Documentation technique : GDM-001

Pour les programmeurs / programmeuses.

Module : codage\_mawa

* Class CodageMawa :

CodageMawa est un système de chiffrement symétrique (qui permet de chiffrer et de déchiffré du texte avec la même clé).

* La méthode master\_key\_generator :

CodageMawa.master\_key\_generator permet de générer une master key.

Fonctionnement de CodageMawa.master\_key\_generator:

Etape 1 : On génère des nombre pseudo aléatoires (mode longueur du tableau de caractère -1 (pour éviter d’avoir le signe ‘Σ’ dans la clé et avoir des erreurs lors de son déchiffrement par le master password)) cryptographiquement sécuriser avec random.SystemRandom().randint(). On enregistre les nombres dans un tableau.

Etape 2 : On remplace chaque nombre par la lettre correspondant dans le tableau.

Etape 3 : On transforme le tableau de lettre en chaine de caractères.

* La méthode encode :

CodageMawa.encode permet de chiffrer du texte.

Fonctionnement de CodageMawa.encode:

Pour faire le chiffrement il faut que la longueur du password soit égale à la longueur de la master key.

Etape 1 : Si la longueur du password est inférieur à la longueur de la master key on incrémente le password avec des lettres pseudo aléatoires préfixé de 'Σ' pour pouvoir les repérer lors du déchiffrement. 'Σ' marque la fin du mot de passe et le début de la chaine aléatoire. Si au contraire c’est la master key qui a une longueur inférieure à la longueur du password alors on incrémente la master key avec ses (longueur password – longueur master key) premières valeurs, (longueur password – longueur master key) fois pour avoir longueur password = longueur master key.

Etape 2 : On change les caractères de la master key et du password par leurs index dans le tableau de caractère (self.elements).

Etape 3 : On change chaque valeur du password par son image par \_\_algorithme (mode longueur du tableau de caractère).

Etape 4 : On additionne le résultat de l'étape 3 par le master key (mode longueur du tableau de caractère).

Etape 5 : On change chaque valeur par le caractère correspondant dans le tableau (self.elements)

* La méthode decode

CodageMawa.decode permet de déchiffrer du texte.

Fonctionnement CodageMawa.decode:

Etape 1 : Si la master key a une longueur inférieure à la longueur du password alors on incrémente la master key avec ses (longueur password – longueur master key) premières valeurs, (longueur password – longueur master key) fois pour avoir longueur password = longueur master key

Etape 2 : On change les caractères de la master key et du chiffré par leurs index dans le tableau de caractère (self.elements).

Etape 3 : On change chaque valeur du chiffré par son image par \_\_inv\_algorithme (mode longueur du tableau de caractère).

Etape 4 : On additionne le résultat de l'étape 2 par le master key (mode longueur du tableau de caractère).

Etape 5 : On change chaque valeur par le caractère correspondant dans le tableau (self.elements)

Etape 6 : On ne garde que les éléments situer avant le 'Σ'.

Module : gestion\_data

Gestion des données de la table gestionnaire.

* La fonction save\_password

Permet de sauvegarder un compte dans la base de données.

Fonctionnement de la fonction save\_password:

Etape 1 : On vérifier si le mot de passe ne contient pas de caractère inconnue (voir liste des caractères autoriser ci-dessous)

Si Etape vérifier est ok :

Etape 2 : On chiffre le mot de passe avec le master key

Etape 3 : On écrit l'id du nouveau mot de passe qui correspond à l'id de l'utilisateur dans la table user, on écrit la plateforme, le nom, le mot de passe chiffré et le mail.

Etape 4 : On exécute la requête.

Etape 5 : On commit pour sauvegarder.

* La fonction lecture

La fonction lecture permet de lire dans la base de données gestionnaire pour afficher toutes les données ou faire de la recherche

Fonctionnement de la fonction lecture :

Mode 1 : afficher tout (return\_all == True)

Etape 1 : On sélectionne tous les mots de passe ayant pour id l'id de l'utilisateur passé en paramètre

Etape 2 : On trie la liste de tuple retourner avec la 3éme valeur (nom)

Etape 3 : On transforme la liste de tuples retourner en une liste de dictionnaires avec la clé ['platforme', 'nom', 'password', 'mail'] chaque dictionnaire de la liste correspond à un compte avec la plateforme le nom, le mot de passe déchiffré et le mail.

Mode 2 : rechercher (return\_all == False)

Etape 1 : On sélectionne tous les mots de passe ayant pou id l'id de l'utilisateur passé en paramètre

Etape 2 : On recherche dans cette liste le compte qui a pour nom le nom passé en en paramètre et qui a pour plateforme la plateforme passée en paramètre.

* La fonction importation

La fonction importation permet d’importer des données à partir d’un fichier csv.

Fonctionnement de la fonction importation :

Etape 1 : On vérifie si fichier\_user existe.

Si Etape 1 vérifier est ok :

Etape 2 : On vérifie si le fichier contient les en-tête ['platforme', 'nom', 'password', 'mail']

Si Etape 2 vérifier est ok :

Etape 3 : On vérifie que les mots de passe ne contiennent pas de caractères inconnus.

Si Etape 3 vérifier est ok :

Etape 4 : On écrit l'id du nouveau mot de passe qui correspond à l'id de l'utilisateur dans la table user, on écrit la plateforme, le nom, le mot de passe chiffré, et le mail.

* La fonction supprimer

La fonction supprimer permet de supprimer des données dans la table gestionnaire.

Fonctionnement de la fonction supprimer :

Mode 1 : supprimer tout (delt\_all == True)

Etape 1 : On supprime tous les données ayant pour id l'id de l'utilisateur.

Mode 2 : supprimer un compte (delt\_all == False)

Etape 1 : On supprime le compte ayant pour id l'id en paramètre, le nom en paramètre, la platforme en paramètre.

* La fonction modifier

La fonction modifier permet de modifier des données de la table gestionnaire.

Fonctionnement de la fonction modifier :

Etape 1 : On chiffre le nouveau mot de passe avec le master key

Etape 2 : On modifie avec UPDATE si l'id ,le nom et la platforme correspondent.

Module : gestion\_user

Gestion de l'utilisateurs (enregistrement, authentification, modification du compte, suppression du compte)

* La fonction sing\_in

Création du compte de l'utilisateur nom, mot de passe.

Fonctionnement de la fonction gestion\_user:

Etape 1 : On vérifie s'il n'y a pas déjà dans la base de données user un utilisateur qui a le nom saisi par l'utilisateur.

Etape 2 : Si Etape 1 == non : on crée une master clé pour l'utilisateur.

Etape 3 : On récupère le hasher du mot de passe de l'utilisateur dans une variable.

Etape 4 : On vérifie que le mot de passe à la même taille que la master key.

Etape 5 : Si Etape 4 == non : on incrémente le mot de passe avec les (longueur master key - longueur mot de passe)

Première lettre de mot de passe (longueur master key - longueur mot de passe) fois pour avoir

Longueur master key - longueur mot de passe = 0

Soit :

Longueur master key = longueur mot de passe

Etape 6 : On enregistre le nom, le mot de passe hasher et la master key chiffré avec le mot de passe dans la base de données user.

Etape 7 : On récupère les in formation de l'utilisateur dans la base de données. On déchiffre la master key avec le mot de passe

Etape 8 : On retourne un dictionnaire avec l'id le nom et la master key de l'utilisateur.

* La fonction login

Connection de l'utilisateur nom, mot de passe.

Fonctionnement de la fonction login :

Etape 1 : On vérifie si le mot de passe de l'utilisateur et son nom sont correctes.

Etape 2 : Si Etape 1 == oui : On vérifie que le mot de passe à la même taille que la master key chiffré.

Etape 3 : Si Etape 2 == non : on incrémente le mot de passe avec les (longueur master key chiffré - longueur mot de passe)

Première lettre de mot de passe (longueur master key chiffré - longueur mot de passe) fois pour avoir

Longueur master key chiffré - longueur mot de passe = 0

Soit :

Longueur master key chiffré = longueur mot de passe

Etape 4 : On déchiffre la master key chiffré

Etape 5 : On retourne un dictionnaire avec l'id le nom et la master key de l'utilisateur.

* La fonction modifier

Modification du compte de l'utilisateur.

Fonctionnement de la fonction modifier :

Etape 1 : On vérifie si le mot de passe de l'utilisateur est correct.

Etape 2 : On vérifie que le nouveau nom n’est pas déjà utilisé par un autre utilisateur.

Etape 3 : On vérifie que le mot de passe à la même taille que la master key chiffré

Etape 4 : Si Etape 2 == non : on incrémente le mot de passe avec les (longueur master key chiffré - longueur mot de passe)

Première lettre de mot de passe (longueur master key chiffré - longueur mot de passe) fois pour avoir

Longueur master key chiffré - longueur mot de passe = 0

Soit :

Longueur master key chiffré = longueur mot de passe

Etape 5 : On déchiffre la master key chiffré.

Etape 6 : On vérifie que le nouveau mot de passe à la même taille que la master key.

Etape 7 : Si Etape 4 == non : on incrémente le nouveau mot de passe avec les (longueur master key - longueur nouveau mot de passe)

Première lettre de mot de passe (longueur master key - longueur nouveau mot de passe) fois pour avoir

Longueur master key - longueur nouveau mot de passe = 0

Soit :

Longueur master key = longueur nouveau mot de passe

Etape 8 : On chiffre le master key avec le nouveau mot de passe.

Etape 9 : On modifie le nom par new\_nom et master key par la nouvelle valeur chiffrée avec new\_password

et master\_password par le hasher de new\_password.

Caractère autoriser :

'a', 'b', 'c', 'd', 'e', 'f', 'g', 'o', 'p', 'q', 'r', 's', 't', 'u', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0', ' ', '!', '@', '#', '$', '%', '^', '&', '\*', '(', ')', '/', '-', '\_', '\', '{', '}', '|', '`', '°','[', ']', '+', '.', '?', ';', ':', '!', '§', '¤', '€', '£', '¨', '<', '>','~', ''', '\"', ',', '=',