8/11/2024 – Hippolyte PASCAL

PROJET DE SECURITE ET PROTECTION DE L'INFORMATION MEDICALE - BIO-5103A

EXERCICE 1 : STRATEGIE DE DEIDENTIFICATION EN 2 ÉTAPES

Etape 1 : Établissement de Santé - Service de Pseudonymisation

Dans la première étape, l'objectif est de supprimer toutes les données personnelles identifiables, tout en maintenant suffisamment d'informations pour les analyses internes au sein de l'hôpital. On souhaite donc générer un pseudo ou identifiant construit à partir des informations personnelles (notamment leur nom, prénom, sexe, date de naissance, adresse, etc.) qui identifie directement le patient. Cet identifiant ne doit jamais sortir de l'hôpital, même dans une version chiffrée. Les données sont traitées de manière qu'aucune personne externe ne puisse les associer directement à un individu spécifique.

J'ai tout d'abord appliqué un chiffrement AES en mode CBC pour le pseudo identifiant regroupant nom, prénom, sexe et date de naissance. Le sexe est lui-même généré sous forme de chiffre premier entre 2 et 100 pour complexité d'avantage le déchiffrement sans clé. (Si le chiffre premier ='H' sinon 'F'). J'ai choisi d'utiliser le mode CBC (Cipher Block Chaining) au lieu du mode ECB (Electronic Codebook) pour chiffrer les données très sensibles, car il offre une meilleure protection contre les failles de sécurité. Le mode CBC ajoute un vecteur d'initialisation (IV) aléatoire à chaque chiffrement, ce qui signifie que même des blocs de texte identiques produiront des résultats chiffrés différents.

Dans le mode ECB, chaque bloc de texte en clair est chiffré indépendamment.

Cela signifie que si deux blocs de texte en clair sont identiques, leurs blocs chiffrés le seront également. Je conserve tout de même l'utilisation du mode ECB qui permet de varier les modes. De plus il demande généralement moins de ressources, ce qui peut le rendre légèrement plus rapide et économe en mémoire dans des cas de chiffrement de donnée modérément sensibles. Je l'applique notamment sur le pseudo adresse du patient.

J'ai ensuite appliqué le hachage SHA-256 auquel est ajouté un sel unique généré aléatoirement pour transformer les pseudo (comme les pseudos identifiant du patient, adresse et numéros de sécurité sociale). Ce sel est une chaîne aléatoire qui rend le hachage de chaque valeur unique, même pour des données identiques.

On conserve finalement les informations nécessaires pour le suivi et les traitements, sans contenir de données directement identifiables. La reconnaissance par données croisées n'est pas omise encore…

Étape 2 : Pseudonymisation pour l'Équipe Statistique

La deuxième étape consiste à pseudonymiser les données du fichier intermédiaire afin de les rendre exploitables pour des études statistiques, sans compromettre la confidentialité des personnes concernées.

Ici aussi on vari entre le mode ECB et CBC de l'algorithme de chiffrement AES notamment pour les dates de consultation, auquel j'applique déjà un calcul en amont du nombre de jours en mois (décimal) qu'il y a entre la naissance du patient et ses consultations. Je combine les dates de traitement « AMMM » avec les code « CIS » renseignant sur leur traitement respectif pour ensuite les chiffrer en mode CBC. Je chiffre individuellement les codes « CIM-10 » indiquant les diagnostiques, et « Code CCAM » les types d'intervention.

Données non nécessaires et source de failles

Lors de la pseudonymisation des adresses, des informations d'identité des patients et des professionnels de santé, ainsi que des divers codes cliniques tels que les codes CIM, CCAM ou CIS, qui regroupent des informations cliniques condensé, il est essentiel de supprimer les données explicites comme les libellés et descriptions associées. Le contenu doit rester pseudonymisé au maximum, limitant les informations descriptives concernant la nature des traitements et des diagnostics pour préserver la confidentialité des patients.

Chiffrement des clés secrètes avec algorythme asymétrique

Une étape clé de mon projet, visant à désidentifier les acteurs concernés, consiste à sécuriser les clés AES générées en les chiffrant à l'aide de l'algorithme asymétrique RSA. Un fichier texte est créé par le code pour stocker les clés AES, chiffrées à l'aide de la clé publique du destinataire prévu. Celui-ci pourra ensuite les déchiffrer à l'aide de sa clé privée, garantissant ainsi la confidentialité des clés AES partagées entre l'expéditeur et le destinataire. Ce processus repose sur un échange préalable des clés publiques entre les parties.

Remarque & axes d'amélioration :

Dans ce projet, la question de l'intégrité et de l'authenticité des contenus n'est pas abordée de manière explicite. Bien que certains pseudonymes soient hachés (notamment les plus sensibles), les données elles-mêmes ne sont que chiffrées en AES.

Une amélioration possible serait de mettre en place un mécanisme de signature électronique. Cela pourrait consister à hacher l'ensemble des données (par exemple avec SHA-256), puis à chiffrer ce hash avec une clé privée RSA (2048 bits). Ainsi, le destinataire serait en mesure de recalculer le hash des données reçues en utilisant le même hachage, puis de vérifier leur intégrité (contenu n'a pas été changé ou falsifié) et l'authenticité de la signature à l'aide de la clé publique de l'expéditeur.

Pour renforcer la sécurité, il serait judicieux de chiffrer la clé privée RSA de chaque utilisateur, par exemple à l'aide de l'algorithme 3DES.

Une alternative intéressante serait de s'appuyer sur un service de PKI (Infrastructure à Clés Publiques) pour gérer les certificats et garantir la fiabilité des échanges.

EXERCICE 2 : DÉIDENTIFICATION DE DONNÉES TEXTUELLES

Approche directe

Mon code vise à anonymiser automatiquement les 5 rapports en chargeant aux préalables des listes de mots sensibles depuis des fichiers spécifiques : noms et prénoms, adresses, termes médicaux, et dates. Cela permet de préparer une base de données de mots à remplacer. (Avec l'aide de Chat GPT)

J'ai ensuite généré des listes de remplacements fictifs crédibles pour chaque type d'information (noms fictifs, adresses fictives, termes médicaux génériques, et dates fictives), pour garder un minimum de cohérence tout en cachant les informations réelles.

Le code détecte et remplace les mots sensibles dans le texte. Les informations sensibles sont identifiées à l'aide d'expressions régulières (ou "regex"), qui reconnaissent des modèles spécifiques pour chaque catégorie (e.g. formats de date avec L'expression \b(\d{2}/\d{2}/\d{4}|\d{4}-\d{2}-\d{2}|\d{2} \w+ \d{4})\b est utilisée pour repérer des dates sous différents formats, tels que "15/03/2010", "2012-07-21", ou "03 décembre 2015")

Cette fonction choisit un élément de remplacement aléatoire approprié en fonction de la catégorie du mot détecté. Dans la même logique, une liste de termes médicaux ou d'adresses peut être utilisée pour trouver des termes médicaux sensibles dans le texte et les remplacer par des termes fictifs.

Approche UNITEX vs Machine Learning (spaCy ou Flair)

Une méthode vue en TP serait d'utiliser UNITEX, qui a malheureusement été pour moi très fastidieux et sans résultat exploitable car je n'étais pas là lors du TP en question… J'ai passé beaucoup de temps à explorer d'autres alternatives, comme des outils de NLP comme SpaCy ou Flair. Ces outils, après un certain entraînement, sont capables de reconnaître des entités nommées (NER), de lemmatiser et de segmenter des textes. Cependant, ils doivent être entraînés sur de larges corpus de texte, ce qui leur permet de généraliser et de reconnaître des motifs linguistiques très vaste. Ce que j'aurais pu approfondir en récoltant plus de rapports médicaux de la même structure.

Je vois donc que UNITEX a son intérêt, car il fonctionne principalement sur des règles prédéfinies et des grammaires lexicales (dictionnaires et automates) utilisant des expressions régulières et des graphes précis. Il est donc beaucoup mieux adapté à de petits corpus de textes courts et structurés.