Internet (IP)).
- Les informations sensibles : informations sur la religion, la santé, l'information financière, photo et tout type qui est considéré comme privé.
- Les données d'utilisation : les données d'utilisation recueillies auprès de l'utilisation d'appareils informatiques ou les actions effectuées (utilisez une imprimante, visitez une page web, etc.)

Xiao et al. [45] explique que la confidentialité est une caractéristique de la vie privée en disant, que la vie privée est le concept global qui implique la protection des données, la législation et la gestion, bien que la confidentialité soit plus spécifique à la protection des informations personnelles identifiables.
Résumant les éléments de preuve dans une définition unique, nous pourrions dire que la confidentialité est un sous-ensemble ou un attribut de la vie privée qui se concentre sur la protection des informations personnelles identifiables et qui doivent être protégées contre la divulgation, la copie ou le vol.

## 2.10 Problématique

La protection des données dans le cloud parait insuffisante, c'est pour cela que même l'identité de l'utilisateur du cloud doit être protégée contre le fournisseur de cloud et les autres clients.

L'un des objectifs les plus importants que nous voulons atteindre dans la sécurité du cloud computing est la confidentialité des utilisateurs. Et cela en protégeant les informations personnelles identifiables contre le fournisseur de service ; ainsi notre

problématique est la suivante : comment peut-on assurer une authentification anonyme des utilisateurs sans être obligés de fournir leurs PII au fournisseur de service ?

## 2.11 Conclusion

Dans ce chapitre, nous avons présenté les questions relatives à la sécurité dans le cloud. Ensuite, nous avons présenté les différentes définitions existantes dans la littérature sur la vie privée et la confidentialité afin de conclure la relation entre les deux concepts. Enfin, nous avons utilisé cette conclusion pour élaborer notre problématique.

Dans le chapitre suivant, nous effectuerons un état de l'art sur quelques travaux qui traitent la confidentialité dans le cloud.

Chapitre 3

# État de l'art

## 3.1 Introduction

Avec le développement du cloud computing, les préoccupations sur la confidentialité des utilisateurs deviennent de plus en plus importantes. Ainsi, les informations personnelles des utilisateurs seront exposées au risque d'un accès non autorisé. La confidentialité des utilisateurs devrait être maintenue lorsque les données sont collectées, stockées ou transmises dans le cloud.

## 3.2 Critères de l'étude critique des solutions étudiées

Afin de bien évaluer les articles et les travaux que nous allons traiter, nous avons établi une liste de critère d'évaluation, qui se compose de confidentialité des PII(Personally Identifiable Information) , facilité d'utilisation (ergonomie) , intraçabilité et authentification sure :

### 3.2.1 Confidentialité des PII

Étant donné que tout système de gestion d'identité utilise des informations sensibles telles que PII, il doit s'assurer que ces informations ne sont partagées qu'entre des entités appropriées dont l'utilisateur fait confiance et qu'elles restent anonymes pour le service de cloud.

### 3.2.2 Facilité d'utilisation

L'un des principaux objectifs des systèmes de gestion d'identité est de faciliter tout processus lié à l'authentification anonyme des utilisateurs. Cela ne peut être réalisé que si les utilisateurs ne sont pas tenus de compléter des procédures complexes ou de gérer des outils compliqués, et ce afin d'interagir avec les services. Pour cela, les systèmes de gestion de l'identité devraient fournir des interfaces conviviales et des procédures intuitives lorsqu'une fonctionnalité liée à l'identité leur est présentée.

### 3.2.3 In-traçabilité

c'est le fait qu'un fournisseur d'identité ne peut connaître les services auxquels un de ses utilisateurs a eu accès. Et dans le cas d'un fournisseur cloud c'est l'incapacité de savoir l'origine de la demande d'accès et son parcours pour arriver à sa destination.

### 3.2.4 Authentification sure

Les mécanismes d'authentification basés sur des secrets partagés tels que l'authentification par mot de passe d'utilisateur commun n'offrent pas suffisamment de protection pour éviter l'usurpation d'identité ou le vol d'identité. Dans les systèmes de gestion d'identité, des mécanismes d'authentification garantissant un certain niveau de sécurité doivent être déployés, tels que ceux basés sur des techniques biométriques ou des certificats numériques, améliorant ainsi le niveau de sécurité de l'ensemble du système.

## 3.3 Classification

Cette section fait l'objet d'une étude sur quelques solutions proposées pour résoudre le problème de confidentialité des utilisateurs dans le cloud. Pour ce faire, nous proposons une classification en deux catégories selon l'utilisation d'une tierce partie de confiance (TTP) et du chiffrement, comme illustrée dans la figure 3.1.

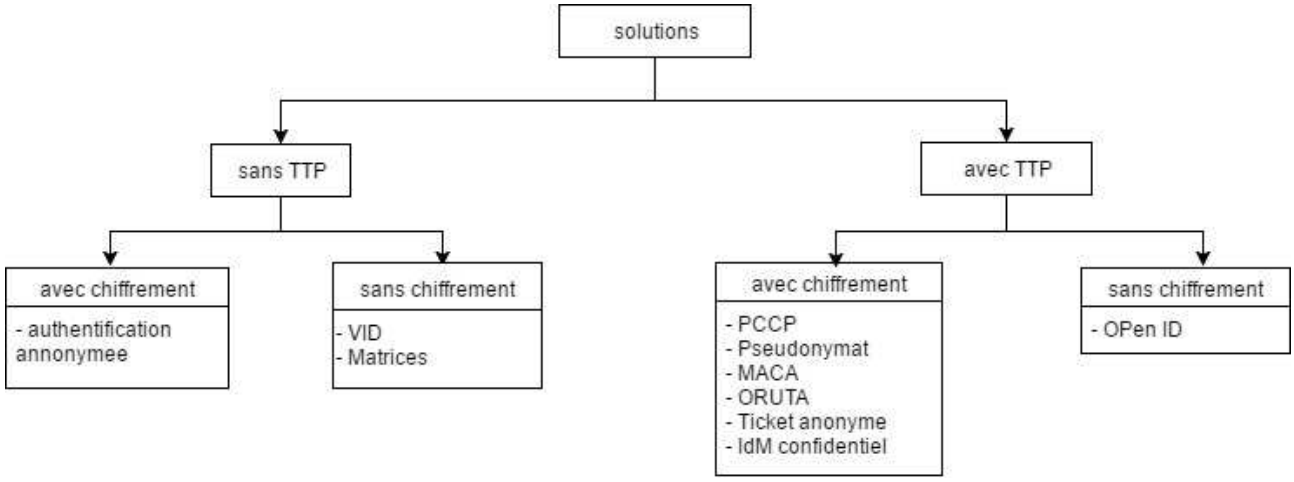


FIGURE 3.1 – Classification des protocoles étudiés.

### 3.3.1 Solutions sans TTP

#### 3.3.1.1 Authentification anonyme

L'article [48] présente un schéma d'authentification anonyme sans certificat, ce système permet d'éviter les blocages des clés dans les schémas d'authentification basés sur des certificats. Cette approche est utilisée dans le commerce électronique, pris en charge par le cloud computing, où les utilisateurs ne veulent pas exposer leurs identités et ils espèrent simplement que les fournisseurs de services sachent qu'ils sont des utilisateurs légitimes.

Avant d'entamer le processus d'authentification, l'utilisateur sélectionne d'abord sa clé privée notée x, après il calculera sa clé publique $PK = g^{1/x}$. Le processus d'authentification anonyme est décrit par les étapes suivantes :

- L'utilisateur envoie une demande d'authentification au serveur cloud qui contient l'adresse de l'utilisateur ;
- Une fois la demande d'authentification est reçue, le serveur cloud génère un nombre aléatoire et transmet à l'utilisateur ;
- Lorsque l'utilisateur reçoit le nombre aléatoire, il démarre le processus d'au-

thentification basé sur l'identité et envoie au serveur cloud $PK = g^{1/x}$ ("g" est un générateur d'un groupe cyclique [20]) ;

- Le serveur cloud sélectionne un nombre $y \in RZ^*P$ et l'envoie à l'utilisateur ;
- L'utilisateur calcule $A = g^{x/y}$ et l'envoie au serveur cloud ;
- Le serveur cloud vérifie si $e(A, PK^y) = 1$ est vrai ("e" est une application bilinéaire [20]). Dans le cas échéant, l'utilisateur démarre l'étape suivante sinon le serveur cloud arrête le processus d'authentification basé sur l'identité ;
- Après avoir vérifié l'identité de l'utilisateur, le serveur cloud et l'utilisateur négocient une clé de session K, qui sera envoyée de manière sécurisée à l'utilisateur ;
- Lorsque l'utilisateur reçoit le message réussi, l'utilisateur et le serveur cloud auront ainsi établi une connexion avec succès. L'utilisateur peut transmettre des données vers le cloud. Au cours de la procédure de transmission, la fonction hash est utilisée pour chiffrer le champ de données à transmettre.
- Lorsque l'utilisateur quitte la session, il envoie un message de déconnexion au serveur cloud . Ce dernier met fin à la connexion.

Le processus d'authentification anonyme est illustré dans la figure 3.2.

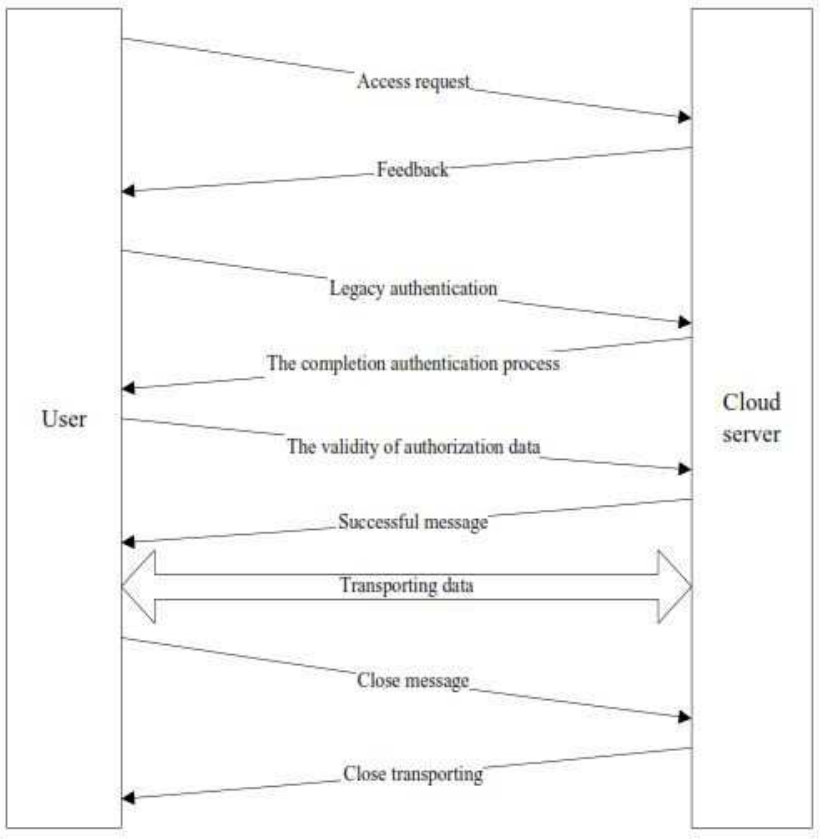


FIGURE 3.2 – Processus général d'authentification anonyme [48].

Dans cette approche la confidentialité des utilisateurs est assurée par l'authentification anonyme. Tandis qu'il présente un inconvénient à savoir la traçabilité.

#### 3.3.1.2 Les identités virtuelles

L'article cité dans [30] présente une approche importante pour la manipulation de la confidentialité des utilisateurs dans les systèmes ubiquitaires. Cela est atteint grâce aux identités virtuelles VID (Virtual IDentities) qui sont utilisées pour dissimuler l'identité réelle de l'utilisateur. Cette approche a été explorée dans le système

Daidalos [44] qui est un projet de recherche européen dont l'objectif est d'assurer le pseudonymat en utilisant les VID.

L'approche Daidalos utilise les préférences de l'utilisateur pour sélectionner les identités virtuelles. Ces VID forment un sous-ensemble d'entités de l'utilisateur qui sont utilisées pour s'authentifier via un service ; c'est-à-dire, tout utilisateur possédant un ensemble des VID est considéré comme un ensemble de différents noms d'utilisateur qui dissimule tout ou une partie de sa véritable identité. Lorsque l'utilisateur demande un service, le système l'authentifie par défaut avec son VID approprié au service demandé. Une fois que l'utilisateur est authentifié, il peut demander le service. Un VID peut être créé de deux façons. Explicitement par l'utilisateur en utilisant une interface graphique ou implicitement par la configuration basée sur des préférences spécifiques.

L'avantage de cette approche est qu'elle assure la confidentialité des utilisateurs par le pseudonymat de Daidalos. Ce dernier utilise un système d'identités virtuelles (VID) pour cacher la véritable identité de l'utilisateur. Cependant, le problème de cette approche consiste à savoir comment déterminer les mesures que l'utilisateur doit engager dans la prise de décision relative à la sélection des identités virtuelles. Si elle est complètement automatique, il sera difficile pour l'utilisateur de modifier en cas de besoin ; si elle est entièrement manuelle, il sera trop difficile pour l'utilisateur de faire le choix du VID.

#### 3.3.1.3 Les Matrices

L'article [22] se base sur la question de l'externalisation sécurisée des services. Il propose une approche d'obfuscation (brouillage) des données pour transformer les données privées en public sans l'utilisation de clé de cryptage dans le contexte du cloud computing. L'objectif principal de cette approche est de permettre aux serveurs cloud d'effectuer un calcul sur des données non chiffrées du client sans révéler leurs valeurs réelles, c'est-à-dire tout en conservant la confidentialité des données.

Dans cette approche, il existe deux niveaux d'abstraction des données sensibles des clients : privé et public. Les données privées définies par "A" sont les données

d'origine qui ne doivent pas être divulguées à d'autres, alors que les données publiques définies par "AP" sont une représentation conservatrice des données privées. Cela signifie que "AP" peut être vu par n'importe qui, et ce n'est pas un secret. Cependant, il est difficile pour quelqu'un d'extraire la valeur privée de sa représentation publique. Pour cela, une méthode simple est proposée, une variation des techniques existantes, qui transforme les données privées "A" en données publiques "AP" avec un minimum d'effort, tout en protégeant la confidentialité de l'entrée sans cryptage à clé publique.

La figure 3.3 représente les deux fonctions complémentaires : Transformation et Extraction qui composent cette approche. Cette approche est conçue pour traiter des données de type entier, plus spécifiquement, un ensemble de matrices et leur multiplication. Les étapes de cette approche sont présentées ci-dessous :

1) Le client transforme les données privées "A" et "B" en données publiques "AP" et "BP" respectivement ;
2) Envoyer "AP" et "BP" au serveur en nuage ;
3) Le serveur cloud multiplie "AP" par "BP" et génère le résultat public "CP", qui sera envoyé au client ;
4) Le client reçoit la production publique "CP" ;
5) Le client applique la fonction d'extraction pour récupérer les données privées "C".

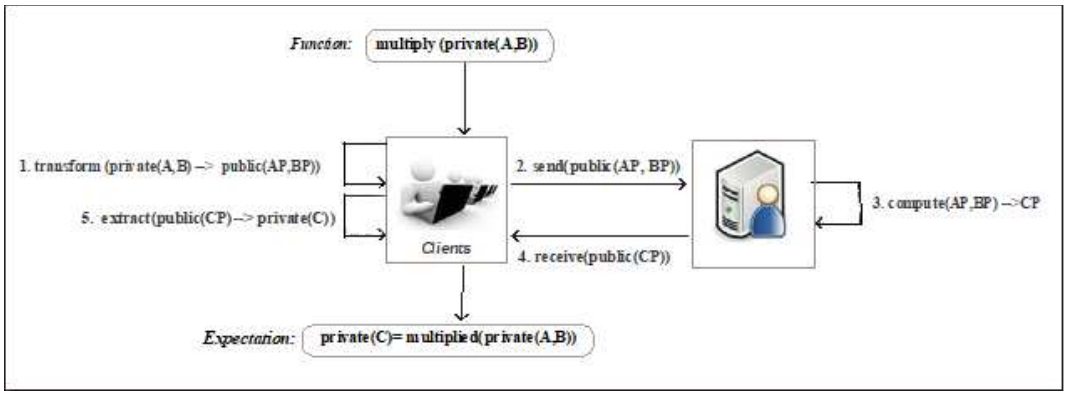


FIGURE 3.3 – Transformation et extraction de données privées et publiques [22].

Cette approche se compose de deux fonctions :

- ▷ **Transformation** : est la fonction de transformation qui modifie les données privées "A" en public "AP" en utilisant des multiplication matricielles successives, comme suit :
  1. Considérons une matrice A de taille $m \times n$ , et une autre matrice B de taille $n \times k$, où A et B sont compatibles pour la multiplication.
  2. Générer une matrice de 1 et 0 telle qu'il s'agit d'une matrice U de taille aléatoire $n * n$.
  3. Effectuez les multiplications comme suit :
     $A \times U = A^S$
     $U^T \times B = B^S$
  4. Générer deux matrices diagonales [1] aléatoires P et Q de taille $m \times m$ et $k \times k$ respectivement. Notons que les tailles sont compatibles avec les tailles des matrices A et B
  5. multiplier les matrices $A^S$ et $B^S$ avec les matrices diagonales inversibles P et Q respectivement comme suit : $A' = P \times A^S$ $B' = B^S \times Q$
  6. Générer une autre matrice diagonale aléatoire D de taille $n \times n$, et obtenir son inverse, $D^{-1}$
  7. Multiplier les éléments suivants :
     $AP = A' \times D$
     $BP = D^{-1} \times B'$
  8. Envoyer AP et BP au serveur en nuage.
- ▷ **Extraction** : Le serveur cloud multiplie AP par BP et génère le résultat CP qui est public. Le serveur renvoie la sortie CP au client. Le client reçoit CP et extrait le C privé qui est supposé être la sortie de $A \times B$. Afin d'extraire les valeurs privées de CP, le client doit calculer l'inverse de P et Q. Pour obtenir les données privées C, le client effectue le calcul suivant :
  $P^{-1} * CP * Q^{-1} = C = A \times B$.

1. Une matrice diagonale : est une matrice carrée dont les termes situés hors de la diagonale principale sont tous nuls [4].

L'avantage de cette approche est que le serveur cloud n'apprend rien sur les valeurs privées des données du client. En d'autres termes, il ne connaît pas la valeur privée cachée dans la sortie qu'il calcule, ,et cela est granit sans la nécessite d'utilisé le cryptage à clé publique. Cependant, cette approche est encore dans ses débuts, il faut plus de recherche sur les techniques alternatives pour protéger et cacher les données sans cryptage. De plus, elle est limitée pour les entiers.

### 3.3.2 Solutions avec TTP

#### 3.3.2.1 Preserving Cloud Computing Privacy

L'article cité dans [34] propose un modèle appelé PCCP qui intègre une architecture à trois niveaux : la couche de consommateur (consumer layer), la couche de mappage d'adresse (Address Mapping layer) et la couche confidentialité préservée( privacy preserved layer), visant à préserver la confidentialité des informations relatives aux utilisateurs de cloud :

- La couche de consommateur traite tous les aspects qui permettent à l'utilisateur d'accéder aux services de cloud.
- La couche de mappage d'adresse crée une correspondance entre les adresses IP (Internet Protocol address) d'origine des utilisateurs avec une adresse IP modifiée. Elle accomplit cela en mappant les adresses IP d'origine aux adresses OTIP (Owned Translated I P address). En utilisant ce mappage, le fournisseur de services cloud est empêché d'accéder aux adresses IP d'origine, la confidentialité de l'adresse IP d'origine de l'utilisateur est donc assurée. Ces OTIP ainsi que l'horodatage sont utilisés pour générer un USID (User Service Dependent Identity)à partir d'un pool d'ID à l'aide d'une fonction de mappage [2].
- La couche confidentialité préservée génère une identité unique USID pour les utilisateurs du cloud. Pour cela un algorithme est proposé pour générer

2. Chaque fonction de mappage définit une méthode différente pour transformer une donnée A en une donnée B.

une identité unique dépendante de l'utilisateur en établissant le mappage des identités des utilisateurs.

La figure 3.4 illustre l'approche PCCP.

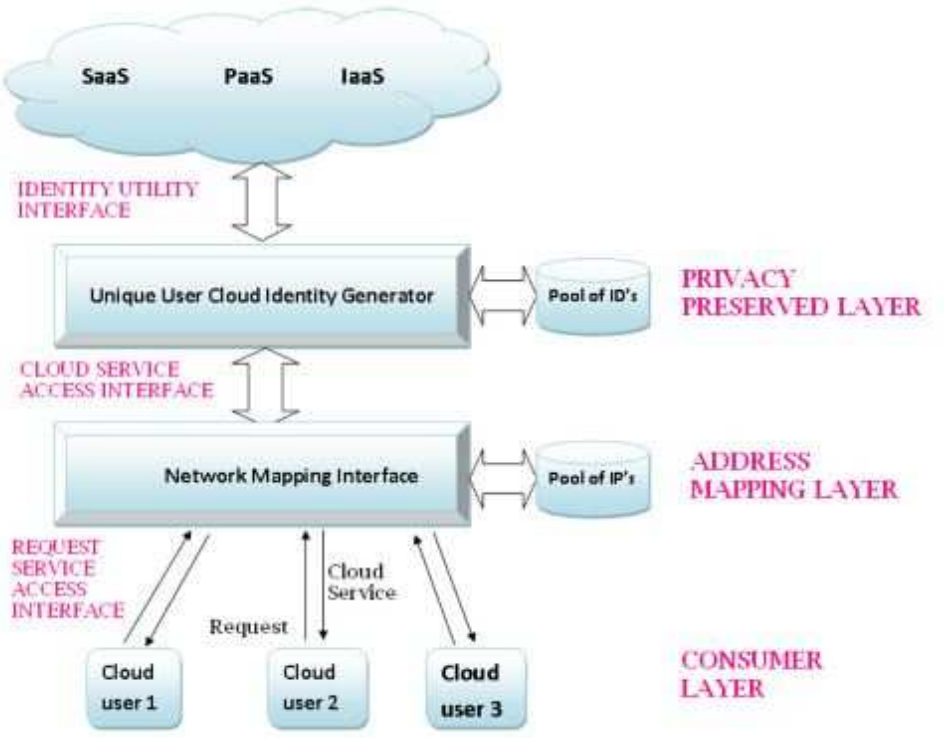


FIGURE 3.4 – Le modèle Preserving cloud computing Privacy [34].

Les différentes couches qui ont été proposées dans ce modèle sont fournies par les fournisseurs de cloud eux-mêmes avec les contributions pertinentes des utilisateurs. Par exemple, la couche confidentialité préservée qui comprend la fonctionnalité de la génération USID nécessite que l'utilisateur fournisse une clé comme entrée qui sera utilisée pour déterminer le degré d'obscurité des données partagées pour que les informations sensibles soient partagées par l'utilisateur de cloud avec le fournisseur cloud sans compromettre la confidentialité des informations personnelles.

Le modèle proposé résout les problèmes de confidentialité des utilisateurs, en empêchant la dérivation de l'identité de l'utilisateur sur la base d'informations secondaires telles que les adresses IP et les identités des utilisateurs. Cependant, l'utilisateur doit contrôlé la visibilité de ses données vis-à-vis du fournisseur cloud ce qui peut compliquer la tâche de l'utilisateur, ainsi l'approche ne sera pas conviviale pour l'utilisateur.

#### 3.3.2.2 Pseudonymat

L'auteur de [37] présente le concept de "Pseudonymisation" pour la protection des données médicales. Ce concept est un processus où l'identificateur d'une personne et ses données personnelles sont remplacés par un pseudonyme.
Il existe deux techniques pour la pseudonymisation :

- Par le service national de pseudonymisation NSP (National Pseudonymisation Service) comme un tiers de confiance qui est conçu pour couvrir les échanges des identifiants locaux entre une source de données personnelles et une destination de stockage.
- Par un algorithme, pour la création d'un pseudonyme d'un identifiant d'une personne de manière locale (interne) ce qui signifie que le pseudonyme est calculé soit à la source de données (chez le client) ou de la destination de stockage (chez le cloud).

Dans les deux cas (NSP ou en interne pseudonymisation) un numéro de pseudonyme doit être calculé ou déterminé à un instant donné. Il y a plusieurs options pour créer un pseudonyme avec un ensemble determiné de données démographiques. Certaines de ces techniques se basent sur le hachage ou le chiffrement du numéro d'identification unique de la personne. D'autres choisissent simplement un nombre aléatoire et le lie avec l'identité. Les algorithmes de hachage et de chiffrement actuels fonctionnent avec 128 bits minimum ce qui pourrait être trop lourd donc ne sont pas appropriés dans le cas décrit, mais un recadrage du résultat à 32 bits peut provoquer des collisions de pseudonyme.

Cependant pour ses techniques de création de pseudonymes, il est difficile d'estimer comment sécuriser ces algorithmes, car la cryptanalyse [3] sur des algorithmes de chiffrement symétriques, algorithmes existants et de hachage ont montré des faiblesses qui peuvent être trouvées et exploitées pour des attaques (calcul du l'identité réelle à partir de pseudonyme). Par conséquent, un algorithme de calcul de pseudo alternative est proposé pour le calcul des pseudonymes à partir d'un identifiant de personne basé sur le chiffrement asymétrique (par exemple : l'algorithme RSA ou le Diffie-Hellman). L'algorithme garantit une distribution pseudo-aléatoire sans collision des pseudonymes. L'algorithme de pseudonymisation agit comme une fonction à sens unique si tous les paramètres de calcul sont gardés secrets.

#### 3.3.2.3 MACA

L'article [25] présente "MACA" (Multi Factor Cloud Authentication), qui est la première approche d'authentification à facteurs multiples (MFA) qui prend en compte la confidentialité des utilisateurs. Dans MACA, le premier facteur est un mot de passe tandis que le deuxième facteur est un profil hybride du comportement de l'utilisateur. Le profil hybride est basé sur le comportement intégré des utilisateurs. MACA comporte quatre composants principaux :

(1) Un programme d'acquisition des profils open source PAP (open-source profile acquisition program) qui s'exécute sur l'hôte local de l'utilisateur ;

(2) Une base de données de profils d'utilisateur UPDB (user profile database) qui stocke les informations de l'utilisateur de manière à préserver la vie privée ;

(3) Un serveur d'authentification AS(authentication server) qui traite et valide la demande de connexion de l'utilisateur dans l'environnement cloud ;

(4) Un serveur cloud (content server).

La première fois qu'un utilisateur utilise MACA avec un site spécifique, il doit passer par le processus d'inscription. Lors de l'inscription, le programme d'acquisition d'informations utilisateur recueille un profil d'utilisateur désigné par "P". Puis,

3. La cryptanalyse est la science qui consiste à décrypter un message chiffré.

il crypte "P" avec la clé publique "FHE pk". Ensuite, l'ID et le mot de passe attribué aux utilisateurs désignés par "uid" et "Psw", ainsi que le profil d'utilisateur cryptographique désigné par "PC" sont transmis à l'AS. Ce dernier interagira avec l'UPDB et insère "PC" dans la base de données du profil utilisateur. Pour lancer une tentative d'authentification, l'utilisateur passe son "uid" et son "Psw" pour l'authentification du premier facteur. L'échec de l'authentification du premier facteur met fin à la conversation. Si l'utilisateur passe la première étape de l'authentification, il passe son nouveau profil utilisateur désigné par "NPC" au serveur AS. Après avoir évalué la différence entre "NPC" et "PC", le serveur AS renvoie une valeur booléenne "AuthResult", indiquant le succès ou l'échec de l'authentification du second facteur. Si la valeur booléenne "AuthResult" est un succès, l'AS enverra un ticket de service de contenu avec "Authresult" à l'utilisateur qui contient un ID de session. Tout utilisateur qui détient un ticket de service valide peut accéder aux services de cloud.

Pour collecter des profils d'utilisateurs, MACA utilise deux programmes d'acquisition de fichiers utilisateur dans C et Python pour Windows et Linux OS, respectivement. Le programme comporte trois étapes principales :

- Récapitulation de données : Le bloc récapitulation de données est responsable de la collecte des informations de l'utilisateur dans une fenêtre coulissante, la collecte pour certaines informations utilisateur se produit en continu et à la fin de chaque période de la fenêtre coulissante les informations collectées sont transmises au bloc de dérivation des fonctionnalités et la récapitulation des données recommence.
- Dérivation de caractéristique : Le bloc dérivation de caractéristique reçoit les données brutes du bloc précédent et extrait les fonctionnalités requises. Une fois que chaque fonctionnalité est prête à être traitée, le bloc de dérivation des entités passe les données au bloc suivant.
- Cryptage par hachage : Le bloc cryptage par hachage est responsable de générer un profil cryptographique basé sur toutes les fonctionnalités disponibles via un hachage flou et cryptage homomorphique. Le profil utilisateur haché flou est désigné par "PF" tandis que le profil d'utilisateur homomorphiquement chiffré est désigné par "PH". Ainsi, le bloc affiche un profil d'utilisateur cryptographique hybride "$PC \longleftarrow \{PF, PH\}$".

En outre, le bloc de dérivation des fonctionnalités continue de produire des informations sur les fonctionnalités à la fin de chaque période de fenêtre.
La figure 3.5 illustre l'approche d'authentification à facteurs multiples MACA.

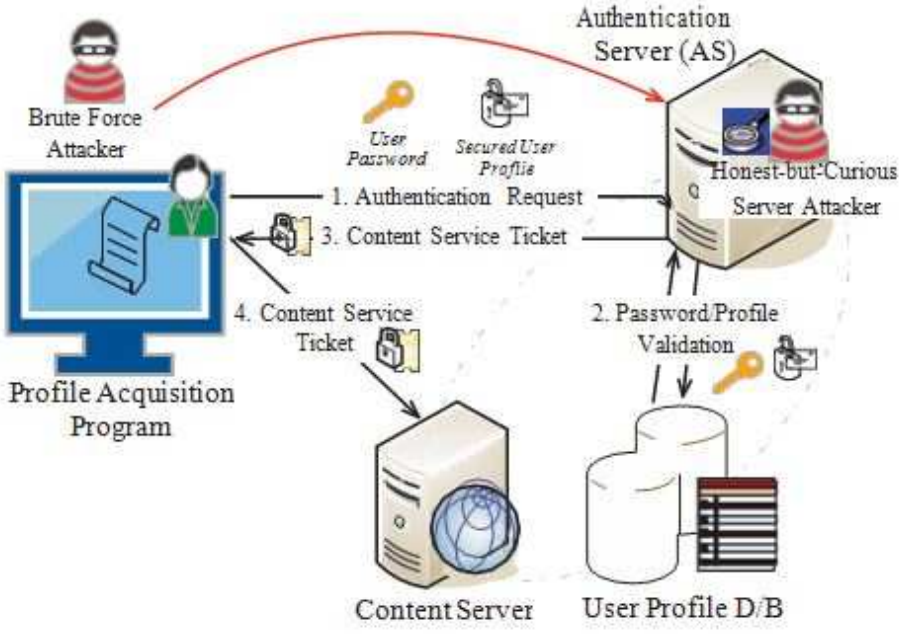


FIGURE 3.5 – Vue d'ensemble de l'approche MACA [25].

L'approche proposée permet de préserver la confidentialité des utilisateurs, tandis que l'un des inconvénients de cette approche est qu'une entité malveillante peut attaquer le système d'authentification multifacteurs par force brute (L'attaque par force brute est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé) ce qui peut conduire à une usurpation d'identité.

#### 3.3.2.4 ORUTA

L'article cité dans [9] présente une approche appelée " Oruta " (One Ring to Rule Them All) pour résoudre le problème de confidentialité des utilisateurs et des données partagées dans le cloud. Oruta comporte trois partis : le serveur cloud, un auditeur tiers (TPA) et un groupe d'utilisateurs. Il existe deux types d'utilisateurs : les propriétaires de données qui sont impliqués dans le processus d'audit et les utilisateurs du groupe qui sont membres du groupe. TPA vérifiera la sécurité des ressources pour le compte des utilisateurs du cloud. Oruta comprend une signature en anneau qui est essentielle pour cacher l'identité des utilisateurs vis-à-vis des vérificateurs externes.

Lorsqu'un utilisateur souhaite accéder aux services cloud. L'utilisateur envoie d'abord une demande de vérification à un auditeur tiers (ou tiers vérificateur). À la réception de la demande, TPA crée un message d'audit et l'envoie au cloud. Après avoir reçu le message d'audit, le serveur cloud vérifiera la demande et générera une preuve d'audit et la transmettra à la TPA. Ce dernier analysera les preuves reçues par le serveur cloud et doit générer un rapport d'audit à l'utilisateur demandé. Le rapport d'audit indiquera si l'utilisateur qui a demandé la vérification est authentifié pour utiliser les données confidentielles dans le cloud. Au cours de cette phase, l'auditeur tiers peut collecter des informations confidentielles sur l'identité des utilisateurs, pour remédier à ce problème, le système Oruta utilise la signature en anneau. La figure 3.6 illustre l'approche Oruta.

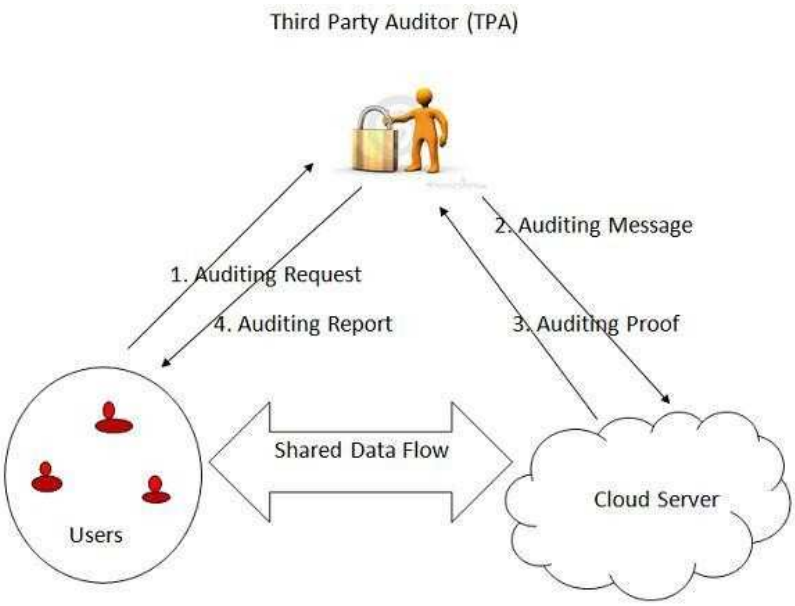


FIGURE 3.6 – Modèle de processus du système Oruta [9].

En utilisant la signature en anneau [4], l'identité de chaque utilisateur est cachée aux vérificateurs publics, c'est à dire cacher le générateur de clé réelle sur chaque bloc. Si le propriétaire de la donnée souhaite récupérer le fichier externalisé à partir du cloud, tous les utilisateurs du groupe doivent fournir leurs clés dans l'ordre correct pour former une structure en anneau. Enfin, les clés des utilisateurs sont traitées pour le téléchargement de fichiers.

En utilisant ORUTA, le vérificateur public ne peut identifier l'identité des utilisateurs sur les données partagées et prend en charge la confidentialité des données. L'avantage d'utiliser la signature en anneau est qu'il est difficile de calculer quelles clés des membres du groupe sont utilisées pour produire la signature.

4. La signature en anneau (en anglais : Ring signature), aussi appelé signature de cercle, est un procédé cryptographique permettant à une personne de signer électroniquement de façon anonyme un message ou un document au nom d'un " cercle ". Les membres de ce cercle sont choisis par l'auteur de la signature et ne sont pas nécessairement informés de leur implication dans la création de la signature électronique.

Ce système prend en charge les groupes dynamiques. Lorsqu'un utilisateur quitte le groupe, le propriétaire des données ne peut pas récupérer les données confidentielles stockées dans un cloud. Ce problème est résolu dans cette approche qui prend en charge les groupes d'utilisateurs dynamiques, où un nouvel utilisateur peut être ajouté en groupe et un membre du groupe existant peut être révoqué lors de la phase d'audit public. L'utilisateur révoqué ne peut pas générer de clés pour la signature en anneau. Lorsqu'un nouvel utilisateur rejoint le groupe sa clé est utilisée pour calculer la nouvelle signature en anneau dans le bloc modifié. Ainsi la confidentialité de l'identité des utilisateurs est conservée dans des groupes dynamiques. D'où le système ORUTA préserve l'identité des utilisateurs.

#### 3.3.2.5 Ticket anonyme

L'article [17] propose une approche pour assurer la confidentialité de l'identité des patients grâces à une authentification anonyme fournissant un accès anonyme dans un cloud e-Health. Cette approche est utilisée dans le domaine médical où la révélation d'identité d'un patient est une violation de sa vie privée, même si ses données médicales sont confidentielles (souvent cryptées). Le but de l'approche est d'obtenir l'autorisation d'accès en fournissant des tickets anonymes pour permettre aux patients de faire des demandes anonymes sur la consommation.

Cette approche comporte trois composants principaux :

(1) l'utilisateur est un patient qui utilise les applications e-santé et les services de stockage ;

(2) Le gestionnaire d'enregistrement (RTM) qui est responsable de l'enregistrement d'un nouvel utilisateur, il doit être en mesure de traiter avec succès l'enregistrement des différents patients.Aussi, il attribue des tickets d'accès anonyme via la signature aveugle [5] basée sur le système RSA ;

(3) Le gestionnaire de services (SM) fournit la consommation de services après avoir reçu Access Ticket (AT) du patient, le Access Ticket est vérifié avec

5. En cryptographie, une signature aveugle, telle que définie par David Chaum1, est une signature effectuée sur un document qui a été masqué avant d'être signé, afin que le signataire ne puisse prendre connaissance de son contenu.

RTM avant de permettre au patient d'accéder et de démarrer la procédure de consommation de services.

Les différentes étapes que les utilisateurs de cloud e-Health (souvent les patients) suivront pour être authentifiés dans ce système est comme indiqué dans la figure 3.7 :

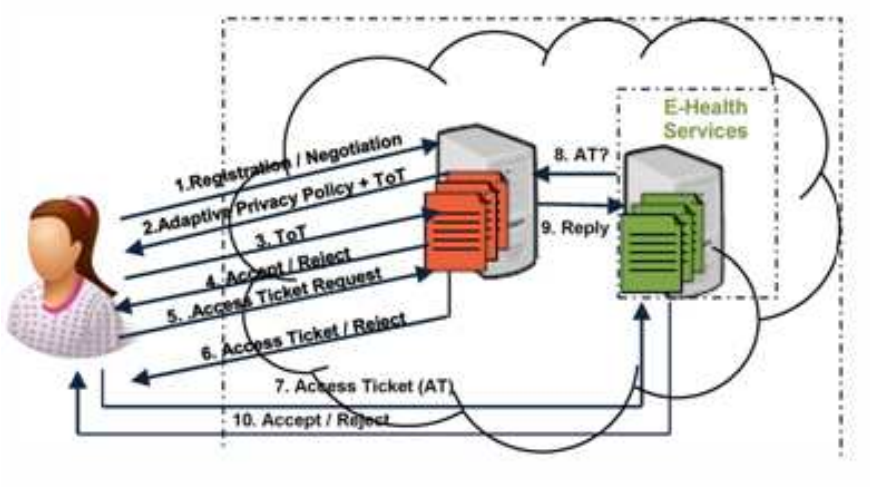


FIGURE 3.7 – Aperçu du schéma d'authentification proposé [17].

**Étape 1** Enregistrement et génération de la politique de confidentialité adaptative du patient : l'objectif de cette étape est de présenter le cloud e-Health aux utilisateurs potentiels (souvent les patients), le SLA et la Politique de confidentialité à leur disposition. Ceci est fait pour les familiariser et les motiver à rejoindre ce cloud e-Health pour bénéficier de ses services, y compris être anonyme. Le résultat sera d'identifier et d'enregistrer les patients concernés et de générer la Politique de confidentialité adaptée aux exigences de chacun. à la fin de cette étape, RTM remet un ticket initial nommé ToT pour "Ticket of Ticket", qui représente la preuve d'inscription.

**Étape 2** Obtention d'un AT "Access Ticket " : Après l'achèvement de l'inscription, le patient obtiendra un ticket anonyme (nommé Anonymous Access Ticket) via la technique de la signature aveugle lui permettant de consommer des services

de cloud ultérieurement. aini chaque utlisateur consomme le service anonymement sans révéler son identité auprès du fournisseur de services cloud(CSP). Ce ticket anonyme est considéré comme l'information d'identification dans ce système d'authentification.

**Étape 3** Consommation du service e-Health souhaité : dans cette étape, le patient devra seulement présenter son ticket d'accès obtenu à l'étape 2 pour demander un service donné (stockage pour accéder à ses données, applications de santé en ligne, etc.). CSP ne sait pas quel patient demande l'accès aux services, mais seulement qu'un patient légitime veut accéder.

L'avantage de cette approche est que même si les deux gestionnaires( RTM et SM) communiquent, la divulgation de l'identité du patient reste difficile à réaliser et cela est atteint grâce à la consommation anonyme de services via des tickets anonymes, ce qui a permis à cette approche d'être indépendant de la notion de confiance qui exige un certain niveau de confiance pour l'assurance de l'anonymat de l'utilisateur. L'inconvénient de cette approche est que l'adresse IP de l'utilisateur est visible pour le fournisseur de services.

#### 3.3.2.6 Le modèle IdM confidentiel

L'article [43] présente un modèle de confidentialité nommé PIdMM (Privacy Idenity Management Model) pour les systèmes de gestion d'identité dans le cloud, ce modèle traite les problèmes de confidentialité, en particulier les menaces sur des données sensibles comme les PII, cela est due essentiellement au manque de contrôle sur la diffusion des attributs des utilisateurs. Ce prototype prend en compte les aspects dynamiques des fédérations, des différentes politiques, des règles de manipulation de données et du cryptage, et il est réalisé à l'aide de OpenID Connect (OIDC) [7].

La gestion de l'identité fédérée permet d'administrer l'accès aux services tout en assurant la confidentialité des PII. Les entités impliquées dans PIdMM sont :

(1) les utilisateurs qui souhaitent accéder à certains services / ressources ;

(2) les fournisseurs d'identité (IdP) qui exécutent des processus d'authentification, gèrent et diffusent les PII des utilisateurs ;

(3) Les fournisseurs de services (SP) qui fournissent le service / ressource.

PIdMM est divisé en couches pour satisfaire toutes les caractéristiques souhaitées de la vie privée dans le contrôle d'accès. L'idée est d'offrir des paramètres de confidentialité prédéfinis par défaut de manière personnalisés, ce qui aide les utilisateurs à déclarer leur niveau de confidentialité préféré, en fonction du service demandé. Le modèle devrait traiter les limites de la législation, les interactions entre les parties en utilisant les politiques et l'audit.

La figure 3.8 présente les interactions du PIdMM dans le cloud. Le flux comprend des environnements dynamiques où des milliers d'utilisateurs dans différents domaines administratifs interagissent avec les systèmes de gestion d'identité (IdM). Cependant, le contrôle d'accès doit être effectué par un système IdM fédéré qui prend en charge les préférences et les profils d'accès des différents utilisateurs. Par conséquent, il est nécessaire que IdM permet à l'utilisateur d'enregistrer les PII dans son IdP, en choisissant des critères d'utilisation et de diffusion de ses données.

- À l'étape 1, l'utilisateur enregistre ses attributs et ses informations d'identification, avec ou sans chiffrement ce qui empêche les IdP d'utiliser ou de divulguer les données PII sans le consentement et la connaissance de l'utilisateur.
- À l'étape 2, IdP permet à l'utilisateur de définir sa politique de confidentialité pour réglementer l'utilisation et la diffusion de ses données. Ainsi, les utilisateurs pourraient accéder aux applications en envoyant uniquement les données nécessaires.
- L'étape 3 Présente à l'utilisateur les modèles convenus de diffusion dynamique de données (DDM) entre SP (Service Provider) et IdP(Identity Provider) ainsi l'utilisateur est informé à l'avance des risques et des paramètres personnel que le fournisseur voudra utiliser. L'IdP présente à l'utilisateur des modèles d'accès possibles disponibles pour les interactions : accès anonyme, accès pseudonyme ou accès avec des attributs minimaux choisis.

- À l'étape 4, après toutes ses définitions, le système regroupe dans un paquet les règles et obligations auxquelles les SP doivent satisfaire.
- À l'étape 5, IdP transmet un paquet contenant des politiques et des obligations au SP, le fournisseur s'assure que les politiques représentant les préférences des utilisateurs seront accomplies.
- Enfin, à l'étape 6, l'utilisateur accède à l'application.

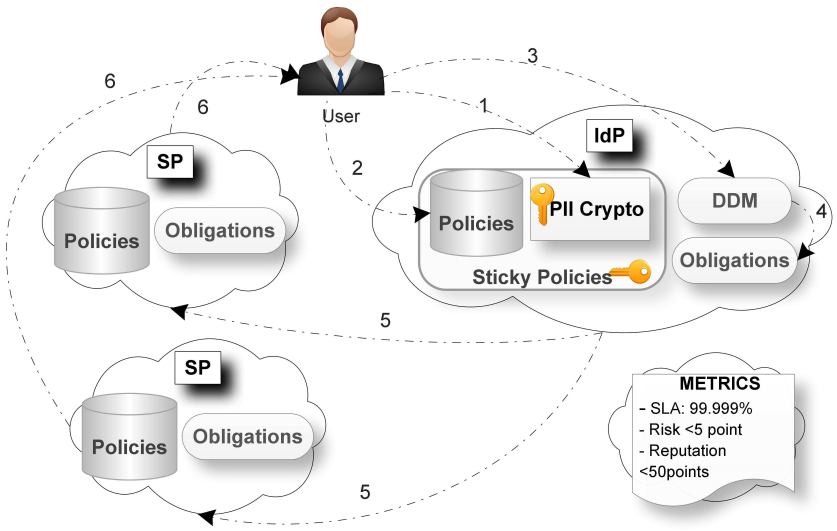


FIGURE 3.8 – Les interactions du modèle IdM [43].

L'avantage de ce modèle est qu'il offre des paramètres de confidentialité prédéfinis par défaut de manière personnalisés comme la possibilité de cryptage des PII ; l'utilisation de méthodes pour cacher les identités telles que l'anonymat ou le pseudo-anonymat. Ainsi, les utilisateurs auront plus de contrôle sur l'utilisation de leurs données personnelles. L'inconvénient de cette approche est que l'utilisateur est un acteur majeure dans ce modèle.

#### 3.3.2.7 OpenID

L'article cité dans [36] présente openID qui est une norme technologique définissant une authentification décentralisée pour permettre aux utilisateurs de se connecter à plusieurs sites avec le même compte. Il est supporté de plusieurs grandes organisations telles que AOL, Google, Microsoft, VeriSign et Yahoo.

Lorsque les utilisateurs créent un compte dans un fournisseur OpenID (c'est-à-dire le fournisseur d'identité) ils reçoivent un identifiant comme une URL (Uniform Resource Locator)ou XRI (Extensible Resource Identifier). Puis, quand ils accèdent à un site Web qui exige une authentification et supporte OpenID, ils peuvent entrer leur identifiant dans le but d'être redirigés vers leur fournisseur OpenID .Il est important de dire que cet identifiant est généralement unique pour chaque utilisateur (par exemple alice.myopenid.net). Par conséquent, le fournisseur de services pourrait tracer l'accès de l'utilisateur final, car il utilise toujours le même identifiant. Dans le fournisseur OpenID, les utilisateurs peuvent utiliser leur nom utilisateur et mot de passe pour effectuer le processus d'authentification. Après vérification des informations d'identification de l'utilisateur, le fournisseur OpenID affiche une page de confirmation où l'utilisateur peut vérifier et sélectionner les informations qui peuvent être partagées avec le fournisseur de service. Enfin les utilisateurs sont redirigés au fournisseur de services et ce dernier peut avoir les informations demandées sur les utilisateurs, tout en gardant la confidentialité du mot de passe et de l'identité réelle. La figure 3.9 présente OpenID.

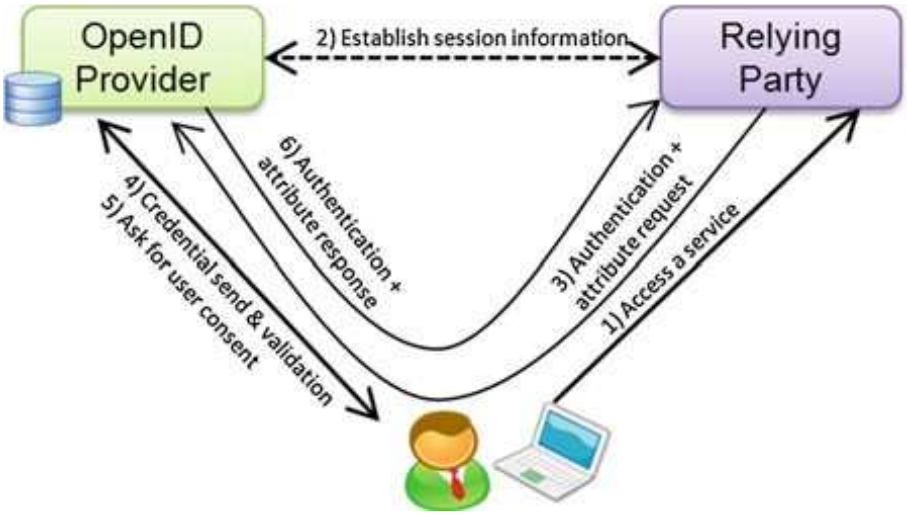


FIGURE 3.9 – Processus général de openid [36].

l'inconvénient de cette norme est que le fournisseur d'identité peut retracer les accès de tous les utilisateurs, car ils doivent générer une affirmation à chaque fois qu'ils ont besoin d'accéder à un fournisseur de services. On a aussi le cas ou le fournisseur d'identité est attaqué donc le mot de passe d'un utilisateur peut être volé, ainsi l'attaquant pourrait avoir accès à son fournisseur service avec le nom de l'utilisateur victime.

## 3.4 Synthèse

Nous avons remarqué que certains protocoles étudiés sont plus performants que d'autres. Afin de mieux comprendre la diversité des protocoles étudiés dans ce chapitre qui traitent le problème de confidentialité des utilisateurs, nous avons proposé une comparaison basée sur les critères cités précédemment. Le tableau résume cette comparaison entre les travaux étudiés dans ce chapitre ; le signe (✓) signifie que le protocole assure la fonction ou résout le problème, contrairement au signe (x).le signe (-) signifie que le problème n'a pas été étudié dans le protocole. Nous avons

ajouté notre proposition, dénommée PIICMM (Personally Identifiable Information Confidential Management Method) , qui sera présenté dans le chapitre suivant.

| | | confidentialité des PII | facilité d'utilisation | intraçabilité | authentification sure |
|---|---|---|---|---|---|
| sans TTP | Authentification anonyme [48] | ✓ | ✓ | x | ✓ |
| | VID [30] | ✓ | x | x | - |
| | Matrices [22] | - | ✓ | - | - |
| avec TTP | PCCP [34] | ✓ | x | ✓ | - |
| | Pseudonymat [37] | ✓ | ✓ | x | ✓ |
| | MACA [25] | ✓ | ✓ | x | x |
| | ORUTA [9] | - | ✓ | x | ✓ |
| | Ticket anonyme [17] | ✓ | ✓ | x | ✓ |
| | IdM confidentiel [43] | ✓ | x | x | ✓ |
| | openID [36] | ✓ | ✓ | x | x |
| | PIICMM | ✓ | ✓ | ✓ | ✓ |

TABLE 3.1 – Comparaison des protocoles étudiés.

## 3.5 Conclusion

Après l'étude de quelques solutions proposées, nous avons pu identifier deux catégories de protocoles : protocoles avec TTP et protocoles sans TTP. Ainsi, nous avons pu soulever précisément les problèmes auxquels nous devons faire face pour obtenir une solution qui répond aux critères cités précédemment et qui présentera de meilleurs résultats.

Le chapitre qui suit fera l'objet de notre contribution qui est basée sur l'utilisation d'une tierce partie de confiance comme gestionnaire d'identité, afin de garantir la confidentialité des PII des utilisateurs.

Chapitre 4

# Proposition et validation de notre système PIICMM

## 4.1 Introduction

Le problème de la vie privée aurait pu être résolu si les lois sur la protection de la vie privée et la propriété de données étaient les mêmes dans tous les pays, ce qui faciliterait l'usage du cloud en toute confiance. Toutefois le "Patriot Act" complique les choses en imposant à toute entreprise dont le siège est aux États-Unis d'ouvrir ses centres de données sur demande du gouvernement américain, même s'ils sont situés dans un autre pays [19]. Par conséquent, la protection des données dans le cloud paraît insuffisante, notamment l'identité de l'utilisateur du cloud qui doit être protégée contre le fournisseur de service et les autres clients, c'est dans ce contexte que nous apportons notre collaboration afin d'améliorer la confidentialité des utilisateurs. Et cela en protégeant les informations personnelles identifiables contre le fournisseur cloud ; ainsi notre problématique est la suivante : comment peut-on assurer une authentification anonyme des utilisateurs sans être obligé de fournir leurs PII au fournisseur de service ?

Nous consacrons ce chapitre à la présentation de notre solution, au choix des différentes technologies sur lesquelles elle s'appuie, ainsi qu'à son fonctionnement illustré par des scénarios d'exécution. Ensuite, nous simulerons la sécurité de notre proposition avec l'outil Scyther afin de vérifier si notre proposition assure une authentification anonyme et sure.

## 4.2 Préliminaires

Nous allons définir quelques notions qui sont la base de la solution proposée dans ce chapitre :

### 4.2.1 Mappage d'adresses IP

Le mappeur crée une correspondance entre les adresses IP d'origine des utilisateurs avec une adresse IP modifiée.
Quand une demande de service arrive au mappeur, R-sa-I (Request Service Access Interface) récupère cette demande et interagit alors avec N-map-I (the Network Mapping Interface), qui est disponible dans le mappeur. Le N-map-I maintient un pool d'adresses IP. Le N-map-I crée alors une correspondance appropriée entre l'adresse IP d'origine de l'utilisateur et une adresse IP modifiée MIP(Modified IP address). En utilisant ce mappage, le fournisseur de services cloud est empêché d'accéder aux adresses IP d'origine. Il est plutôt en mesure d'accéder uniquement aux adresses MIP. Dans ce processus, la confidentialité de l'adresse IP d'origine de l'utilisateur est assurée [34].

### 4.2.2 Signature aveugle cas de RSA

Nous adopterons la signature aveugle, qui est une technique permettant de consommer anonymement un service. L'objectif est de fournir des tickets anonymes. Cette méthode peut être utilisée dans cloud computing pour permettre aux utilisateurs de cloud de faire des requêtes de recherches anonymes lors de l'utilisation des services.
L'idée générale est que certaines parties (par exemple une tierce partie de confiance) émettent des tickets (également appelés information d'identification) aux clients pour utiliser anonymement un service.
Une construction simple d'une signature aveugle est basée sur le système RSA. Soient p, q deux grands nombres premiers, et N = p.q ; et soient e, d deux entiers satisfaisant : $e.d = 1 mod \phi(N)$.

Soit Pk = (N, e) est la clé publique du signataire, authentiquement connu par le propriétaire du message m à signer.
Soit Sk = (N, d) la clé privée du signataire.
Une signature aveugle pour un message m peut être obtenue comme suit :

1. Le détenteur du message m, sélectionne d'abord un entier aléatoire r et calcule le message aveuglé $blind(m) = H(m).r^e(modN)$ où : H est une fonction de hachage. Ensuite, il envoie ce message au signataire.
2. Le signataire calcule la signature numérique pour le blind(m) :
   $S' = (blind(m))^d(modN)$, tel que : $(blind(m))^d = (H(m).r^e)^d = H(m)^d r^{e.d} = H(m)^d.r(modN)$
   Puisque r est inconnu du signataire et choisi d'une manière aléatoire, le message reste complètement secret.
3. Le propriétaire reçoit $S' = H(m)^d.r$ et le multiplie par $r^{-1}$. La signature résultante $S = S'.r^{-1} = H(m)^d(modN)$, est alors valable pour le message m.

## 4.3 Notre proposition

Notre proposition est basée sur les deux approches précédemment citées dans [34] et [17], elle repose sur une architecture centralisée et utilise une tierce partie de confiance appelée IdP(Identity Provider) caractérisée par la capacité de stocker les PII chiffrés des utilisateurs.

Notre solution est nommée méthode de gestion confidentielle des PII (PIICMM : Personally Identifiable Information Confidential Management Method ). Elle présente les différentes opérations faites par un client afin d'obtenir un ticket d'accès anonyme. Ce dernier lui permettra de s'authentifier et d'utiliser des applications cloud, tout en étant non identifiable. Ainsi le fournisseur de service ne saura rien sur l'identité réelle de l'utilisateur, sauf qu'il est un utilisateur légitime.

### 4.3.1 Exigences

Dans cette section, nous présentons certaines exigences que notre méthode doit prendre en charge afin de garantir une meilleure confidentialité des PII :

- Contrôle des PII : il s'agit de la notification concernant l'utilisation des attributs.
- Minimisation : l'utilisateur transmet uniquement les informations nécessaires pour accéder au service.
- Convivial : c'est le fait que le système de gestion d'identité facilite tout processus lié à l'authentification anonyme. En effet, cela ne peut être réalisé que si les utilisateurs ne sont pas tenus de compléter des procédures complexes ou de gérer des outils compliqués lors de l'authentification.

### 4.3.2 Entités

Les entités impliquées dans notre modèle sont :

▷ L'utilisateur : un être humain (client), qui est le consommateur d'un service.

▷ Request Service Access Interface(R-sa-I) : c'est un service fourni par le cloud permettant de récupéré les requêtes destinées au cloud et interagit avec le service concerné par la requête.

▷ Le fournisseur d'identités (IdP) : un service fourni par le cloud permettant l'enregistrement et l'authentification d'un utilisateur lors de l'accès à un service. Dans notre système, nous envisageons que ce fournisseur agira comme une tierce partie de confiance et il doit être en mesure de traiter avec succès l'enregistrement des différents clients, de plus il a pour tâche d'attribuer des tickets d'accès anonymes via la technique de signature aveugle aux différents clients enregistrés.

▷ Le fournisseur de service (SP) : c'est un service fourni par le cloud. Il permet aux utilisateurs authentifiés d'accéder au service.

▷ Le mappeur : un service fourni par le cloud, permettant de masquer/démasquer l'adresse IP de l'utilisateur.

### 4.3.3 Phases d'exécutions

Le modèle proposé s'exécute en deux phases essentielles : l'enregistrement et l'authentification.

- ▷ **Enregistrement :** Lorsqu'un utilisateur soumet une demande d'enregistrement au cloud, le R-sa-I récupère cette demande et la passe à IdP. Ce dernier affiche les services enregistrés dans sa base de données et leurs politiques à l'utilisateur. Celui-ci aura ensuite le choix d'accepter ou non les politiques de confidentialité du service choisi. S'il accepte, il transmet les PII demandés dans la politique de confidentialité à IdP. Ce dernier enregistre les PII dans sa base de données et envoie un ticket d'enregistrement à l'utilisateur. Voir la figure 4.1.

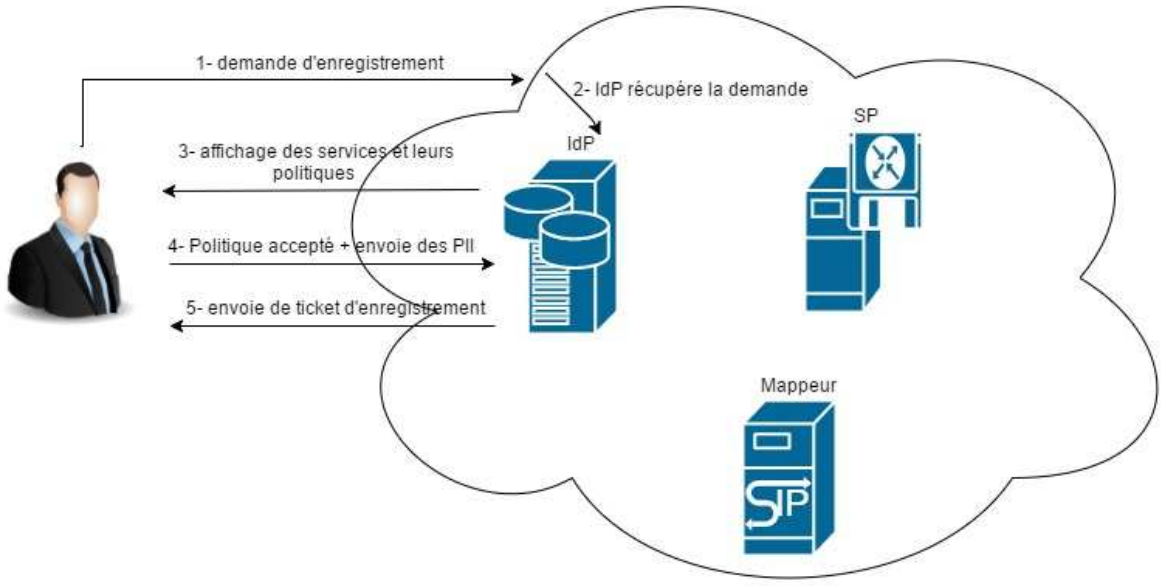


FIGURE 4.1 – Enregistrement d'un utilisateur

**Remarque :** les services proposés dans IdP sont enregistrés au préalable.

- ▷ **L'authentification :** Lorsqu'un utilisateur souhaite accéder à un service, il demande un ticket d'accès 'AT' en envoyant son ticket d'enregistrement

à IdP. Ensuite, IdP accepte ou rejette la demande selon la validité de TR .Si la demande est acceptée, IdP envoie le 'AT' à l'utilisateur, l'utilisateur retransmet son 'AT' au SP (cette requête passera par le mappeur afin de masquer l'adresse IP originale). SP vérifie la validité de 'AT' en l'envoyant à IdP, le cas échéant, SP fournit le service à l'utilisateur.
Si l'utilisateur souhaite avoir accès à un service supplémentaire, SP demande des attributs supplémentaires à IdP. Ce dernier retransmet la demande à l'utilisateur qui pourra accepter ou rejeter la demande. Voir la figure 4.2.

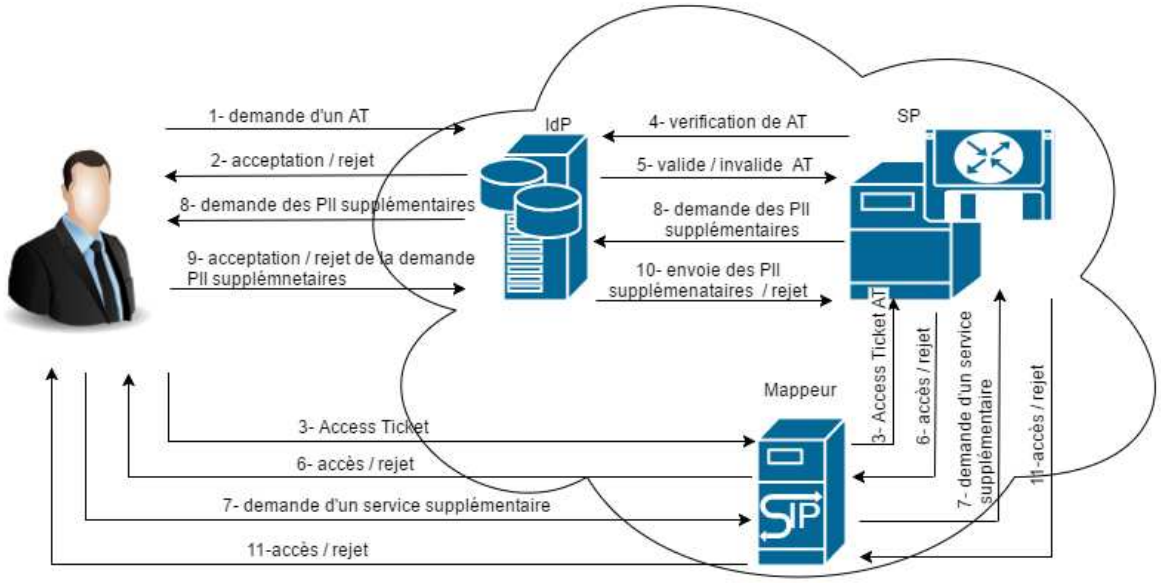


FIGURE 4.2 – Authentification et demande d'un service.

**Remarque :** Quand l'utilisateur souhaite accéder au même service supplémentaire une autre fois, une demande d'autorisation sera envoyée par IdP à l'utilisateur. Ce cas sera expliqué dans la section 4.3.7

### 4.3.4 Algorithmes de fonctionnement

Dans cette partie nous allons décrire les différentes variables et les messages utilisés dans les algorithmes cités ci-dessous :

**Les variables :**

- PII : attributs de l'utilisateur de type enregistrement ;
- PII-supp : attributs supplémentaires demandés par un service de type enregistrement ;
- KU+/KU- : clé publique / clé privée de l'utilisateur de type entier ;
- KIdP+/KIdP- : clé publique / clé privée de IdP de type entier ;
- KIdP+=(N,e), KIdP-=(N,d) ;
- TR : ticket d'enregistrement (identifiant) ;
- KSIdP+/ KSIdP- : clé publique /clé privée de signature de IdP de type entier ;
- KSP+/KSP- : clé publique/clé privée de SP de type entier ;
- AT :ticket d'accès ( Access Ticket ) ;
- VérifierTR : booléen (vrai si TR est valide, faux sinon) ;
- VérifierAT : booléen (vrai si AT est valide, faux sinon) ;
- H(m) : fonction de hachage appliquée au message m ;
- rep : variable qui prend soit la valeur valide ou invalide, initialement égale à Nil ;
- Z : variable qui prend soit la valeur positive ou négative, initialement égale à Nil ;
- SD : variable qui prend soit la valeur SSDD(service supplémentaire déjà demandé ) ou SSJD (service supplémentaire jamais demandé), elle est initialisée à SSJD ;

**Les messages :**

- **(demande-enregistrement-c)** : transporte la demande d'enregistrement d'un utilisateur à un service dans le cloud ;
- **(demande-enregistrement, KIdP+(TR))** : l'utilisateur passe son TR à IdP ;

  KIdP+(TR) : désigne le ticket d'enregistrement de l'utilisateur auprès

de IdP ;

- **(enregistrement, KIdP+(PII))** : l'utilisateur confirme son enregistrement en acceptant les politiques d'accès du service souhaité ;
  KIdP+(PII) : désigne les PII demandés par Idp lors de l'enregistrement au service voulu ;
- **(demandeTR)** : IdP demande TR pour que l'utilisateur (déjà enregistré) s'enregistre à un autre service auprès de IdP ;
- **(enregistré, KU+(KSIdP-(TR)))** : IdP finalise l'enregistrement en envoyant KU+(KSIdP-(TR)) ;
  KU+(KSIdP-(TR)) : désigne le ticket d'enregistrement de l'utilisateur auprès de IdP ;
- **(demandeAT, KIdP+(TR))** : l'utilisateur demande un service en envoyant son ticket d'enregistrement à IdP ;
  KIdP+(TR) : désigne le ticket d'enregistrement de l'utilisateur au service SP auprès de IdP ;
- **(vérificationTR, rep)** : IdP confirme la validité de ticket d'enregistrement ;
  rep : prend la valeur valide si le ticket d'enregistrement est vérifié, valeur invalide sinon ;
- **(signatureAvg, Blind(m))** : l'utilisateur génère une signature aveugle pour IdP ;
  Blind(m) : désigne le message aveuglé transmis par l'utilisateur ;
- **(réponseAT, S')** : Idp passe l'AT avec le message signe contenant S' ;
  S' : désigne le message signé contenant la valeur d'AT et qui peut être extraite que de la part de l'utilisateur légitime ;
- **(demande-service, KIdP+(AT))** : l'utilisateur redirige sa demande d'accès au SP en envoyant son AT de session ;
- **(demande-vérificationAT, AT)** : le SP demande la vérification d'AT reçue en le passant à IdP ;
- **(vérificationAT, rep)** : IdP confirme la validité de ticket d'accès ;
- **(paquet-service)** : désigne le paquet qui contient le service ;
- **(paquet-service-supp)** : le paquet qui contient le service supplémentaire ;

- **(demande-service-supp)** : demander un service supplémentaire au SP de la part de l'utilisateur ;
- **(demande-PII-supp)** : IDP demande des PII supplémentaires à l'utilisateur pour lui fournir un service supplémentaire ;
- **(réponsePII-supp, X)** : l'utilisateur répond par les PII-supp demande pour avoir accès au service supplémentaire ;

$$X : \begin{cases} KIdP + (PII - supp), \ si\ l'utilisateur\ passe\ ses\ PII - supp\ à\ IdP; \\ KSp + (PII - supp), \ si\ IdP\ passe\ les\ PII - supp\ à\ SP. \end{cases}$$

- **(demande-autorisation)** : IdP demande l'autorisation de fournir les PII-supp lors d'accès au service supplémentaire ;
- **(reponsedemande-autorisation, Z)** : l'utilisateur répond à la demande d'autorisation ;

$$Z : \begin{cases} accept, \ si\ l'utilisateur\ accepte; \\ refus, \ sinon. \end{cases}$$

### 4.3.5 Enregistrement d'un utilisateur

Lorsqu'un utilisateur soumet une requête d'inscription au cloud, R-SA-I la redirige vers IdP. Nous distinguons deux cas :

- Un utilisateur s'enregistre pour la première fois dans IdP : dans ce cas le IdP générera un TR pour l'utilisateur ,c'est la preuve que l'utilisateur s'est enregistré dans le cloud. Ce TR sera ensuite sauvegardé dans une base de données de IdP.
- Un utilisateur déjà enregistré dans IdP qui veut avoir accès à un autre service proposé par IdP : dans ce cas IdP demandera le TR de l'utilisateur afin d'éviter la génération de plusieurs TR pour un même utilisateur.

L'algorithme 1 représente la procédure effectuée, ainsi la figure 4.3 illustre le diagramme d'échanges.

**Algorithm 1** Enregistrement d'un utilisateur

**utilisateur :**
envoyer *(demande-enregistrement-c)* à IdP ;
**Lors** de *l'affichage de l'interface* de IdP faire
choisir un service ;
**Si** (politiques de service accepté) alors
chiffrer les PII avec la clé KIdP+ ;
envoyer *(enregistrement, KIdP+(PII))* à IdP ;
**Sinon**
Annuler la demande d'enregistrement ;
**Fsi ;**
**Fait.**
**Lors** de la réception de *(demandeTR)* depuis IdP faire
envoyer *(demandeenregistrement, KIdP+(TR))* à IdP ;
**Fait.**
**Lors** de réception de *(enregistré, KU+(KSIdP-(TR,n)))* depuis IdP faire
déchiffrer KU+(KSIdP-(TR)) avec KU- et KSIdP+ ;
**Fait.**
**IdP :**
**Lors** de la réception de la *(demande-enregistrement-c)* depuis l'utilisateur faire
**Cas1 :** l'utilisateur s'enregistre pour la première fois
Affichage de l'interface contenant les services enregistrés dans IdP
et leurs politiques pour l'utilisateur ;
**Cas 2 :** utilisateur déjà enregistré à un service auprès de IdP
envoyer *(demandeTR)* à l'utilisateur ;
**Fcas ;**
**Fait.**
**Lors** de la réception de *(demandeenregistrement, KIdP+(TR))* depuis l'utilisateur faire
Affichage de l'interface contenant les services enregistrés dans IdP et leurs politiques pour l'utilisateur ;
**Fait**
**Lors** de réception de *(enregistrement, KIdP+(PII))* depuis l'utilisateur faire
**Cas1 :** si l'utilisateur s'enregistre pour la première fois
enregistrer les PII dans la BDD ;
générer un TR ;
enregistrer le ticket (TR) dans sa BDD ;
signer le ticket (TR) avec KSIdP- ;
chiffrer (KSIdP-(TR) avec KU+ ;
envoyer *(enregistré, KU+(KSIdP-(TR)))* à l'utilisateur ;
**Cas 2 :** si l'utilisateur est déjà enregistré
enregistrer les PII dans la BDD ;
envoyer enregistré avec succès a user ;
**Fcas ;**
**Fait.**

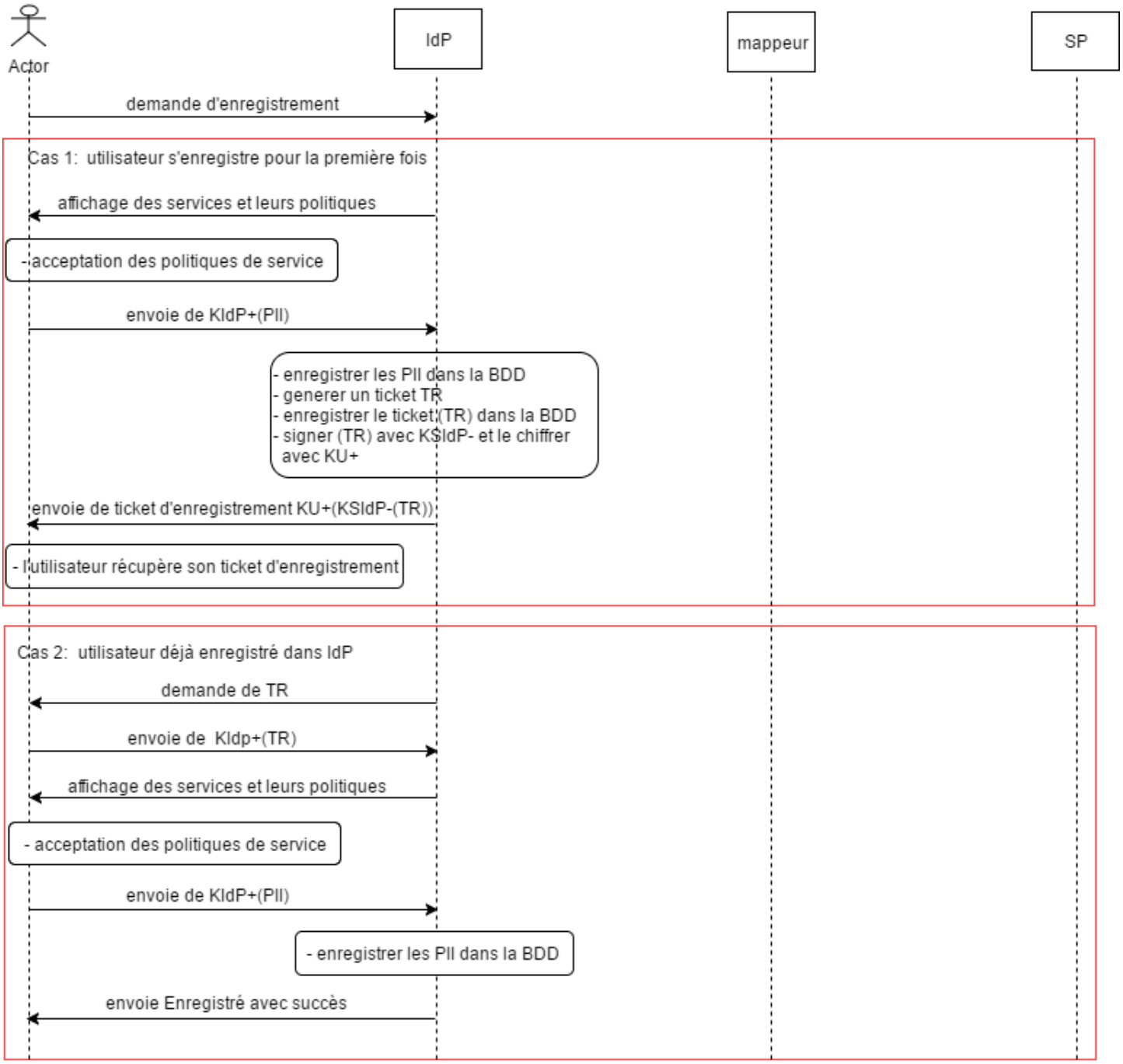


FIGURE 4.3 – Diagramme d'échange du scénario d'exécution d'enregistrement.

### 4.3.6 Authentification d'un utilisateur

Lorsqu'un utilisateur souhaite accéder à un service de cloud, il envoie une demande d'un Access Ticket 'AT' contenant (TR) à IdP (R-SA-I se charge de rediriger la demande en question vers sa destination). L'algorithme 2 représente la procédure effectuée, ainsi la figure 4.4 illustre le diagramme d'échanges.

**Algorithm 2** Authentification d'un utilisateur

**utilisateur :**
**Lors**de la décision d'accès à un service faire
chiffrer (TR) avec KIdP+ ;
envoyer *(demandeAT, KIdP+(TR))* à IdP ;
**Fait.**
**Lors** de la réception de *(vérificationTR, rep)* depuis IdP faire
**Si** (rep = valide) alors
générer un message 'm' et un nombre aléatoire r premier avec N ;
calculer $Blind(m) = H(m)r^e mod N$ ;
envoyer *(signatureAvg, Blind(m))* à IdP ;
**Sinon**
Authentification échouée ;
**Fsi ;**
**Fait.**
**Lors** de la réception de *(réponseAT, S')* depuis IdP faire
calculer $AT = S' * r^{-1}$ ;
envoyer *(demande-service, KIdP+(AT))* au mappeur ;
**Fait.**
**IdP :**
**Lors** de la réception de *(demandeAT, KIdP+(TR))* depuis l'utilisateur faire
déchiffrer KIdP+(TR) avec KIdP- ;
vérification de (TR) ;
**Si** (VérifierTR) alors
envoyer *(vérificationTR, valide)* à l'utilisateur ;
**Sinon**
envoyer *(vérificationTR, invalide)* à l'utilisateur ;
**Fsi ;**
**Fait.**

**Lors** de la réception de *(signatureAvg, Blind(m))*depuis l'utilisateur faire
calculer $S' = Blind(m)^d mod N$ ;
envoyer *(réponseAT, S')* à l'utilisateur ;
**Fait.**
**Lors** de la réception de *(demande-vérificationAT, KIdP+(AT))* depuis SP faire
déchiffrer KIdP+(AT) avec KIdP- ;
vérification de AT ;
**Si** (VérifierAT) alors
envoyer *(vérificationAT, valide)* à SP ;
**Sinon**
envoyer *(vérificationAT, invalide)* à SP ;
**Fsi ;**
**Fait.**
**Mappeur :**
**Lors** de la réception *(demande-service, KIdP+(AT))* depuis l'utilisateur faire
masquer l'adresse IP ($@IP \longrightarrow @MIP$) ;
retransmettre *(demande-service, KIdP+(AT))* à SP ;
**Fait.**
**Lors** de la réception de ( *paquet-service*) depuis SP faire
démasquer l'adresse ($@MIP \longrightarrow @IP$) ;
retransmettre (*paquet-service*) à l'utilisateur ;
**Fait.**
**SP :**
**Lors** de la réception de *(demande-service, KIdP+(AT))* depuis mappeur faire
envoyer *demande-vérificationAT , KIdP+(AT))* à IdP ;
**Fait.**
**Lors** de réception de *(vérificationAT, rep)* depuis IdP faire
**Si**(rep = valide) alors
fournir le service à l'utilisateur en envoyant le (*paquet-service*) au mappeur ;
**Sinon**
accès refusé ;
**Fsi ;**
**Fait.**

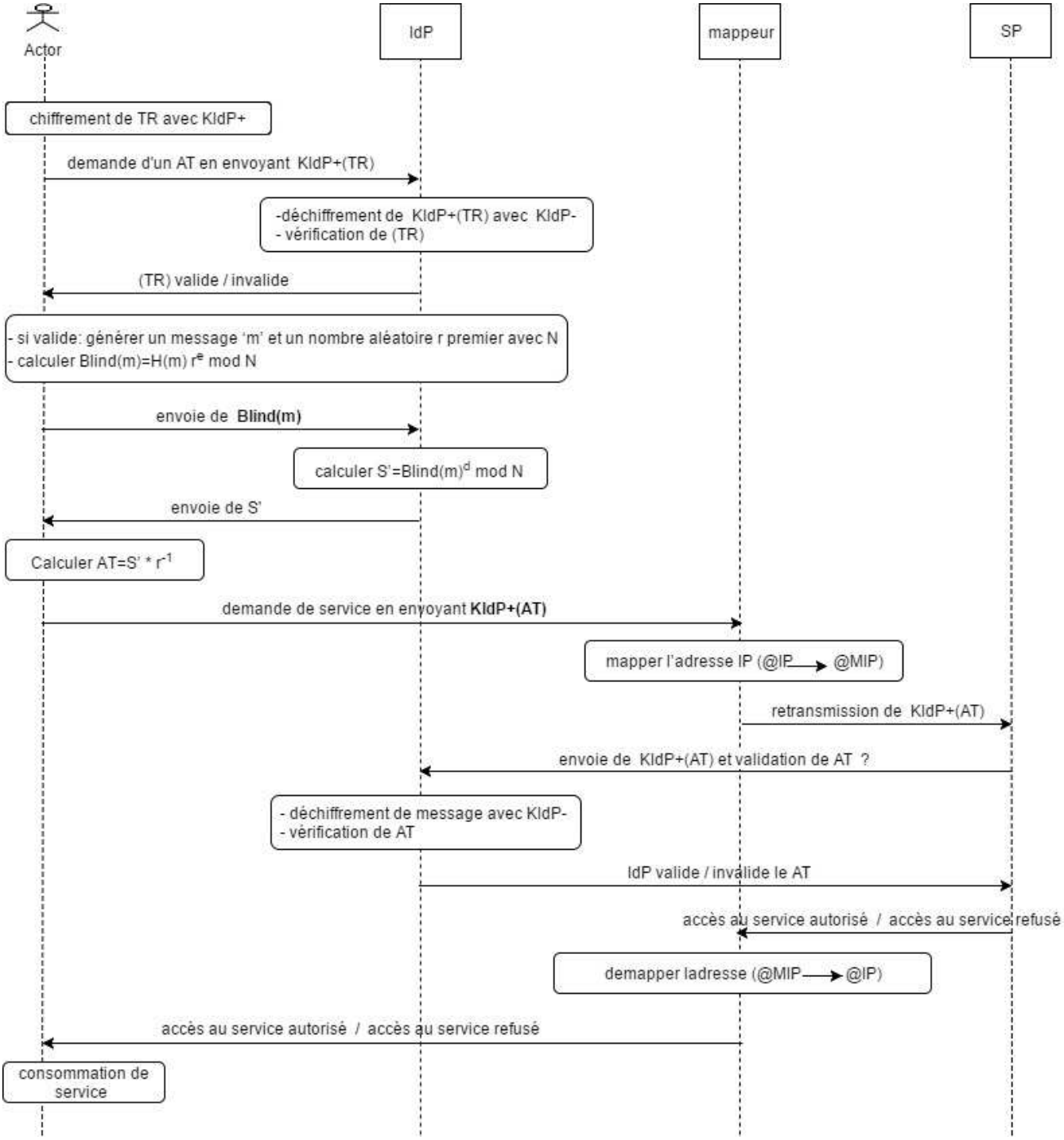


FIGURE 4.4 – Diagramme d'échange du scénario d'exécution d'authentification.

### 4.3.7 Demande d'un service supplémentaire

Lorsqu'un utilisateur déjà authentifié souhaite demander un service supplémentaire, deux cas de figure peuvent se présenter :

- Un utilisateur qui n'a jamais demandé un service supplémentaire.
- Un utilisateur qui a déjà demandé ce service supplémentaire.

L'algorithme 3 représente la procédure effectuée, ainsi la figure 4.5 illustre le diagramme d'échanges.

**Algorithm 3** Demande d'un service supplémentaire

**utilisateur :**
envoyer *(demande-service-supp)* au mappeur ;
**Lors** de la réception de *(demandePII-supp)* depuis IdP faire
  Si (demande accepté) alors
    SD=SSDD ;
    chiffré les PII demandés PII-supp avec KIdP+ ;
    envoyer *(réponsePII-supp ,KIdP+(PII-supp))* à IdP ;
  Sinon
    terminer ;
  Fsi.
**Fait.**
**Lors** de la réception de *(demande-autorisation)* depuis IdP faire
  Si (demande d'autorisation acceptée) alors
    envoyer *(reponsedemande-autorisation, accept)* à IdP ;
  Sinon
    envoyer *(reponsedemande-autorisation, refus)* à IdP ;
  Fsi.
**Fait**
**Idp :**
**Lors** de la réception de *(demande-PII-supp)* depuis SP faire
  **Cas1 :** SD =SSJD
  envoyer *(demande-PII-supp)* à l'utilisateur ;
  **Cas2 :** SD=SSDD à IdP ;
  envoyer *(demande-autorisation)* à l'utilisateur ;
  Fsi.
**Fait**

**Lors** de la réception de *(réponsePII-supp, KIdP+(PII-supp))* depuis l'utilisateur faire
déchiffré KIdP+(PII-supp) avec KIdP- ;
enregistré les PII-supp dans la BDD ;
chiffrer PII-supp avec KSP+ ;
retransmettre *(réponsePII-supp, KSP+(PII-supp))* à SP ;
Fsi.
**Fait.**
**Lors** de la réception de *(reponsedemande-autorisation, Z)* depuis l'utilisateur faire
Si (Z=‘accept’) alors
Chiffrer PII-supp avec KSP+ ;
envoyer *(réponsePII-supp ,KSP+(PII-supp))* à SP ;
Sinon
Rien a faire ;
Fsi.
**Fait**
**Mappeur :**
**Lors** de la réception de *(demande-service-supp)* depuis l'utilisateur faire
masquer l'adresse IP ($@IP \longrightarrow @MIP$) ;

retransmettre *(demande-service-supp )* à SP ;
**Fait.**
**Lors** de la réception de *(paquet-service-supp)* depuis SP faire
démasquer ($@MIP \longrightarrow @IP$) ;

retransmettre *(paquet-service-supp)* à l'utilisateur ;
**Fait.**
**SP :**
**Lors** de la réception de *(demande-service-supp)* depuis le mappeur faire
envoyer *(demande-PII-supp )* à Idp ;
**Fait.**
**Lors** de la réception de *(réponsePII-supp ,KSP+(PII-supp))* Idp faire
déchiffré KSP+(PII-supp) avec KSP- ;

envoyer *(paquet-service-supp)* à Idp ;
**Fait.**

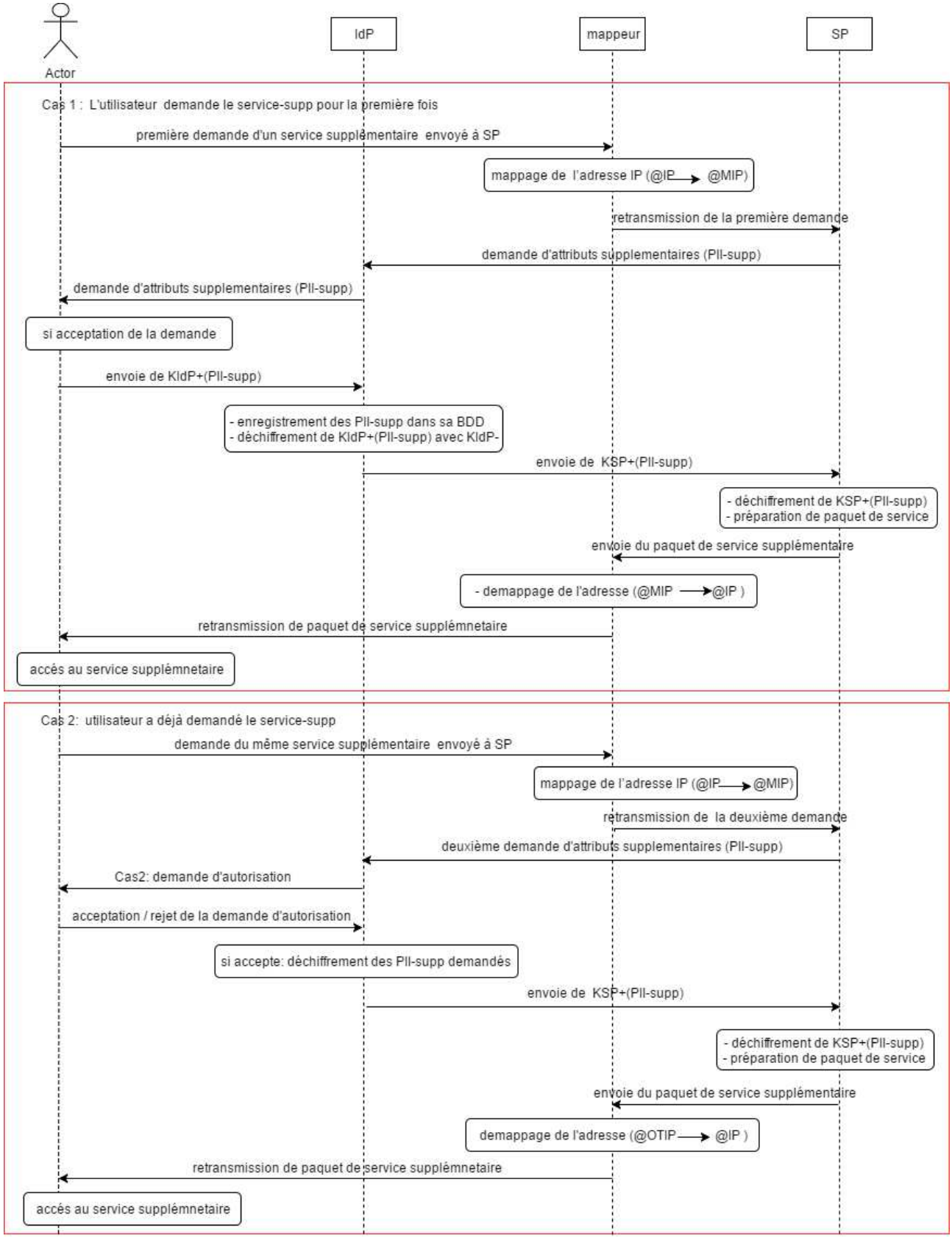


FIGURE 4.5 – Diagramme d'échange d'une demande d'un service supplémentaire.

### 4.3.8 Exemple illustratif :

Nous illustrons les différentes phases d'exécution de notre proposition avec un exemple pratique afin de fournir une meilleure compréhension sur les algorithmes cités auparavant. Pour ce faire, nous supposons le scénario suivant :
Lorsqu'un utilisateur souhaite s'enregistrer au service "Facebook" proposé pat notre IdP. Il souhaitera par la suite effectuer un achat (service supplémentaire) dans un jeu proposé par "Facebook Gameroom".

**Lors de l'enregistrement au service Facebook** (on considère le premier cas : utilisateur s'enregistre pour la première fois dans IdP), l'utilisateur soumet une demande d'enregistrement au cloud, le R-SA-I redirige la demande vers IdP. Ce dernier transmet les politiques du "Facebook" a l'utilisateur. Ensuite, l'utilisateur accepte ou non ces politiques de confidentialité : s'il accepte, il transmet les PII demandés dans la politique de confidentialité à IdP pour être enregistrés dans sa base de données. Un ticket d'enregistrement TR est envoyé à l'utilisateur comme preuve d'enregistrement dans le cloud. Le TR sera sauvegardé dans la base de données de IdP pour les prochaines utilisations.

**Lors de l'authentification pour l'accès au service "Facebook"** . L'utilisateur soumet une demande d'un Access Ticket 'AT' contenant son 'TR' au cloud, le R-SA-I redirige la demande vers IdP. Si le TR fourni par l'utilisateur est valide, IdP accepte la demande en envoyant à l'utilisateur un 'AT'. Ce dernier retransmet son 'AT' au "Facebook" (cette requête sera redirigée par R-SA-I vers le mappeur afin de masquer l'adresse IP originale et retransmise au "Facebook"). "Facebook" vérifie la validité de 'AT' en l'envoyant à IdP, dans le cas échéant "Facebook" fournit le service à l'utilisateur.

**Lorsque l'utilisateur déjà authentifié souhaite effectuer un achat** dans un jeu proposé par "Facebook Gameroom" (service supplémentaire), nous distinguons deux cas :

- L'utilisateur n'a jamais effectué d'achat dans "Facebook Gameroom" : dans ce cas, "Facebook Gameroom" envoie une demande d'attributs supplémentaires (numéro de carte, date d'expiration, code de sécurité, code postale, pays , etc.)

à IdP qui retransmet la demande à l'utilisateur. Ce dernier envoie ses attributs supplémentaires demandés à IdP. IdP les enregistre dans sa base de données avant de les retransmettre à "Facebook Gameroom". Ainsi l'utilisateur aura effectué son achat avec succès.

– L'utilisateur a déjà effectué un achat dans "Facebook Gameroom" : dans ce cas, "Facebook Gameroom" envoie une demande d'attributs supplémentaires (numéro de carte, date d'expiration, code de sécurité, code postal, pays , etc.) à IdP. Ensuite, IdP transmet une demande d'autorisation à l'utilisateur. Ce dernier envoie sa réponse à la demande d'autorisation (accept ou refus). Dans le cas échéant, IdP transmet les attributs supplémentaires demandés (déjà sauvegardé lors du premier achat) à "Facebook Gameroom", ainsi l'utilisateur aura effectué son achat avec succès.

## 4.4 Vérification et validation

Afin de valider notre proposition, nous avons effectué une vérification avec "Scyther" (voir annexe) qui est un outil spécifié dans l'analyse de la sécurité des protocoles. Nous nous référons à [17], [10] et [11] où Scyther a été utilisé pour analyser la sécurité de leurs travaux.

### 4.4.1 Paramètres de vérification

la vérification a été réalisée en utilisant un langage propre à l'outil Scyther, qui modélise les protocoles du point de vue d'entités participantes. Chaque entité est un rôle introduit qui contient une déclaration de valeurs. Ceux-ci peuvent être 'fresh' si ce rôle les a générés ou de type 'variable' si la valeur est envoyée par une autre entité. Chacune des déclarations comprend également un type, qui est l'un des suivants [38] : 'Nonce' pour modéliser les paramètres de variantes de temps (comme fresh n :Nonce) ou 'Ticket' qui est la forme la plus générale pour la modélisation des champs de texte. Scyther permet la possibilité de créer de nouveaux types.

L'échange de message est modélisé dans les différents rôles par 'send'et 'recv'. De

plus, des contraintes sont déclarées dans chaque rôle, elles spécifient les propriétés de sécurité que Scyther évaluera pour le protocole. La forme générale d'une contrainte est Claim(I, type_Claim) où I est l'identifiant du rôle dans lequel les contraintes sont déclarées et type_Claim indique le type de contrainte qui peut être : 'Secret', 'Aliveness' ou 'Weakagreement' [38].

Dans notre cas, nous avons suivi la démarche suivante :tout d'abord, nous avons modélisé l'algorithme distribué de notre protocole en déclarant les trois rôles : user, IdP et SP. Ensuite, nous avons décrit, dans chaque rôle, l'ensemble des événements d'envoie et de réception défini dans type de messages correspondant à notre algorithme. Enfin, nous avons ajouté les contraintes de sécurité que notre protocole doit respecter. Sans ces contraintes Scyther ne saura pas ce qui doit être vérifié.

Les contraintes de sécurité choisies sont : la contrainte 'secret' de ticket d'enregistrement 'TR' et de ticket d'accès 'AT' généré, la contrainte 'aliveness' et la contrainte 'weakagreemnt'. Cela offre une protection suffisante pour éviter l'usurpation d'identité dans note système d'authentification, par conséquent la caractéristique de l'authentification sure sera respectée.

### 4.4.2 Résultats obtenus

Dans cette section, nous détaillons le résultat obtenu par Scyther à travers l'interface graphique. Dans la fenêtre de résultat, Scyther affiche la propriété (contrainte) à vérifier sur chaque ligne. Cette dernière est divisée en quatre colonnes (claim ,Statuts , Comments , Patterns) :

**Claim :** est devisé à son tour en quatre sous colonne : La première colonne présente le nom du protocole ; la seconde présente le rôle ; dans la troisième colonne, Scyter associe à chaque exigence un identifiant sous la forme " Nom_prtocole,étiquette_exigence " ; la quatrième colonne présente la propriété (contrainte).

**Statuts :** Affiche le résultat du processus de vérification : il produira un échec en cas d'attaque autrement dit lorsque la propriété (contrainte) n'est pas respectée, et OK lorsque la propriété est respectée.

**Comments :** sert à expliquer davantage l'état des résultats. En particulier, la co-

lonne contient des phrases simples.
**Patterns :** en cas d'échec sur une propriété (contrainte), Scyther produit un graphe pour schématiser les attaques. Les nœuds dans ce graphe représentent les événements de communication et les flèches l'ordre de ces événements. Ce graphe est orienté verticalement, ce qui facilite la compréhension de la chronologie des événements lorsque plusieurs sessions se chevauchent. Cette dernière colonne ne figure pas sur notre interface, car toutes les propriétés ont été respectées ainsi aucune attaque n'est signalée.

La Figure 4.6 illustre les résultats obtenus en vérifiant notre protocole avec l'outil Scyther.
Dans notre protocole nous avons obtenu des OK sur toutes les lignes des propriétés (contrainte) : secret, aliveness et weakagreeent, ce qui se traduit par le fait que notre proposition offre une authentification anonyme et sure pour les utilisateurs, ainsi les attributs de l'utilisateur PII, les tickets d'enregistrements TR, les tickets d'accès AT sont bien protégés contre tout autre tiers pendant l'enregistrement et lors d'accès aux services.

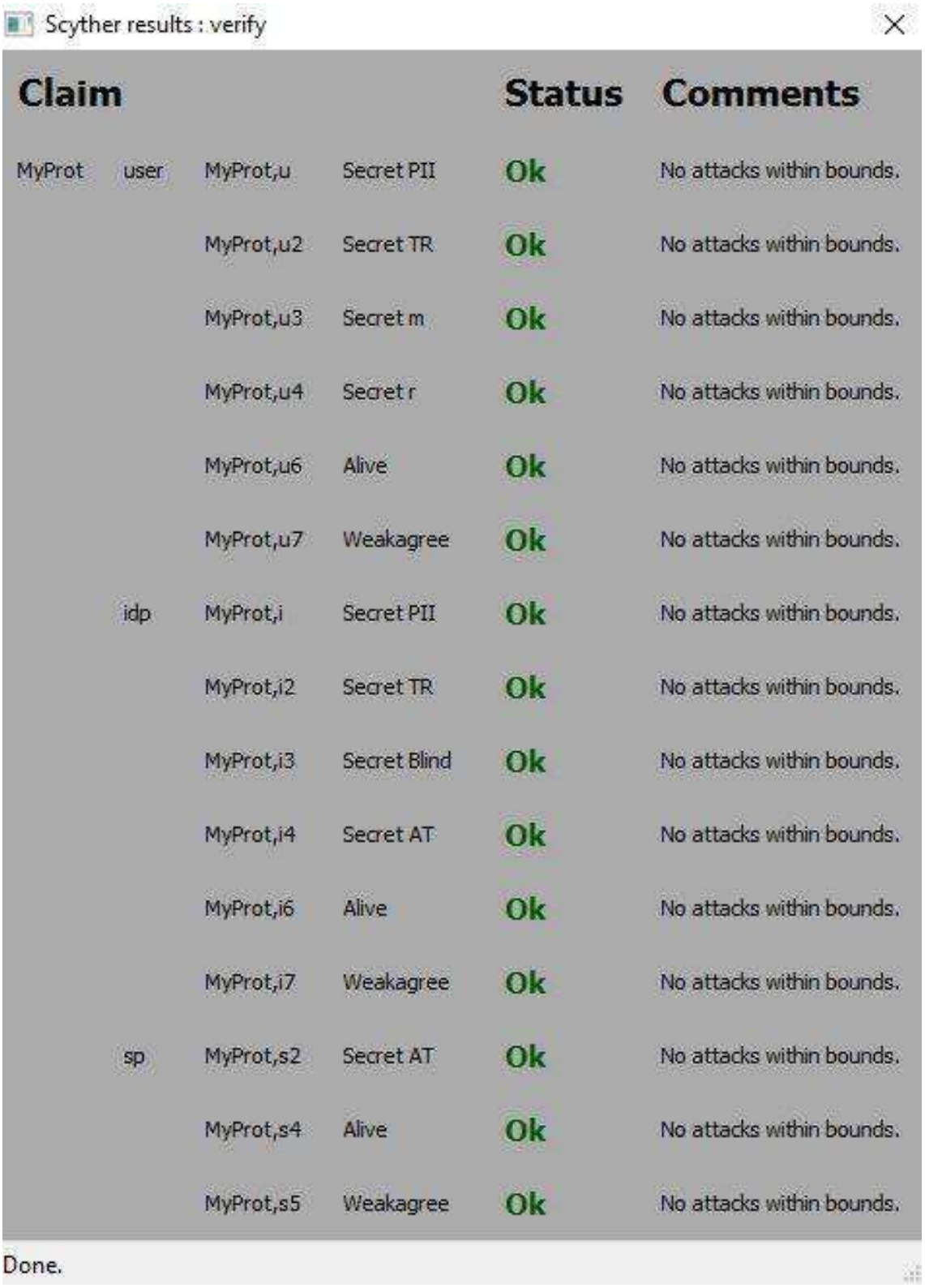


FIGURE 4.6 – Resultat de verification avec l'outil scyther.

### 4.4.3 Synthése

Notre protocole respecte les exigences déjà citées dans la section 4.3.1 :

- Contrôle des PII :l'utilisateur donne son autorisation avant de passer les attributs demandés pour un fournisseur de service.
- Minimisation : l'utilisateur transmet uniquement les attributs nécessaires lors de son enregistrement et le TR avec AT lors de son accès au service .
- Convivial : notre système de gestion d'identité facilite tout processus lié à l'authentification anonyme et cela est réalisé, car les utilisateurs ne sont pas tenus à compléter des procédures complexes ou de gérer des outils compliqués lors de l'authentification.
- In-traçabilité : l'adresse IP de l'utilisateur est masqué pour le fournisseur de services et ce en utilisant le mappeur d'adresse.

## 4.5 Conclusion

Dans ce chapitre nous avons proposé une méthode de gestion confidentielle des PII qui est une amélioration des travaux cités dans [17] et [34]. Ensuite, nous avons détaillé les différents scénarios que notre modèle aura à exécuter. Enfin, nous avons clôturé notre chapitre avec une validation qui consiste à vérifier si le critère "authentification anonyme" cité dans le troisième chapitre a été respecté dans l'exécution de notre proposition. Les résultats obtenus ont démontré que notre méthode offre une authentification anonyme et sure, ainsi l'utilisateur pourra consommer de manière anonyme des services.

# Conclusion et perspectives

Le cloud computing est une grande évolution technologique dans le domaine informatique. C'est un paradigme dans lequel les entreprises peuvent stocker leurs données à distance et accéder aux services partout là où ils en ont besoin et à moindres coûts. Toutefois, externaliser ses ressources informatiques apporte aussi son lot de risques notamment en termes de sécurité des informations sensibles, en particulier la confidentialité des informations personnelles identifiables (PII) des utilisateurs.

Après avoir présenté l'état de l'art traitant la confidentialité dans le cloud, nous avons dégagé les points faibles et points forts des différents protocoles étudiés afin de mieux établir notre solution. Nous avons proposé une solution de protection des PII des utilisateurs, afin de garantir la confidentialité des utilisateurs dans le cloud. Notre proposition appelée "méthode de gestion confidentielle des PII" (PIICMM) contient un fournisseur d'identité (IdP), considéré comme une tierce partie de confiance. IdP permet l'enregistrement et l'authentification d'un utilisateur lors de l'accès à un service, de plus il attribue des tickets d'accès anonymes aux différents clients enregistrés. Aussi, un mappeur qui permet de masquer l'adresse IP de l'utilisateur vis-à-vis du fournisseur de service.

Notre proposition PIICMM, permet d'une part, une authentification anonyme, en utilisant un ticket d'enregistrement et un autre ticket d'accès lors de l'accès au service, d'autre part elle garantit l'intraçabilité de l'utilisateur en masquant son adresse IP. Nous avons ensuite détaillé les différentes phases d'exécution de notre proposition. Enfin nous avons validé notre proposition avec l'outil de vérification automatique Scyther.

En perspective, il serait intéressant de :

- Simuler et implémenter notre modèle.
- Tester notre approche sur un environnement cloud réel.

# Annexe

## 1. Définition

Scyther est un outil automatique spécifié dans l'analyse de la sécurité des protocoles, il est développé par Cas Cremers et disponible pour Linux, Windows et Mac OS X [38]. Scyther peut être téléchargé à partir de [14].

Scyther est un outil pour l'analyse formelle des protocoles de sécurité sous l'hypothèse parfaite de cryptographie, dans laquelle on suppose que toutes les fonctions cryptographiques sont parfaites : l'adversaire n'apprend rien d'un message crypté à moins qu'il connaisse la clé de décryptage. L'outil peut être utilisé pour trouver des problèmes qui découlent de la façon dont le protocole est construit [15].

## 2. Fichier d'entrée minimal

Les éléments de base dans un fichier d'entrée Scyther sont des définitions de protocole. L'exemple présenté ci-dessous définit un protocole appelé "ExempleProtocole" qui a deux rôles, 'I' et 'R' en les répertoriant entre parenthèses après le nom du protocole. Notez que nous n'avons pas encore défini le comportement de ces rôles : ces comportements sont définis dans les crochets après le rôle I et le rôle R du rôle correspondant [15].

```
protocol ExempleProtocole(I,R) {
role I { }
role R { }
}
```

# 3. Terme atomique [15]

Un terme atomique peut être n'importe quel identifiant, qui est habituellement une chaîne de caractères alphanumériques.

## 3.1 Valeurs fraîchement générées

De nombreux protocoles de sécurité reposent sur la génération de valeurs aléatoires. Ils peuvent être spécifiés en les déclarants dans une définition de rôle en utilisant la déclaration fresh. Par exemple, pour générer une valeur aléatoire Na de type Nonce, nous spécifions :

```
role X(...) { fresh Na : Nonce ;
send_1(X,Y,Na) ;
}
```

## 3.2 Variables

Les rôles peuvent utiliser des variables pour stocker les termes reçus. Par exemple, pour recevoir un nonce dans une variable avec le nom Na, nous spécifions :

```
role Y(...) { var Na : Nonce ;
recv_1(X,Y,Na) ;
}
```

**Remarque** : Il est à noter que les variables qui sont générées par les rôles sont déclarées avec le mot clé fresh, les autres sont définies avec le mot clé “var”.

# 4. Types prédéfinis [15]

- Fonction : un type qui définit un terme de fonction qui peut prendre une liste de termes en tant que paramètre. Par défaut, il se comporte comme une fonction de hachage donné par terme h (x) où h est du type Fonction.
- Nonce : un type standard qui est souvent utilisé à l'intérieur de l'outil pour définir une valeur aléatoire.
- Ticket : une variable de type Ticket peut être remplacée par n'importe quel terme.

# 5. Type utilisateur [15]

Il est possible de définir un nouveau type. Cela peut être fait en utilisant la commande "usertype" :

```
usertype MyAtomicMessage;
protocol X(I,R) {
role I { var y : MyAtomicMessage;
recv_1(I,R, y ); }
}
```

# 6. Clés asymétriques [15]

Une infrastructure à clé publique (PKI) est prédéfinie : sk(X) désigne la clé privée de X et pk(X) désigne la clé publique correspondante. Par exemple, considérez le terme suivant :

```
{ ni } pk(I);
```

Ce terme représente le cryptage d'un terme ni par la clé pk(I). Ce terme ne peut être décrypté que par un agent (role) qui connait la clé secrète sk(I).

# 7. Fonctions de hachage [15]

Les fonctions de hachage sont essentiellement des cryptages avec une fonction, dont l'inverse n'est connu de personne. Ils peuvent être utilisés par une déclaration globale d'un identifiant comme étant une fonction de hachage, par exemple :

```
hashfunction H1 ;
```

# 8. Événements

## 8.1 Recevoir et envoyer des événements [15]

Les événements recv et send enregistrent la réception et l'envoi d'un message, respectivement. Dans la plupart des cas, chaque événement d'envoi aura un événement recv correspondant. Nous spécifions cette correspondance en donnant à ces événements la même étiquette, indiquée par un indice.
Si un événement send ou recv n'a pas d'événement correspondant, Scyther affichera un avertissement. Pour surcharger cet avertissement, l'étiquette peut être préfixée par un point d'exclamation ! , par exemple :

```
send_!1(I,n, LeakToAdversary ) ;
```

## 8.2 Événements d'exigences et propriétés de sécurité

Les événements d'exigences sont utilisés dans les spécifications de rôle pour modéliser les propriétés de sécurité prévues. Par exemple, l'événement d'exigence suivant signifie que Ni est censé être secret [15].

```
claim(I, Secret, Ni) ;
```

Il existe plusieurs types d'exigences prédéfinis.

- Secret : cette exigence nécessite un terme de paramètre. Le secret de ce terme est exigé [15].

- SKR : La condition de vérification pour cette réclamation équivaut à l'exigence Secret, son but est de marquer le terme de paramètre en tant que clé de session [15].
- Aliveness : "Nous disons qu'un protocole garantit à un initiateur 'A' la vivacité d'un autre agent B si, chaque fois que A (agissant comme initiateur) complète une séquence du protocole, apparemment avec le répondeur B, alors B avait précédemment exécuté le protocole" [38]. Autrement dit cette affirmation implique que, à la fin d'un protocole, les participants sont garantis que tous les autres participants utilisent le protocole [12].
- Weak Agreement : Weak Agreement : cette affirmation suppose qu'à la fin de l'exécution du protocole, l'initiateur du protocole est persuadé que le répondeur de protocole exécute le protocole, mais superficiellement [12].

**Remarque** : Si une propriété de protocole est incorrecte, il existe au moins une attaque sur le protocole. Un bouton s'affiche à côté de la revendication permettant d'afficher les attaques sur cette dernière.

# Bibliographie


[1] http://www.yeswecloud.fr/cloud/avantages-et-inconvenients-du-cl\
\oud-computing-2-810.html, (Consulté le 09/11/ 2016).

[2] http://www.ideecloud.com/dossiers/les-avantages-du-cloud-computin\
\g-les-entreprises/, (Consulté le 09/11/2016).

[3] https://www.babelio.com/auteur/-OCDE/233056, (Consulté le 14/06/2017).

[4] Matrice diagonale. http://uel.unisciel.fr/mathematiques/calculmat1/calculmat1_ch01/co/apprendre_ch1_01_09.html, (Consulté le 15/06/2017).

[5] Cloud computing. https://missarte.wordpress.com/les-inconvenients-du-cloud-computing/, (Consulté le 20/06/ 2017).

[6] La vérite sur le cloud. http://www.computerland.fr/la-verite-sur-le-cloud/, (Consulté le 20/06/ 2017).

[7] Welcome to openid connect. http://openid.net/connect/, (Consulté le 31/05/2017).

[8] H. Alddin, S. Ahmed, M. F. Zolkipli, and B. Zolkipli. Data security issues in cloud computing : Review. *International Journal of Software Engineering and Computer Systems*, vol 2 :58–65, Février 2016.

[9] R.S. Anjali and A. Ravikumar. Preserving privacy in public auditing for shared cloud data. volume Vol 2, pages 1–6, Coimbatore, India, 26-27 août 2016. International Conference on Inventive Computation Technologies (ICICT).

[10] D. Basin and C. Cremers. Degrees of security : protocol guarantees in the face of compromising adversaries. In *Proceedings of the 24th international conference*, page 1–18, Berlin, Heidelberg, 2010. Springer-Verlag.

[11] D. Basin, C. Cremers, and S. Meier. Provably repairing the iso/iec 9798 standard for entity authentication. In *Proceedings of the First international conference on Principles of Security and Trust*, page 129–148, Berlin, Heidelberg, 2012. Springer-Verlag.

[12] M. Bilal and S. Kang. Time-assisted authentication protocol. `https://arxiv.org/ftp/arxiv/papers/1702/1702.04055.pdf`, (Consulté le 09/06/2017).

[13] T. Chardonnens. Les enjeux du cloud computing en entreprise. Mémoire de licence, Département d'informatique, Université de Fribourg Suisse, 2012.

[14] C. Cremers. The scyther tool. `http://www.cs.ox.ac.uk/people/cas.cremers/scyther/index.html`, (Consulté le 07/06/2017).

[15] C.J.F. Cremers. Scyther user manual. `https://profs.info.uaic.ro/~cbirjoveanu/web/Ps/Scyther/scyther-manual.pdf`, (Consulté le 09/06/2017).

[16] D.Chen and H.Zhao. Data security and privacy protection issues in cloud computing. volume Vol 1, pages 647–651, Shenyang, Chine, 23-25 Mars 2012. Conférence internationale sur l'informatique et l'ingénierie électronique.

[17] A. Djellalbia, S. Benmeziane, S. Bensimessaoud, and N. Badache. Anonymous authentication scheme in e-health cloud environment. pages 47–52, Barcelone, Espagne, 5-7 Dec 2016. The 11th International Conference for Internet Technology and Secured Transactions (ICITST).

[18] D.Martinez. Privacy and confidentiality issues in cloud computing architectures. Mémoire de master en informatique, Université polytechnique, Catalogne, 2013.

[19] G.Plouin. *Tout sur le cloud personnel*. paris, edition dunod edition, 2013.

[20] A. Guillevic. Arithmetic of pairings on algebraic curves for cryptography. `https://tel.archives-ouvertes.fr/tel-00921940/document`, (Consulté le 31/05/2017).

[21] J.Figer. L'informatique en nuage [cloud computing] mode ou révolution ? `http://www.figer.com/Publications/nuage.htm#548`, (Consulté le 30/9/2016).

[22] K.M. Khan and M. Shaheen. Secure cloud services : Matrix multiplication revisited. pages 9–14, Sydney, Australia, 3-5 Dec 2013. IEEE 16th International Conference on Computational Science and Engineering.

[23] K.Jeffery and B.Neidecker-Lutz. The future of cloud computing : Opportunities for european cloud computing beyond. Rapport du groupe d'experts, 2010.

[24] K.Maioua and A.Mansouri. Approche basée agents mobiles intelligents dans un environnement de cloud computing. Mémoire master académique en informatique, Université Kasdi Merbah, Ouargla, 2014.

[25] W. Liu, A. S. Uluagac, and R. Beyah. Maca : A privacy preserving multi-factor cloud authentication system utilizing big data. pages 518–523, Toronto, 27 Avr-2 Mai 2014. IEEE Conference on Computer Communications Workshops.

[26] M.Tebaa. *Chiffrement Homomorphe appliqué au Cloud Bancaire*. Thèse de doctorat en informatique, Université Mohammed V, Rabat, Maroc, 2015.

[27] M.Zhou, R.Zhang, W.Xie, W.Qian, and A.Zhou. Security and privacy in cloud computing : a survey. pages 105–112, Ningbo, China, 1-3 Nov 2010. Sixième conférence internationale sur la sémantique.

[28] N.Grevet. Le cloud computing : évolution ou révolution ? pourquoi,quand,comment et surtout faut-il prendre le risque ? Mémoire de recherche en informatique, école supérieure d'informatique, Versailles, Paris, 2009.

[29] O.Levina, J.Oetting, and Y.Chou. Enforcing confidentiality in a saas cloud environment. pages 90–93, Belgrade, Serbie, 22-24 Nov 2011.

[30] E. Papadopoulou, S. McBurney, N.Taylor, M.H. Williams, and Y.A.Shaaban. User preferences to support privacy policy handling in pervasive/ubiquitous systems. *International Journal on Advances in Security*, vol 2(no 1) :62–71, 2009.

[31] S. Pearson. Taking account of privacy when designing cloud computing services. pages 44–52, Vancouver, BC, Canada, 23 May 2009.

[32] T. Probst. Evaluation et analyse des mécanismes de sécurité des réseaux dans les infrastructures virtuelles de cloud computing. `https://tel.archives-ouvertes.fr/tel-01216609/document`, (Consulté le 05/02/2017).

[33] Q.Zhang, L.Cheng, and R.Boutaba. Cloud computing : state-of-the-art and research challenges. *Journal of internet services and applications*, vol 1 :7–18, 2010.

[34] S.M Rahaman and M. Farhatullah. A framework for preserving privacy in cloud computing with user service dependent identity. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, pages 133–136, Chennai India, August 3-5 2012.

[35] R.Bohn, J.Messina, F.Liu, J.Tong, and J.Mao. Nist cloud computing reference architecture. rapport de recherche, National Institute of Standards and Technology, Etats-Unis, 2011.

[36] D Recordon and R Drummond. Openid 2.0 : a platform for user-centric identity management. Proceedings of the second ACM workshop on digital identity management, pages 11–16, Alexandria, Virginia, USA, 03 Nov 2006. CCS Computer and Communications Security.

[37] U. Roth. A generalized view on pseudonyms and domain specific local identifiers. *International Journal on Advances in Security*, vol 7(no 3 and 4) :76–92, 2014.

[38] L. Schmid. Improving the iso/iec 11770 standard. `http://crypto-protocol.nict.go.jp/data/eng/ISOIEC_Protocols/11770-2_3/ISO11770_LaraSchmid_ETHZ_BSc_thesis.pdf`, (Consulté le 09/06/2017).

[39] J. Sen. Security and privacy issues cloud computing. *Innovation Labs, Tata Consultancy Services Ltd,Kolkata, INDIA*.

[40] S.Pearson and A.Charlesworth. Accountability as a way forward for privacy protection in the cloud. pages 131–144, Pékin, Chine, 1-4 Dec 2009. Premiére conférence internationale sur le cloud computing.

[41] T.Mather, S.Kumaraswamy, and S.Latif. *cloud security and privacy : An enterprise perspective on risks and compliance*. edition O'Reilly Media, 2009.

[42] V.Kherbache, M.Moussalih, Y.Kuhn, and A.Lefort. Cloud computing. Mémoire de master en informatique, Institut universitaire de technologie, Nancy, France, 2010.

[43] J. Werner and C. M. Westphall. A model for identity management with privacy in the cloud. pages 463–468, Messine, Italie, 27-30 Juin 2016. IEEE Symposium on Computers and Communication (ISCC).

[44] M. H. Williams, N. K. Taylor, I. Roussaki, P.Robertson, B. Farshchian, and K. Doolin. *Exploiting the knowledge economy : issues, applications and case studies*, chapter Developing a Pervasive System for a Mobile Environment, pages 1695–1702. ios press edition, oct 2006.

[45] Z. Xiao and Y. Xiao. Security and privacy in cloud computing china. *IEEE Communications Surveys and Tutorials(journal en ligne)*, 15, 2013.

[46] X.Ma. Security concerns in cloud computing. pages 1069–1072, Chongqing, Chine, 17-19 Aug 2012. Quatrième conférence internationale sur les sciences informatiques.

[47] Z.Aouameur and H.Tahrine. Comparaison et mise en place des plateformes de cloud computing : Openstack et eucalyptus. Mémoire master académique en informatique industrielle, Université Kasdi Merbah, Ouargla, 2013.

[48] Z. Zhi-hua, L. Zian-jun, J. Wei, Z. Yong, and G. Bei. An new anonymous authentication scheme for cloud computing. pages 896–898, Melbourne, Australia, 14-17 Juillet 2012. The 7th International Conference on Computer Science and Education.

## RÉSUMÉ

Le cloud computing est une grande évolution technologique dans le domaine informatique. C'est un paradigme dans lequel les entreprises peuvent stocker leurs données à distance et accéder aux services partout là où ils en ont besoin et à moindres coûts. Toutefois, externaliser ses ressources informatiques apporte aussi son lot de risques notamment en termes de sécurité des informations sensibles, en particulier la confidentialité des informations personnelles identifiables (PII) des utilisateurs. Dans notre travail, après une étude critique menée sur les solutions proposées dans la littérature, nous proposons une solution permettant d'une part une authentification anonyme, en utilisant un ticket d'enregistrement et un autre ticket d'accès lors de l'accès au service, d'autre part elle garantit l'intraçabilité de l'utilisateur en masquant son adresse IP. Grâce à l'outil de vérification automatique Scyther, nous avons démontré que notre proposition offre une authentification anonyme et sure en protégeant les PII, les tickets d'enregistrements et les tickets d'accès contre tout autre tiers pendant l'enregistrement et lors d'accès aux services.


**Mots clés :** cloud computing, authentification anonyme, confidentilaité, PII.



## ABSTRACT

Cloud computing is a great technological evolution in the computer science field. It is a paradigm where companies can store their data remotely and access services wherever they need it and at lower costs. However, outsourcing its IT resources also brings its share of risks including the security of sensitive information, especially the confidentiality of the Personally Identifiable Information (PII) of the users. In our work, after a critical study of the solutions proposed in the literature, We propose a solution allowing an anonymous authentication using a registration ticket and another access ticket when accessing the service, but also it guarantees the untraceability of the user by hiding its IP address. Thanks to the scyther automatic verification tool, we have demonstrated that our proposal offers anonymous and secure authentication by protecting PII, registration tickets and access tickets against any third party during registration and access to services.


**Key words :** cloud computing, anonymous authentication, confidentiality, PII.