Cloud Computing, Sécurité et Dissimulation de Données

Christian Delettre* - Karima Boudaoud - Michel Riveill
Laboratoire I3S-UNSA/CNRS, Rainbow
Sophia-Antipolis, France
{delettre, karima, riveill}@polytech.unice.fr

Résumé — **Le cloud computing est un nouveau paradigme fournissant des ressources logicielles et matérielles selon les besoins des clients. Cependant, il présente de nouveaux risques de sécurité comme la confidentialité des données stockées dans des bases de données du cloud. Ce dernier risque est un point crucial incontournable puisque les mécanismes de chiffrement ne sont pas suffisants afin de garantir une confidentialité forte des données. Dans cet article, nous discuterons des risques de sécurité liés au cloud computing et nous nous concentrerons principalement sur le problème de la confidentialité des données dans un contexte de cloud pour l'e-commerce. Nous décrirons la solution d'un composant de dissimulation de données que nous proposons afin de résoudre ce problème. Grâce à notre évaluation de performances, nous constaterons qu'il dissimule avec succès les données des utilisateurs légitimes et les protège contre les attaques potentielles.**

Cloud Computing, Sécurité, Confidentialité, Base de données, Dissimulation de données

I. INTRODUCTION

Depuis quelques années, le monde informatique a vu se populariser un nouveau paradigme qui est le cloud computing. Bien que l'idée du cloud computing n'est pas nouvelle et a vu sa première énonciation en 1960 par John McCarthy : « *computation may someday be organized as a public utility* », ce paradigme a pris son essor depuis 2002 grâce à Amazon. Celui-ci possédant une capacité de stockage et de traitement supérieure à ses besoins, il décida de les revendre. Nous pouvons actuellement distinguer deux grands types d'acteurs de clouds, ceux issus du web comme Amazon, Salesforce.com, Google, et ceux issus de l'IT comme IBM, Microsoft, Sun, HP ou encore Oracle. Ces deux types d'acteurs mettent à disposition les couches du cloud computing permettant le développement et la mise en ligne d'applications comme par exemple la plateforme de développement Force.com de Salesforce.com ou encore l'utilisation de l'application Gmail chez Google. En plus de ces acteurs, il existe également des acteurs spécialisés dans les solutions de gestion de la couche physique du cloud computing comme VMware, Citrix ou encore EMC. La firme de consulting OpenCrow a réalisé une taxonomie du cloud computing [1] qui regroupe l'ensemble des offres de services existantes pour le cloud computing et quelques fournisseurs de solutions. Ainsi, le cloud computing repose sur plusieurs technologies comme le *grid computing*, les *utilities computing*, *SOA*, le *web 2.0*, la *virtualisation* et les *accès haut-débit* et se caractérise par : ***1) Une informatique distribuée*** où les applications sont stockées dans un nuage de serveurs décentralisés auxquelles on accède avec une connexion Internet et un navigateur web ; ***2) Une forte extensibilité*** des applications, des plateformes ou encore des infrastructures ; ***3) Une forte élasticité*** des ressources le constituant qui peuvent être alloué dynamiquement en fonction des besoins, ceci rendu possible par leur virtualisation ; ***4) Une forte tolérance aux pannes*** d'une ou des ressources constituant le cloud computing ; ***5) Un business modèle*** où le client paye en fonction des ressources qu'il utilise.

Bien que le cloud computing possède un grand nombre d'avantages comme la flexibilité et la réduction des coûts, d'importantes questions concernant la sécurité restent ouvertes : Quelle est la garantie de la disponibilité des services ? Comment récupérer mes données ? Comment sont-elles sécurisées ? Qui y accède et comment ? Comment tracer ces accès ? Comment assurer la traçabilité de mes données ? Où seront traitées et stockées mes données ? De plus, par essence même du cloud computing, les données peuvent être dispersées de part le monde. Ainsi, nous pouvons également nous demander si nous serons conformes aux lois et aux réglementations des pays dans lesquels nos données seront stockées. La sécurité est donc un point capital du cloud computing qui permettra une adoption plus franche par les entreprises si celle-ci est correctement prise en compte et que les défis actuels en termes de sécurité sont relevés.

Dans la suite de cet article, nous présenterons synthétiquement en section 2 le cloud computing et en section 3 la prise en compte de la sécurité dans celui-ci. Puis, en section 4, nous présenterons une nouvelle problématique de sécurité concernant la confidentialité des données dans les bases de données du cloud avant de proposer, en section 5, une brève comparaison du point de vue de la sécurité entre différentes bases de données traditionnelles (telles que Oracle, MySQL) et celles du cloud. En section 6, nous identifierons les propriétés devant être remplies pour la dissimulation des données d'un utilisateur légitime et décrira également le composant de dissimulation de données. En section 7, nous donnerons un aperçu de l'évaluation de performance de notre composant. Enfin, la section 8 conclura cet article avec quelques pistes d'améliorations.

II. QU'EST-CE QUE LE CLOUD COMPUTING ?

Le cloud computing est qualifié comme la cinquième génération d'architecture. Dans la littérature, il existe une multitude de définitions plus ou moins floues du cloud computing. Cependant, il en existe une qui fait autorité, celle

* Thèse CIFRE cofinancé par l'ANRT et la société d'e-commerce Vivadia.

issue de l'U.S. National Institute of Standards and Technology (NIST) [2]. La vue du NIST sur le cloud computing repose sur trois modèles de services, quatre modèles de déploiement et cinq caractéristiques essentielles. Les trois modèles de services définis par le NIST sont :

- **Software as a service (SaaS)** : l'utilisateur contrôle seulement des configurations d'applications (exemples : Google Apps, Salesforce.com) ;
- **Platform as a service (Paas)** : l'utilisateur contrôle aussi les environnements d'hébergements (exemples : Microsoft Azure, Force.com, Google App Engine) ;
- **Infrastructure as a service (IaaS)** : l'utilisateur contrôle tout sauf l'infrastructure des Datacenter (exemples : Amazon EC2, Xen).

À ces modèles de services s'additionnent quatre modèles de déploiement :

- **Cloud public** : l'infrastructure du cloud est accessible par le grand public ou par les grands groupes d'industries et l'infrastructure appartient à une organisation vendant des services de cloud computing ;
- **Cloud privé** : l'infrastructure du cloud est accessible à une organisation seule et peut être gérée par elle-même ou par un tiers. Elle peut lui être interne ou externe ;
- **Cloud communautaire** : l'infrastructure du cloud est partagée entre différentes organisations formant une communauté ayant les mêmes préoccupations. L'infrastructure peut être gérée par ces organisations ou par un tiers. Elle peut leur être interne ou externe ;
- **Cloud hybride** : l'infrastructure du cloud est une combinaison de deux modèles de déploiement ou plus (public, privé ou communautaire) où chaque type de cloud est considéré comme une entité unique. Néanmoins, tous les clouds sont interconnectés en utilisant des technologies standardisées ou des technologies propriétaires mais interopérables afin de permettre une portabilité des données/applications.

Enfin, les cinq caractéristiques essentielles se résument de la manière suivante : la notion de service à la demande, un large accès réseau (mobile, laptop, PDA), l'utilisation de ressources multi-tenant, un dimensionnement rapide et un service mesuré et facturé à l'usage.

La Cloud Security Alliance (CSA) a proposé le détail technologique des trois couches de services (IaaS, PaaS, SaaS) dans [3]. Ici, IaaS contient le Datacenter lui même et les machines physiques, mais également une couche d'abstraction matérielle, des connectivités permettant de délivrer les ressources et enfin une couche API (application programming interface ou interface de programmation applicative) permettant de gérer l'infrastructure. PaaS quant à lui rajoute une couche d'intégration permettant le développement d'applications spécifiques. Enfin, SaaS offre un environnement complet permettant d'utiliser des applications pour les utilisateurs finaux. En complément de ces efforts réalisés par le NIST et le CSA, il est important de se rappeler que le cloud computing a également introduit un nouveau modèle économique dont nous pouvons en résumer simplement l'idée avec le principe que le client ne paye que ce qu'il consomme et dont un exemple est donné dans [4] avec l'offre de cloud d'Amazon. Ainsi, la définition donnée par le NIST et les efforts du CSA afin de détailler davantage les technologies utilisées dans les couches de services du cloud permettent de comprendre correctement le concept de cloud computing. Dès lors, nous nous interrogeons sur la fiabilité réelle et la sécurité régnant au sein d'un cloud. Existe-t-il une politique de confidentialité, de traçabilité et de vie privée des données dans le cloud ? De manière générale, est-il possible de contrôler ses données ? Existe-il des possibilités d'audit des clouds ? Ceci est autant de questions que la définition précédente ne prend pas en compte et qu'il est important de préciser dans l'avenir afin de définir correctement et plus finement le cloud computing.

III. LES PROBLÈMES DE SÉCURITÉ DANS LE CLOUD COMPUTING

A. Principaux problèmes de sécurité

Le tableau 1, extrait de l'étude faite dans [5], met en évidence les principaux problèmes devant correctement être traités dans le cloud computing. Comme la colonne **Total** le montre, il en ressort que sur sept critères, six d'entre eux concernent plus de 70% des petites et moyennes entreprises (PME) interrogées dont la confidentialité des données est en tête des préoccupations.

L'European Network and Information Security Agency (ENISA) a identifié trente-cinq risques pour le cloud computing listés dans [6]. Ces risques sont regroupés en quatre sections : ***1) Risques organisationnels et politiques*** : tous les risques liés à la gestion du cloud, à son interopérabilité, aux respects de conformité, etc. ; ***2) Risques techniques*** : tous les risques liés à l'utilisation d'un cloud : interception des données en transit, attaques, échec de l'isolation des ressources, etc. ; ***3) Risques légaux*** : tous les risques liés au plan légal comme le changement de juridiction des données, etc. ; ***4) Risques non spécifiques au cloud*** : ensemble des risques courants dans les systèmes informatiques comme la gestion du réseau (congestion, utilisation non optimale), la perte ou compromission de logs, etc.

De ces risques, l'ENISA en a extrait les huit plus importants dont cinq concernent directement ou indirectement la confidentialité des données : l'échec de l'isolation, la gestion des interfaces compromises, la protection des données, la suppression non sécurisée ou incomplète des données et les attaques de l'intérieur. Similairement à l'ENISA, le CSA a identifié treize domaines d'intérêts en matière de sécurité des clouds répartis en trois sections [3] : ***1) Architecture du cloud*** : traite de l'architecture du cloud computing et de sa sécurisation ; ***2) Administration du cloud*** : traite des domaines législatifs et légaux, des audits, du management et de l'interopérabilité/migration des clouds ; ***3) Exploitation du cloud*** : traite principalement des problématiques de sécurité comme la gestion des incidents, des accès, du chiffrement, etc.

TABLEAU I. PRINCIPALES PRÉOCCUPATIONS DE SÉCURITÉ AU SUJET DES CLOUDS

Critères	Très important	Stoppant	Total
Confidentialité des données des sociétés	30,9%	63,6%	94,5%
Vie privée	43,9%	43,9%	87,8%
Disponibilité des services et/ou des données	47,3%	40,0%	87,3%
Intégrité des services et/ou des données	42,6%	44,4%	87,0%
Perte de contrôle des services et/ou des données	47,2%	28,3%	75,5%
Manque de responsabilité des fournisseurs en cas d'incidents de	43,1%	29,4%	72,5%
Répudiation	47,9%	8,3%	56,2%

De ces domaines, le CSA en a extrait les sept plus grandes menaces dont cinq concernent encore la confidentialité des données [7] : les interfaces de programmation d'applications peu sécurisées, les attaques de l'intérieur, la perte/fuite de données, les vulnérabilités technologiques du multi-tenant et le détournement du trafic et des services de comptes. Cependant, plusieurs de ces risques de sécurité ne sont pas spécifiques au cloud computing [8]. Ainsi, la sécurité se décompose principalement en deux niveaux : 1) le premier niveau traite du bon fonctionnement du cloud computing lui-même et de son infrastructure et 2) le second niveau traite de la sécurité liée aux données circulantes et stockées dans le cloud. En termes d'implications techniques sur l'architecture des clouds, ces risques imposent donc de résoudre plusieurs défis technologiques comme la gestion du grand nombre de clés de chiffrement, un contrôle d'accès très fin, une fédération des identités, etc.

B. *Quelques exemples d'attaques*

Comme nous l'avons déjà mentionné, le cloud computing repose sur un ensemble de technologies préexistantes le rendant donc sensible aux attaques ciblant chacune de ces technologies. Par exemple, concernant le web 2.0, il existe un nombre important d'attaques comme le montre [9] qui réalise un état de l'art sur ces diverses techniques. Ces attaques concernent principalement les couches PaaS et SaaS. Comme autres types d'attaques, nous pouvons citer par exemple : «*Joanna Rutkowska's Red and Blue Pill exploits*» [10] : cette attaque permet d'installer un backdoor sur un système cible par le biais de la virtualisation (IaaS) ; «*Kortchinksy's CloudBurst presentations*» [11] : cette attaque permet d'exécuter du code sur le système hôte à partir d'un système invité de ce host (IaaS) ; «*Distributed denial-of-service (DDoS)*» [12], [13] : cette attaque a pour but de rendre des services indisponibles (PaaS et SaaS). En ce qui concerne les attaques des hyperviseurs serveur, bien qu'à l'heure actuelle elles restent marginales, il est fort probable qu'elles se généralisent d'ici quelques années. Ces attaques sont d'autant plus dangereuses qu'elles s'attaquent à la couche la plus basse du cloud computing et donnent la possibilité de mettre hors de service un cloud tout entier. De plus, la facilité d'utilisation d'un cloud permet de renforcer certaines attaques préexistantes comme : «*Zeus botnet*» [14] : Réussite de l'hébergement et de l'exécution du module central (command and control) du botnet Zeus spécialisé dans le vol de données sensibles (bancaires, mots de passe) sur la plateforme de cloud d'Amazon (IaaS et PaaS) ; «*Spam Attack*» [15] : Attaque issue de l'utilisation du cloud d'Amazon qui par le biais de spams à très grande échelle a permis d'installer des chevaux de Troie sur les machines cibles des spams. Cette attaque a pu être réalisée grâce à la facilité que procure le cloud d'obtenir des ressources informatiques (IaaS et PaaS). Enfin, l'apparition du cloud computing a également fait naître de nouvelles attaques comme l'utilisation d'un déni de service distribué dont le but n'est plus de bloquer un service mais d'augmenter artificiellement les ressources allouées pour le propriétaire du service par le cloud [16]. Ceci est réalisé dans le but d'augmenter considérablement les coûts d'utilisation du cloud pour le propriétaire du service afin de le mener à une faillite financière. Cette attaque peut être également réalisée sur un cloud entier afin de toucher tous les utilisateurs du cloud.

Ainsi, le cloud computing doit pouvoir se protéger contre de nombreuses attaques : celles issues de chacune des technologies préexistantes qui ont permis son émergence, mais également de celles qui ont évoluées et de celles qui sont apparues en même temps que le cloud computing. De plus, comme il l'est souligné dans [17], il est nécessaire de définir de nouvelles méthodes d'évaluation des risques afin de prendre en compte l'aspect dynamique du cloud computing.

IV. LA DISSIMULATION DE DONNÉES COMME UN NOUVEAU PROBLÈME DE CONFIDENTIALITÉ DES DONNÉES

A. *Principe du problème*

Comme nous avons pu le voir précédemment, la confidentialité des données est un élément crucial. Elle couvre la manière dont les données sont stockées, dont elles circulent, mais aussi dont elles sont détruites. Ainsi, même si nous supposons que ces différents points sont parfaitement maitrisés, il reste tout de même possible que ces données puissent offrir des informations fortement utiles à des tierces personnes malveillantes. Imaginons qu'une entreprise d'e-commerce utilise un cloud. Le fournisseur du cloud soucieux de son client lui offre les garanties maximales de confidentialité de ses données. Dans ce cadre, les données de l'e-commerçant seront principalement les données relatives à ses clients et aux commandes qui sont des données critiques pour lui et nécessitant donc un niveau de confidentialité très élevé. Ces données seront donc chiffrées et stockées dans une base de données relationnelle. L'accès à ces données sera également fortement restreint, par exemple à l'administrateur de la base de données, mais également aux techniciens intervenant sur les serveurs physiques constituant le cloud. Imaginons maintenant que l'administrateur de la base de données soit peu scrupuleux et fournisse régulièrement une copie de la base de données de l'e-commerçant à un autre e-commerçant concurrent moyennant finance. Bien que les données soient chiffrées, il reste possible pour l'e-commerçant concurrent de tirer un grand nombre de statistiques de ces données.

Imaginons maintenant, que l'e-commerçant concurrent a récupéré la base de données de l'e-commerçant une première fois juste avant que celui-ci lance une nouvelle campagne publicitaire afin d'augmenter ses ventes. Une fois la campagne de l'e-commerçant finie, l'e-commerçant concurrent récupère à nouveau la base de données. Ainsi, en faisant la différence

entre le nombre de tuples pour les commandes de la première base de données et la seconde, l'e-commerçant concurrent peut avoir une idée précise de l'impact de la campagne publicitaire et si les résultats sont concluants, il peut alors se lancer dans une campagne similaire en sachant qu'un gain substantiel sera obtenu en termes de ventes par exemple. Ainsi, même si les données sont chiffrées, il est toujours possible d'obtenir des statistiques pertinentes. Ici, nous avons pris l'exemple d'un administrateur de bases de données peu scrupuleux, mais il peut en être de même avec un technicien du Datacenter ou un attaquant s'introduisant sur les serveurs du cloud ou le site web de l'e-commerçant comme présenté dans la partie suivante.

B. Cas concret d'un exploit pouvant mener à ce problème

Par exemple, les PME utilisent très souvent une application web de commerce électronique gratuite et libre de type OSCommerce ou Magento qui sont facilement disponibles sur Internet. Ces e-commerçants utilisent bien souvent ces applications de commerce électronique afin de mettre rapidement en place leur activité et elles ne nécessitent pas de posséder des connaissances approfondies dans le domaine de l'informatique. Ces applications sont également bien souvent hébergées sur les serveurs d'un hébergeur qui leur fournit un environnement web prêt à l'emploi qui est rarement optimisé pour leurs besoins. Ainsi, cette apparente simplicité de monter son activité d'e-commerçant fait que la part des compétences informatiques est réduite au strict minimum et bien souvent les notions de sécurité sont peu ou mal prises en compte, laissant par exemple souvent des sites web n'ayant jamais subit l'application de correctifs de sécurité pour combler les failles de sécurité. De ce fait, pour une personne malveillante, il est possible de d'attaquer facilement un tel site. En effet, avec ces solutions prêtes à l'emploi et publiques, il est très facile d'y découvrir des failles de sécurité sur Internet comme pour OSCommerce. Dans ses précédentes versions, OSCommerce souffrait d'une faille de sécurité exploitable par le biais de son « FileManager », ce qui correspondant à une faille de sécurité dans la couche SaaS. Grâce à cette faille, il était possible de passer outre le mécanisme d'identification et de créer par son biais des scripts sans qu'il soit nécessaire d'avoir les accès FTP du site. De plus, par le fait qu'OSCommerce soit open-source, il est très facile de se rendre compte que les codes d'accès à la base de données sont contenus dans un fichier localisé dans le répertoire du site. Par ce même fait, il est possible de savoir où sont situées les données comme celles des commandes et quelle est le nom des tables concernées. Ainsi, il devient très facile de créer un script utilisant ces codes d'accès pour accéder à la base de données, puis d'en extraire le contenu, comme l'ensemble des données des commandes. Tant qu'il ne sera pas prouvé formellement que ces applications sont sans failles de sécurité, il sera toujours probable de trouver des failles de sécurité qui permettront d'accéder à des données sensibles. Face à ce problème, il est donc nécessaire de posséder une solution permettant, en plus de celle du chiffrement, de fausser les tentatives possibles de statistiques sur ces données afin de garantir un fort niveau de confidentialité des données ou autrement dit une confidentialité forte des données. Dans la section suivante, nous allons voir les propriétés de sécurité offertes par les bases de données les plus populaires du marché. Nous verrons également si le problème de confidentialité mis en évidence dans cette section est pris en compte dans celles-ci.

V. SÉCURITÉ DES BASES DE DONNÉES

Dans cette étude, nous avons tenté de comprendre la mise en œuvre de la sécurité dans les bases de données mais également savoir quelles bases de données fournies le plus haut niveau de sécurité. Pour cette étude, nous avons choisi les bases de données libres et propriétaires les plus populaires comme Microsoft SQL, Oracle, IBM DB2, MySQL, PostgreSQL, mais également celles issues du cloud computing comme Amazon SimpleDB, Google DataStore et Azure SQL. Nous avons comparé ces bases de données du point de vue de la sécurité au travers des dix critères suivants (voir résultat fourni en tableau 2) :

- **Identification et authentification des utilisateurs** : Il s'agit d'identifier les utilisateurs de la base de données, de manière sûre et non ambiguë, pour mettre en œuvre des mécanismes de contrôles d'accès, en fonction des droits de chaque utilisateur.
- **Robustesse de l'identification et authentification** : Il s'agit de garantir qu'il est très difficile d'usurper une identité comme vérifier la robustesse des mots de passe.
- **Séparation des droits** : Il s'agit de permettre de distinguer différents types d'utilisateurs avec des actions prédéfinies afin de séparer par exemple les tâches d'exploitation des données, des tâches de maintenance de la base de données.
- **Contrôle d'accès aux données** : Il s'agit de n'autoriser l'accès aux données stockées qu'aux personnes autorisées. Ce contrôle doit permettre différents modes d'accès (lecture et/ou écriture) et une granularité variable (une base, une ou plusieurs tables).
- **Intégrité et confidentialité des données stockées** : Il s'agit de s'assurer que seuls les utilisateurs autorisés peuvent modifier les données stockées dans le serveur hébergeant le système de gestion de bases de données (SGBD) et les données sensibles.
- **Chiffrement des communications** : Il s'agit de protéger les échanges réseau en assurant l'intégrité et, si nécessaire, la confidentialité des requêtes et des données échangées entre différents équipements mettant en œuvre ou utilisant le service de base de données, comme les échanges entre les postes clients (utilisateurs finaux ou administrateurs) et le serveur hébergeant le SGBD ou encore entre serveurs dans le cas d'un SGBD distribué (réplication de données).
- **Camouflage des données** : Il s'agit de dissimiler des données réelles au sein de données factices afin de fausser le volume de données réelles dans le cadre de données en production, tout en gardant la possibilité de retrouver facilement les données réelles.
- **Masquage des données** : Utilisation d'un processus irréversible pour remplacer les données sensibles en

s'assurant que les données originales ne peuvent pas être recherchées ni récupérées. Cette propriété est très importante quand les données sont utilisées pour le développement d'applications et des tests.

- **Services d'audit** : Il s'agit de la journalisation des événements relatifs aux accès au SGBD et la protection en intégrité de ces journaux. Ce type de service est nécessaire, pour permettre un contrôle à posteriori des opérations effectuées afin d'éviter que certains utilisateurs outrepassent leurs droits.
- **Certification** : Il s'agit de la certification EAL (Evalution Assurance Level) définie sur sept niveaux permettant d'évaluer une application informatique (applications civiles : 1 à 4+ ; militaire : 5 à 7).

TABLEAU II. COMPARATIF DE BASES DE DONNÉES AVEC DES CRITÈRES DE SÉCURITÉ

✔ Bonne prise en compte
! Prise en compte imparfaite
✖ Aucune prise en compte
- Aucune information trouvée

	Traditionnelles					Cloud Computing		
	Propriétaires			Libres		Propriétaires		
	Microsoft SQL	Oracle	IBM DB2	Mysql	Postgre SQL	Amazon SimpleDB	Google DataStore	Azure SQL
Identification/authentification des utilisateurs	✔	✔	✔	✔	✔	✔	✔	✔
Robustesse de l'identification/authentification	✔	✔	✔	✖	✔	✔	✔	✔
Sépration des droits (rôles)	✔	✔	✔	!	✖	!	!	!
Contrôle d'accès aux données	✔	✔	✔	✔	✔	!	!	✔
Intégrité/confidentialité des données stockées	✔	✔	✔	!	✔	✖	✖	✖
Chiffrement des communications	✔	✔	✔	✔	✔	✔	✔	✔
Camoufflage des données	✖	✖	✖	✖	✖	✖	✖	✖
Masquage des données	✖	✔	✖	✖	✖	✖	✖	✖
Services d'audit	✔	✔	✔	✖	✖	✔	-	✖
Certification	EAL 1+	EAL 4+	EAL 4+	✖	EAL 1	✖	✖	✖

Nous pouvons constater avec le tableau 2 que les bases de données traditionnelles possèdent un haut niveau de sécurité. Ceci est particulièrement vrai pour les bases de données propriétaires comme Microsoft SQL et Oracle qui sont plus mature que celles proposées dans les clouds étant relativement nouvelles. Nous pouvons également voir que les bases de données issues du cloud ne proposent pas nativement la confidentialité des données qui est à la charge de l'utilisateur. De même, bien souvent dans les bases de données issues du cloud, comme dans Amazon SimpleDB, le contrôle d'accès aux données est présent mais pas à une granularité assez fine. L'accès à une base de données est associé à un compte utilisateur ou rôle qui a tous les droits sur celle-ci. Enfin, nous pouvons constater qu'il n'existe aucun mécanisme de dissimulation de données permettant de se prémunir du problème de statistiques exposé dans la partie précédente. La seule base de données proposant un service dans ce sens est celle d'Oracle, mais ne s'applique que pour produire des données fictives dans le cadre de tests. Ainsi, dans la partie suivante, nous allons présenter notre solution de dissimulation de données permettant de fausser toutes statistiques sur des données en production.

VI. BASES DE DONNÉES CLOUD ET DISSIMULATION DES DONNÉES

A. *Propriétés à respecter pour la dissimulation de données*

Afin de pouvoir fausser tous les résultats issus de statistiques réalisées sur des données non légitimes, nous devons injecter dans la base de données des données artificielles (*factices*) en nous assurant de certaines propriétés :

1) Il ne doit y avoir aucune perte de données réelles. Afin d'assurer cette propriété, nous n'autoriserons que l'insertion de données artificielles.

2) Il doit être facile et rapide pour le propriétaire légitime des données de retrouver l'ensemble ou un sous ensemble de ses données réelles dans la base de données que celles-ci soient chiffrées ou non.

3) Il doit être très difficile pour une tierce personne non légitime de trouver l'ensemble ou un sous ensemble des données réelles du propriétaire légitime que celles-ci soient chiffrées ou non. Afin d'assurer cette propriété et la précédente, nous devons disposer d'un système de marquage des données efficace. Pour ce faire, nous allons nous inspirer de la technique de « watermarking » [18] des bases de données relationnelles. Cette technique consiste à insérer une signature invisible et permanente à l'intérieur des données de la base de données en dégradant celles-ci de manière non réversible. Le but premier de cette méthode est de lutter contre la fraude et d'assurer la protection des droits de propriété intellectuelle. Dans notre cas, nous utiliserons un dérivé de cette méthode pour son utilité secondaire qui est de pouvoir identifier les données réelles. Afin de marquer l'ensemble des données réelles qui seront insérées dans la base de données, nous allons nous inspirer de la méthode de « watermarking » décrite dans [19] et [20]. Ce processus de marquage des données utilise une clé sécrète que seul le propriétaire légitime des données connait, ce qui lui permet de retrouver facilement l'ensemble de ses données. Sans connaissance de cette clé secrète, il est donc très difficile de différencier les données réelles des données artificielles.

4) La génération de données artificielles doit prendre en compte l'évolution naturelle du nombre d'entrées. Afin de respecter cette propriété, nous proposons de majorer par une valeur fixée à l'avance le nombre de données insérées dans la base de données afin de donner l'illusion d'un nombre de données à peu près constant.

5) La dissimulation des données doit se réaliser tout ou en partie en temps réel. Afin de respecter cette propriété, il est nécessaire d'insérer des données artificielles lorsque des données réelles sont insérées, car il n'est pas possible de prédire quand une tentative de vol des données sera réalisée sur les données. De ce fait, il ne doit pas être possible de réaliser des statistiques précises à n'importe quel moment.

6) Les données artificielles générées doivent paraîtres réelles et de même type que les données à insérer afin qu'en cas de réussite du cassage du chiffrement des données, il soit toujours très difficile de retrouver les données réelles. Nous entendons ici par « paraître réel » qu'un utilisateur humain ou non ne peut pas différencier une donnée réelle d'une donnée artificielle (comme le nom d'une personne) sans une analyse poussée notamment en réalisant des recoupements d'informations (par exemple en recherchant dans un annuaire l'existence du nom avec l'adresse postale associée). Afin de respecter cette propriété, il existe à l'heure actuelle des générateurs de données libres permettant de générer divers

types de données artificielles principalement utilisés dans le cadre de tests, comme par exemple le générateur de données disponible sur www.generatedata.com. Cependant, le paraître réel des données générées est très limité. De ce fait, nous allons nous inspirer et réutiliser en partie ce générateur afin de réaliser un générateur de données satisfaisant à notre condition de réalité des données.

Ainsi, afin d'assurer ces six propriétés, que nous avons définies dans le cadre de ce travail, nous avons imaginé une solution permettant de dissimuler les données qui est présentée dans les sous-sections suivantes.

B. Composant de sécurité de dissimulation de données

La notion de *composant de sécurité* a été utilisée dans différents systèmes d'exploitation (Windows, Linux, etc.) et par plusieurs développeurs d'applications afin d'isoler certaines fonctionnalités de sécurité dans des modules. Suivant le contexte, ces modules peuvent représenter des algorithmes de chiffrement comme la bibliothèque JCA (Java Cryptography Architecture), des mécanismes d'authentification comme dans GSSAPI (Generic Security Services Application Program Interface) ou encore des modèles de contrôle d'accès comme avec la bibliothèque LSM (Linux Security Modules) pour les systèmes à base de noyau Linux. En comparaison, dans [21] un composant de sécurité est défini comme un composant applicatif remplissant une propriété de sécurité comme la confidentialité, l'intégrité, l'authenticité, le contrôle d'accès ou encore la non-répudiation. Dans le cadre de ce travail, comme nous définissons la dissimulation de données comme une propriété de sécurité, nous utiliserons par conséquent la définition de [21] pour concevoir un composant de sécurité de dissimulation de données afin de sécuriser les données stockées dans une base de données d'un cloud. Notre composant de sécurité de dissimulation de données est composé de trois sous-composants : *le composant de prédiction, de génération de données et de marquage de données*.

Le principe du *sous-composant de prédiction* consiste à définir le nombre de vecteurs de données artificielles à insérer en plus du vecteur marqué afin de dissimuler nos données réelles. Pour ce faire, nous avons utilisé un modèle prédictif basique, mais rapide et performant. Ce modèle se décompose en deux étapes. La première étape consiste à dissimuler, en temps réel, nos données réelles dans un ensemble restreint de données artificielles en fonction de deux paramètres α et β, ou α ($0 < \alpha \leq 1$) représente la probabilité que n vecteurs de données artificielles soient générés pour chaque insertion d'un vecteur de données réelles avec $n = random(0, \beta)$ et $\beta \in N^+$. La seconde étape se réalise à postériori, c'est-à-dire lors du déclenchement d'un événement comme par exemple la fin d'un pas de temps ou encore l'atteinte d'une valeur limite de vecteurs de données insérés afin de finaliser correctement la dissimulation des données. Pour cela, nous définissons un paramètre τ ($\tau \in N$) représentant le nombre de vecteurs de données artificielles et réelles déjà insérés, qui permet d'indiquer au composant, si sa valeur est positive, de déclencher cette phase. La génération des nouveaux vecteurs de données se réalise alors par le biais de deux nouveaux paramètres λ ($\lambda \in N^+$) et θ ($\theta \in N^+$), où λ représente un objectif de nombre de vecteurs de données réelles et artificielles à avoir insérés au déclenchement de l'événement et θ la différence entre le nombre de vecteurs de données déjà insérés (i.e. τ) et l'objectif λ à atteindre ce qui permet une génération des θ vecteurs de données artificielles manquants. Ensuite, avec une probabilité de 1-α, n autres vecteurs de données artificielles sont insérés, où $n = random(0, \varepsilon)$ et $\varepsilon \in N^+$. L'insertion de ces vecteurs de données supplémentaires permet en cas de découverte des paramètres du modèle prédictif de ne pas retrouver le nombre précis de vecteurs de données réelles insérés.

Le *sous-composant de génération de données* permet de générer le nombre de vecteurs de données artificielles donné par le modèle prédictif à l'aide du générateur de données que nous avons développé. Cette génération des données doit s'effectuer suivant la langue des données réelles notée *l*, afin d'augmenter le réalisme des données artificielles. Enfin, l'objectif du *sous-composant de marquage de données* est de marquer le vecteur de données à insérer. Ce vecteur est défini comme suit : $V = \langle (d_1, t_1, m_1), (d_2, t_2, m_2), \cdots, (d_k, t_k, m_k) \rangle$ où d_i est la $i^{ème}$ donnée du vecteur, t_i en est son type, m_i signifie si la donnée d_i doit être marquée ($m_i = 1$) ou non ($m_i = 0$) et *k* le nombre de champs dans le vecteur de données. Les méthodes traditionnelles de « *watermarking* » marquent les données selon les données elles mêmes et un élément secret connu seulement par le propriétaire légitime des données. Ainsi, lors de la vérification du marquage, si le nombre de données présent n'est pas suffisant, alors il n'est plus possible de vérifier avec certitude que les données appartiennent bien au propriétaire légitime des données. Cependant, dans le cadre de ce travail, nous devons pouvoir à partir d'un vecteur de données savoir avec certitude que ce vecteur est un vecteur de données réelles ou artificielles sans avoir connaissance des autres vecteurs de données. En outre, généralement les méthodes de « *watermarking* » reposent sur une dégradation partielle non réversible des données. Or, ceci n'est pas toujours acceptable, comme par exemple pour certains types d'informations concernant un client comme son nom, son adresse, etc. qui sont cruciales pour l'identifier de manière unique. Ainsi, nous avons besoin d'une méthode de marquage fonctionnant en temps réel et dégradant de manière réversible les données. Afin de marquer les vecteurs de données réelles, nous utilisons le même principe que les méthodes de « *watermarking* », c'est-à-dire l'utilisation d'une clé privée connue seulement du propriétaire légitime des données. La méthode consiste donc à réaliser un hachage de cette clé privée concaténée à la donnée devant être marquée puis d'extraire, de ce haché, les *m* premiers bits qui seront concaténés à la donnée elle-même (i.e. les données devant être marquées). Ce même principe est réalisé sur les données artificielles hormis que nous altérons les *m* bits afin que lors de la détection, ces vecteurs de données artificielles ne soient pas reconnus comme réels.

Ainsi, notre composant de dissimulation de données (voir Fig. 1) possède les entrées données en tableau 3 et fourni en sortie l'ensemble des vecteurs de données réelles et artificielles qui seront insérés dans la base de données, soit $\{V_k, [V]_n\}$.

TABLEAU III. ENTRÉES DU COMPOSANT DE DISSIMULATION DE DONNÉES

Entrée	Description
v	le vecteur de données réelles
k	la clé privée utilisée pour le marquage des données
m	la taille en bits de la marque
l	la langue utilisée pour la génération des données
α et β	avec α la probabilité que n vecteurs de données artificielles soient générés, où $n = random(0, \beta)$
τ	ayant une valeur positive au déclenchement d'un événement
λ	l'objectif du nombre de vecteurs de données réelles et artificielles à avoir générés à la fin du déclenchement de l'évènement
ε	la probabilité 1-α que n vecteurs de données artificielles soient générés, où $n = random(0, \varepsilon)$

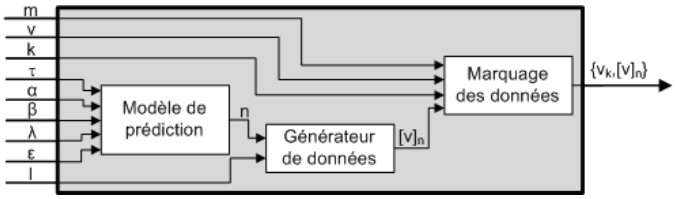

Figure 1. Composant de dissimulation de données

VII. ÉVALUATION DE PERFORMANCE

Afin d'évaluer les performances de notre composant, nous avons réalisé des tests de performance concernant la durée de trois phases : la génération des données (i.e. la génération des données artificielles), le marquage des données (i.e. marquer les vecteurs de données réelles) et enfin l'extraction des données (i.e. l'extraction des vecteurs de données réelles parmi un ensemble de vecteurs de données). Nous avons aussi évalué l'impact de notre composant de dissimulation de données sur les données générées. Pour tous ces tests, nous avons fixé le nombre de données de chacun des vecteurs à 11. Les tests ont été réalisés sur une machine dotée d'un processeur Intel Core 2 Duo T9600 (2,8 GHz) ayant 1 Go de mémoire vive.

Dans le premier test (voir Fig. 2), nous avons testé le temps de génération des vecteurs de données artificielles. Dans le cadre de ce test, nous avons généré plusieurs types complexes de données, par exemple un type nom, un type adresse, un type email, etc. Sur la Fig. 2, nous pouvons voir que le temps de génération de 50 vecteurs de données reste très rapide avec un temps de génération de l'ordre de 20 ms. Dans le second test, nous avons évalué le temps de marquage des données réelles par le sous-composant de marquage (voir Fig. 3) avec le marquage de deux données par vecteur de données. Sur la Fig. 3, nous pouvons observer que le marquage se réalise extrêmement rapidement avec le marquage de 1000 vecteurs de données en moins de 30 ms. Dans le troisième test, nous avons évalué le temps d'extraction des vecteurs de données réelles par le composant de marquage (voir Fig. 4). Comme dans le test précédent, nous avons choisi de marquer deux données dans chacun des vecteurs de données. Pour chaque ensemble de vecteurs testé, 80 pourcents des vecteurs ont été marqués aléatoirement comme des vecteurs de données réelles. Sur la Fig. 4, nous pouvons observer que l'extraction des vecteurs de données réelles se réalise extrêmement rapidement avec l'extraction de 800 vecteurs parmi un ensemble de 1000 vecteurs de données en moins de 10 ms. Dans un autre test, nous avons évalué l'impact de notre composant de dissimulation de données sur les données générées. La Fig. 5 montre les résultats de l'utilisation du composant de dissimulation de données sur une période d'une heure, tandis que la Fig. 6 montre le résultat sur une période de 24 heures. Dans le cadre de ce test, nous avons utilisé un jeu réel de données de transactions bancaires d'un site d'e-commerce et nous avons défini les paramètres du composant comme ceci : m=8, α=0.3, β=1, λ=90, ε=20. Pour ce test, une seule donnée par vecteur est marquée dans chacun des vecteurs et la génération des données est réalisée avec des données de la langue française. La taille de la clé privée est de 15 octets et l'événement utilisé par le sous-composant de prédiction est le changement d'heure. Sur la Fig. 5, la courbe bleue (annotée *sans)* représente les données sans l'utilisation de notre composant de dissimulation de données, la courbe rouge (annotée *avec)* celles des données avec utilisation du composant de dissimulation de données et la courbe verte (annotée *ratio)* le ratio entre les deux. Nous pouvons ainsi voir que l'utilisation de notre composant de dissimulation de données perturbe très bien les données permettant ainsi une bonne dissimulation des données réelles avec un taux de génération de données artificielles relativement faible. Sur la Fig. 6, nous avons utilisé le même code de couleurs et nous pouvons observer le même résultat que précédemment en ce qui concerne la dissimulation des données. De plus, nous pouvons voir qu'aucune corrélation ne peut être réalisée entre les données réelles et artificielles comme en observant les résultats donnés à 11h et 17h. Ici, les valeurs sont les mêmes alors que le nombre de données réelles n'est pas identique.

Le principal défaut de la méthode utilisée pour générer les données artificielles est qu'un taux relativement important de celles-ci est généré durant les périodes creuses, par exemple entre 1 heure et 5 heures comme nous pouvons le voir sur la Fig. 6. Cependant, en définissant deux jeux de paramètres pour le modèle prédictif suivant les plages horaires, il est tout à fait possible de réduire significativement ce surplus de données artificielles, comme dans l'exemple donné en Fig. 7. Enfin, en ce qui concerne le sous-composant de prédiction, s'agissant d'un calcul arithmétique simple et de générations aléatoires de nombres, le temps pour effectuer le calcul est de l'ordre d'une milliseconde.

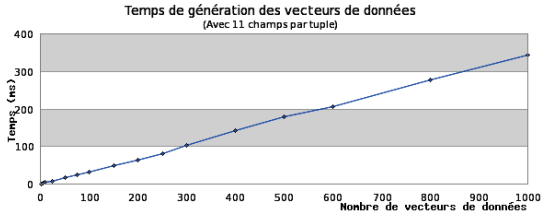

Figure 2. Temps de génération des vecteurs artificiels

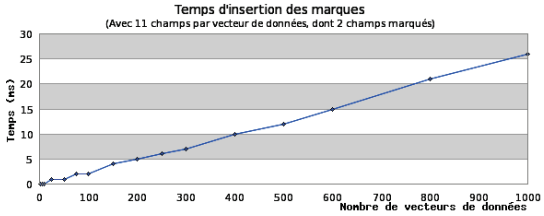

Figure 3. Temps d'insersion des marques

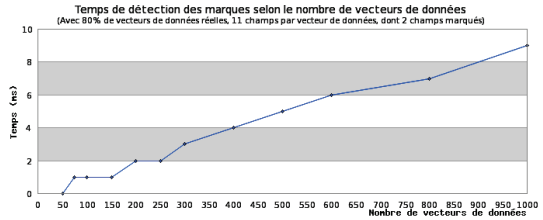

Figure 4. Temps de detection des marques

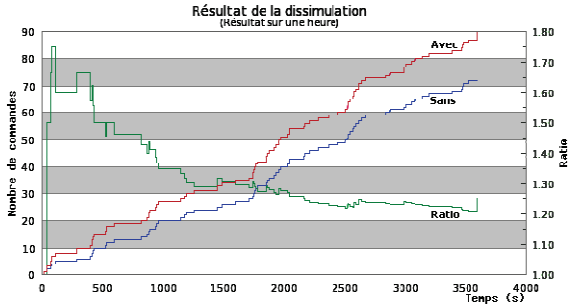

Figure 5. Résultat de la dissimulation sur une periode d'une heure

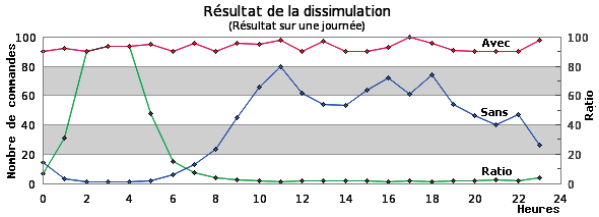

Figure 6. Résultat de la dissimulation sur une periode d'une journée

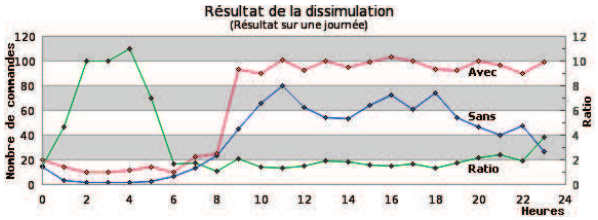

Figure 7. Résultat de la dissimulation avec une adaptation des paramètres

VIII. CONCLUSION ET TRAVAUX FUTURS

Dans cet article, nous avons tout d'abord présenté brièvement le paradigme du cloud computing. Puis, nous avons présenté les principaux problèmes de sécurité que pose le cloud computing et qui doivent être pris en compte afin de permettre une adoption plus franche par les PMEs. Ensuite, nous nous sommes focalisés sur le problème de confidentialité des données stockées dans les bases de données du cloud, tout particulièrement lors de l'utilisation d'un cloud dans le cadre du e-commerce. Afin de résoudre ce problème, nous avons proposé un composant de dissimulation de données que nous avons mise en œuvre et testé en termes de performance. Nous souhaitons souligner le fait que même si notre composant a été conçu afin de sécuriser les bases de données dans les clouds, il peut être également utilisé pour les bases de données traditionnelles, pour lesquelles, à notre connaissance, il n'existe pas de solution similaire. Lors des tests d'évaluation, nous avons pu mesurer l'impact de notre composant. Bien que notre solution soit efficace, il est nécessaire d'améliorer la méthode de marquage des données afin d'éviter la concaténation de la marque à la donnée. Ainsi, nous avons travaillé sur une méthode de marquage qui utilise le principe de la clef secrète mais qui permet une dégradation réversible des données contrairement aux méthodes de « watermarking » existantes. Comme travaux futurs, nous projetons de tester notre composant dans un environnement réel afin d'estimer son impact en termes de ressources processeurs et mémoires.

RÉFÉRENCES

[1] Open Crowd. Cloud Taxonomy, Landscape, Evolution. June 8, 2010. http://www.opencrowd.com/assets/images/views/views_cloud-tax-lrg.png.

[2] P.Mell and T. Grance. NIST definition of cloud computing. National Institute of Standards and Technology. October, 2009.

[3] Cloud Security Alliance. Security Guidance for Critical Areas of Focus in Cloud Computing V2.1. Technical report. Cloud Security Alliance. December, 2009.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. 2010. A view of cloud computing. *Commun. ACM, vol.* 53, April 4, 2010, pp. 50-58.

[5] G. Hogben. Privacy, Security and Identity in the Cloud. ENISA. OASIS/EEMA eIdentity conference, London. June, 2010.

[6] ENISA. Benefits, risks and recommendations for information security. European Network and Information Security Agency. November, 2009.

[7] J. Archer, A. Boehme, D. Cullinane, P. Kurtz, N. Puhlmann, J. Reavis. Top Threats to Cloud Computing V1.0. Technical Report. Cloud Security Alliance. March, 2010.

[8] Y. Chen, V. Paxson and R.H. Katz. 2010. What's New About Cloud Computing Security? Technical Report No. UCB/EECS-2010-5. http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.html

[9] F. Batard, K. Boudaoud, M. Kamel. Web 2.0 Security: State of the Art. In Proceedings of the 5th Conference On Security in Network Architecture and Information Systems (SARSSI), Rocquebrune Cap-Martin, French Riviera, France. May, 2010.

[10] J. Rutkowska. 0wning Xen in Vegas! The Invisible Things Lab's blog. July 7, 2008. http://theinvisiblethings.blogspot.com/_2008/07/0wning-xen-in-vegas.html

[11] K. Kortchinsky. CLOUDBURST. BlackHat USA 2009, Las Vegas.

[12] J. Mirkovic and P. Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. SIGCOMM Comput. Commun. Rev., vol. 34,2. Apr. 2004, pp. 39-53.

[13] R. Fléchaux. Amazon : un premier client du Cloud victime d'une attaque par déni de service. LeMagIT. October 6, 2009. http://www.lemagit.fr/article/cloud-computing-sla-panne-amazon-ec2-cloud-ddos/4460/1/amazon-premier-client-cloud-victime-une-attaque-par-deni-service/

[14] D. Danchev. Zeus crimeware using Amazon's EC2 as command and control server. ZDNet. December, 2009. http://www.zdnet.com/blog/security/zeus-crimeware-using-amazons-ec2-as-command-and-control-server/5110

[15] B. Krebs. Amazon: Hey Spammers, Get Off My Cloud! The Washington Post. July 1, 2008. http://voices.washingtonpost.com/securityfix/2008/07/amazon_hey_spammers_get_off_my.html

[16] S. H. Khor and A. Nakao. sPoW: On-Demand Cloud-based eDDoS Mitigation Mechanism. In Hot Topics in Dependency WorkShop. 2009.

[17] Burton S. Kaliski, Jr. and Wayne Pauley. 2010. Toward risk assessment as a service in cloud environments. In Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (HotCloud'10). USENIX Association, Berkeley, CA, USA, 13-13.

[18] R. Agrawal, P. J. Haas, and J. Kiernan. Watermarking relational data : framework, algorithms and analysis. VLDB J., 12(2) :157-169, 2003

[19] F. H. Wang, X. Cui, Z. Cao. A Speech Based Algorithm for Watermarking Relational Databases. International Symposiums on Information Processing. 2008.

[20] Z. Yong, N. Xia-mu, A. Khan, L.Qiong, H. Qi. A novel method of watermarking relational databases using character string. AIA'06: Proceedings of the 24th IASTED international conference on Artificial intelligence and applications .ACTA Press. 2006, pp. 120-124.

[21] N. Nobelis. Une architecture pour le transfert électronique sécurisé de document. PhD These, University of Nice - Sophia Antipolis, GLC, RAINBOW. December 15th, 2008.