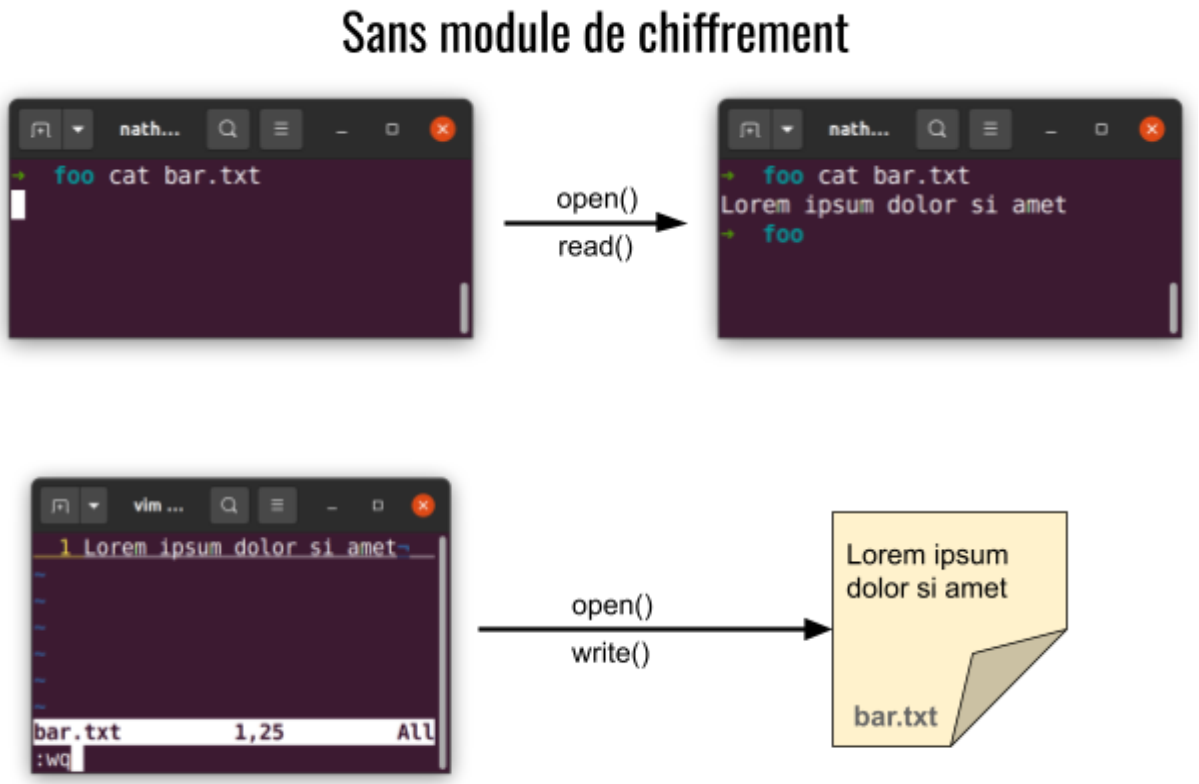
Chiffreur/déchiffreur

Module kernel Linux

Surcharge des appels systèmes de lecture/écriture de fichiers

L'idée est de remplacer les appels systèmes *open(), close(), write()* et *read()* par des fonctions customisées qui auront pour but de wrapper les appels systèmes correspondant avec une surcouche de chiffrement/déchiffrement.



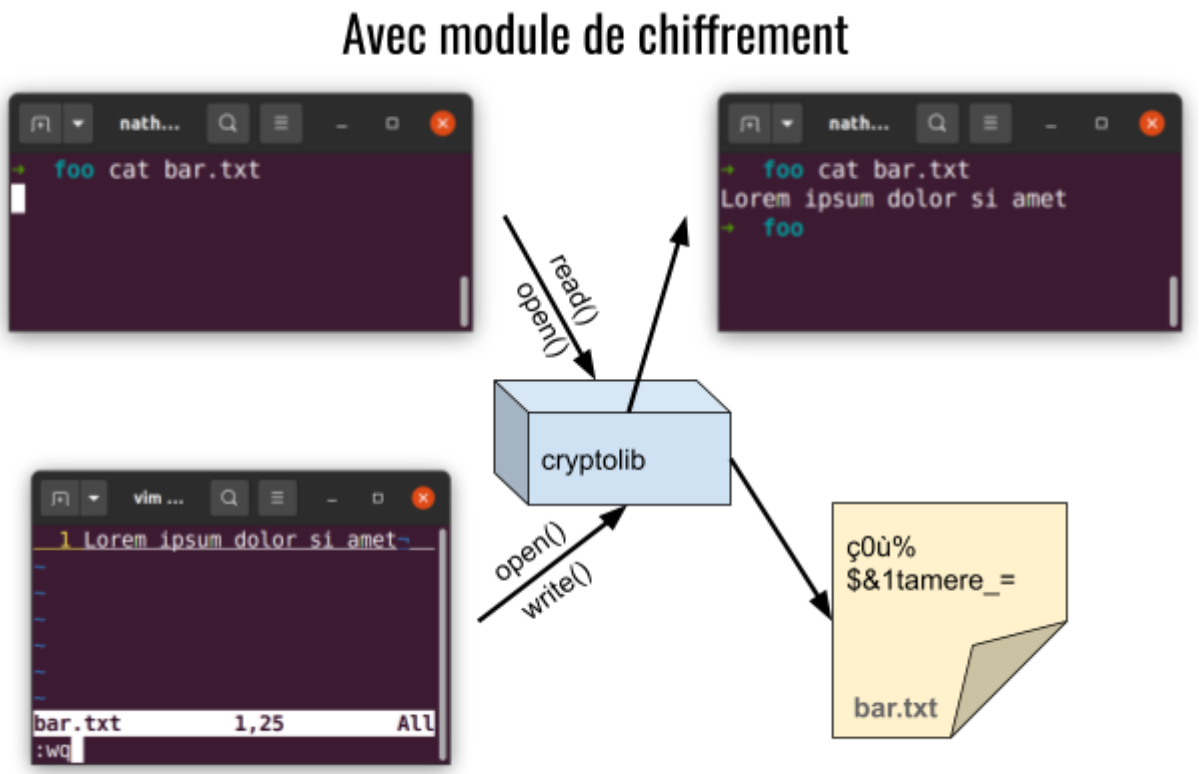


Table de correspondance

Cette table sera mise à jour par nos wrappers. Elle contiendra une liste de l'ensemble des fichiers qui ont été impactés par notre module. L'intérêt de cette table est que lors de la suppression du module, il nous suffira de la parcourir pour déchiffrer les fichiers et rendre le système accessible sans le module.

Zone sécurisée

Cette zone apparaît comme un dossier aux yeux de l'utilisateur. Mais ce dernier ne pourra l'ouvrir qu'en passant par un programme spécial qui lui demandera une clé de déchiffrement. Si la clé est correcte, le dossier sera déchiffré dans une zone tampon, et tous les changements apportés à cette zone devront être aussi apportés dans la zone chiffrée.

Il faudra s'assurer aussi que la zone déchiffrée soit bien supprimée après utilisation. Soit en la plaçant dans /tmp, ou alors en écrivant une fonction qui lors de la connexion ou de la déconnexion va supprimer toutes les zones déchiffrées.

Chiffrement de tout le système

Le but de cette fonctionnalité est de remplir la table de correspondance décrite ci-dessus avec des chemins considérés comme critiques (exemple : /root, /home, ...).

Gestion des clés de chiffrement

L'idée est d'avoir un mot de passe qui sera stocké en RAM (et vérifiable via un hash sur disque dur) qui permettra de chiffrer et déchiffrer l'ensemble des données redirigées par notre module. Ce mot de passe pourrait être le mot de passe de l'utilisateur en question, ou bien un autre mot de passe demandé via une autre entrée.