TER : Découverte du langage Go

Clément CAUMES & Mehdi MTALSI-MERIMI

UFR des Sciences Versailles - M1 Informatique

- Réalisation d'une carte de référence
- Proposition d'un ensemble d'exercices d'apprentissage
- Réalisation d'une application exemple

Bases du langage

- Briques du langage (Variables, Constantes, Pointeurs...)
- Boucles
- Instructions de branchement conditionnel
- Instructions de branchement non conditionnel
- Collections et Opérations sur les collections
- Structure
- Appels de suite d'opérations (Fonctions, Méthodes et Interfaces)
- Ligne de commande et Arguments

Principales bibliothèques standards

- Bibliothèque 'fmt' (entrées-sorties de l'utilisateur)
- Bibliothèque 'errors' (gestion des exceptions)
- Bibliothèque 'os' (gestion des processus, manipulation de permissions de fichiers)
- Bibliothèque 'io' (interaction avec les fichiers)
- Bibliothèque 'strings' (manipulation de chaînes de caractères)
- Bibliothèque 'time' (mesure et gestion du temps)
- ...

Principales bibliothèques tierces

- Bibliothèque 'debug'
- Bibliothèque 'mobile'
- ...

Bases du langage

- Commandes (compilation, exécution, installation, tests unitaires, documentation)
- Compilation (création d'une application avec plusieurs packages)
- Tests unitaires (utilisation de tests pour vérifier des portions de code)
- Documentation (godoc)

Références pour produire notre carte de référence

- Introduction : `https://fr.wikipedia.org/wiki/Go_(langage)`
- Site officiel Golang : `https://golang.org/`
- Bases du langage Go : `https://www.tutorialspoint.com/go/index.htm`
- Illustrations de code Go : `https://gobyexample.com/`
- Goroutines : `https://blog.fedora-fr.org/metal3d/post/Go-et-les-goroutines-introduction-au-langage`
- Go vs C++ : `https://www.scriptol.fr/programmation/go.php`
- Application Android : `https://play.google.com/store/apps/details?id=in.intelitech.golang&hl=en_US`

TD1 : Installation & Manipulation des bases du langage

- Exercice 1 : Installation de l'environnement Golang et réalisation du premier programme Go
- Exercice 2 : Calcul de conversions d'un temps-secondes en temps exprimé en heures, minutes et secondes
- Exercice 3 : Réalisation de fonctions récursives (Fibonacci et factorielle)
- Exercice 4 : Test naïf de primalité
- Exercice 5 : Test de nombres amicaux
- Exercice 6 : Création d'une structure de données pour représenter une fraction et réalisation des fonctions de calculs de fractions

TD2 : Manipulation de structures complexes

- Exercice 1 : Implémentation de tris (à bulles et fusion) de tableaux
- Exercice 2 : Implémentation d'un annuaire électronique et première utilisation de méthodes
- Exercice 3 : Implémentation de listes et de méthodes manipulant ces dernières

TD3 : Manipulation avancée de bibliothèques Go

- Exercice 1 : Manipulation de la bibliothèque 'image' pour dessiner un damier, puis une fractale de Mandelbrot
- Exercice 2 : Manipulation de la bibliothèque 'io' pour implémenter une fonction qui copie le contenu d'un fichier dans un autre fichier et une fonction de lecture des métadonnées d'un fichier

TD4 : Manipulation des outils de développement

- Exercice 1 : Implémentation et manipulation de polynômes en utilisant les outils de développement (compilation, plusieurs packages, documentation godoc, tests unitaires, gestion des exceptions)

Application exemple : goshield

Il est primordial de prendre conscience de l'importance de chiffrer ses propres communications privées. En effet, avec l'émergence des réseaux informatiques, il y a de plus en plus de risques d'avoir ses communications personnelles surveillées. D'où l'importance d'utiliser des applications de chiffrement. Le but est de créer une application en ligne de commande pour le chiffrement et le déchiffrement de fichiers/dossiers. Son utilisation pourrait être, par exemple, de chiffrer le contenu de dépôt git afin de le rendre illisible pour le public.

Contraintes

- utilisation des outils de développement acquis lors de la lecture de la carte de référence (tests unitaires, compilation de packages, documentation godoc ...)

Exemple d'utilisation

L'application propose deux outils différents :

- chiffrer une liste de fichiers/dossiers avec la commande : **goshield –encrypt -p projet dossier1/sous-dossier2/ fichier1.txt image.png**
- déchiffrer une liste de fichiers/dossiers avec la commande : **./goshield –decrypt -p projet dossier1/sous-dossier2/ fichier1.txt.gsh image.png.gsh**

Définitions

L'algorithme de chiffrement choisi pour goshield est l'AES (Advanced Encryption Standard). C'est l'un des algorithmes symétriques les plus sécurisés puisqu'aucune attaque n'a été démontrée (mise à part l'attaque par force brute qui n'est pas réalisable avec la puissance de calcul actuelle).
GoShield proposera AES-256 (avec une clé sur 256 bits pour être le plus sécurisé possible).
De plus, l'utilisation du mode opératoire CBC est le plus efficace afin de chiffrer un fichier de taille variable.

Etapes du chiffrement goshield

Le chiffrement d'un fichier consiste en plusieurs étapes :

- écriture de la signature GOSHIELD.
- génération et écriture du sel cryptographique pseudo-aléatoire.
- génération et écriture du vecteur d'initialisation (IV).
- écriture de la taille du dernier bloc en octets.
- chiffrement et écriture de chaque bloc chiffré en utilisant AES-256 avec CBC. Le contenu de ce chiffrement sera écrit dans un fichier avec le nom du clair initial concaténé à l'extension goshield (.gsh).

Etapes du déchiffrement goshield

Le déchiffrement d'un fichier consiste en plusieurs étapes :

- vérification de la bonne extension .gsh.
- vérification de la signature GOSHIELD censée apparaître sur les 8 premiers octets.
- lecture du sel cryptographique et calcul de la clé en concaténant le sel avec le mot de passe choisi par l'utilisateur lors du déchiffrement.
- lecture de la valeur du vecteur d'initialisation (IV).
- lecture de la taille du dernier bloc en octets. Cela permettra d'enlever le padding sur le dernier bloc.
- déchiffrement et écriture de chaque bloc déchiffré en utilisant AES-256 avec CBC.

Pourquoi avoir choisi le langage Go pour cette application ?

Le langage Go présente un intêret pour ce type d'application :

- il s'agit d'un langage facile à comprendre et qui est aussi puissant que certains langages bas niveaux tel que le langage C.
- Go est également très intéressant pour la programmation "multithread" qui a été utilisée pour cette application. En effet, la gestion de concurrence peut être maintenue aisément grâce à sa facilité d'utilisation.
- Golang propose de nombreuses bibliothèques. Pour l'application goshield, la bibliothèque crypto nous a permis de travailler sur le chiffrement AES.