Chiffrement de données au repos

Ken Beer

Ryan Holland

Novembre 2014

Table des matières

Résumé

Les politiques organisationnelles ou les réglementations sectorielles ou gouvernementales peuvent nécessiter l'utilisation du chiffrement au repos pour protéger vos données. Avec sa nature flexible, Amazon Web Services (AWS) vous permet de choisir entre de nombreuses options différentes qui répondent à vos besoins. Ce livre blanc présente les différentes méthodes de chiffrement de vos données au repos disponibles à l'heure actuelle.

Introduction

Amazon Web Services (AWS) propose une plateforme de cloud computing évolutive et sécurisée, à la disponibilité élevée, qui vous offre la flexibilité pour créer une large gamme d'applications. Si vous avez besoin d'une couche supplémentaire de sécurité pour les données que vous stockez dans le cloud, de nombreuses options existent pour le chiffrement de données au repos, des solutions de chiffrement d'AWS totalement automatisées aux options manuelles côté client. Le choix des bonnes solutions dépend du service AWS que vous utilisez et de vos exigences pour la gestion des clés. Ce livre blanc présente de nombreuses méthodes de chiffrement de données au repos dans AWS. Les liens vers des ressources supplémentaires sont fournis pour mieux comprendre comment mettre en œuvre réellement les méthodes de chiffrement ayant fait l'objet de discussions.

La clé du chiffrement : Qui contrôle les clés ?

Le chiffrement d'un système nécessite trois composants : (1) des données à chiffrer ; (2) une méthode de chiffrement des données à l'aide d'un algorithme cryptographique ; et (3) des clés de chiffrement à utiliser avec les données et l'algorithme. La majorité des langages de programmation modernes fournissent des bibliothèques avec de nombreux algorithmes cryptographiques disponibles, comme AES (Advanced Encryption Standard). Le choix du bon algorithme implique l'évaluation des exigences en matière de sécurité, de performances et de conformité spécifiques à votre application. Bien que le choix d'un algorithme de chiffrement soit important, la protection des clés contre les accès non autorisés est essentielle. La gestion de la sécurité des clés de chiffrement est souvent réalisée à l'aide d'une infrastructure de gestion des clés (KMI). Une KMI se compose de deux sous-composants : la couche de stockage qui protège les clés en texte brut et la couche de gestion qui autorise l'utilisation de la clé. Une manière fréquente de protéger les clés dans une KMI est d'utiliser un module de sécurité matérielle (HSM). Un HSM est un appareil de stockage et de traitement des données dédié qui réalise des opérations cryptographiques à l'aide des clés sur l'appareil. Un HSM fournit généralement des preuves ou une résistance contre les détériorations, pour protéger les clés contre une utilisation non autorisée. Une couche d'autorisation basée sur le logiciel contrôle les personnes pouvant administrer le HSM et quels utilisateurs ou applications peuvent utiliser quelles clés dans le HSM.

Lorsque vous déployez le chiffrement pour de nombreuses classifications de données dans AWS, il est important de comprendre exactement qui a accès à vos clés ou données de chiffrement et dans quelles conditions. Comme le montre la figure 1, il existe trois modèles différents de fournir la méthode de chiffrement et la KMI.

- Vous contrôlez la méthode de chiffrement et l'ensemble de la KMI.
- Vous contrôlez la méthode de chiffrement, AWS fournit le composant de stockage pour la KMI et vous vous chargez de la couche de gestion de la KMI.
- AWS contrôle la méthode de chiffrement et l'ensemble de la KMI.

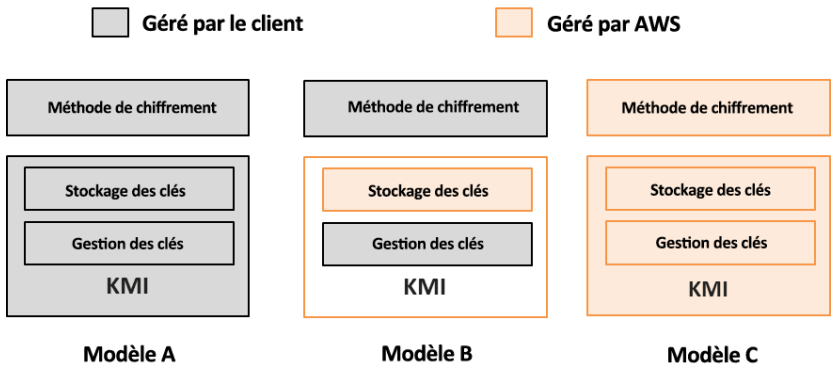

Figure 1 : Modèles de chiffrement dans AWS

Modèle A : Vous contrôlez la méthode de chiffrement et l'ensemble de la KMI

Dans ce modèle, vous utilisez votre propre KMI pour générer, stocker et gérer l'accès aux clés, ainsi que pour contrôler toutes les méthodes de chiffrement dans vos applications. Cet emplacement physique de la KMI et la méthode de chiffrement peuvent être extérieurs à AWS ou se trouver dans l'instance Amazon EC2 (Amazon Elastic Compute Cloud) que vous possédez. La méthode de chiffrement peut être une combinaison d'outils open source, de SDK AWS ou de logiciel et/ou matériel tiers. La propriété de sécurité importante de ce modèle est que vous avez le contrôle complet des clés de chiffrement et de l'environnement d'exécution qui utilise ces clés dans le code de chiffrement. AWS n'a pas accès à vos clés et ne parvient pas à effectuer le chiffrement ou le déchiffrement en votre nom. Vous êtes chargé du stockage, de la gestion et de l'utilisation corrects des clés, pour garantir la confidentialité, l'intégrité et la disponibilité de vos données. Les données peuvent être chiffrées dans des services AWS de la manière décrite dans les sections suivantes.

Amazon S3

Vous pouvez chiffrer des données à l'aide d'une méthode de chiffrement souhaitée, puis vous téléchargerez les données chiffrées à l'aide de l'API Amazon S3 (Amazon Simple Storage Service). Les principaux langages d'application incluent des bibliothèques cryptographiques qui vous permettent de réaliser le chiffrement dans vos applications. Deux outils open source habituellement disponibles sont Bouncy Castle et OpenSSL. Après avoir chiffré un objet et stocké en toute sécurité la clé dans votre KMI, l'objet chiffré peut être téléchargé sur Amazon S3 directement avec une demande PUT. Pour déchiffrer ces données, vous émettez la demande GET dans l'API Amazon S3, puis vous transférez les données chiffrées vers votre application locale pour qu'elle les déchiffre.

AWS constitue une alternative à ces outils de déchiffrement open source avec le client de chiffrement Amazon S3, qui est un ensemble d'API open source intégrées aux SDK AWS. Ce client vous fournit une clé depuis votre KMI pouvant être utilisée pour chiffrer ou déchiffrer vos données dans le cadre de l'appel vers Amazon S3. Le SDK tire parti de vos JCE (Java Cryptography Extension) dans votre application pour utiliser votre clé symétrique ou asymétrique comme entrée et chiffrer l'objet avant de le télécharger dans Amazon S3. Le processus est inversé lorsque le SDK est utilisé pour récupérer un objet. L'objet chiffré téléchargé depuis Amazon S3 est transféré au client avec la clé de votre KMI. Le JCE sous-jacent dans votre application déchiffre l'objet.

Le client de chiffrement Amazon S3 est intégré aux SDK AWS pour Java, Ruby et .NET, et est un remplacement transparent pour le code cryptographique que vous aviez avec votre application qui interagit avec Amazon S3. Bien qu'AWS fournisse la méthode de chiffrement, vous contrôlez la sécurité de vos données car vous contrôlez les clés à utiliser pour ce moteur. Si vous utilisez le client de chiffrement d'Amazon S3 sur site, AWS n'a jamais accès à vos clés ou données non chiffrées. Si vous utilisez le client dans une application exécutée sur Amazon EC2, une bonne pratique consiste à transférer des clés vers le client à l'aide d'un transport sécurisé (p. ex., Secure Sockets Layer [SSL] ou Secure Shell [SSH]) depuis votre KMI pour aider à garantir la confidentialité. Pour plus d'informations, consultez la documentation du SDK AWS pour Java et la section Utilisation du chiffrement côté client dans le *Manuel du développeur d'Amazon S3*. La figure 2 montre comment ces deux méthodes de chiffrement côté client fonctionnent pour les données Amazon S3.

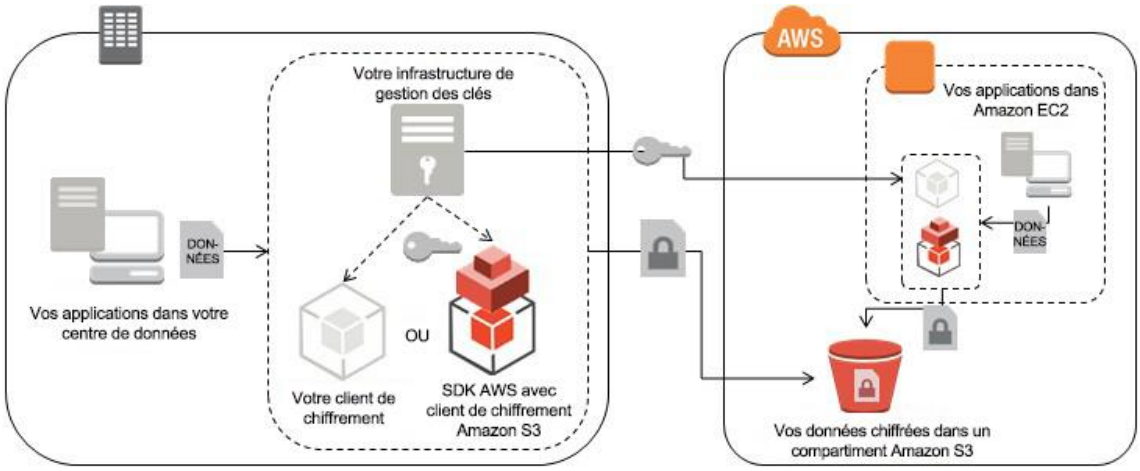

Figure 2 : Chiffrement côté client Amazon S3 depuis le système sur site ou votre application Amazon EC2

Des solutions tierces sont disponibles, qui permettent de simplifier le processus de gestion de clé lors du chiffrement des données dans Amazon S3. CloudBerry Explorer PRO pour Amazon S3 et CloudBerry Backup proposent tous les deux une option de chiffrement côté client qui utilise un mot de passe défini par l'utilisateur sur le schéma de chiffrement, afin de protéger les fichiers stockés dans Amazon S3. Pour les besoins de chiffrement par programmation, SafeNet ProtectApp for Java s'intègre à la KMI SafeNet KeySecure pour proposer un chiffrement côté client dans votre application. La KMI KeySecure fournit un stockage de clé sécurisé et l'application des politiques pour des clés transmises au client Java ProtectApp compatible avec le SDK AWS. La KMI KeySecure peut être exécutée comme une appliance sur site ou une appliance virtuelle dans Amazon EC2. La figure 3 montre comment la solution SafeNet peut être utilisée pour chiffrer des données stockées sur Amazon S3.

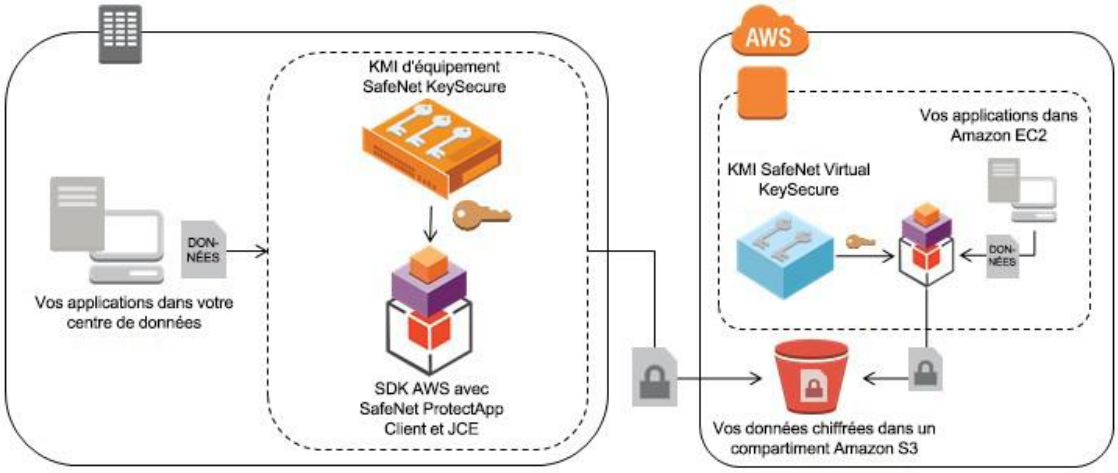

Figure 3 : Chiffrement côté client Amazon S3 depuis le système sur site ou votre application dans Amazon EC2 à l'aide de la KMI SafeNet ProtectApp et SafeNet KeySecure

Amazon EBS

Le service Amazon EBS (Amazon Elastic Block Store) fournit des volumes de stockage au niveau du bloc destinés à être utilisés avec les instances Amazon EC2. Les volumes Amazon EBS sont liés au réseau et persistent indépendamment de la vie d'une instance.

Étant donné que les volumes Amazon EBS sont présentés sur une instance comme un périphérique de stockage en mode bloc, vous pouvez tirer parti de la plupart des outils de chiffrement standard pour un chiffrement au niveau du système de fichiers ou du bloc. Certaines solutions fréquentes de chiffrement open source au niveau du bloc pour Linux sont *Loop-AES*, *dm-crypt* (avec ou sans) *LUKS* et *TrueCrypt*. Chaque solution fonctionne sous la couche du système de fichiers à l'aide de pilotes d'appareils d'espace de noyau pour réaliser le chiffrement et le déchiffrement des données. Ces outils sont utiles lorsque vous voulez que toutes les données écrites sur un volume soient chiffrées, quel que soit le répertoire dans lequel les données sont stockées.

Une autre option serait d'utiliser le chiffrement au niveau du système de fichiers, qui fonctionne en empilant un système de fichiers chiffré au-dessus d'un système de fichiers existant. Cette méthode est typiquement utilisée pour le chiffrement d'un répertoire spécifique. *eCryptfs* et *EncFs* sont deux exemples d'outils de chiffrement open source basés sur Linux au niveau du système de fichiers.

Ces solutions nécessitent que vous fournissiez des clés, manuellement ou depuis votre KMI. Une mise en garde importante avec les outils de chiffrement au niveau du bloc et du système de fichiers est qu'ils ne peuvent être utilisés que pour chiffrer des volumes de données qui ne sont pas des volumes de démarrage Amazon EBS. Cela est dû au fait que ces outils ne vous permettent pas de rendre une clé approuvée automatiquement disponible sur le volume de démarrage au démarrage.

Le chiffrement de volumes Amazon EBS liés à des instances Windows peut être réalisé à l'aide de *BitLocker* ou de *Encrypted File System (EFS),* ainsi que d'applications open source comme TrueCrypt. Dans tous les cas, vous devez toujours fournir ces clés pour ces méthodes de chiffrement et vous pouvez uniquement chiffrer des volumes de données.

Il existe des solutions partenaires d'AWS qui permettent d'automatiser le processus de chiffrement des volumes Amazon EBS, ainsi que pour la fourniture et la protection des clés nécessaires. Trend Micro SecureCloud et SafeNet ProtectV sont deux produits partenaires qui chiffrent des volumes Amazon EBS et incluent une KMI. Les deux produits peuvent chiffrer des volumes de démarrage en plus des volumes de données. Ces solutions prennent également en charge des cas d'utilisation dans lesquels les volumes Amazon EBS sont fixés à des instances Amazon EC2 d'un groupe Auto Scaling. La figure 4 montre comment les solutions SafeNet et Trend Micro peuvent être utilisées pour chiffrer des données stockées sur Amazon EBS à l'aide de clés gérées sur site, via un logiciel en tant que service (SaaS) ou via un logiciel exécuté sur Amazon EC2.

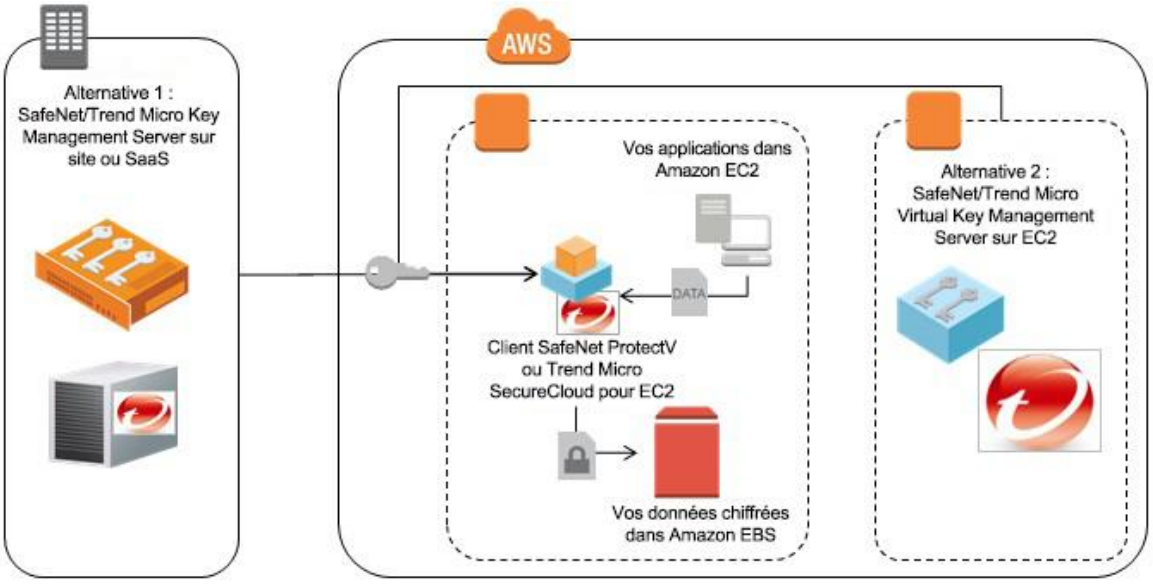

Figure 4 : Chiffrement dans Amazon EBS à l'aide de SafeNet ProtectV ou de Trend Micro SecureCloud

AWS Storage Gateway

AWS Storage Gateway est un service qui connecte en toute sécurité une appliance logicielle sur site à Amazon S3. Il peut être exposé à votre réseau comme un disque iSCSI pour faciliter la copie de données depuis d'autres sources. Les données sur les volumes de disque liés à AWS Storage Gateway seront automatiquement téléchargées sur Amazon S3 en fonction de la politique. Vous pouvez chiffrer des données source sur les volumes de disque à l'aide de n'importe quelle méthode de chiffrement de fichier décrite précédemment (p. ex., Bouncy Castle ou OpenSSL) avant qu'elles n'atteignent le disque. Vous pouvez également utiliser un outil de chiffrement au niveau du bloc (p. ex., BitLocker ou dm-crypt/LUKS) dans le point de terminaison iSCSI exposé par AWS Storage Gateway pour chiffrer toutes les données sur le volume de disque. En outre, deux solutions partenaires AWS, Trend Micro SecureCloud et SafeNet StorageSecure, peuvent assurer le chiffrement et la gestion des clés pour le volume de disque iSCSI exposé par AWS Storage Gateway. Ces partenaires fournissent une solution facile à cases à cocher pour le chiffrement de données et la gestion des clés nécessaires, dont la conception est similaire au fonctionnement de leurs solutions de chiffrement Amazon EBS.

Amazon RDS

Le chiffrement des données dans Amazon RDS (Amazon Relational Database Service) à l'aide de la technologie côté client nécessite que vous réfléchissiez à la manière dont vous voulez que les requêtes de données fonctionnent. Étant donné qu'Amazon RDS n'expose pas le disque lié qu'il utilise pour le stockage des données, le chiffrement des données transparent à l'aide de techniques décrites dans la section précédente sur Amazon EBS n'est pas disponible pour vous. Toutefois, un chiffrement sélectif des champs de base de données dans votre application peut être réalisé à l'aide de l'une des bibliothèques de chiffrement standard mentionnées précédemment (p. ex., Bouncy Castle, OpenSSL) avant que les données ne soient transférées à votre instance d'Amazon RDS. Alors que ces données de champ spécifiques ne prennent pas facilement en charge des requêtes de plage dans la base de données, les requêtes basées sur les champs qui ne sont pas chiffrés peuvent toujours retourner des résultats utiles. Les champs chiffrés des résultats retournés peuvent être déchiffrés par votre application locale pour une présentation. Pour prendre en charge des requêtes plus efficaces des données chiffrées, vous pouvez stocker un HMAC à clés d'un champ chiffré dans votre schéma et vous pouvez fournir une clé pour la fonction de hachage. Les requêtes suivantes des champs protégés qui contiennent le HMAC des données recherchées ne divulgueront pas les valeurs en texte brut dans la requête. Cela permet à la base de données d'effectuer une requête sur base des données chiffrées dans votre base de données sans divulguer les valeurs en texte brut dans la requête. Toute méthode de chiffrement que vous choisissez doit être réalisée sur votre propre instance d'application avant que les données ne soient envoyées à l'instance d'Amazon RDS.

CipherCloud et Voltage Security sont deux partenaires d'AWS proposant des solutions qui simplifient la protection de la confidentialité des données dans Amazon RDS. Les deux fournisseurs peuvent chiffrer les données à l'aide du chiffrement qui conserve le format (FPE) qui permet d'insérer du texte chiffré dans la base de données sans interrompre le schéma. Ils prennent également en charge des options de tokenisation avec des tables de recherche intégrées. Dans tous les cas, vos données sont chiffrées en tokenisées dans votre application avant d'être écrites dans l'instance Amazon RDS. Ces partenaires proposent des options pour l'indexation et la recherche dans les bases de données avec des champs chiffrés ou tokenisés. Les données non chiffrées ou non tokenisées peuvent être lues depuis la base de données par d'autres applications sans devoir distribuer de clés ou de tables de mapping à ces applications pour déverrouiller les champs chiffrés ou tokenisés. Par exemple, vous pouvez déplacer des données d'Amazon RDS vers la solution d'entreposage des données Amazon Redshift et exécuter des requêtes sur des champs non sensibles, tout en conservant les champs sensibles chiffrés ou tokenisés. La figure 5 montre comment la solution Voltage peut être utilisée avec Amazon EC2 pour chiffrer des données avant qu'elles ne soient écrites sur l'instance Amazon RDS. Les clés de chiffrement sont extraites de la KMI Voltage située dans votre centre de données par le client Voltage Security qui est exécuté sur vos applications sur Amazon EC2.

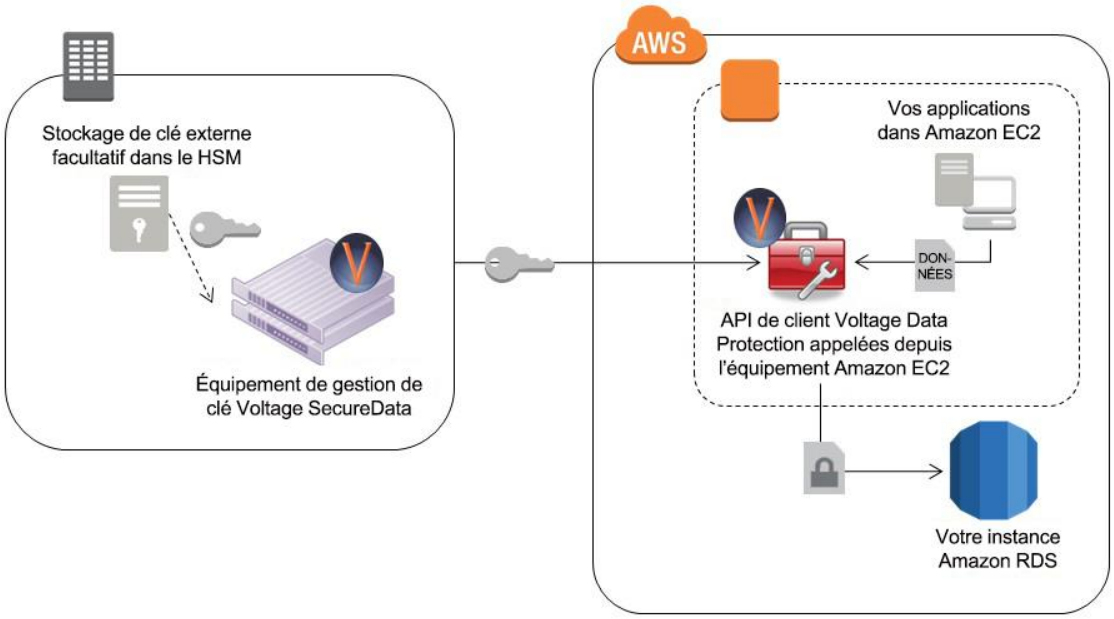

Figure 5 : Chiffrement de données dans vos applications Amazon EC2 avant leur écriture sur Amazon RDS à l'aide de Voltage SecureData

CipherCloud for Amazon Web Services est une solution dont le fonctionnement est similaire à celui du client Voltage Security pour des applications exécutées sur Amazon EC2 qui doivent envoyer des données chiffrées vers et depuis Amazon RDS. CipherCloud fournit un pilote JDBC qui peut être installé sur l'application, qu'il soit exécuté sur Amazon EC2 ou dans votre centre de données. En outre, la solution CipherCloud for Any App peut être déployée comme une passerelle en ligne pour intercepter des données lorsqu'elles sont envoyées vers et depuis votre instance Amazon RDS. La figure 6 montre comment la solution CipherCloud peut être déployée de cette manière pour chiffrer ou tokeniser les données provenant de votre centre de données avant qu'elles ne soient écrites sur l'instance Amazon RDS.

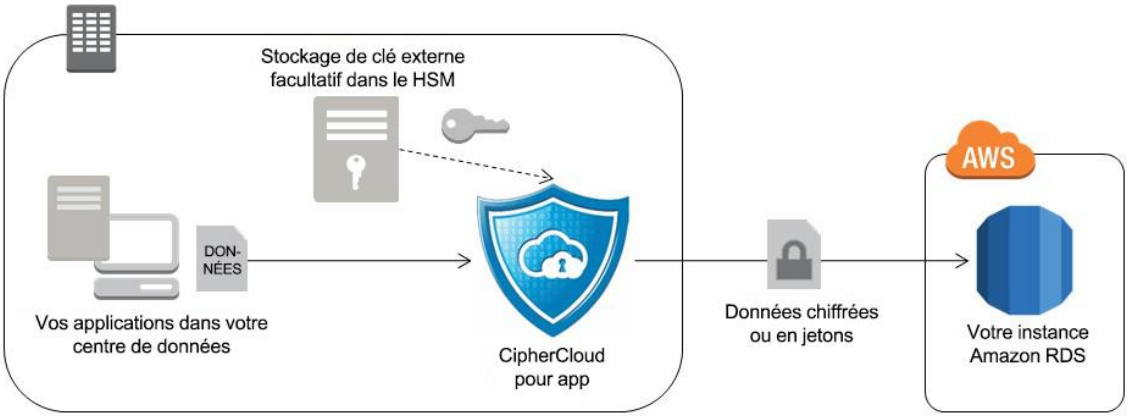

Figure 6 : Chiffrement de données dans votre centre de données avant leur écriture sur Amazon RDS à l'aide de CipherCloud Encryption Gateway

Amazon EMR

Amazon EMR (Amazon Elastic MapReduce) propose une mise en œuvre facile à utiliser de Hadoop sur Amazon EC2. La réalisation du chiffrement pendant l'opération MapReduce implique le chiffrement et la gestion des clés à quatre points distincts :

1. les données source ;
2. le système de fichiers distribué Hadoop (HDFS) ;
3. la phase de remaniement ;
4. les données de sortie.

Si les données source ne sont pas chiffrées, cette étape peut être ignorée et SSL peut être utilisé pour protéger les données qui transitent vers le cluster Amazon EMR. Si les données source sont chiffrées, votre tâche MapReduce devra pouvoir déchiffrer les données lorsqu'elles sont assimilées. Si votre flux de travail utilise Java et si les données source se trouvent sur Amazon S3, vous pouvez utiliser l'une des méthodes de déchiffrement du client décrites dans les sections précédentes sur Amazon S3.

Le stockage utilisé pour le point de montage HDFS est le stockage éphémère des nœuds de cluster. En fonction du type d'instance, il peut y avoir plusieurs montages. Le chiffrement de ces points de montage nécessite l'utilisation d'un script d'amorçage Amazon EMR qui procèdera comme suit :

- arrêter le service Hadoop ;
- installer un outil de chiffrement du système de fichiers sur l'instance ;
- créer un répertoire chiffré pour monter le système de fichiers chiffré par dessus les points de montage existants ;
- redémarrer le service Hadoop.

Par exemple, vous pourriez réaliser ces étapes à l'aide du package open source eCryptfs et d'une clé éphémère générée dans votre code sur chaque montage HDFS. Vous ne devez pas vous inquiéter du stockage permanent de cette clé de chiffrement car les données qu'elle chiffre ne subsistent pas après la durée de vie de l'instance HDFS.

La phase de remaniement implique le transfert de données entre les nœuds de cluster avant l'étape de réduction. Pour chiffrer ces données en transit, vous pouvez activer SSL avec une option de configuration d'amorçage Hadoop lorsque vous créez votre cluster.

Enfin, pour permettre le chiffrement des données de sortie, votre tâche MapReduce doit chiffrer la sortie à l'aide d'une clé provenant de votre KMI. Ces données peuvent être envoyées vers Amazon S3 pour être stockées sous forme chiffrée.

Modèle B : Vous contrôlez la méthode de chiffrement, AWS fournit le composant de stockage pour la KMI et vous vous chargez de la couche de gestion de la KMI

Ce modèle est similaire au Modèle A car vous gérez la méthode de chiffrement. Mais il est différent du Modèle A car les clés sont stockées dans une appliance AWS CloudHSM et pas dans un système de stockage de clé que vous gérez sur site. Alors que les clés sont stockées dans l'environnement AWS, elles ne sont pas accessibles pour les employés d'AWS. Cela est dû au fait que vous seul avez accès aux partitions cryptographiques dans le HSM dédié pour utiliser les clés. L'appliance AWS CloudHSM possède des mécanismes physiques et logiques de réponse et de détection d'effraction qui déclenchent une remise à zéro de l'appliance. La remise à zéro efface la mémoire volatile du HSM dans laquelle les clés en cours de déchiffrement sont stockées et détruit la clé qui chiffre les objets stockés, ce qui a pour conséquence que toutes les clés du HSM sont inaccessibles et irrécupérables.

Lorsque vous déterminez si l'utilisation d'AWS CloudHSM convient à votre déploiement, il est important de comprendre le rôle joué par un HSM dans le chiffrement de données. Un HSM peut être utilisé pour générer et stocker des clés et effectuer des opérations de chiffrement et de déchiffrement, mais il n'utilise pas de fonction de gestion du cycle de vie de clé (p. ex., politique de contrôle d'accès, rotation des clés). Cela signifie qu'une KMI compatible peut être nécessaire en plus de l'appliance AWS CloudHSM avant le déploiement de votre application. La KMI fournie peut être déployée sur site ou sur Amazon EC2 et communiquer de manière sécurisée avec l'instance AWS CloudHSM via SSL pour protéger les données et les clés de chiffrement. Étant donné que le service AWS CloudHSM utilise des appliances SafeNet Luna, tout serveur de gestion de clés qui prend en charge la plateforme SafeNet Luna peut également être utilisé avec AWS CloudHSM. Toutes les options de chiffrement décrites pour les services AWS dans le Modèle A peuvent fonctionner avec AWS CloudHSM tant que la solution prend en charge la plateforme SafeNet Luna. Cela vous permet d'exécuter votre KMI dans l'environnement de calcul AWS en conservant une racine fiable dans une appliance matérielle à laquelle vous êtes le seul à pouvoir accéder.

Les applications doivent pouvoir accéder à votre appliance AWS CloudHSM dans un Amazon Virtual Private Cloud (Amazon VPC). Le client AWS CloudHSM, fourni par SafeNet, interagit avec l'appliance AWS CloudHSM pour chiffrer les données de votre application. Ensuite, les données chiffrées peuvent être envoyées à un service AWS à des fins de stockage. Les applications de base de données, de volume de disque et de chiffrement de fichier peuvent toutes être prises en charge par AWS CloudHSM et votre application personnalisée. La figure 7 montre comment la solution AWS CloudHSM fonctionne avec vos applications sur Amazon EC2 dans un Amazon VPC.

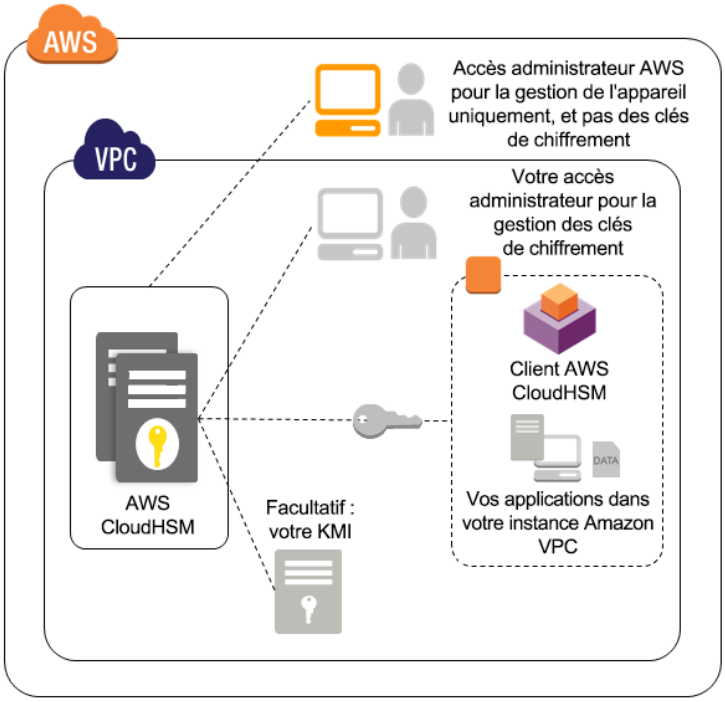

Figure 7 : AWS CloudHSM déployé sur Amazon VPC

Pour permettre la disponibilité et la durabilité maximales des clés dans votre appliance AWS CloudHSM, nous recommandons le déploiement de plusieurs applications AWS CloudHSM sur des zones de disponibilité ou avec une appliance SafeNet Luna sur site que vous gérez. La solution SafeNet Luna prend en charge la réplication sécurisée des clés sur des appliances. Pour plus d'informations, consultez AWS CloudHSM sur le site Web d'AWS.

Modèle C : AWS contrôle la méthode de chiffrement et l'ensemble de la KMI

Dans ce modèle, AWS propose un chiffrement côté serveur de vos données, en gérant en toute transparence la méthode de chiffrement et les clés.

AWS Key Management Service (KMS)

AWS Key Management Service (KMS) est un service de chiffrement géré qui vous permet d'allouer et d'utiliser des clés pour le chiffrement de vos données dans des services AWS et vos applications. Des clés principales dans AWS KMS sont utilisées d'une manière similaire à celle des clés principales dans un HSM. Après la création des clés principales, elles sont conçues pour ne jamais être exportées du service. Les données peuvent être envoyées vers le service pour être chiffrées ou déchiffrées dans une clé principale spécifique de votre compte. Cette conception vous donne un contrôle centralisé des utilisateurs qui peuvent accéder à vos clés principales pour chiffrer et déchiffrer des données, et vous permet de vérifier cet accès. AWS KMS est intégré, de manière native, à d'autres services AWS notamment Amazon EBS, Amazon S3 et Amazon Redshift pour simplifier le chiffrement de vos données dans ces services. Des SDK AWS sont intégrés à AWS KMS pour vous permettre de chiffrer des données dans vos applications personnalisées. Pour les applications qui doivent chiffrer des données, AWS KMS propose une disponibilité globale, une faible latence et une durabilité supérieure pour vos clés. Visitez https://aws.amazon.com/kms/ ou téléchargez le livre blanc KMS Cryptographic Details White Paper pour en savoir plus.

AWS KMS et d'autres services qui chiffrent vos données utilisent directement une méthode appelée chiffrement d'enveloppe pour offrir un équilibre entre performances et sécurité. La figure 8 décrit le chiffrement d'enveloppe.

1. Une clé de données est générée par le service AWS au moment où vous demandez le chiffrement de vos données.

2. La clé de données est utilisée pour chiffrer vos données.

3. La clé de données est ensuite chiffrée avec une clé de chiffrement de clé unique pour le service qui stocke vos données.

4. La clé de données chiffrée et les données chiffrées sont ensuite stockées par le service de stockage AWS en votre nom.

Figure 8 : Chiffrement d'enveloppe

Les clés de chiffrement de clé utilisées pour chiffrer les clés de données sont stockées et gérées séparément des données et des clés de données. Des contrôles d'accès stricts sont placés sur les clés de chiffrement conçues pour empêcher une utilisation non autorisée par les employés d'AWS. Lorsque vous devez accéder à vos données en texte brut, ce processus est inversé. La clé de données chiffrée est déchiffrée à l'aide de la clé de chiffrement de clé. Ensuite, la clé de données est utilisée pour déchiffrer vos données.

Les services AWS suivants proposent une variété de fonctions de chiffrement que vous pouvez choisir.

Amazon S3

Il existe trois manières de chiffrer vos données dans Amazon S3 à l'aide du chiffrement côté serveur.

1. **Chiffrement côté serveur** : Vous pouvez définir un indicateur API ou activer la case à cocher dans AWS Management Console, pour que les données soient chiffrées avant leur écriture sur le disque dans Amazon S3. Chaque objet est chiffré à l'aide d'une clé de données unique. Comme protection supplémentaire, cette clé est chiffrée avec une clé principale à rotation périodique gérée par Amazon S3. Le chiffrement côté serveur Amazon S3 utilise des clés AES (Advanced Encryption Standard) 256 bits pour les clés d'objet et principales. Cette fonction n'entraîne pas de supplément au tarif que vous payez pour utiliser Amazon S3.

2. **Chiffrement côté serveur à l'aide de clés fournies par le client** : Vous pouvez utiliser votre propre clé de chiffrement lorsque vous téléchargez un objet vers Amazon S3. Cette clé de chiffrement est utilisée par Amazon S3 pour chiffrer vos données à l'aide d'AES-256. Après le chiffrement de l'objet, la clé de chiffrement que vous avez fournie est supprimée du système Amazon S3 qui l'a utilisée pour protéger vos données. Lorsque vous récupérez cet objet depuis Amazon S3, vous devez fournir la même clé de chiffrement dans votre demande. Amazon S3 vérifie que la clé de chiffrement correspond, puis il déchiffre l'objet et vous renvoie l'objet. Cette fonction n'entraîne pas de supplément au tarif que vous payez pour utiliser Amazon S3.

3. **Chiffrement côté serveur à l'aide de KMS :** Vous pouvez chiffrer vos données dans Amazon S3 en définissant une clé principale AWS KMS sur votre compte que vous voulez utiliser pour chiffrer la clé d'objet unique (appelée clé de données dans la figure 8) qui chiffrera votre objet. Lorsque vous téléchargez votre objet, une demande est envoyée à KMS pour créer une clé d'objet. KMS génère cette clé d'objet et la chiffre à l'aide de la clé principale que vous avez spécifiée plus tôt. Ensuite, KMS renvoie cette clé d'objet chiffrée avec la clé d'objet en texte brut à Amazon S3. Le serveur Web Amazon S3 chiffre votre objet à l'aide de la clé d'objet en texte brut et stocke l'objet désormais chiffré (avec la clé d'objet chiffrée) et supprime la clé d'objet en texte brut de la mémoire. Pour récupérer cet objet chiffré, Amazon S3 envoie la clé d'objet chiffrée vers AWS KMS. AWS KMS déchiffre la clé d'objet à l'aide de la clé principale correcte et renvoie la clé d'objet (en texte brut) déchiffrée à S3. Avec la clé d'objet en texte brut, S3 déchiffre l'objet chiffré et vous le renvoie. Pour connaître le tarif de cette option, reportez-vous à la page de tarification AWS Key Management Service.

Amazon EBS

Lors de la création d'un volume dans Amazon EBS, vous pouvez choisir de le chiffrer à l'aide d'une clé principale AWS KMS sur votre compte, qui chiffrera la clé de volume unique qui permettra de chiffrer votre volume EBS. Après avoir fait votre choix, le serveur Amazon EC2 envoie une demande authentifiée à AWS KMS pour créer une clé de volume. AWS KMS génère cette clé de volume, la chiffre à l'aide de la clé principale et renvoie la clé de volume en texte brut et la clé de volume chiffrée au serveur Amazon EC2. La clé de volume en texte brut est stockée dans la mémoire pour chiffrer et déchiffrer toutes les données entrant ou sortant de votre volume EBS attaché. Lorsque le volume chiffré (ou les instantanés chiffrés provenant de ce volume) doit être à nouveau attaché à une instance, AWS KMS est appelé pour déchiffrer la clé de volume chiffrée. AWS KMS déchiffre cette clé de volume chiffrée à l'aide de la clé principale correcte et renvoie la clé de volume déchiffrée à Amazon EC2.

Amazon Glacier

Avant leur écriture sur le disque, les données sont toujours automatiquement chiffrées à l'aide de clés AES 256 bits uniques pour le service Amazon Glacier qui sont stockées dans des systèmes distincts sous contrôle d'AWS. Cette fonction n'entraîne pas de supplément au tarif que vous payez pour Amazon Glacier.

AWS Storage Gateway

AWS Storage Gateway transfère vos données sur AWS via SSL et stocke les données chiffrées au repos dans Amazon S3 ou Amazon Glacier à l'aide de leurs schémas de chiffrement côté serveur respectifs.

Amazon EMR

S3DistCp est une fonction d'Amazon EMR qui déplace de grandes quantités de données d'Amazon S3 vers HDFS, de HDFS vers Amazon S3 et entre des compartiments Amazon S3. S3DistCp prend en charge la capacité à demander à Amazon S3 d'utiliser un chiffrement côté serveur lorsqu'il écrit des données EMR dans un compartiment Amazon S3 que vous gérez. Cette fonction n'entraîne pas de supplément au tarif que vous payez pour Amazon S3 pour stocker vos données Amazon EMR.

Oracle sur Amazon RDS

Vous pouvez choisir d'établir une licence pour l'option Oracle Advanced Security pour Oracle sur Amazon RDS pour tirer parti des fonctionnalités natives TDE (Transparent Data Encryption) et NNE (Native Network Encryption). Le module de chiffrement Oracle crée des clés de chiffrement de données ou de clé pour chiffrer la base de données. Les clés de chiffrement de clé spécifiques à votre instance Oracle sur Amazon RDS sont chiffrées par une clé principale AES 256 bits à rotation périodique. Cette clé principale est unique pour le service Amazon RDS et est stockée dans des systèmes distincts sous contrôle d'AWS.

Microsoft SQL Server sur Amazon RDS

Vous pouvez choisir d'allouer le chiffrement TDE (Transparent Data Encryption) pour Microsoft SQL Server sur Amazon RDS. Le module de chiffrement SQL Server crée des clés de chiffrement de données ou de clé pour chiffrer la base de données. Les clés de chiffrement de clé spécifiques à votre instance SQL Server sur Amazon RDS sont chiffrées par une clé principale régionale AES 256 bits à rotation périodique. Cette clé principale est unique pour le service Amazon RDS et est stockée dans des systèmes distincts sous contrôle d'AWS. Cette fonction n'entraîne pas de supplément au tarif que vous payez pour utiliser Microsoft SQL Server sur Amazon Glacier.

Amazon Redshift

Lorsque vous créez un cluster Amazon Redshift, vous pouvez choisir de chiffrer toutes les données dans des tables créées par l'utilisateur. Trois options sont disponibles pour le chiffrement côté serveur d'un cluster Amazon Redshift.

1. Dans la première option, des blocs de données (et sauvegardes) sont chiffrés à l'aide de clés AES 256 bits aléatoires. Ces clés sont elles-mêmes chiffrées à l'aide d'une clé de base de données AES 256 bits aléatoire. Cette clé de base de données est chiffrée par une clé principale de cluster AES 256 bits unique pour votre cluster. La clé principale de cluster est chiffrée à l'aide d'une clé principale régionale à rotation périodique unique pour le service Amazon Redshift et stockée dans des systèmes distincts sous le contrôle d'AWS. Cette fonction n'entraîne pas de supplément au tarif que vous payez pour utiliser Amazon Redshift.

2. Dans la deuxième option, la clé principale de cluster AES 256 bits utilisée pour le chiffrement de vos clés de base de données est générée dans votre AWS CloudHSM ou à l'aide d'une appliance SafeNet Luna HSM sur site. Cette clé principale de cluster est ensuite chiffrée par une clé principale qui ne quitte jamais votre HSM. Au démarrage du cluster Amazon Redshift, la clé principale de cluster est déchiffrée dans votre HSM et utilisée pour déchiffrer la clé de base de données, qui est envoyée aux hôtes Amazon Redshift pour résider uniquement dans la mémoire pendant la durée de vie du cluster. Si le cluster redémarre, la clé principale de cluster est à nouveau récupérée depuis votre HSM, elle n'est jamais stockée sur le disque en texte brut. Cette option vous permet de mieux contrôler la hiérarchie et le cycle de vie des clés utilisées pour chiffrer vos données. Cette fonction n'entraîne pas de supplément au tarif que vous payez pour Amazon Redshift (et AWS CloudHSM si vous choisissez cette option pour le stockage de clés).

3. Dans la troisième option, la clé principale de cluster AES 256 bits utilisée pour le chiffrement de vos clés de base de données est générée dans AWS KMS. Cette clé principale de cluster est ensuite chiffrée par une clé principale dans AWS KMS. Au démarrage du cluster Amazon Redshift, la clé principale de cluster est déchiffrée dans AWS KMS et utilisée pour déchiffrer la clé de base de données, qui est envoyée aux hôtes Amazon Redshift pour résider uniquement dans la mémoire pendant la durée de vie du cluster. Si le cluster redémarre, la clé principale de cluster est à nouveau récupérée depuis l'appliance de sécurité renforcée dans AWS KMS, elle n'est jamais stockée sur le disque en texte brut. Cette option vous permet de définir des contrôles précis de l'accès et de l'utilisation de vos clés principales et de vérifier ces contrôles via AWS CloudTrail. Pour connaître le tarif de cette option, reportez-vous à la page de tarification AWS Key Management Service.

Outre le chiffrement de données générées dans votre cluster Amazon Redshift, vous pouvez également charger des données chiffrées sur Amazon Redshift depuis Amazon S3, qui ont déjà été chiffrées à l'aide d'Amazon S3 Encryption Client et des clés que vous fournissez. Amazon Redshift prend en charge le déchiffrement et le rechiffrement de données entre Amazon S3 et Amazon Redshift pour protéger le cycle de vie complet de vos données.

Ces fonctions de chiffrement côté serveur sur plusieurs services d'AWS vous permettent de chiffrer vos données facilement et simplement en définissant une configuration dans AWS Management Console ou en faisant une demande de CLI ou d'API pour le service AWS donné. L'utilisation autorisée des clés de chiffrement est gérée de manière automatique et sécurisée par AWS. Étant donné que l'accès non autorisé à ces clés pourrait entraîner la divulgation de vos données, nous avons créé des systèmes et des processus avec des contrôles d'accès forts qui limitent le risque d'accès non autorisé et ces systèmes ont été vérifiés par des audits tiers pour respecter les certifications de sécurité dont SOC 1, 2 et 3, PCI-DSS et FedRAMP.

Conclusion

Nous avons présenté trois modèles différents pour la gestion des clés de chiffrement et l'endroit où elles sont utilisées. Si vous prenez en charge la méthode de chiffrement et la KMI, vous pouvez bénéficier d'un contrôle granulaire sur la manière dont vos applications chiffrent les données. Toutefois, ce contrôle granulaire a un coût, aussi bien au niveau de l'effort de déploiement que de l'incapacité à intégrer des services AWS aux méthodes de chiffrement de vos applications. Vous pouvez également choisir un service opéré qui permet un déploiement plus facile et une plus grande intégration aux services de cloud AWS. Cette option permet le chiffrement des cases à cocher pour de nombreux services qui stockent vos données, le contrôle de vos propres clés, le stockage sécurisé de vos données et une possibilité de vérification de toutes les tentatives d'accès aux données.

Le tableau 1 résume les options disponibles pour le chiffrement de données au repos sur AWS. Nous recommandons que vous déterminiez quel modèle de chiffrement et de gestion de clé est le plus approprié à vos classifications de données dans le cadre du service AWS que vous utilisez.

	Méthode de chiffrement et KMI			
	Modèle A		**Modèle B**	**Modèle C**
Service AWS	Solutions côté client à l'aide de clés gérées par le client	Solutions partenaires côté client avec KMI pour des clés gérées par le client	Solutions côté client pour des clés gérées par le client dans AWS CloudHSM	Chiffrement côté serveur à l'aide de clés gérées par AWS
Amazon S3	Bouncy Castle, OpenSSL, Amazon S3 Encryption Client dans le SDK AWS pour Java	SafeNet ProtectApp for Java	Application personnalisée Amazon VPC-EC2 intégrée au client AWS CloudHSM	Chiffrement côté serveur d'Amazon S3, chiffrement côté serveur avec les clés fournies par le client ou chiffrement côté serveur avec AWS Key Management Service
Amazon Glacier	N/A	N/A	Application personnalisée Amazon VPC-EC2 intégrée au client AWS CloudHSM	Toutes les données sont automatiquement chiffrées à l'aide du chiffrement côté serveur

	Méthode de chiffrement et KMI			
	Modèle A		Modèle B	Modèle C
AWS Storage Gateway	Niveau Bloc Linux : - Loop-AES, dm-crypt (avec ou sans LUKS) et TrueCrypt Système de fichiers Linux : - eCryptfs et EncFs Niveau Bloc Windows : -TrueCrypt Système de fichiers Windows : - BitLocker	Trend Micro SecureCloud, SafeNet StorageSecure	N/A	Chiffrement côté serveur Amazon S3
Amazon EBS	Niveau Bloc Linux : - Loop-AES, dm-crypt+LUKS et TrueCrypt Système de fichiers Linux : - eCryptfs et EncFs Niveau Bloc Windows : -TrueCrypt Système de fichiers Windows : - BitLocker, EFS	Trend Micro SecureCloud, SafeNet ProtectV	Application personnalisée Amazon VPC-EC2 intégrée au client AWS CloudHSM	Chiffrement Amazon EBS avec AWS Key Management Service
Oracle sur Amazon RDS	Bouncy Castle, OpenSSL	CipherCloud Database Gateway et Voltage SecureData	Application personnalisée Amazon VPC-EC2 intégrée au client AWS CloudHSM	Transparent Data Encryption (TDE) et Native Network Encryption (NNE) avec licence Oracle Advanced Security facultative TDE pour Microsoft SQL Server
Microsoft SQL Server sur Amazon RDS	Bouncy Castle, OpenSSL	CipherCloud Database Gateway et Voltage SecureData	Application personnalisée Amazon VPC-EC2 intégrée au client AWS CloudHSM	N/A
Amazon Redshift	N/A	N/A	Clusters chiffrés Amazon Redshift avec votre clé principale gérée dans AWS CloudHSM ou Safenet Luna HSM sur site	Clusters chiffrés Amazon Redshift avec clé principale gérée par AWS
Amazon EMR	eCryptfs		Application personnalisée Amazon VPC-EC2 intégrée au client AWS CloudHSM	S3DistCp à l'aide du chiffrement côté serveur Amazon S3 pour protéger les données stockées de manière permanente

Tableau 1 : Résumé des options de chiffrement des données au repos

Références et suggestions de lecture

- Bibliothèque cryptographique Java Bouncy Castle http://www.bouncycastle.org/
- Bibliothèque cryptographique OpenSSL http://www.openssl.org/
- CloudBerry Explorer PRO pour le chiffrement d'Amazon S3 http://www.cloudberrylab.com/amazon-s3-explorer-pro-cloudfront-IAM.aspx
- Chiffrement de données côté client avec le SDK AWS pour Java et Amazon S3 http://aws.amazon.com/articles/2850096021478074
- Produits de chiffrement SafeNet pour Amazon S3, Amazon EBS et AWS CloudHSM http://www.safenet-inc.com/
- Trend Micro SecureCloud http://www.trendmicro.com/us/enterprise/cloud-solutions/secure-cloud/index.html
- CipherCloud for AWS et CipherCloud for Any App http://www.ciphercloud.com/
- Voltage Security SecureData Enterprise http://www.voltage.com/products/securedata-enterprise/
- AWS CloudHSM https://aws.amazon.com/cloudhsm/
- AWS Key Management Service https://aws.amazon.com/kms/
- Key Management Service Cryptographic Details White Paper https://d0.awsstatic.com/whitepapers/KMS-Cryptographic-Details.pdf
- Amazon EMR S3DistCp pour le chiffrement de données dans Amazon S3 http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/UsingEMR_s3distcp.html
- Transparent Data Encryption pour Oracle dans Amazon RDS http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.Options.html#Appendix.Oracle.Options.AdvSecurity
- Transparent Data Encryption pour Microsoft SQL Server dans Amazon RDS http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_SQLServer.html#SQLServer.Concepts.General.Options
- Chiffrement Amazon Redshift http://aws.amazon.com/redshift/faqs/#0210
- Blog de sécurité AWS http://blogs.aws.amazon.com/security

Révisions de documents

Novembre 2013 : première version

Novembre 2014 :

- Introduction de la section sur AWS Key Management Service (KMS) et Amazon EBS dans le Modèle C
- Sections mises à jour dans le Modèle C pour Amazon S3, Amazon Redshift