Système de stockage-en-ligne de photos avec confidentialité des données personnelles

Luis Montalvo, Serge Defrance, Fréderic Lefebvre, Nicolas Le Scouarnec, and Patrick Pérez

Technicolor Rennes,
1, Av. de Belle Fontaine - 35576 Cesson-Sevigné - France
prenom.nom@technicolor.com

Résumé Dans un système de stockage-en-ligne de photos, le fournisseur de service de stockage cherche à optimiser son volume de stockage et l'utilisateur cherche à garder ses données personnelles confidentielles. Ces deux intérêts peuvent entrer en conflit. En effet, si les utilisateurs confient leurs collections de photos en clair au fournisseur du service de stockage, celui-ci peut identifier les images identiques et n'en stocker qu'une seule copie, et ceci quels que soient les propriétaires de photos identiques ; en revanche, la confidentialité des données personnelles est compromise. Cependant, si les utilisateurs du service de stockage chiffrent leurs images avant de les envoyer au fournisseur du service, celui-ci ne peut plus identifier les images identiques si celles-ci ont été chiffrées avec des clefs différentes. Dans cette contribution, nous proposons un système de stockage-en-ligne qui réconcilie ces deux intérêts en apparence conflictuels. Le fournisseur du service de stockage-en-ligne a la capacité d'identifier non seulement les images identiques mais aussi les images similaires, même si elles sont chiffrées avec des clefs différentes, sans compromettre la confidentialité des données personnelles.

Keywords: stockage-en-ligne, confidentialité de données personnelles, empreinte d'image, chiffrement convergent, plus-proche-voisin, de-duplication de données.

1 Introduction

Les services commerciaux de stockage-en-ligne tels que Carbonite [1], Crashplan [2], Dropbox [3], et Mozy [4] deviennent de plus en plus populaires, principalement en raison de la protection qu'ils offrent face à certains risques (feu, inondation, vol) qu'une simple sauvegarde locale ne permet pas de prévenir [5].

Dans le but d'optimiser l'espace de stockage et les délais de téléchargement de fichiers, les fournisseurs du service de stockage (*FSS*) peuvent appliquer des techniques de de-duplication aux données des utilisateurs. Quelques *FSS* appliquent ces techniques non seulement aux données appartenant au même compte utilisateur (de-duplication intra-compte) mais également aux données appartenant à différents comptes utilisateurs (de-duplication inter-compte) [5]. La gestion de

de-duplication intra et inter-compte s'effectue sur les données en clair. Cependant, cette gestion de données en clair pose le problème de la confidentialité des données personnelles.

Un remède à la manipulation des données en clair est le chiffrement des données. Les utilisateurs pourraient chiffrer les images avant de les envoyer au fournisseur du service de stockage-en-ligne. Malheureusement, un même *message-en-clair* chiffré avec deux clefs différentes produit deux *messages-chiffrés* différents. La confidentialité des données personnelles est garantie mais le *FSS* ne peut pas détecter que les deux *message-chiffrés* correspondent à un même *message-en-clair* et il ne peut plus optimiser l'espace de stockage [6].

Dans cette contribution, nous proposons un système de stockage-en-ligne pour photos ou autres données visuelles, qui réconcilie les intérêts du *FSS* et de l'utilisateur, c'est-à-dire, un système de stockage qui permet au *FSS* d'appliquer des méthodes de de-duplication, tout en respectant la confidentialité des données personnelles.

La suite de cette contribution est organisée de la manière suivante. La section 2 présente les objectifs de notre proposition de service de *stockage-en-ligne* pour photos, et passe en revue les principes des deux technologies clefs utilisées dans notre proposition, à savoir : le *chiffrement convergent* [7] et l'*empreinte d'image*. La section 3 présente les détails de notre proposition. La section 4 décrit une méthode pour améliorer la précision globale des requêtes sur la base de données des images. La section 5 montre comment notre proposition contribue à mieux garantir la confidentialité des données personnelles des utilisateurs du service de *stockage-en-ligne*. Finalement, la section 6 présente nos conclusions.

2 Motivation et revue des technologies utilisées dans notre proposition

Notre but est de développer un système de stockage-en-ligne pour photos qui permet au fournisseur du service de stockage (*FSS*) de détecter des images identiques dans des collections d'images d'un même utilisateur ou de plusieurs utilisateurs différents, mais qui, en même temps, garantit la confidentialité des données personnelles.

Prenons l'exemple d'un fournisseur de service de stockage-en-ligne souhaitant proposer un service de mutualisation d'archivage d'images entre plusieurs utilisateurs. Les utilisateurs peuvent utiliser le service pour archiver leurs collections de photos mais ils peuvent également l'employer pour partager soit l'ensemble, soit une partie de leurs collections avec d'autres utilisateurs autorisés.

Afin de garantir la confidentialité des données personnelles et de garantir la capacité du *FSS* à détecter les copies strictement identiques ou proches dans les collections de photos des utilisateurs, le système de stockage-en-ligne doit avoir les caractéristiques suivantes :

- Le *FSS* doit avoir accès seulement aux images chiffrées des utilisateurs et le *FSS* ne doit pas pouvoir les déchiffrer.

– Le *FSS* doit pouvoir détecter que deux images chiffrées correspondent à deux images strictement identiques.
– Seul les utilisateurs autorisés peuvent déchiffrer l'ensemble ou une partie des images chiffrées qui se trouvent archivées dans le compte d'un utilisateur.
– Sur demande d'un utilisateur autorisé, le *FSS* doit pouvoir effectuer des requêtes d'images similaires, par la méthode dite du *plus-proche-voisin*, dans les collections des photos des utilisateurs sans avoir accès aux images en clair. Le résultat de telles requêtes doit être équivalent au résultat que l'utilisateur aurait obtenu s'il avait exécuté les mêmes requêtes sur une collection de photos non chiffrées.

Le système de stockage-en-ligne proposé dans cette contribution s'appuie sur deux briques technologiques principales : le *chiffrement convergent* et l'*empreinte d'image*. Le *chiffrement convergent* permet la gestion de doublons sans manipulation de données en clair. L'*empreinte d'image* permet la recherche d'images similaires dans une collection d'images.

2.1 Chiffrement convergent

Le *chiffrement convergent* [7] permet au fournisseur du service de stockage (*FSS*) d'appliquer des méthodes de de-duplication de fichiers même si les fichiers sont chiffrés avec des clefs différentes. Dans la suite, nous assumons que le lecteur est familier avec les principes de base des chiffrements symétrique et asymétrique [8].

Pour chiffrer un fichier avec la technique de *chiffrement convergent* (voir la Figure 1), Alice (l'expéditeur) procède de la façon suivante : Alice calcule une valeur de hachage cryptographique, tel que SHA-256, du contenu du fichier (*message-en-clair*) ; puis, Alice utilise cette valeur de hachage cryptographique comme clé pour chiffrer, avec un algorithme symétrique, le contenu du fichier ; finalement, Alice utilise la clef publique de Bob (le récepteur) pour chiffrer, avec un algorithme asymétrique, la valeur de hachage cryptographique. Le couple <*hachage-chiffré, contenu-chiffré*> constitue le *message-chiffré* et il est envoyé au système de *stockage-en-ligne*.

Bob peut déchiffrer le fichier de la façon suivante : d'abord, il déchiffre le *hachage-chiffré* avec sa clef secrète et puis, il déchiffre le *contenu-chiffré* avec la valeur de *hachage-en-clair* (H dans la Figure 1) comme clef.

Puisque le contenu du fichier est chiffré avec sa propre valeur de hachage cryptographique comme clef, le *contenu-chiffré* est indépendant des clefs utilisées pour protéger la valeur de hachage, il n'est dépendant que du *contenu-en-clair*. Par conséquent, le fournisseur du service de stockage (*FSS*), sans connaissance des clefs secrètes des utilisateurs, peut détecter que deux fichiers sont strictement identiques et les de-dupliquer. La Figure 2 présente d'une manière formelle le principe de *chiffrement convergent* [7].

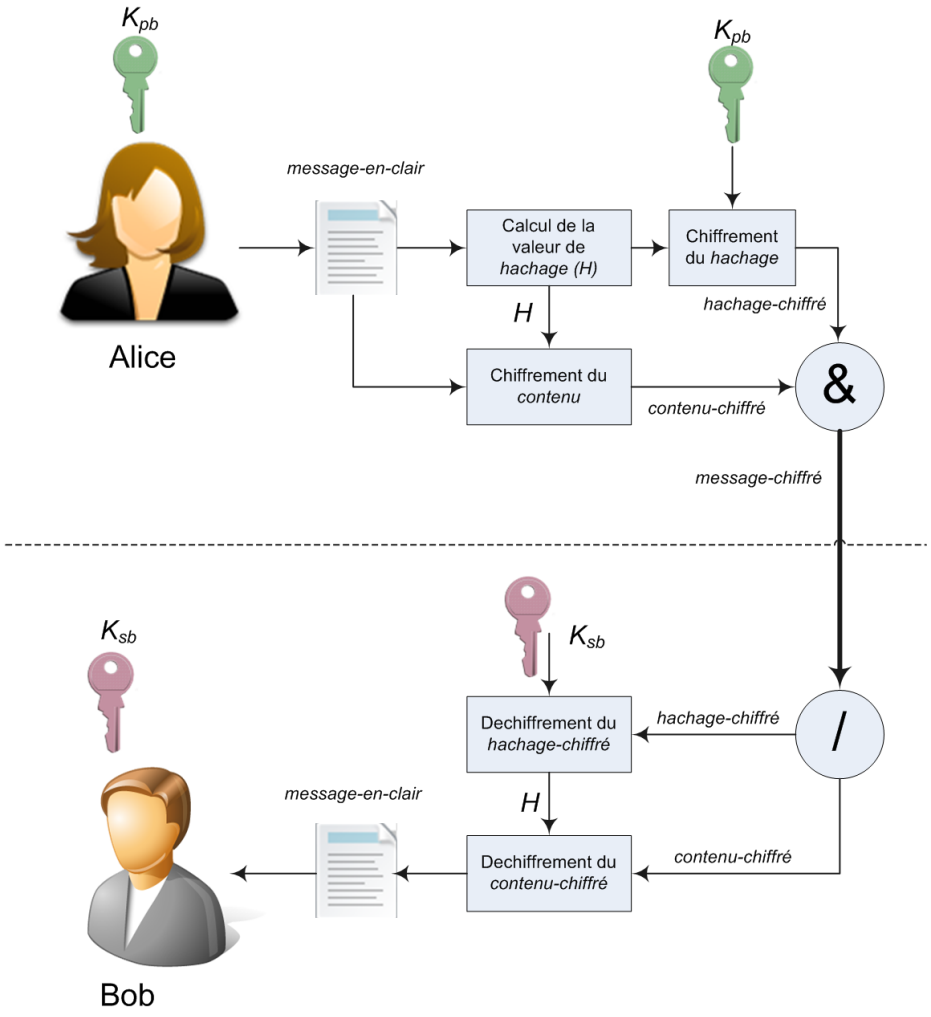

Figure 1. Principe du chiffrement convergent.

Configuration :
SHA : Fonction de calcul de valeur de hachage cryptographique.
E_s, D_s : Fonctions de chiffrement/déchiffrement à clef symétrique.
E_a, D_a : Fonctions de chiffrement/déchiffrement à clef asymétrique.
Clef secrète de Bob : K_{sb}.
Clef publique de Bob : K_{pb}.
Chiffrement :
Message-en-clair : $M \in Z_n$
$H = SHA(M)$
$\{M\}_H = E_s\{H\}(M)$
$\{H\}_{K_{pb}} = E_a\{K_{pb}\}(H)$
$C =< \{H\}_{K_{pb}}, \{M\}_H >$
Déchiffrement :
Message-chiffré : $C =< \{H\}_{K_{pb}}, \{M\}_H >$
$H = D_a\{K_{sb}\}(\{H\}_{K_{pb}})$
$M = D_s\{H\}(\{M\}_H)$

Figure 2. Description formelle du chiffrement convergent.

2.2 Empreinte d'image

La recherche de doublons a comme objectif l'optimisation du stockage chez le *FSS* et elle fait appel à des techniques de hachage cryptographique, e.g. SHA-256. Le résultat de la fonction de hachage change radicalement si un seul bit d'entrée change. En conséquence, les techniques de hachage cryptographique sont utiles pour identifier des copies strictement identiques (bit à bit) d'un fichier mais elles sont inutiles pour la recherche d'images visuellement similaires (par exemple de résolutions différentes), ou pour la recherche d'images strictement identiques mais encodées dans de formats différents (e.g., BMP et JPEG).

Pour résoudre le problème de la recherche de contenu similaire, on fait appel à des fonctions d'*empreinte d'image*, appelée aussi *descripteur d'image*. Ces descripteurs ont la particularité d'être tolérants à certaines distorsions des images. Il existe deux grands classes de descripteurs :

- L'*approche globale*, telle que l'histogramme des niveaux de gris, décrit le contenu de l'image dans son ensemble. L'algorithme d'extraction du *descripteur d'image* est rapide et le *descripteur* est tolérant aux distorsions communes de l'image mais souvent insuffisamment discriminant pour des comparaisons fines.
- L'*approche locale*, à base, par exemple, de *points d'intérêt*, décrit le contenu de l'image comme une collection d'empreintes de fragments de cette image. Ce type d'approche est généralement plus lourd mais conduit à un *descripteur d'image* à la fois riche et tolérant aux distorsions communes. Ces descriptions riches donnent également souvent lieu à une *version agrégée* de taille fixe dans laquelle la disposition spatiale des fragments est omise et leur apparence est décrite de façon grossière.

La similitude entre deux images A et B se détermine par un calcul de distance entre les descripteurs des deux images.

Le passage à l'échelle, c.a.d., la recherche de similitude entre une image A et l'ensemble des images d'une bibliothèque d'images est beaucoup plus complexe. Il nécessite la mise en place d'un système efficace pour résoudre le problème, dit du *plus-proche-voisin*, défini comme suit : *Soit une collection de « points de données » et un « point de requête » dans un espace métrique de dimension « n », trouver le « point de donnée » qui est le plus proche du « point de requête »* [9].

Souvent, ce sont les k plus proches voisins qui sont retournés par le système. La manière habituelle de mettre en application un tel système est la suivante. Un ensemble de descripteurs, dit collection de *points de données*, est calculé sur une bibliothèque de photos donnée. Ensuite, quand une requête de similitude est lancée, l'empreinte de l'image de requête est calculée afin d'obtenir le *point de requête*, et ensuite les *points de données* les plus proches du *point de requête* sont déterminés [9].

L'efficacité d'une recherche du *plus-proche-voisin* est évaluée en fonction des mesures dites de *rappel* et de *précision* :

- *rappel* : ratio entre le nombre d'éléments corrects parmi les éléments retournés et le nombre total d'éléments corrects dans la base de données.
- *précision* : ratio entre le nombre d'éléments corrects parmi les éléments retournés et le nombre total d'éléments retournés.

Ces mesures dépendent essentiellement de l'algorithme d'*empreinte d'image* et de l'algorithme de recherche du *plus-proche-voisin* (exact ou approché).

Les algorithmes d'*empreinte d'image* existants sont divers et variés. Par exemple, *BoF* (*Bag of Features*) [10] et *VLAD* (*Vector of Locally Aggreated Descriptors*) [11], sont tous deux basés sur une agrégation de descripteurs *SIFT* [12] associés à des points d'intérêt extraits de l'image et quantifiés. Comme algorithmes d'indexation/recherche de descripteurs nous avons également : *LSH* (*Locality-Sensitive Hashing*) [13] et *Hamming Embedding* [14].

Dans cette contribution, nous définissons l'*empreinte d'image* comme un vecteur de taille fixe Z_n appartenant à un espace métrique. Pour rappel, la norme de la différence de deux tels vecteurs fournit une mesure de distance. Une des normes de vecteur les plus populaires est la norme Euclidienne (norme L_2) mais d'autres normes de vecteur existent et pourraient être employées.

Il est important de mentionner l'influence de la dimension *« n »* de l'empreinte d'image sur l'efficacité d'indexation des bibliothèques de photos numériques à large échelle, et sur la *précision* et le *rappel* de la requête de la base de données. Les empreintes d'images à grande dimension fournissent habituellement une meilleure *précision* et un meilleur *rappel* que les empreintes d'images à petite dimension, mais elles sont en général plus difficiles à indexer pour une recherche, exacte ou approchée, efficace (en quantité de mémoire parcourue et en coût de calcul pour réaliser la recherche). La capacité de discrimination d'une empreinte de petite dimension est inférieure mais peut se révéler suffisante dans certains contextes [11].

3 Notre proposition

La Figure 3 présente l'architecture générale du système de stockage-en-ligne proposé dans cette contribution. Dans un souci de clarté, nous limitons les acteurs du système au minimum nécessaire pour présenter les principes de base de notre proposition : *FSS* représente le fournisseur du service de stockage, et Alice représente l'utilisateur qui souhaite partager de photos avec deux autres utilisateurs (nommés Bob et Charlie). De plus, nous ne faisons pas apparaitre les mécanismes de signatures permettant de s'assurer de l'authenticité des clés publiques des différents utilisateurs et des images transmises : nous nous limitons à la description de la solution assurant la confidentialité.

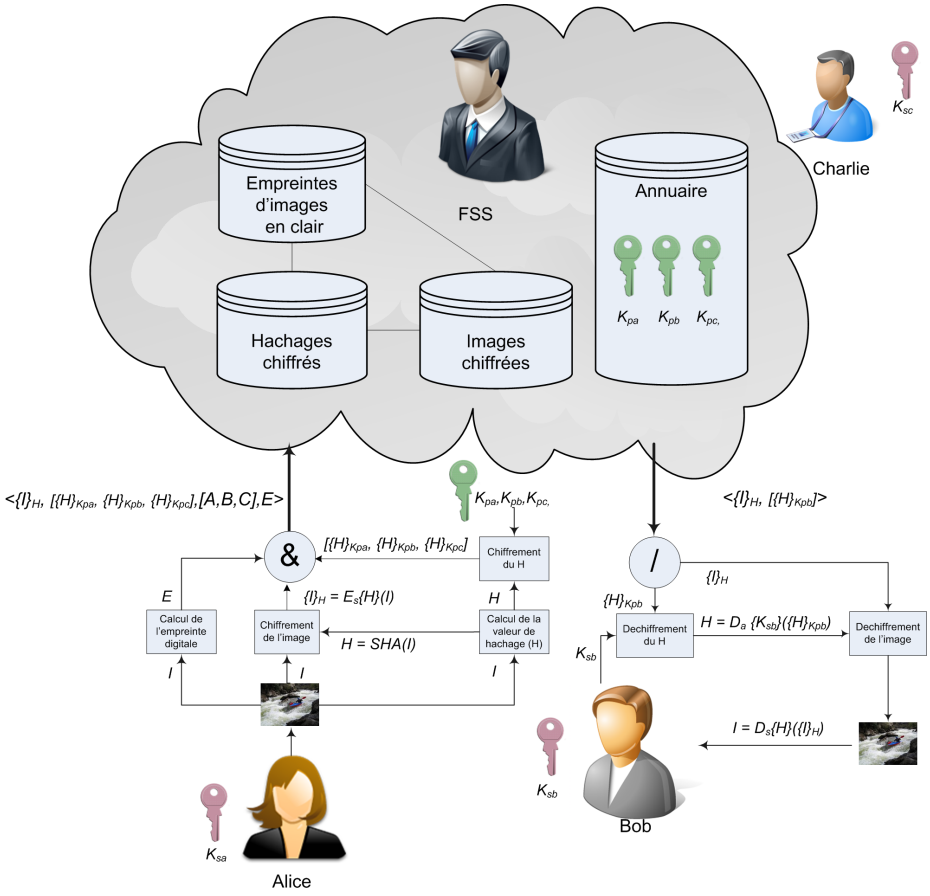

Figure 3. Système de stockage-en-ligne avec détection de doublons d'images et garantie de la confidentialité des données personnelles.

3.1 Mise en place du partage sécurisé de photos entre les utilisateurs

Dans la suite, nous supposons qu'Alice souhaite stocker et partager des photos avec Bob et Charlie, en utilisant le service de stockage fourni par le *FSS*. Au préalable, les utilisateurs génèrent des clefs de chiffrement asymétriques de la façon suivante :

- Chaque utilisateur u génère une paire de clefs $< K_{su}, K_{pu} >$ (clef secrète, clef publique).
- Chaque utilisateur garde secrète sa clef K_{su} de façon à être le seul autorisé à déchiffrer les images et à les lire.
- Chaque utilisateur publie sa clef publique K_{pu} vers les autres utilisateurs, par exemple via un annuaire ou un système classique de partage sécurisé de clefs publiques (e.g., *GPG* ou *PGP*) afin qu'ils puissent chiffrer les images.

3.2 Envoi d'une image chez le fournisseur du service de stockage

Dans cette section, nous décrivons le processus mis en œuvre quand Alice envoie une image au système de stockage-en-ligne.

- Alice calcule la valeur de hachage cryptographique H de l'image I.
- Alice chiffre l'image I, en utilisant la valeur de hachage cryptographique H de l'image comme clef, et obtient l'image chiffrée $\{I\}_H$.
- Pour chaque utilisateur u à qui elle souhaite donner l'accès, Alice chiffre la valeur de hachage cryptographique H de l'image en utilisant la clef publique K_{pu}, et obtient la valeur de hachage cryptographique chiffrée $\{H\}_{K_{pu}}$.
- Alice calcule l'empreinte E de l'image.
- Alice envoie le quadruplet $< \{I\}_H, [\{H\}_{K_{pa}}, \{H\}_{K_{pb}}, \{H\}_{K_{pc}}], [A, B, C], E >$ au *FSS*. L'avant dernier champ $[A, B, C]$ correspond à la liste des utilisateurs U (ici Alice, Bob et Charlie) pouvant accéder à cette image.

Si Alice souhaite utiliser le système du *FSS* pour stocker sans partager les photos, elle se contentera de les chiffrer avec sa propre clé publique K_{pa}, transmettant ainsi le triplet $< \{I\}_H, [\{H\}_{K_{pa}}], [A], E >$. De plus, la solution donnée pour le partage vers plusieurs utilisateurs (Alice, Bob et Charlie) est adaptée à de petits groupes d'utilisateurs. Dans le cas de grand groupes d'utilisateurs et de réseaux sociaux, il faut s'appuyer sur des mécanismes de cryptographie de groupe [15], [16].

3.3 Téléchargement d'une image vers l'utilisateur

Le processus opéré quand Bob veut rapatrier chez lui une image du système de stockage-en-ligne est le suivant. Alice et Charlie peuvent aussi récupérer l'image en utilisant le même processus.

- Bob reçoit le triplet $< \{I\}_H, [\{H\}_{K_{pb}}] >$ de la part du *FSS*.
- Bob déchiffre la valeur de hachage cryptographique chiffrée $\{H\}_{K_{pb}}$ de l'image, en utilisant sa clef secrète K_{sb}, et obtient la valeur de hachage cryptographique H de l'image.

– Bob déchiffre l'image chiffrée $\{I\}_H$, en utilisant la valeur de hachage cryptographique H de l'image comme clef, et obtient l'image en clair I.

3.4 Gestion de la base de données

Le quadruplet $< \{I\}_H, [\{H\}_{K_{pa}}, \{H\}_{K_{pb}}, \{H\}_{K_{pc}}], [A, B, C], E >$ envoyé par les utilisateurs au *FSS* est stocké dans une base de données. Il est important de souligner que l'espace de mémoire nécessaire pour stocker la partie $< [\{H\}_{K_{pa}}, \{H\}_{K_{pb}}, \{H\}_{K_{pc}}], [A, B, C], E >$ de l'information est négligeable par rapport à l'espace nécessaire pour stocker l'image chiffrée $\{I\}_H$.

Puisque l'image $\{I\}_H$ est chiffrée avec sa propre valeur de hachage cryptographique comme clef, les contenus chiffrés sont identiques, indépendamment des clefs publiques employées par les utilisateurs pour chiffrer la valeur de hachage cryptographique H de l'image et le *FSS*, sans connaissance des clefs (nécessaires pour déchiffrer les images), peut détecter que deux images sont identiques et n'en stocker qu'une seule. En d'autres termes, le *FSS* peut optimiser l'espace de stockage de son système en évitant le stockage redondant d'images identiques et ceci aussi bien au niveau du compte de chaque utilisateur qu'au niveau du système dans son ensemble.

Il est important de noter également que les données que le *FSS* manipule afin d'identifier les copies exactes présentes dans la base de données d'images sont celles des images chiffrées $\{I\}_H$ et que les clefs H nécessaires pour déchiffrer ces images sont chiffrées avec une fonction de chiffrement asymétrique dont le *FSS* ne connait pas les clefs car les utilisateurs (Alice, Bob et Charlie) les gardent secrètes. Par conséquent, la confidentialité des données personnelles est garantie.

3.5 Requêtes d'images similaires

Une caractéristique très importante d'une bibliothèque de photos est la possibilité, pour les utilisateurs autorisés, d'interroger la base de données d'images en fonction de son contenu. Par exemple, les utilisateurs devraient pouvoir rechercher dans la bibliothèque de photos des images presque identiques ou des images semblables à une image présentée au système comme exemple.

Comme expliqué dans la section 2.2, la manière habituelle de répondre à une telle exigence est d'associer une empreinte (un vecteur Z_n appartenant à un espace métrique) à chacune des images de la bibliothèque de photos. Afin de déterminer si deux images sont quasi-identiques ou semblables, l'utilisateur calcule la distance Euclidienne (norme L_2) entre les empreintes correspondant aux deux images et il compare ce résultat à un seuil donné. Puisque le *FSS* a accès aux empreintes en clair des images, le *FSS* peut, sur demande des utilisateurs, lancer de requêtes sur la base de données des images. Nous devons mentionner que nous supposons que le *FSS* ne peut obtenir aucune information, concernant l'image en clair, par sa connaissance de l'empreinte en clair de l'image. Ceci implique que l'empreinte E de l'image ne permet pas de reconstituer l'image ou même une approximation de celle-ci. Ainsi, on doit favoriser les empreintes de petite dimension pour limiter la fuite d'information vers le *FSS*.

4 Amélioration de la précision de requêtes

Comme mentionné dans la section 2.2, dans le cas d'une bibliothèque numérique de photos à large échelle, la précision dans les requêtes obtenue avec une empreinte de petite dimension pourrait ne pas être satisfaisante. Ceci signifie que le nombre de *faux-positifs* reçu par Bob pourrait être élevé. Nous proposons le mécanisme suivant afin d'améliorer la précision finale dans les requêtes du système de stockage-en-ligne.

Les techniques d'empreinte d'image à base d'agrégation de descripteurs locaux, telles que *BoF* (*Bag of Features*) [10] et *VLAD* (*Vector of Locally Aggregated Descriptors*) [11] produisent des vecteurs de taille fixe qui peuvent être ensuite quantifiés et/ou réduits en dimension pour un stockage et une recherche efficaces. Par conséquent, nous proposons que les utilisateurs, pour chaque photo, stockent chez le *FSS* l'information $< \{I\}_H, [\{H\}_{K_{pa}}, \{H\}_{K_{pb}}, \{H\}_{K_{pc}}], [A, B, C], \{S\}_H, V >$, où :

- $\{I\}_H$ est l'image chiffrée.
- $[\{H\}_{K_{pa}}, \{H\}_{K_{pb}}, \{H\}_{K_{pc}}]$ est la valeur de hachage chiffrée.
- $[A, B, C]$ est la liste des utilisateurs autorisés à accéder à l'image.
- $\{S\}_H$ est le vecteur de grande dimension, donnant la description géométrique et photométrique de chacun des points d'intérêt détectés dans l'image, chiffré avec la même clef H que celle employée par Alice pour chiffrer l'image.
- V est un vecteur de petite dimension, non chiffré.

Maintenant, supposons que Bob souhaite rechercher, dans la base de données d'images chiffrées, l'ensemble des images similaires à une image de référence I'. Le processus qui a lieu est le suivant :

- Bob calcule les deux empreintes V' et S' correspondant à l'image I'.
- Bob envoie au *FSS* l'empreinte de petite dimension V' de l'image de référence et son identité B.
- Le *FSS* utilise l'empreinte d'image de petite dimension V' de l'image de référence pour rechercher, dans la base correspondante, l'ensemble d'empreintes d'images les plus proches de V'. Cette recherche est restreinte aux images auxquelles Bob a accès en utilisant la liste d'utilisateurs autorisés associée à chaque image.
- Le *FSS* envoie à Bob les triplets $< [\{H\}_{K_{pb}}], \{S\}_H, V >$ (empreintes des images positives) correspondant à l'ensemble de signatures courtes proches de V trouvées à l'étape précédente. En raison de la faible précision de la recherche effectuée par le *FSS*, Bob reçoit un certain nombre d'empreintes d'images qui sont des *faux positifs*.
- Bob déchiffre les empreintes d'images de grande dimension $\{S\}_H$ des images reçues afin d'obtenir les empreintes d'images en clair S qu'il emploie pour éliminer les *faux positifs* de l'ensemble d'images qu'il a reçu en provenance du *FSS*. Bob élimine les *faux positifs* en recherchant dans l'ensemble d'empreintes d'images reçues les plus proches de S de l'image de référence. Pour identifier les *faux positifs*, Bob peut utiliser une méthode itérative de type *RANSAC* (RANdom SAmple Consensus) [17] permettant

de raffiner la comparaison sur la base de la disposition des points d'intérêt dans l'image (étape dite de post-vérification géométrique).
– Une fois les *faux positifs* éliminés, Bob demande au *FSS* de lui envoyer les images sélectionnées $\{I\}_H$.

Notez que le *FSS* a accès en clair seulement aux empreintes d'images de petite dimension. Par conséquent, il peut effectuer des requêtes rapides sur la bibliothèque de photos chiffrées mais il ne peut pas en général extraire d'information exploitable sur le contenu.

Notez également que la faible précision de la requête, effectuée par le *FSS*, est améliorée par Bob en employant les empreintes de grande dimension des images reçues qui sont stockées d'une manière chiffrée dans le service de stockage-en-ligne. Ces signatures, contrairement aux précédentes, peuvent dans certains cas donner accès à une visualisation assez parlante des images en clair [18], voire à une reconstruction précise pour certains types d'images [19].

5 Compromis entre confidentialité et fonctionnalité

Dans cette section, nous discutons du nécessaire compromis entre garanties de confidentialité et fonctionnalités offertes par le *FSS*. Nous situons les systèmes proposés dans les sections 3 et 4 par rapport à un système naïf n'utilisant pas de chiffrement et par rapport à un système utilisant du chiffrement non-convergent sur toutes les données afin de garantir une très forte confidentialité des données personnelles. Les systèmes suivants sont comparés :

Tout en clair : Une approche naïve au stockage d'image chez le *FSS* consiste à stocker toutes les données en clair, ce qui n'offre aucune garantie de confidentialité mais permet au *FSS* d'offrir une large gamme de fonctionnalités.

Empreinte en clair : Afin d'offrir un minium de confidentialité tout en conservant les fonctionnalités offertes par le *FSS*, dans notre première proposition décrite à la section 3, nous proposons de chiffrer l'image en nous appuyant sur du *chiffrement convergent*, tout en laissant l'empreinte en clair (e.g., Alice transmet $< \{I\}_H, [\{H\}_{K_{pu}}], [U], E >$ au *FSS*). Cependant, une telle approche présente trois possibles fuites d'information qui seront détaillées ci-dessous : reconstruction partielle de l'image, détection d'images similaires et détection d'images identiques.

Empreinte partiellement chiffrée : Afin de contrer une éventuelle reconstruction partielle de l'image, nous proposons dans la section 4 de séparer l'empreinte en deux composantes : une composante de petite dimension qui ne permet pas de reconstruire l'image et une composante chiffrée de grande dimension qui permet d'améliorer la précision des requêtes. Dans cette solution, Alice transmet $< \{I\}_H, [\{H\}_{K_{pu}}], [U], \{S\}_H, V >$. Cette approche présente toujours deux fuites possibles d'information, à savoir la détection d'images similaires (avec une précision moindre que dans le cas précédent) et la détection d'images identiques.

Empreinte chiffrée : Afin d'éviter une détection d'images similaires, il convient de ne pas transmettre l'empreinte (ou une partie de celle-ci) en clair (e.g., Alice transmet $< \{I\}_H, [\{H\}_{K_{pu}}], [U], \{S\}_H, \{V\}_H >$). Cependant, cette approche reste exposée à la fuite de détection d'images identiques.

Tout chiffré : Enfin, une garantie de confidentialité très élevée [1] peut être offerte en abandonnant le chiffrement convergent (e.g., Alice transmet au *FSS* l'information suivante $< \{I\}_R, [\{R\}_{K_{pu}}], [U], \{S\}_R, \{V\}_R >$, R étant une clé symétrique générée aléatoirement pour chaque image). Le système n'est plus vulnérable aux attaques précédemment mentionnées mais il ne sert plus que de coffre fort, aucun service de recherche d'images similaires ou identiques ne peut être offert.

Les fuites éventuelles d'information mentionnées précédemment sont décrites ci-dessous par importance décroissante :

Images en clair : En l'absence de chiffrement, le *FSS* peut accéder directement aux images en clair.

Reconstruction partielle de l'image : Lorsqu'une empreinte de grande dimension E est disponible en clair, le *FSS* peut effectuer une reconstruction partielle de l'image en s'appuyant sur les informations contenues dans l'empreinte [18].

Images similaires : Si le *FSS* a une photo de référence en clair, il peut déterminer que des utilisateurs ont des photos similaires à la photo de référence dans leurs collections hébergées chez lui. Ceci est possible du fait que le *FSS* possède la base de données des *empreintes d'images* en clair et qu'en conséquence il peut réaliser des recherches de *plus-proches-voisins* dans cette base de données.

Images identiques : De la même façon, si le *FSS* a une photo de référence en clair, il peut déterminer si les utilisateurs ont des copies exactes de cette photo de référence, stockées dans ses serveurs. Ceci est possible du fait de l'utilisation du *chiffrement convergent* pour chiffrer les images des utilisateurs. Ainsi, il pourrait « cerner les intérêts » de certains de ses utilisateurs à partir de quelques images. Cette fuite d'information rend nos propositions vulnérables aux attaques *à clair connu* de la part du *FSS*. En effet, si l'utilisateur n'est pas l'auteur des images mais ne fait que stocker-en-ligne des images récupérées de source publique, alors le *FSS* pourrait lui-même récupérer beaucoup d'images de source publique et appliquer un *chiffrement convergent* dessus pour se doter d'un dictionnaire d'images chiffrées.

Il est important de souligner que les trois types de fuites d'information (reconstruction partielle d'image, images similaires et images identiques) sont contrôlés et qu'ils résultent d'un compromis choisi par l'utilisateur en échange de services supplémentaires au simple service d'hébergement.

Le Tableau 1 résume les différents systèmes décrits avec les compromis entre le niveau de services que le *FSS* peut offrir et les garanties de confidentialité

1. Tant que la cryptographie utilisée n'est pas cassée.

offertes aux utilisateurs. Les services que le *FSS* peut proposer dépendent de la quantité d'information disponible en clair chez lui et de la forme du chiffrement.

Nos propositions (Empreinte en clair et Empreinte partiellement chiffrée) offrent de bons compromis entre le niveau de services et les garanties de confidentialité des données personnelles pour les utilisateurs.

Approche	Recherche d'images		Optimisation	Nb. des fuites	Fuite éventuelle
	Identiques	Similaires	stockage *FSS*	éventuelles	la plus importante
Tout en clair	Oui	Oui	Oui	4	Images en clair
Empreinte en clair	Oui	Oui	Oui	3	Reconstruction partielle
Empr. part. chiffrée	Oui	Oui	Oui	2	Images similaires
Empreinte chiffrée	Oui	Non	Oui	1	Images identiques
Tout chiffré	Non	Non	Non	0	Aucune

Table 1. Compromis entre confidentialité et fonctionnalité.

6 Conclusions

Nous avons présenté un système de stockage-en-ligne de photos efficace qui réconcilie les intérêts, jusqu'à présent contradictoires, du fournisseur de service de stockage-en-ligne et des utilisateurs du système de stockage. D'une part, le système garantit la confidentialité des données personnelles en stockant les images des utilisateurs chiffrées avec une clef à laquelle le fournisseur du service de stockage-en-ligne n'a pas accès. D'autre part, la méthode de chiffrement (*chiffrement convergent* [7]) employée pour chiffrer les images est telle que deux images identiques chiffrées avec deux clefs différentes produisent des contenus identiques ; en conséquence, le fournisseur du service de stockage a la capacité d'identifier que deux images sont exactement identiques et de les stocker dans l'espace correspondant à une seule image. Cette détection d'images identiques peut s'effectuer, aussi bien à l'intérieur d'un même compte utilisateur (de-duplication intra-compte) qu'au niveau des comptes d'utilisateurs différents (de-duplication inter-compte).

Sur demande d'un utilisateur autorisé, le fournisseur du service de stockage a aussi la capacité d'interroger l'intégralité de la base de données d'images chiffrées et d'identifier les images similaires à une image de référence fournie par l'utilisateur, et ceci toujours en garantissant la confidentialité des données personnelles. Ceci est possible du fait que le système de stockage-en-ligne stocke une empreinte d'image en clair de chacune des images. Cette empreinte d'image doit répondre à l'exigence que sa connaissance en clair ne permette pas la reconstitution, même partielle, de l'image [18].

Bien que nous nous soyons concentrés sur un système de stockage-en-ligne de **photos**, ce système de stockage pourrait également être employé pour le

stockage de **vidéos**. Dans ce cas, les empreintes d'images seraient remplacées par des empreintes vidéos, souvent de même nature mais exploitant la redondance temporelle du contenu visuel.

7 Remerciements

Nous souhaitons remercier les différents relecteurs pour leurs remarques qui nous ont permis d'améliorer la qualité de cette contribution.

Références

1. Carbonite : http ://www.carbonite.fr.
2. Crashplan : http ://www.crashplan.com.
3. Dropbox : http ://www.dropbox.com.
4. Mozy : http ://www.mozy.com.
5. Hu, W., Yang, T., Matthews, J. : The good, the bad and the ugly of consumer cloud Storage. ACM Operating Systems Review. **vol. 44** (2010) 110–115.
6. Storer M., Greenan K., Long D., Miller E. : Secure data deduplication. Proceedings of the 4th ACM International Workshop on Storage Security and Survivability. (2008).
7. Douceur J., Adya A., Bolosky W., Simon D., Theimer M. : Microsoft Research / Microsoft Corporation. **MSR-TR-2002-30** (2002).
8. Schneier, B. : Cryptographie appliquée : Algorithmes, protocoles et codes source en C. 2ème édition. Vuibert.
9. Beyer K., Goldstein J., Ramakrishnan R., Shaft U. : When is "nearest neighbor" meaningful ?. Springer-Verlag. **ICDT '99** (1999) 217–235.
10. Sivic J., Zisserman A. : Video Google : A text retrieval approach to object matching in videos. Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03). **vol. 2** (2003) 1470–1477.
11. Jegou H., Douze M., Schmid C., Pérez P. : Aggregating local descriptors into a compact image representation. Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition. (2010) 3304–3311.
12. Lowe D.G. : Object recognition from local scale-invariant features. Proceedings of the Seventh International Conference on Computer Vision (ICCV'1999). **vol. 2** (1999) 1150–1157.
13. Indyk P., Motwani R. : Approximate nearest neighbor - towards removing the curse of dimensionality. Proceeding of the thirtieth annual ACM Symposium on Theory of Computing. Dallas (May 1998) 604–613.
14. Sibiryakov A. : High-entropy Hamming Embedding of local image descriptors using random projections. Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP'09). (2009).
15. Steiner, M. : Secure Group Key Agreement. Ph.D. Thesis. Universität des Saarlandes (2001).

16. Banks L., Wu S. : All Friends are NOT Created Equal : An Interaction Intensity based Approach to Privacy in Online Social Networks. Proceedings of the 2009 International Conference on Computational Science and Engineering. **vol. 4**, 970-974, (CSE-2009).

17. Fischler M., Bolles R. : Random Sample Consensus : A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM. **vol. 24, no. 6** (June 1981) 381–395.

18. Weinzaepfel Ph., Jégou H., Pérez P. : Reconstructing an image from its local descriptors. Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. (CPVR-2011).

19. Ross A., Shah J., Jain A. : Towards Reconstructing Fingerprints From Minutiae Points. Proceedings of SPIE Conference on Biometric Technology for Human Identification II. **vol. 5779** (March 2005) 68–80.