GOSTCRYPT

2017&2018

Suiveur : Éric Filiol

MEMBRES DU PROJET
Antoine HÉBERT - Quentin VARO - William LARDIER

esiea
ECOLE D'INGENIEURS
DU MONDE NUMERIQUE

ABSTRACT - ENGLISH

“In 2014, the well-known cross platform whole disk encryption software Truecrypt was discontinued. Truecrypt was used by many people, companies and governments across the world to protect their sensitive data, so its brutal disappearance left a void. That is why, soon after, the GostCrypt project, using safer GOST cipher algorithms was launched. Throughout the year, we contributed to this project, in order to make it more relevant for today’s needs. After having analyzed the project source code, we decided to rewrite GostCrypt from scratch using the current version of GostCrypt as an inspiration. In this report, we describe how we carried out these objectives to be able to release this whole new version by the end of the year. This new version with a different software architecture and an extensive documentation will be the base for the development of new features in the future.”

ABSTRACT - FRANÇAIS

“En 2014, le développement du logiciel de chiffrement de volume de données multi-plateforme Truecrypt a été arrêté. Truecrypt était utilisé par de nombreuses personnes, entreprises et gouvernements à travers le monde pour protéger leurs données sensibles. Sa brutale disparition a donc laissé un vide. C’est pourquoi peu après, le projet GostCrypt utilisant les algorithmes de chiffrement plus sûrs GOST a été lancé. Tout au long de cette année 2017, nous avons contribué à ce projet afin de le rendre plus adapté aux besoins d’aujourd’hui. Après avoir analysé le code source du projet, nous avons décidé de réécrire GostCrypt de zéro en nous inspirant de la version actuelle. Dans ce rapport, nous décrirons comment nous avons réalisé ces objectifs afin de publier cette toute nouvelle version en fin d’année. La nouvelle version de Gostcrypt que nous avons développée avec une architecture logicielle revue et une documentation complète pourra servir de base pour le développement de nouvelles fonctionnalités dans le futur.”

TABLE DES MATIERES - PST GOSTCRYPT

#1 - REMERCIEMENTS

Après déjà un an sur ce projet, nous tenons à exprimer nos sincères remerciements à Éric Filiol, suiveur de ce projet, pour le temps qu'il nous a consacré, l'opportunité qu'il nous a donnée de continuer notre projet l'été dernier grâce au stage et pour ses précieux conseils. Nous souhaitons aussi remercier Baptiste David et plus généralement les membres du laboratoire CVO qui nous ont été de bon conseil pour améliorer la sécurité de GostCrypt. Nos remerciements s'adressent aussi à notre école l'ESIEA qui nous a permis de consacrer plus de temps à notre projet grâce au statut d'étudiant espoir-recherche qu'elle nous a donné.

#2 - INTRODUCTION

- Le contexte du projet -

Initié fin 2013 par Éric Filiol, **GostCrypt** se veut une alternative au défunt projet TrueCrypt. Plus que jamais, la sécurisation des données personnelles est au centre des préoccupations numériques de chacun : Comment protéger de façon sûre des données sensibles ? Je ne fais pas confiance à l'hégémonie américaine, quelles solutions de chiffrement existent ?

Dans les années 2000, une solution s'est imposée : TrueCrypt. C'était un logiciel open source sans le statut de logiciel libre qui permettait le chiffrement à la volée de volumes de données. TrueCrypt permettait entre-autre de créer un volume virtuel chiffré de données, de chiffrer des clés USB ou encore de chiffrer l'intégralité d'une partition d'un système d'exploitation. L'objectif initial du logiciel était de combler le manque de solution pour le système Windows XP.

Alors que le support apporté par Microsoft de Windows XP prenait fin, les développeurs de TrueCrypt annoncèrent la fin du développement du logiciel, et par la même occasion dirent que leur projet était corrompu et qu'il fallait plutôt utiliser BitLocker, la solution propriétaire de Microsoft connue pour ses backdoors [1] . Bien évidemment, la communauté des chercheurs et des utilisateurs de TrueCrypt ne l'a pas cru, et différents projets visant à faire prospérer TrueCrypt ou à protéger les dernières versions sûres de l'ingérence Américaine virent le jour. Parmi celles-ci, on notera **VeraCrypt**, proposé par des Français et qui apporta diverses mises à jour de sécurité.

Figure 1 - Logo de VeraCrypt

GostCrypt est également un *fork*[2] de TrueCrypt. Mais il ne fait pas confiance aux algorithmes de chiffrements UK/USA : AES, Serpent et Twofish ne font plus partie du logiciel. À la place, il utilise des algorithmes de chiffrement alternatifs, notamment l'algorithme russe : GOST Grasshopper. GostCrypt permet finalement de faire la même chose que TrueCrypt.

- Les objectifs du projet -

Le projet que nous menons consiste en le développement de différents points. Avec M. Filiol, nous sommes

[1] Portes dérobées permettant la fuite d'informations vers Microsoft, et donc les États-Unis.

[2] Projet nouveau se basant sur un ancien projet aujourd'hui arrêté

partis du constat que GostCrypt héritait de la licence restrictive de TrueCrypt et qu'il propose un code source vulnérable (voir bibliographie : *Attaques et Audits de sécurité pour TrueCrypt, VeraCrypt et GostCrypt*). De plus, l'interface utilisateur de la version stable actuelle est vieille, non ergonomique et chargée d'éléments inutiles ou inutilisés ; le projet en lui-même possédant deux versions différentes suivant le système d'exploitation, malgré l'utilisation de technologies sensées être multiplateformes (*wxWidget*).

À l'occasion de ce Projet Scientifique et Technique (PST), nous avons pour objectif de redévelopper l'intégralité de GostCrypt, d'une part pour nous affranchir de la licence, mais aussi pour améliorer considérablement la structure du code et résoudre les problèmes de sécurité connus du logiciel. Un code source clair, documenté et bien structuré nous permettra de retrouver une communauté qui pourra s'investir dans le développement de GostCrypt.

Nous implémentons également une nouvelle interface graphique moderne et dynamique afin de réconcilier la simplicité d'utilisation avec le monde de la sécurité informatique. Un nouveau site web proposera GostCrypt au téléchargement, ainsi que diverses documentations plus ou moins techniques. En particulier, nous prévoyons de mettre à disposition une documentation innovante sous forme de graphe. Nous espérons par ce biais ressouder une communauté autour du projet.

Également, l'ajout de nouveaux algorithmes de chiffrement est prévu.

Figure 2 - Logo de GostCrypt 2

Pour ce projet, nous formons une équipe de quatre étudiants : Antoine Hébert, Louis Béclair, William Lardier et Quentin Varo. Le projet est suivi par M. Filiol. Nous avons au travers de ce PST développer nos compétences de développement logiciel, d'architecture logiciel, de développement d'interface, de développement sécurisé et plus généralement nous avons pu acquérir de l'expérience dans le développement d'un projet important (développement, réunions, tests, etc.).

- L'organisation du projet –

GostCrypt n'est pas seulement notre PST, c'est aussi notre projet espoir recherche. Dans le cas de Louis, il ne fait pas partie de l'équipe PST cette année, mais participe au projet via son statut d'étudiant chercheur.

Pour développer le projet GostCrypt, nous avons décidé que chaque membre travaillerait sur une tache différente tout en pouvant se rendre disponible pour apporter son soutien au bon déroulement d'une autre. Voici comment est organisé le travail :

Antoine :

Antoine se charge de remplacer l'ancien code hérité de TrueCrypt (Core, Volume, FuseService) car le code récupéré l'année dernière par le groupe a été jugé trop ancien et donc obsolète.

William :

William travaille sur la gestion de l'interface utilisateur (en langage QML/JavaScript/C++) pour que le logiciel soit intuitif et facile d'utilisation pour tout type d'utilisateur (professionnel ou bien privé). Il s'occupe aussi du développement du site web de GostCrypt.

: Quentin

Quentin est en charge de détecter des failles et des problèmes dans le code à l'aide d'outils de détection (CppCheck, Flawfinder, Valgrind, American Fuzzy Lop) puis de renforcer le code en l'optimisant.

: Louis

Louis (qui ne fait pas partie de notre groupe PST) travaille sur le sujet en tant que projet espoir recherche et s'occupe de la gestion de GostCrypt en ligne de commande. Il travaille aussi avec Antoine sur la gestion de la partie Volume.

GostCrypt est développé en utilisant le gestionnaire de version **Git**. Cela nous permet de synchroniser notre travail puisque l'on développe sur des ordinateurs séparés. Jumelé à Git, nous avons mis en place une plateforme en ligne afin d'organiser au mieux notre travail via des « Todo » (à faire), Issues, Graphiques ou encore Historisation, etc. Cette plateforme s'appelle **GitLab** Community Edition (GitLab CE). C'est un logiciel libre qui permet de gérer des dépôts Git, de mettre en place des examens de code, de renforcer la collaboration avec des demandes de fusion, un système de ticket et un wiki intégré à chaque projet.

Nous utilisons des issues taguées afin de gérer la liste des choses à faire, à corriger, à demander à notre suiveur lors de la prochaine réunion ou afin de répartir les tâches entre les membres du projet.

Tout ceci est hébergé sur un serveur distant appartenant à Antoine Hébert. Ces technologies mises en place nous permettent de travailler à distance, en autonomie et de mettre à jour notre logiciel rapidement tout en ayant une vision claire et précise de l'état d'avancement du projet.

Pour clore cette partie, le modèle mis en place (GIT & GitLab) convient parfaitement à notre façon de travailler, car nous nous sommes réparti les tâches. Cela nous donne une grande autonomie tout en restant dépendant du bon déroulement de chacun de nos travaux. De plus, lors d'un problème difficile à résoudre, nous pouvons tous intervenir dessus si besoin.

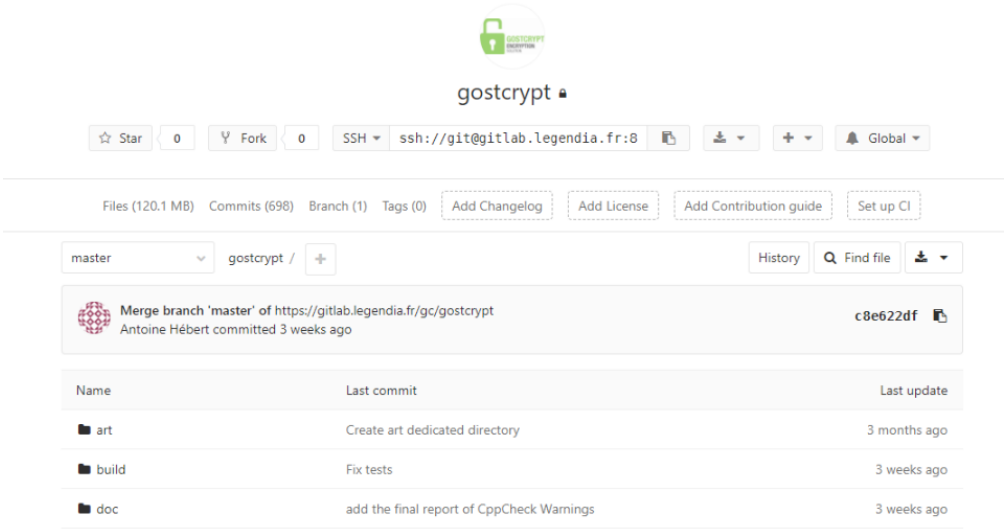


Figure 4 - La platforme GitLab où est hébergé de façon sécurisée le projet GostCrypt 2

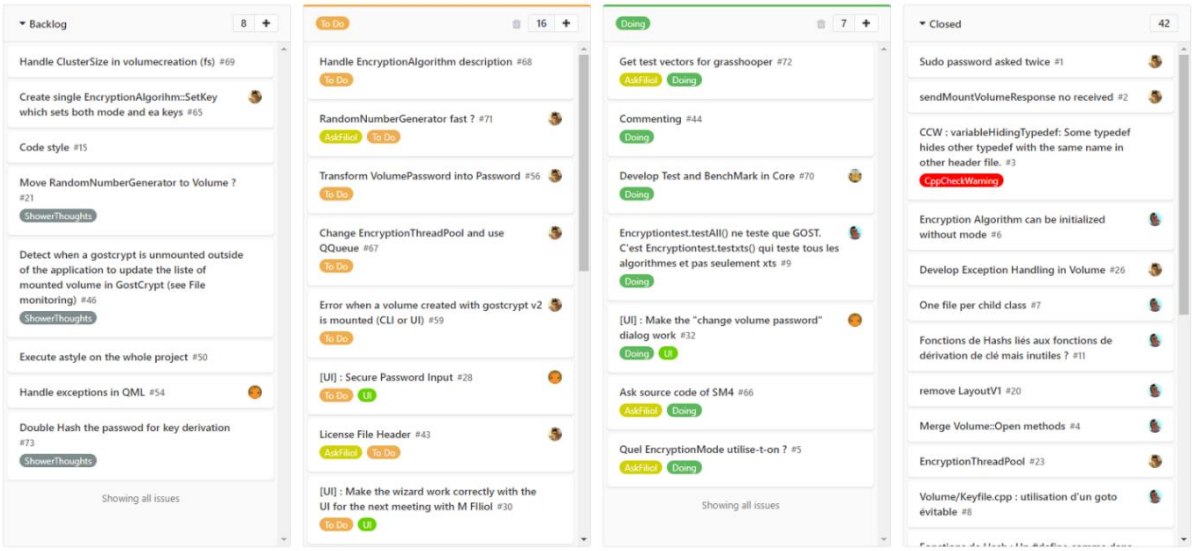


Figure 3 (2) - Gestion des taches à effectuer et problèmes à résoudre

#3 - LE TRAVAIL EFFECTUE

- La refonte de l'ancien code -

--- Contexte du projet au départ

Lorsque nous avons repris GostCrypt, l'interface graphique fût la première chose que nous avons voulu changer. En effet, l'interface était vieille, et très peu pratique à l'utilisation, surtout pour un utilisateur non initié à l'informatique. L'interface était difficile à cerner, nombre d'options, boutons ou parties du logiciel apparaissaient à l'écran sans que l'on ne s'en serve fréquemment. Ajouté à cela, la bibliothèque utilisée pour cette interface était wxWidget. Son implémentation ne respectait aucun modèle d'architecture logicielle : on trouvait dans un même fichier du code pour l'interface et du code pour effectuer des opérations. Enfin, bien que wxWidget soit régulièrement mis à jour, même aujourd'hui, la version utilisée par GostCrypt était une ancienne version, ce qui n'est pas bon pour un logiciel de sécurité.

GostCrypt, tel que nous l'avons récupéré, n'était pas du tout documenté. Il n'existait aucune documentation présentant l'architecture, les classes, les fonctions. Le logiciel a évolué au fur et à mesure des années de développement sans *refactoring* du code. Le refactoring, c'est le principe de retravailler le code d'un programme sans y ajouter de fonctionnalité ni de bogue. GostCrypt (et surtout TrueCrypt dont il hérite) a souffert de ce problème, et le code s'en est vu devenir bien plus complexe que nécessaire. Cela affecte directement la difficulté avec laquelle un développeur peut reprendre le projet (en rappelant qu'il n'y a pas de documentation), et diminue considérablement la facilité de maintenance du logiciel. Comme présenté dans l'audit de TrueCrypt, certaines vulnérabilités du logiciel existent et ne pouvaient pas être corrigées avec l'architecture qu'avait TrueCrypt, il fallait la faire évoluer, ou la réécrire.

L'ancien code de GostCrypt présentait un certain nombre d'anti-patrons (antipatterns) :

- Le problème dit « Yo-Yo », qui vient d'une complexité excessive des héritages de classes en C++, et qui donc ralentit énormément un développeur qui devra sans cesse jongler entre les définitions de classes afin de comprendre le fonctionnement du programme.
- Des dépendances circulaires entre les modules GostCrypt. De nombreuses fonctions d'un module A de GostCrypt nécessitait la présence d'un module B et inversement.
- Un des projets de GostCrypt, nommé « Platform », réimplémentait inutilement de nombreuses fonctions déjà proposées par le système d'exploitation.

De plus, dans cette ancienne version de GostCrypt, les clés de chiffrement utilisées pour gérer les volumes étaient stockées dans la mémoire paginée. Le problème avec cette mémoire paginée, c'est qu'elle est « *swappable* », c'est-à-dire que si un événement matériel survient, il est possible que cette mémoire se retrouve sur le disque dur à la demande du système, ce qui corrompt donc la sécurité et la confidentialité de la clé de chiffrement.

Une licence restrictive

Comme nous l'avons vu dans l'introduction, GostCrypt est un fork de TrueCrypt. Bien que le code source de TrueCrypt ait été distribué dans le passé, il reste sous sa licence unique à TrueCrypt. Celle-ci n'est pas libre, ni open source (d'après *l'Open Source Initiative*) et de nombreuses distributions GNU/Linux n'intègrent pas TrueCrypt dans les dépôts libres. Ces restrictions ont été pensées pour la sécurité, mais cela rend le logiciel non-libre. De ce fait, le développement collaboratif de TrueCrypt devient impossible.

Pour GostCrypt, nous voyons les choses autrement. Nous voulions utiliser une licence libre, pour les nombreux avantages qu'elle pourra offrir au projet.

Parmi ces derniers, il y a évidemment la possible création d'une communauté de développement autour du projet. C'est pourquoi, il était nécessaire de changer de licence.

Un projet non-multiplateforme

Les codes sources étaient à l'origine différents suivant le type de système d'exploitation (Windows ou Linux/Mac/Unix). Pourtant de nombreux copier-coller étaient présents entre ces deux versions. La version proposée pour Windows était encore moins bien organisée que pour Linux, notamment à cause d'un très gros fichier qui possédait l'intégralité des codes de l'interface graphique, ce qui le rendait difficilement lisible. Enfin, le fonctionnement sous-jacent était assez différent entre les deux versions : Windows se servait d'un driver pour gérer les volumes, tandis que ce n'était pas le cas de Linux qui utilise la bibliothèque Fuse.

--- Solutions envisagées

a. Réécrire tout le code

Comme nous venons de le voir, GostCrypt héritait de la licence de TrueCrypt non-libre. L'architecture logicielle rendaient l'évolution du projet très compliquée, et surtout une analyse complète de la sécurité du projet allait être impossible (ou tout du moins très compliquée). Le code effectuait beaucoup de sauts de fonctions (« *code spaghetti* »). Il y avait quand même de bonnes idées dans son principe de fonctionnement, en particulier pour la manipulation des archives chiffrées, avec un format permettant le déni plausible de sous-archives cachées.

Ainsi, nous avons décidé de réécrire totalement GostCrypt.

Cette solution est coûteuse en quantité de travail à effectuer, mais nous pensons que cela sera profitable pour l'avenir du projet sur bien des aspects. Dans un premier temps, réécrire totalement GostCrypt nous offre la possibilité de le documenter au fur et à mesure, et d'avoir une maîtrise totale de notre projet. De plus, cela nous permet de repasser sur une licence libre. Le code source peut aussi être simplifié, et cela n'est que profitable, puisque si davantage de développeurs sont en mesure de comprendre le code, la communauté de développement du projet ne fera que grandir.

b. La licence CeCill

Nous avons choisi la licence CeCill v2 pour GostCrypt 2. Nous avons dans un premier temps comparé les différentes licences open-source (voir l'annexe n°3). Après quoi nous avons fait le choix de CeCill pour ses différentes caractéristiques :

- **Copyleft** : Nous autorisons le projet à être copié, mais nous refusons que dans l'avenir GostCrypt puisse voir arriver une restriction du droit à la copie, à l'étude ou à de nouvelles évolutions non-libres.
- **Non Copyfree** : Si le projet est repris par d'autres personnes, ils ne pourront pas se l'approprier.
- **No Sublicensing** : Nous n'acceptons pas que le projet possède d'autres licences non-compatible.
- **No secret modification** : Si des modifications sont apportées au projet, l'équipe de développement principale doit être tenue au courant.
- **GPL compatibility** : Cette licence est compatible avec la licence GPL.
- **French Law** : Cette licence est régie par la loi Française.

c. L'utilisation de Qt

Nous avons voulu profiter de cette réécriture pour utiliser le *framework* Qt. Ce dernier nous permet d'abord de concevoir une interface moderne et interactive avec la technologie QML. L'utilisation d'un framework puissant et éprouvé comme Qt permet de se décharger de la couche d'abstraction nécessaire pour rester multi-plateforme et des nombreuses fonctions du module Platform de TrueCrypt. Si bien qu'il deviendra à l'avenir bien plus simple

pour des étudiants de reprendre le projet.

--- Le travail effectué

a. Architecture globale du logiciel

Pour commencer, nous avons dû imaginer et concevoir la nouvelle architecture du logiciel. Pour cela, nous nous sommes inspirés de l'ancienne, qui fonctionnait sur le principe de 5 modules :

- Le module UI pour l'interface utilisateur,
- Le module Core, ou cœur de GostCrypt, appelé par l'interface lors de chaque opération (montage/démontage de volume, etc.),
- Volume, qui est un module interagissant directement avec le conteneur chiffré qui prend en charge le chiffrement / déchiffrement,
- Le module Fuse, qui permet la gestion du système de fichier virtuel de GostCrypt permettant de chiffrer/déchiffrer à la volée.

Dans la version précédente de GostCrypt, un cinquième module existait : Platform. Le but de ce module était de réimplémenter des fonctionnalités du système ou de les encapsuler afin de rendre GostCrypt multi-plateforme. Ce module contenait notamment des fonctions de gestion des chaînes de caractères, de gestion des threads et mutex, etc. Nous avons choisi de retirer ce module, car il a pu être remplacé par les fonctionnalités du framework multi-plateforme Qt.

Ces modules seront présentés dans leur nouvelle version dans les autres parties de ce chapitre. Quant au module UI, il sera traité dans le chapitre suivant.

Durant le développement de GostCrypt, nous nous sommes concentrés sur un module à la fois sans toucher aux autres. Cela nous obligeait parfois à revenir dans un module déjà redéveloppé pour modifier la façon dont il utilisait les modules nouvellement développés mais nous pensons qu'il s'agissait d'un mal nécessaire. En effet, il nous semblait impossible de faire évoluer tout le code en même temps. Une autre solution aurait été de tout développer sans tester, tant que tous les modules n'avaient pas été redéveloppés ; mais cela n'était pas raisonnable.

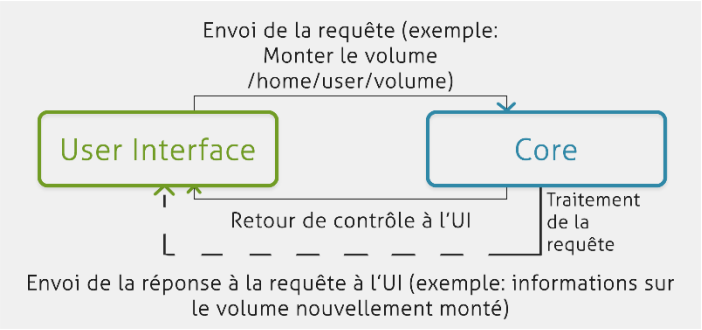
b. Le côté asynchrone

Nous avons fait le choix de tirer parti de la possibilité offerte par Qt de faire de la programmation évènementielle. C'est à dire que certaines communications entre les modules sont asynchrones. Le module interface utilisateur, pour la partie interface graphique, fonctionne de manière asynchrone de par sa nature même.

Par exemple, la communication entre le module UI et le module Core est asynchrone, cela se présente comme présenté sur le graphique ci-dessous.

Nous avons choisi de rendre cette communication asynchrone car le fonctionnement interne de Core est déjà asynchrone dans le cas où l'utilisateur ne lance pas directement GostCrypt avec l'utilisateur root. En effet le module Core va gérer le lancement du processus avec

les privilèges root et la communication avec celui-ci de manière asynchrone.



Communication asynchrone

c. Le module Core

Nous avons choisi de totalement redévelopper le module Core d'une façon différente de l'ancienne version. Ce dernier est chargé d'implémenter les différentes fonctionnalités de base de GostCrypt (monter un volume, démonter un volume, créer un volume, changer le mot de passe d'un volume, etc.) en utilisant des méthodes plus proches de la machine (bas niveau) de Volume. Le module Fuse est utilisé pour lancer le processus en tâche de fond s'occupant du système de fichier virtuel.

La principale difficulté dans ce module est la gestion des fonctions qui nécessite les droits administrateurs. Nous devons exécuter ces fonctions dans un autre processus appartenant à l'utilisateur root. Ce module doit donc gérer le lancement, l'arrêt et la communication avec ce processus.

Dans l'ancien module, cette gestion semblait ne pas avoir été prévue initialement, mais avoir été ajoutée après coup. En effet, un objet nommé « CoreProxy » prenait la place de l'objet Core afin de rediriger les appels de méthodes vers un autre processus, si nécessaire. Cependant, un accès direct à l'objet Core était quand même possible sans que cela ne fût nécessaire. Les fonctions de l'interface utilisateur utilisaient alors l'une ou l'autre des deux méthodes possibles. Nous avons remarqué que certaines fonctionnalités de Core qui avaient besoin d'interactions avec l'utilisateur n'étaient pas accessibles au travers de l'objet CoreProxy.

Après réflexion, nous avons donc décidé de garder ce principe, mais en corrigeant les problèmes de conception de l'ancien module. Dans le Core actuel, cela fonctionne différemment : l'utilisateur de Core, c'est-à-dire le module UI, va utiliser un pointeur sur un objet CoreBase, qui est une classe abstraite. Cette classe possède une méthode « request » appelée à chaque fois que l'utilisateur effectue une action de base de GostCrypt. En paramètre, cette méthode prend un objet de type CoreRequest décrivant l'action à effectuer et les différentes options nécessaires. Lorsque la requête a été traitée, une méthode spécifique à l'interface utilisateur va être appelée avec pour paramètre le résultat de la requête.

Si GostCrypt a été lancé directement avec les droits *root*, le pointeur sur « CoreBase » utilisé par l'UI pointera vers un objet de la classe fille « CoreRoot » qui implémente toutes les fonctionnalités directement.

Dans ce cas, nous n'avons pas besoin de lancer un autre processus. Sinon, si GostCrypt ne possède pas les droits *root*, ce pointeur pointera vers un objet de l'autre classe fille « CoreUser ».

Cette classe va faire automatiquement appel aux méthodes de « CoreBase » lorsque les droits *root* ne sont pas nécessaires, et elle va transmettre la requête à l'objet « CoreServiceHandler » si les droits root sont nécessaires .

Ce dernier objet, « CoreServiceHandler », est en charge de la gestion du service Core, c'est-à-dire du processus qui va exécuter les requêtes avec les droits *root*. Il gère le lancement, l'arrêt et la communication avec ce processus. Le processus lancé avec l'utilisateur Root correspond à l'objet « CoreService ». Il va directement utiliser l'objet « CoreRoot » pour exécuter les requêtes. Lorsque la requête est traitée, il va retransmettre au processus principal la réponse à la requête. En cas d'erreur, l'objet CoreService intercepte l'exception et la retransmet au processus principal, qui va alors la gérer de son côté comme si l'événement s'était passé dans le processus principal.

Afin de faire communiquer nos deux processus, nous avons choisi de garder le même principe de fonctionnement que l'ancienne version du Core. Nous utilisons l'entrée et la sortie standard du processus *root* sur lesquels nous sérialisons les objets de type « CoreRequest » et « CoreResponse ». Après réflexion, nous sommes venus à la conclusion que c'était la méthode de communication inter-processus multi-plateforme la plus sécurisée. Avec cette méthode, le contenu de ce qui est transmis n'est pas stocké dans une mémoire paginée, donc la mémoire ne pourra pas se retrouver sur le disque dur. La mémoire utilisée se situe dans un espace mémoire dédié dans le noyau.

Ce module fût le second que nous avons redéveloppé, après le module UI. Il a représenté une part importante du travail effectué.

d. Le module FuseService

Le module Fuse permet de gérer le processus qui tourne en arrière-plan en charge du chiffrement et du déchiffrement à la volée d'un volume GostCrypt. Pour cela, un système de fichier spécifique est développé dans GostCrypt grâce à la librairie FUSE.

Ce module fonctionne de façon semblable au module Core que nous venons de voir. Lorsque le module Core veut monter un volume, il fait appel à la méthode « mount » de « FuseServiceHandler ». Cet objet va lancer le processus Fuse développé dans la classe « FuseService ». C'est dans le processus Fuse que les premières étapes du montage du volume vont être exécutées : ouverture du fichier ou de la partition, déchiffrement du header, montage du système de fichier GostCrypt, etc. Une fois que ces premières étapes ont été effectuées, le processus fuse va se dupliquer (« fork »). Le processus fils issue de cette duplication va se mettre en tâche de fond pour gérer le système de fichier GostCrypt et chiffrer/déchiffrer les données au fur à mesure qu'elles sont écrites/lues. Le processus parent issu de cette duplication va envoyer les informations nécessaires au processus Core et se terminer. Le processus Core va ensuite continuer le montage du volume en utilisant le processus fuse à travers son système de fichier pour accéder au volume déchiffré.

La version précédente du module Fuse n'utilisait pas un service spécifique et se dupliquait directement depuis le processus Core. Nous avons préféré utiliser un service dédié à Fuse pour ne pas continuer à utiliser toute la mémoire nécessaire pour le fonctionnement du processus Core après l'arrêt de GostCrypt (il est inutile de garder l'interface graphique en mémoire, par exemple). Nous pensons que cela contribue aussi à rendre le code du projet plus clair.

e. Le module Volume

Le module volume est le dernier module sur lequel nous avons travaillé. C'est le module le plus bas niveau dans GostCrypt. Il permet d'interagir avec les fichiers et/ou partitions chiffrés. Il contient le code source nécessaire pour lire le format de fichier de GostCrypt.

Lorsque nous avons analysé le code source de la version précédente de GostCrypt, nous avons trouvé que ce module était plutôt bien conçu et qu'il n'y avait pas de grands changements à apporter. Nous avons donc décidé de ne pas réécrire depuis zéro ce module, mais plutôt de le faire évoluer à partir de sa version existante. La version précédente de ce module utilisait beaucoup le module « Platform » que nous voulions supprimer. Nous avons donc dû adapter tous les endroits où ce module était appelé et nous avons fait appel au framework Qt à la place.

Nous avons aussi supprimé certaines fonctionnalités qui n'étaient plus utilisées dans GostCrypt, notamment la prise en charge de certains formats et modes de chiffrement historique. En effet, certains formats n'étaient déjà plus utilisés depuis longtemps dans TrueCrypt, et aucune archive chiffrée avec ce format a pu être créée avec GostCrypt. La rétrocompatibilité devenait donc inutile. La nouvelle version de GostCrypt ne prend donc en charge que le dernier format d'archive chiffré. Nous réfléchissons à développer un outil permettant de convertir les archives d'anciens formats au nouveau afin de ne pas bloquer d'éventuels utilisateurs qui les utiliseraient toujours.

Suite à de nombreuses petites modifications progressives, le module Volume a maintenant beaucoup évolué par rapport à celui que nous avions récupéré.

f. La documentation du code avec Doxygen

Pour documenter le code source, nous avons choisi d'utiliser Doxygen. Ce logiciel permet de générer automatiquement une documentation logicielle sous la forme d'un ensemble de pages HTML ou d'un fichier PDF en lisant le code source contenant des commentaires formatés d'une certaine façon. Pour obtenir une documentation plus complète, nous devons ajouter la description de chaque classe et de chaque méthode, afin d'aider un potentiel lecteur ou repreneur du projet à comprendre plus rapidement le code du projet ainsi que son architecture.

Nous n'avons pas documenté chaque fonction au fur et à mesure que nous développions, comme nous aurions dû le faire. Cela nous a cependant permis de développer plus vite et de gagner du

temps lorsque nous devions modifier à plusieurs reprises l'organisation du code source. Bien sûr, une fois le code finalisé, nous l'avons documenté.

Les modules UI, Core et Fuse sont maintenant complétement documentés, le module Volume par contre reste encore seulement partiellement documenté. Au cours du second semestre nous avons passé beaucoup de temps sur les modules UI, Core et Fuse pour les tester, définitivement fixer le code source et les documenter de manière exhaustive. Le travail restant se concentrera donc maintenant sur le module Volume.

- Interface et Interactions Utilisateur -

--- Contexte du projet au départ

Dans sa version 7.1a, TrueCrypt utilisait la bibliothèque graphique *wxWidget*. C'est une bibliothèque libre et s'adaptant bien suivant les systèmes d'exploitation, mais qui souffre d'un grand manque de dynamisme et de modernité. Alors que de plus en plus de logiciels optent pour des interfaces modernes voire du *web-like*, il était important que l'interface utilisateur soit repensée. En plus de l'apparence visuelle, l'interface de TrueCrypt (qui rappelons-le est celle de l'actuel GostCrypt) manque cruellement de simplicité d'utilisation. Bien des options ne sont pas expliquées ou sont inutiles dans le cadre d'une utilisation concrète du logiciel.

Le code source de l'ancien GostCrypt implémentait donc son interface avec cette librairie, mais le schéma Modèle-Vue-Contrôleur (MVC) n'était pas respecté. Si bien qu'il n'était pas rare de lire certains fichiers possédant à la fois du code pour développer l'interface, et dans la même fonction un code qui faisait le traitement logique des entrées utilisateur. Dissocier l'interface du noyau de GostCrypt devenait pressant, afin de considérablement en simplifier son développement comme son évolutivité.

Si l'on met de côté la partie technique de l'ancienne interface, elle gardait un problème majeur : elle n'était pas intuitive. Si nous souhaitons voir GostCrypt être utilisé par un maximum d'utilisateurs, il faut prendre en compte que la majorité n'a pas forcément de connaissances en informatique, encore moins en sécurité. Nombreux sont les utilisateurs qui n'ont que des données à protéger et ne souhaitent pas « perdre du temps » à comprendre comment fonctionne GostCrypt. C'est d'ailleurs le même constat que pour les applications sur smartphone. Malheureusement, l'ancienne interface présentait de nombreuses options aux noms obscurs, et plus simplement la fenêtre principale était trop chargée.

Pour ces raisons, M. Filiol a fixé comme objectif de refaire cette interface. Nous étions donc libres de décider de la technologie, de l'apparence et des modifications.

Ce travail est assez indépendant dans le développement global de GostCrypt. Développer une interface, ce n'est pas seulement du code, ce sont aussi de nombreuses maquettes, essais, réflexions, refontes. Il y a un aspect artistique. Il faut également penser à créer quelque chose de portable, de *beau*, et sans oublier les personnes ayant une vision altérée (daltonisme).

NB : Ce qui sera visuellement présenté dans cette section pourra encore changer d'ici à la sortie de la version stable de GostCrypt2.

--- Solutions envisagées

Avant de rentrer dans le détail de la nouvelle interface, attardons-nous sur le choix de la technologie que nous avons pris. Comme dit ci-avant, M. Filiol nous a demandé une refonte de l'interface de l'ancien GostCrypt.

Dans un premier temps, nous avions récupéré un projet d'interface graphique réalisé par un ancien Espoir Recherche, utilisant la librairie Qt (en C++). A cette époque, nous n'avions pas encore décidé de recoder l'intégralité de GostCrypt, mais nous savions que nous utiliserions Qt pour développer le logiciel.

L'avantage de cette technologie C++ est qu'elle est multiplateforme : Qt offre une couche d'abstraction et permet le développement d'interfaces complètes en utilisant les différentes classes offertes. Malheureusement, le développement restait en C++ et donc les possibilités pour développer quelque chose de graphiquement dynamique restaient limités.

De plus, était-il vraiment nécessaire de coder l'interface si bas niveau ? Alors que nous utilisions l'architecture MVC, devions-nous vraiment nous forcer à gérer la mémoire à la main ? Cette architecture permet aussi de ne pas avoir à se soucier de la sécurité dans le code de l'interface (hormis un détail que nous verrons dans la partie suivante).

Il existe de nombreuses technologies plus haut niveau compatibles avec le langage C++ qui permettent de développer des interfaces avec bien plus de fonctionnalités. Notre travail initial fût de décider laquelle utiliser.

Les critères de choix étaient dans l'ordre : *user-friendly*, compatible Qt, richesse des rendus possibles, dynamisme des éléments visuels, rendu moderne, multiplateforme, pas de risque de sécurité. Sans énumérer toutes les possibilités, nous avons opté pour le langage Qt Meta Language (QML), déjà géré par Qt.

Le QML est un langage interprété permettant de développer des interfaces utilisateur. Il faut néanmoins dissocier le QML1 qui se base sur les classes C++ de Qt du QML2 utilisant un réel moteur de rendu web pour afficher les *components*[3]. D'apparence proche du format JSON, le QML permet d'utiliser du JavaScript pour la partie animation/logique de l'interface. Il est aussi possible de mettre en *instance* une classe C++ afin d'accéder à ses fonctions membres. La communication avec le C++ se fait avec des *signaux*.

Grace à l'IDE[4] Qt, il est possible de créer son interface graphiquement à la souris. Mais dans un souci d'optimisation, d'animations et d'organisation du code, toute l'interface qui va être présentée par la suite a été développée à la main, en QML.

Bien sûr, dans le cadre du projet GostCrypt, utiliser une telle technologie

[3] Objets QML utilisables pour créer l'interface. Par exemple : Button, ListView, DropArea, etc.

[4] Environnement de développement

« à l'aveugle » n'est pas une bonne chose. Quid de la gestion mémoire des mots de passe entrés par l'utilisateur ? Pour justifier notre choix, nous avons étudié en profondeur la technologie QML, et en particulier la gestion de la mémoire par le *Garbage Collector*. Toutes ces informations sont présentées dans l'annexe n°1.

--- Le travail effectué

1. La première version de l'interface

Comme expliqué précédemment, l'interface graphique a d'abord été développée en C++, à l'aide des classes proposées par la librairie Qt. Très vite, nous sommes arrivés à la conclusion que cela n'était pas viable au vu du nombre de fenêtres à implémenter. En effet, coder bas niveau quelque chose de dynamique et beau est extrêmement onéreux en temps et en lignes de codes, pour un résultat qui n'est pas forcément celui escompté.

2. Le choix du QML et les maquettes

Lorsque nous avons choisi d'utiliser le QML, il fallait d'abord dessiner l'esquisse de ce qui deviendra l'interface de GostCrypt 2. Pour ce faire, nous avons utilisé des logiciels d'infographie (Adobe Photoshop). Voici ci en haut à droite ce que nous obtenions.

L'interface a depuis bien changé, mais cette maquette nous a au moins permis de choisir la charte graphique de GostCrypt (ses couleurs, les fonctionnalités à afficher sur l'écran principal), et de préciser le type de fenêtre que nous souhaitions.

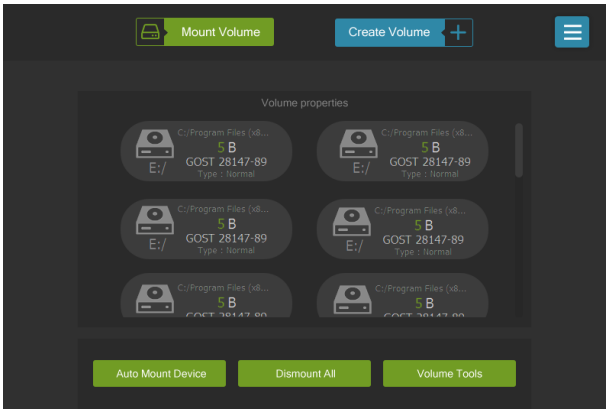


Figure 5 – La maquette de l'interface de GostCrypt 2

Dans le choix des couleurs, nous avons apporté une attention toute particulière au daltonisme. Il serait en effet dommage de priver d'utilisation toute une partie des utilisateurs parce que les couleurs ne sont pas adaptées. Pour cela, il existe des simulateurs et certaines règles (de couleurs, de contraste) pour nous aider.

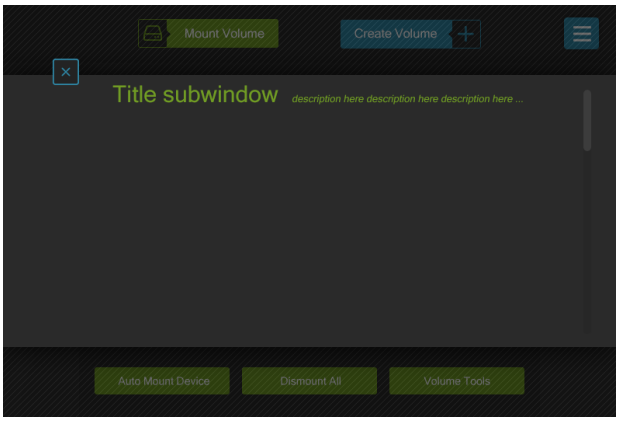


Figure 6 - Les sous fenêtres (concept)

Aussi, en développant la maquette, nous avons pensé qu'il serait plus agréable pour l'utilisateur de ne pas avoir d'innombrables autres fenêtres qui s'ouvriraient à la moindre opération – comme c'était le cas dans TrueCrypt.

Pour y remédier, nous avons décidé de mettre en place des « **sous-fenêtres** », qui s'ouvrent à l'intérieur de la fenêtre principale et permettent à l'utilisateur de s'y retrouver.

3. La refonte du contenu de l'interface

Avoir une charte graphique c'est bien, mais savoir quoi mettre dans l'interface, c'est mieux. Avant de commencer à développer la nouvelle interface, nous avons dû prendre un temps de réflexion pour réduire sa complexité et le nombre de clics.

Voici une liste non exhaustive des changements effectués :

Ancienne interface	Nouvelle interface
Fenêtre principale	
Liste simple des emplacements disponibles pour monter un volume (prend la majeure partie de la place même si aucun volume n'est monté).	Affiche un message de bienvenue tant qu'aucun volume n'est monté, avec une zone de glisser-déposer afin de rapidement monter le volume de son choix. Lorsqu'un volume est monté, l'interface se transforme et affiche correctement la liste des volumes montés. Les volumes montés ont une apparence plus attractive et des raccourcis y sont proposés.
Présence de 10 boutons différents, 6 menus.	Réduction du nombre de boutons à 6, dont 2 pour l'aide et des raccourcis. 6 boutons affichés.
Affichage du formulaire d'ouverture de volume GostCrypt directement dans la fenêtre principale.	Déplacement du formulaire d'ouverture dans une sous-fenêtre et simplifications.
Sous-fenêtres	
De nombreuses fenêtres supplémentaires étaient ouvertes par GostCrypt 1 : messages d'erreur, boites de dialogue pour monter un volume, sélectionner des fichiers, etc.	Hormis les fenêtres d'ouverture de fichier (géré par le système d'exploitation), les messages d'erreur, fenêtres annexes apparaissent directement dans la fenêtre principale comme un calque.

Monter un volume : L'utilisateur était obligé de faire 5 clics. - Sélectionner le volume, l'ouvrir - Cliquer sur « Mount volume » - Entrer le mot de passe dans la fenêtre annexe - Cliquer sur « OK »	Si l'utilisateur utilise la fonctionnalité de glisser-déposer ou le raccourci des volumes en favoris, le nombre de clics passe à 2. S'il ouvre à la main son volume, GostCrypt 2 économise un clic (4).
Menus	
Menus inutiles, inutilisés ou obsolètes.	Les menus ont été épurés, et s'affichent dynamiquement dans un bandeau latéral.

Bien que cette liste n'est ici détaillée qu'à titre d'illustration, toutes les fenêtres de GostCrypt ont été repensées, voire totalement changées afin d'améliorer au maximum l'expérience utilisateur, et surtout de permettre à n'importe quel profil, expérimenté ou non, de pouvoir se servir de GostCrypt tout en comprenant ce qu'il fait.

4. Le développement de la nouvelle interface

Lorsque tout ce qui a été présenté ci-dessus fût terminé, nous fûmes fin-prêts à commencer le développement de la nouvelle interface.

Dans un premier temps, nous avons appris à coder en QML. La technique s'est ensuite affinée au fur et à mesure du développement de l'interface.

Le langage QML offre une fonctionnalité très intéressante : chaque fichier QML peut être considéré comme un « Component » (comprendre un objet qui peut être affiché dans une fenêtre QML). Grace à cela, il est possible de développer des objets réutilisables, comme des boutons personnalisés par exemple. QML permet même de mettre du contenu (texte, image) dynamique dans ces *Components*. C'est exactement ce qui a été fait : nous avons créé nos objets basiques (bouton, radio, liste, menu, sous-fenêtre, etc.) que nous avons pu réutiliser au maximum, afin d'éviter la redondance du code.

Afin de faire communiquer l'interface QML avec la partie C++, nous avons vu qu'il existe des signaux, ou des objets C++ mis en contexte. Pour GostCrypt, nous utilisons les deux méthodes. La première offre la possibilité de faire de l'asynchrone : l'interface n'attend pas la fin d'un calcul pour s'actualiser, mais reste totalement utilisable et est notifiée lorsqu'un événement lui est envoyé par le C++. La seconde méthode est bloquante et devient intéressante dans le cadre de la traduction ou de la mise à jour/récupération des préférences utilisateur.

Les traductions et préférences utilisateurs

Pour permettre une mise à jour en temps réel de la traduction, l'interface

doit communiquer avec une classe C++ qui gère automatiquement la traduction des textes. Ces traductions sont préalablement placées dans des fichiers de langue générés par un outil interne à l'IDE Qt. Ces traductions seront proposées à différents traducteurs de par le monde et ils pourront facilement les éditer grâce à Qt Linguist, un outil de traduction externe proposé par Qt.

Les préférences utilisateur utilisent le même principe. Qt offre une classe qui permet de sauvegarder des paramètres en dehors du logiciel. L'interface communique de façon synchrone avec une classe C++ pour enregistrer ces préférences ou les récupérer, et les afficher en temps réel correctement.

La communication asynchrone

Pour le reste, l'interface est asynchrone. Lorsque l'utilisateur souhaite effectuer une action, le QML va envoyer un *signal* au processus C++. Ce dernier effectuera alors le traitement à part, et lorsqu'il aura terminé, renverra à l'interface un ou plusieurs signaux la notifiant du bon (ou mauvais) déroulement de l'opération. Côté QML, le fichier *SignalManager* gère les signaux, et *GraphicUserInterface* pour la partie C++.

Le sous-fenêtres

Les sous-fenêtres sont assez spéciales. Pour ne pas surcharger la mémoire et l'interface, nous ne pouvons pas charger toutes les sous fenêtres de l'interface, même si elles sont à l'intérieur. Pour cela, seul l'élément conteneur des sous-fenêtres est préchargé. Son contenu est par contre chargé dynamiquement grâce au Component « Loader » qui va lire à la volée le fichier QML demandé. Notons toutefois que même si le QML n'est pas formellement compilé, des fichiers pseudo-compilés sont toutefois générés afin d'optimiser la vitesse de lecture.

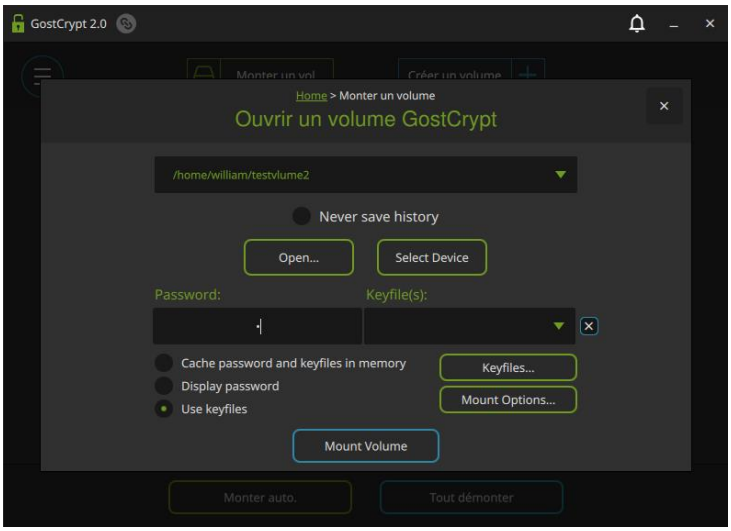


Figure 7 - Sous fenêtre dans GostCrypt 2 (QML)

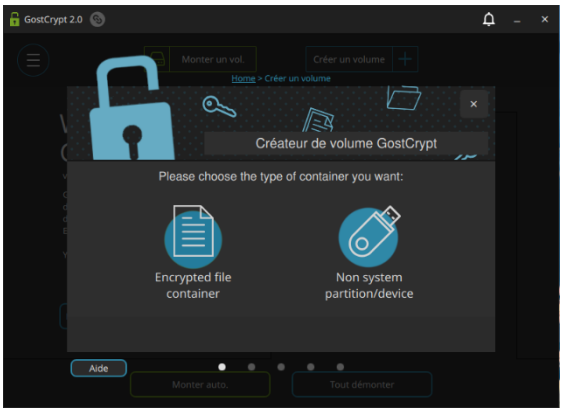
Les animations

Le QML utilise un moteur de rendu web. Plus précisément, il utilise le processeur graphique *GPU* afin d'accélérer le rendu graphique. Il est donc tout à fait possible d'utiliser des animations de déplacement, de changement d'opacité, etc. Les possibilités sont infinies, et nous avons pu profiter de toutes ces fonctionnalités pour implémenter une interface très dynamique, fluide et agréable pour les yeux. Parmi les animations, il y a celles de l'écran principal, lorsqu'un volume est ajouté, ou bien encore le déplacement « smooth » du menu.

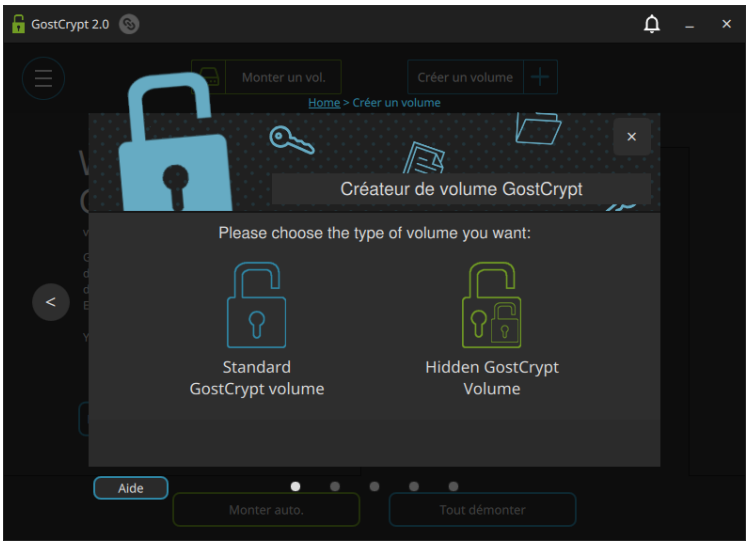
Le *Wizard*[5]

En plus de permettre le montage de volumes, GostCrypt permet d'en créer. Dans l'ancienne version, le Wizard était un programme exécutable dissocié de l'exécutable de GostCrypt. Ce Wizard est en fait une sorte d'installateur mais qui permet de créer son volume GostCrypt par étapes. L'utilisateur choisit l'emplacement de la sauvegarde, sa taille, les algorithmes de chiffrement à utiliser, son mot de passe, etc.

simplifié les étapes, pour ne pas importuner un utilisateur qui connaîtrait déjà les nombreuses remarques affichées entre les étapes.

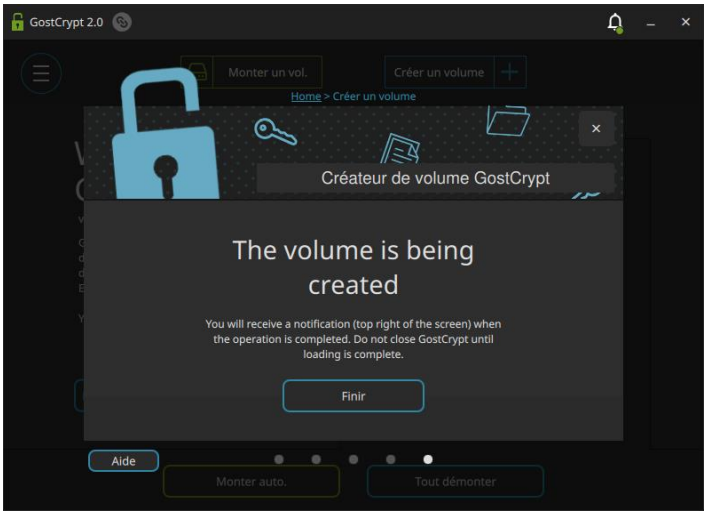


Fenêtre d'accueil du wizard



Des choix visuellement simples

Dans GostCrypt 2, nous avons réalisé un important travail de simplification. D'une part, au lieu de compiler un second exécutable, nous intégrons directement le Wizard dans une sous-fenêtre. D'autre part, nous avons diminué et

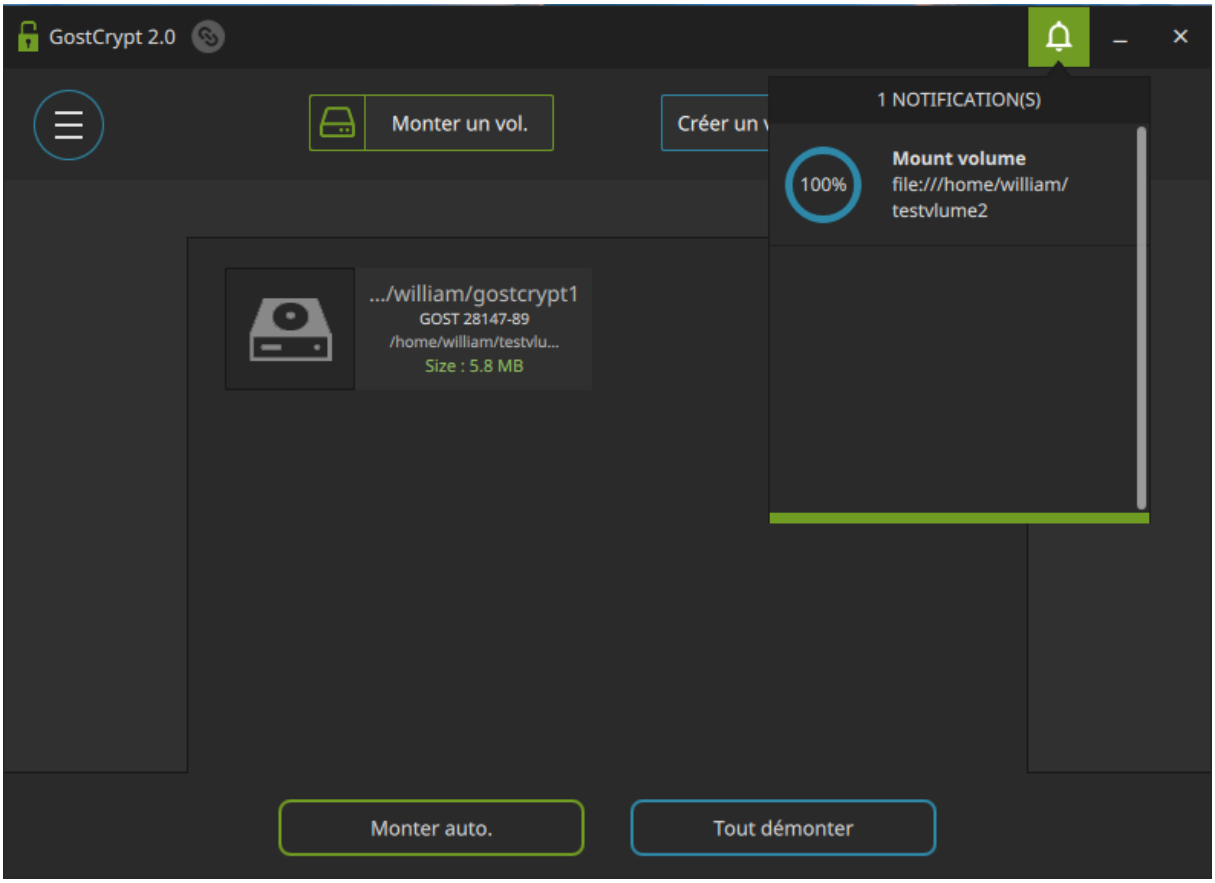


Fenêtre de fin du wizard

[5] Créateur de volume GostCrypt

Les notifications

Enfin, puisque GostCrypt 2 est majoritairement asynchrone, pourquoi faire attendre l'utilisateur lorsqu'une tâche prend du temps ? Avec un système de notifications, nous notifions l'utilisateur des actions en cours en temps réel (mise à jour d'une barre de progression), des erreurs survenues et surtout nous lui permettons de continuer à utiliser GostCrypt même si des tâches importantes sont en cours, ce qui n'était pas possible dans l'ancienne version. Voici un rendu visuel :



5. Les travaux effectués au second semestre

Au niveau de l'interface, nous avons implémenté diverses améliorations et ajouté les dernières fonctionnalités.

Premièrement, les fenêtres de Benchmark ont été implémentées puis reliées au Core. Cette option présente de façon visuelle les vitesses de chiffrement et de déchiffrement des différents algorithmes de chiffrement utilisés. Globalement, ces types de fenêtres peuvent être accessibles depuis différents endroits du programme. Il existe donc des liens entre les sous-fenêtres, ce qui permet une navigation plus intuitive.

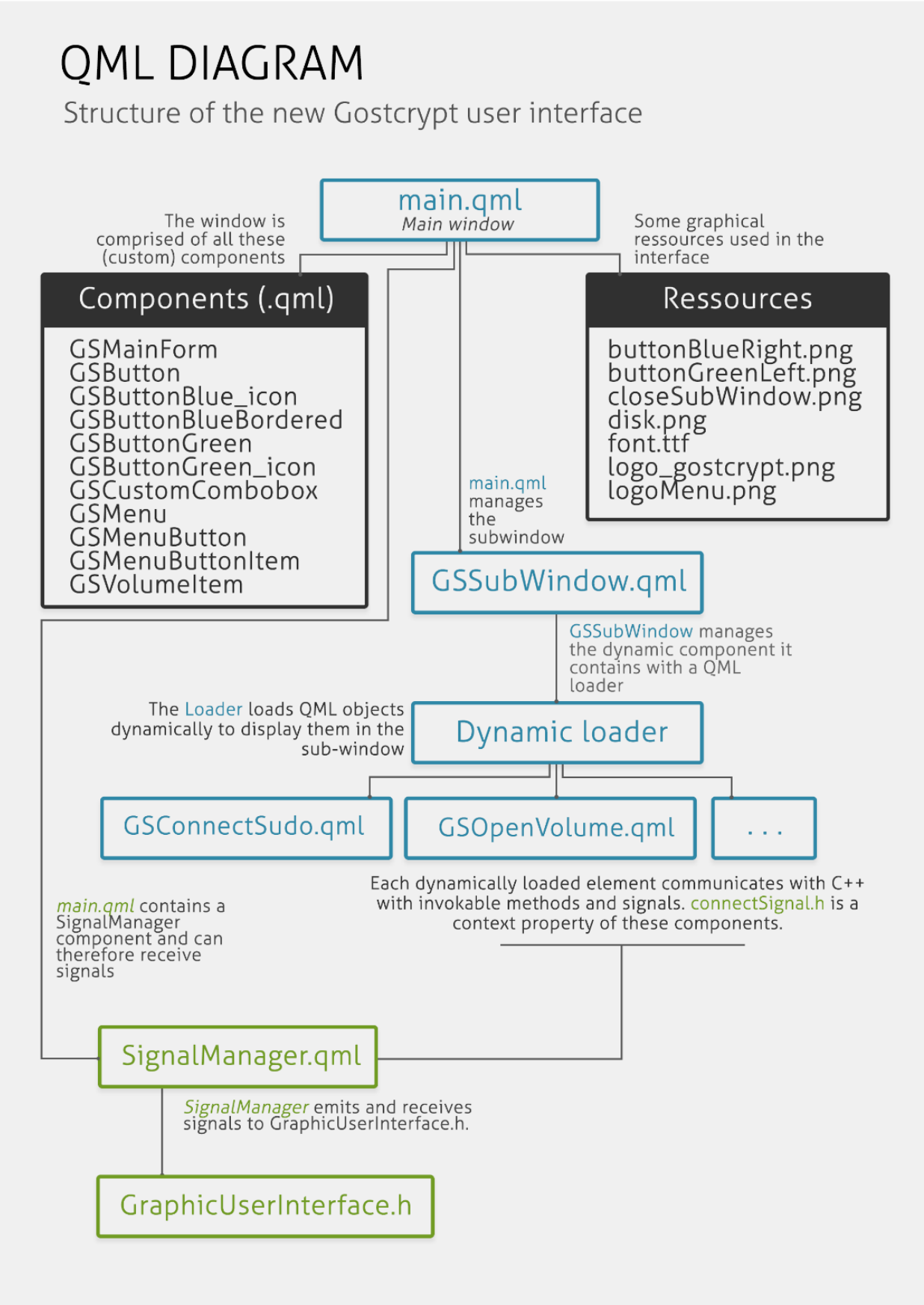
Des optimisations ont été faites au niveau du QML, afin de réduire la taille du code et les calculs inutilement faits en Javascript. En effet, avec une meilleure compréhension de la technologie QML, certains calculs faits initialement en javascript pouvaient être fait directement en QML, et donc par la carte graphique.

Au niveau des améliorations effectuées, on notera entre-autre la possibilité pour l'utilisateur de mieux gérer les keyfiles utilisées pour monter ou créer un volume. Grâce à une fenêtre des keyfiles « sauvegardées », il est possible de rapidement les récupérer lorsque l'on monte un volume, et de gérer ces dernières pour en ajouter ou en supprimer au choix. Le wizard a également été amélioré, en particulier sa structure javascript afin d'assurer une plus grande facilité d'ajout de contenus. Ce dernier a également été totalement relié à l'interface et différents tests nous ont permis de nous assurer de la robustesse de cette partie de l'interface.

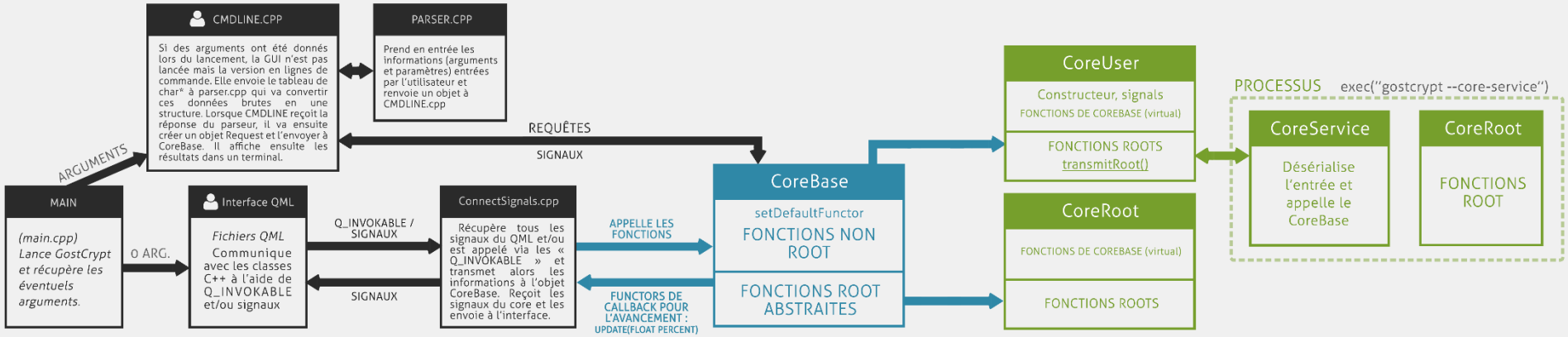
De plus, la possibilité de changer le thème des couleurs de l'interface a été développé, pour à terme proposer aux utilisateurs des thèmes à télécharger afin de diversifier les visuels selon les goûts.

Finalement, l'interface est complète et en attente des dernières fonctionnalités de GostCrypt. Des tests plus approfondis seront aussi réalisés.

Pour présenter plus facilement l'architecture utilisée, voici deux diagrammes résumant les idées principales (architecture QML et architecture globale) :



NOUVELLE ARCHITECTURE - GOSTCRYPT 2.0



Tous les projets originels de GostCrypt sont centralisé en un projet QT, donc tout se compile depuis un seul fichier *.pro.

L'interface est développée en QML via les librairies de QT. Elle est multiplateforme et minimise/simplifie au maximum la complexité de l'expérience utilisateur. Si l'utilisateur lance GostCrypt depuis un terminal, il aura la posibilité d'utiliser le logiciel via des lignes de commandes

ConnectSignals permet de relier l'interface avec le Core de GostCrypt en appelant les fonction associées de CoreBase suite à la réception de signaux/l'exécution d'une fonction Q_INVOKABLE depuis le QML. Renvoie des signaux au QML pour indiquer la fin d'exécution d'une fonction.

CoreBase se divise en deux parties : la première possède un ensemble de fonctions appelées lors du lancement de GostCrypt en mode non root. La seconde possède également des fonctions mais utilisées lorsque GostCrypt est lancé en mode root (ces fonctions sont ici abstraites).

Dans CoreUser, lorsqu'on souhaite appeler une fonction nécessitant d'être root, la classe va appeler transmitRoot(). Cette fonction va lancer un nouveau processus dans lequel gostcrypt sera relancé avec l'option --core-service:

exec("gostcrypt --core-service")

Ce nouveau processus ne possèdera pas de GUI, simplement un objet CoreService qui va communiquer avec un second objet CoreRoot. En redirigeant les entrées et sorties standards, le processus principal va sérialiser ses requêtes pour les envoyer à CoreService. Ce dernier désérialise, demande à CoreRoot d'efeectuer l'action et renvoie les inofrmations à CoreUser.
Si GostCrypt est déjà root par défaut, alors CoreBase va utiliser directement CoreRoot.

--- Objectifs de cette analyse

Comme nous travaillons sur le développement d'un logiciel de sécurité destiné à protéger des données sensibles, nous devons nous assurer de sa sécurité.

Pour ce faire, nous devons analyser et lister les failles potentielles dans le code source de GostCrypt et ainsi penser notre code afin d'optimiser notre logiciel et de le rendre imperméable aux attaques ou à une fuite de donnée qui pourrait, compromettre la fiabilité de nos conteneurs chiffrés.

Pour rappel, le but principal de GostCrypt est de pouvoir créer des partitions chiffrées sur lesquelles un utilisateur peut créer ou déposer n'importe quel contenu qu'il soit privé, sensible ou factice et donc s'assurer que personne ne pourra y accéder. Pour le déchiffrer, il faudra obligatoirement posséder le ou les mots de passes requis. Il ne faut donc absolument pas qu'une personne malveillante puisse retrouver le mot de passe en trouvant et exploitant une faille dans le code source de GostCrypt.

Pour faire ces analyses nous utilisons différents outils que nous décrirons dans la partie suivante du rapport. Il est important de garder à l'esprit qu'il faut faire des analyses fréquentes avec des outils à jour car de nouvelles failles peuvent être découvertes au fil du temps.

Il ne faut en aucun cas laisser de côté un avertissement réalisé par ces outils et même si dans certains cas il peut s'agir d'un faux positif, il faut s'en assurer pour ne pas conserver de vulnérabilités. Il s'agit d'un travail fastidieux mais nécessaire.

Dans notre cas nous utilisons deux types d'outil d'analyse de code : des outils d'analyse statique et d'autres d'analyse dynamique. La différence entre ces deux types d'outils réside dans leur fonctionnement. Une analyse statique va étudier le comportement du logiciel et tenter de détecter des erreurs à l'exécution sans vraiment exécuter le logiciel, alors qu'une analyse dynamique va tester divers aspects du logiciel en l'exécutant. C'est pourquoi il faut dans ce dernier cas faire des tests avec des entrées différentes afin d'obtenir un maximum de résultats/profils d'exécution, et ainsi exploiter au mieux ces outils.

Plus précisément, les outils d'analyse statique repèrent par exemple les variables non utilisées, les problèmes syntaxiques, quelques problèmes d'allocation mémoire, voire détectent si certaines optimisations du code sont possibles. De leur côté, les outils d'analyse dynamique peuvent repérer les fuites de mémoire, les dépassements de tampon ou encore les situations de compétition (aussi appelé « race condition »).

--- Solutions envisagées

Afin de réaliser ces analyses nous avons recherché quels outils étaient les plus adaptés à notre travail. Avec les conseils de M. Filiol, nous avons décidé de prendre deux outils d'analyse statique et deux outils analyse dynamique. Ainsi nous avons retenu CppCheck et Flawfinder en analyse statique et, Valgrind et American Fuzzy Lop en analyse dynamique.

Il est intéressant d'utiliser plusieurs outils d'analyse différents puisqu'ils ne vont pas repérer les mêmes types de problèmes.

Ci-dessous, une brève description de chacun des outils que nous avons utilisés.

Description des différents outils d'analyse :

CppCheck : C'est un outil d'analyse statique pour des programmes codés en C/C++. Il vérifie l'utilisation des variables, les classes, les fonctions dépréciées par Open Group, les problèmes d'allocation mémoire, les librairies ou encore les fonctions inutilisées.

Flawfinder : C'est le second outil d'analyse statique. Il se concentre sur de la « prévention » et fait un audit sur la syntaxe pour mettre en évidence les fonctions qui ne sont pas sécurisées.

Valgrind : C'est un outil d'analyse dynamique du code spécialisé dans la gestion de mémoire. Il met en évidence les erreurs liées à la mémoire comme par exemple le dépassement d'indice dans un tableau ou encore une double libération de mémoire. Il permet en outre d'obtenir un rapport en fermeture du programme de toutes les zones mémoires qui n'ont pas été désallouées.

American Fuzzy Lop : C'est le second outil d'analyse permettant de tester la robustesse du logiciel. Il permet de faire du « *fuzzing* ». Le fuzzing consiste à tester un logiciel en injectant dans ses entrées des données aléatoires. SI le programme échoue, par exemple en plantant ou en générant une erreur, alors il y a des défauts à corriger.

--- Le Travail effectué

Dans un premier temps nous avons décidé d'analyser notre code à l'aide de CppCheck. L'installation de l'outil est très simple car il fonctionne comme un plugin, il suffit de télécharger le package de CppCheck associé à la version de Qtcreator sur le site de Sourcefroge.net. Une fois installé, l'outil est très simple à utiliser : on le retrouve dans l'onglet « outils », puis dans la rubrique « externe » et enfin dans « analyse statique ». Lors de l'analyse, CppCheck va envoyer en sortie toutes les erreurs ou failles qu'il aura pu détecter pour le projet QT actif.

Le rapport CppCheck que nous avons obtenu nous a permis de corriger un certain nombre d'erreurs de code statique, que ce soit de réelles erreurs ou des faux positifs. Tout ce rapport a été documenté et complété des corrections que nous avons effectuées sur notre code.

Nous avons donc rédigé un rapport listant toutes les erreurs rencontrées (regroupées par type), le code posant problème et la solution pour y remédier. Ce document (disponible en annexe) a pour but de mettre en évidence les différentes modifications effectuées pour résoudre des erreurs mais aussi pour nous servir de base si nous venions à rencontrer l'une de ces erreurs à nouveau.

Bien entendu, après chaque modification du code, nous avons retesté notre logiciel et relancé une nouvelle analyse avec CppCheck afin de nous assurer du bon fonctionnement de GostCrypt et que les erreurs venant d'être fixées ne sont plus visibles en sortie d'analyse.

Nous avons aussi effectué les analyses avec Valgrind - qui est déjà préinstallé sur Qt Creator sous forme de deux outils différents. On trouve ainsi Memcheck, qui vérifie que la mémoire est bien libérée et Callgrind qui permet de profiler les fonctions (c'est à dire qu'il vérifie les problèmes relatifs à l'exécution des fonctions). Pour utiliser un de ces deux outils il suffit de le sélectionner dans la rubrique « analyse ».

L'analyse avec Memcheck, qui a dû l'être effectuée en ligne de commande nous a permis de tester chaque fonctionnalité proposée par notre projet. Ainsi nous nous sommes assurés que pour chaque commande (avec les différentes options implémentées pour chacune d'elles) il n'y avait pas de problème d'allocation de mémoire et surtout qu'elle a bien été libérée à la fin de l'exécution. Ainsi, on s'assure qu'aucune donnée intelligible ou non puisse être récupérée pour la réutiliser à des fins malveillantes.

Enfin, l'analyse avec Callgrind, nous a permis de vérifier le comportement de notre logiciel au travers des différentes fonctions appelées. Ainsi, nous nous sommes assurés qu'aucune action non désirée ne survienne.

Nous préférons cependant détailler plus amplement Flawfinder et American Fuzzy Lop après avoir réalisé davantage d'analyses au terme de l'année 2018.

Pour terminer, nous avons mis en place des « testcases ». Puisque GostCrypt possède un mode de fonctionnement en ligne de commande, il est devenu intéressant d'automatiser des tests divers afin de nous assurer via une liste d'actions que l'entièreté du logiciel fonctionne. Ces testcases sont développés en Bash.

- Création du site web de GostCrypt -

--- Contexte du projet au départ

Lorsque nous avions repris le projet GostCrypt, il était prévu de réaliser une nouvelle interface pour moderniser le projet. Mais dans la continuité de nos actions (réécriture complète du logiciel), il était devenu obligatoire de repenser le site web de GostCrypt afin d'y inclure les nouvelles documentations, et rendre l'accès au logiciel plus simple et plus attractif.

--- Solutions envisagées

Afin de limiter au maximum les risques liés à l'hébergement d'un site web d'un projet de sécurité, nous avons cherché les technologies sécurisées. Après nos recherches, nous avons pensé qu'il aurait été plus judicieux de développer le site en statique. Un site en statique n'utilise pas de base de données, et donc il n'est pas possible d'y effectuer des actions malveillantes. De plus en plus de projets sensibles ou de sites web utilisent ce format de site statique pour se prémunir des attaques.

Bien sûr, il existe aujourd'hui des outils complets qui permettent de créer ces sites statiques avec une certaine forme de dynamisme en local. C'est ce que propose par exemple le framework Jekyll, qui offre la possibilité au développeur de créer son site web en suivant un certain formalisme, avec la possibilité de dynamiquement créer des pages lors de la « compilation » du site. Finalement, il est possible de facilement maintenir un site web en y ajoutant des articles sans avoir à modifier tous les liens : tout se fait automatiquement.

Nous avons choisi d'utiliser Jekyll parce qu'il est aujourd'hui le framework le plus abouti pour développer des sites statiques, et il jouit d'une très grande communauté, en cas de problème avec son utilisation.

--- Le travail effectué

Premièrement, nous avons commencé par nous documenter sur Jekyll. Bien que la documentation soit bien faite, il reste assez long d'en comprendre toutes les fonctionnalités.

Ensuite, il a fallu dessiner l'apparence du site : à l'image de l'interface graphique, nous avons imaginé une interface moderne et simple. Les différentes pages ont été créées, donc la page des « articles » qui permettra de publier les différentes notes de release ou informations utiles pour les utilisateurs. Bien sûr, on y trouvera également deux documentations : une documentation utilisateur, volontairement simplifiée, qui se concentrera sur la simple utilisation du logiciel ainsi que les possibilités qu'il offre. La seconde documentation est technique : elle sera adressée surtout aux développeurs qui souhaiteraient contribuer au projet et qui de fait pourront avoir une vision d'ensemble

rapide du logiciel. Cette documentation prendra la forme d'un « graphe » afin de hiérarchiser les informations et permettre de rapidement voir les détails de fonctions sans avoir à chercher partout.

Finalement notre site sera hébergé et remplacera l'ancien lorsque nous choisirons de mettre en ligne GostCrypt pour sa première version Béta.

#4 - CONCLUSION

Cette année a été riche en avancées sur GostCrypt. Alors que l'année dernière nous avions surtout passé du temps sur l'analyse et la compréhension du code source de l'ancienne version de GostCrypt et la conception de la future version, cette année nous avons pu travailler plus concrètement sur la nouvelle version de GostCrypt.

Aujourd'hui, nous pouvons dire que nous avons terminé la réécriture de GostCrypt, avec tous les modules redéveloppés, et toutes les fonctions de GostCrypt fonctionnelles. Le logiciel est donc presque prêt à être publié, il nous reste plus qu'à finir de documenter le code source et à apporter quelques dernières améliorations qui ont été planifiées sur notre plateforme, notamment des améliorations concernant la sécurité du logiciel.

Cette réécriture a considérablement réduit la quantité de code dans le projet. Son développement futur sera simplifié. Il est devenu bien plus facile de comprendre GostCrypt. Enfin, ce nouveau code est documenté de manière exhaustive et une documentation utilisateur a été rédigée. Pour finir, cette nouvelle version inclut une interface graphique beaucoup plus moderne, avec une interactivité se rapprochant des sites web modernes. Elle saura séduire, nous l'espérons, les futurs utilisateurs qu'ils soient novices ou confirmés.

En août, nous avons mis en place un nouvel outil de développement collaboratif basé sur le système de gestion de versions (Git) que nous utilisions déjà. Cette plateforme nous permet d'organiser notre travail et de partager des notes entre les membres du projet. Nous pensons que l'utilisation de cette plateforme nous permettra de continuer à travailler ensemble sur le projet dans les mois à venir, et permettra aux nouveaux étudiants qui rejoindront le projet de travailler avec nous de manière efficace.

#5 - PREVISIONS POUR LA SUITE DU PROJET

GostCrypt est un projet espoir recherche Ainsi, ce projet continuera même si les échéances académiques (PST) sont terminées. Pour tous les membres de notre équipe, GostCrypt est un projet important puisque nous y travaillons depuis plus d'un an déjà.

Le projet GostCrypt est conséquent et demandera encore du travail. Les objectifs principaux qui étaient de réécrire le code hérité de TrueCrypt ainsi qu'une nouvelle interface ont été complétés, ce qui est une étape très importante pour la suite du projet.

A l'avenir, nous comptons mettre en ligne notre projet dans une version béta, afin d'avoir des retours d'utilisateurs et renforcer la sécurité de notre code, si besoin. Nous espérons que cette nouvelle solution open source moderne saura séduire la communauté libriste, mais également le grand public. Même si GostCrypt n'est aujourd'hui développé que sur Linux, nous souhaitons à l'avenir le porter sur Windows, et ainsi offrir à tous les utilisateurs une vraie solution de chiffrement de données personnelles.

Bien sûr, ces étapes iront de pair avec une mise en avant du projet sur les sites spécialisés dans un premier temps, puis directement sur les plateformes de téléchargement usuelles. A terme, la communauté que nous gagnerons sera, nous l'espérons, active dans le développement futur de GostCrypt, afin d'en améliorer sa sécurité, ses performances, ou encore son interface graphique.

Parmi le travail que nous souhaitons encore faire sur GostCrypt à l'avenir, nous aurons dans un premier temps à finaliser nos tests (via des tests unitaires des diverses fonctionnalités du logiciel) ainsi que nos analyses statiques ou dynamiques du code.

La mise en place d'une procédure suivant davantage les recommandations d'une bonne utilisation de GIT sera mise en place. Premièrement, les développeurs du projet ne pourront plus développer directement sur la branche MASTER, afin de n'y placer que les versions stables et disponibles au téléchargement. De plus, si le nombre de contributeurs au projet augmente, GostCrypt pourrait être mis sur la plateforme GitHub, et une hiérarchie sera mise en place (Pull requests).

En termes d'améliorations, nous souhaitons apporter à GostCrypt les fonctionnalités suivantes :

- Possibilité de modifier la taille d'un volume GostCrypt même après qu'il eût été créé.

- Possibilité d'inclure non pas un, mais un nombre indéterminé de volumes cachés à l'intérieur d'un volume GostCrypt.

Puisque nous avons créé une base de site web, il faudra alors s'en servir afin de proposer aux utilisateurs une documentation avancée et simple, en particulier pour les

éventuels utilisateurs de Windows qui n'auraient pas de notions avancées en informatique.

Au niveau de l'interface graphique, nous allons développer notre propre objet pour récupérer les mots de passe utilisateur, afin que la mémoire soit gérée depuis la partie C++ (qui est plus bas niveau que le QML). Ce Component sera nommé **SecureInputTextField** et gèrera les mots de passe des volumes ainsi que l'entrée du mot de passe administrateur.

Enfin, nous avons implémenté la possibilité d'utiliser des thèmes pour GostCrypt : il serait alors intéressant de proposer un créateur de thème GostCrypt afin d'attirer d'éventuels UX designers.

#6 - WEBOGRAPHIE

Qt5 Documentation, The Qt Company website, ©2017. Disponible à l'adresse : http://doc.qt.io/qt-5/index.html

Git – Book, The entire Git book written by Scott Chacon and Ben Straub. Disponible à l'adresse : https://git-scm.com/book/en/v2

GitLab Documentation, created with Nanoc. Disponible à l'adresse : https://docs.gitlab.com/ee/README.html

C++ Reference, ©2000-2017, cplusplus.com. Disponible à l'adresse : http://www.cplusplus.com/reference/

How I compiled TrueCrypt 7.1a for Win32 and matched the official binaries, Xavier de Carné de Carnavalet, Concordia. Disponible à l'adresse : https://madiba.encs.concordia.ca/~x_decarn/truecrypt-binaries-analysis/

Libfuse, Doxygen documentation. Disponible à l'adresse : http://libfuse.github.io/doxygen/

Develop your own filesystem with FUSE, Sumit Singh, October 14, 2014, IBM. Disponible à l'adresse : https://www.ibm.com/developerworks/library/l-fuse/index.html

Fuse FileSystem development: Security Concerns, Jan 1, 2011, Joseph J. Pfeiffer. Disponible à l'adresse : https://www.cs.nmsu.edu/~pfeiffer/fuse-tutorial/html/security.html

CppCheck 1.81 Manual. Disponible à l'adresse : http://cppcheck.sourceforge.net/manual.pdf

Qt Creator documentation. Disponible à l'adresse : http://doc.qt.io/qtcreator/

QtProjectTool, command line tools over a Qt (.pro) project. Disponible à l'adresse : https://sourceforge.net/projects/qtprojecttool/

VeraCrypt 1.18 Security Assessment, October 17, 2016, Quarkslab. Disponible à l'adresse : https://ostif.org/wp-content/uploads/2016/10/VeraCrypt-Audit-Final-for-Public-Release.pdf

Wikipédia, **définition de Fuzzing**. Disponible à l'adresse : https://fr.wikipedia.org/wiki/Fuzzing

#7 - CONTRIBUTIONS DES MEMBRES DU PROJET

William Lardier :

Je suis sur le projet GostCrypt depuis Février 2017 en tant qu'étudiant Espoir Recherche. Depuis, ma contribution s'est surtout orientée au niveau du développement de la nouvelle interface et de tout ce que cela implique. Ayant les compétences en graphisme pour assumer ce rôle, je me charge entièrement de la partie QML et de son dialogue avec le C++. De plus, j'ai pris part dans la réflexion de l'architecture logiciel du nouveau *Core* et je travaille étroitement avec Antoine et Quentin afin de mettre en place des deux côtés les solutions les plus adaptées et évolutives. Je suis responsable de l'expérience utilisateur, je conduis donc la réflexion pour rendre le logiciel plus facile et intuitif à utiliser.

Lorsque nous avons accueilli Quentin en septembre, j'ai pris le temps de lui présenter GostCrypt dans son ensemble, afin qu'il ait rapidement une bonne vision du projet et qu'il connaisse/maîtrise les différentes technologies et terminologies que nous employions alors.

En plus de cela, je travaille avec Antoine sur les problèmes que nous rencontrons durant le développement dans la partie C++ et qui ne sont bien souvent pas triviaux à corriger, à l'image des deadlocks.

Lors du second semestre, j'ai pu me concentrer sur la réalisation du site web de GostCrypt, en utilisant le framework Jekyll qu'il a fallu au préalable comprendre.

Voici une liste non exhaustive des contributions que j'ai faites cette année :

Côté C++ : Création des classes C++ pour gérer la traduction et mise à jour du QML pour permettre la traduction automatique. Classe C++ pour gérer les préférences utilisateur. Création de la structure du *GraphicUserInterface.h/.cpp*. Les signaux du nouveau Core sont correctement reliés à l'interface via des signaux. Débogage avec Antoine de différents problèmes.

Wizard : Création du Wizard, création des 12 sous fenêtres et mise en place d'un arbre de possibilités suivant les choix de l'utilisateur dans ce wizard. Envoi des choix via des signaux au C++ qui traite correctement les demandes. Amélioration du rendu graphique du Wizard à l'aide d'images, animations et simplifications des sous-fenêtres.

Open volume : Nouvelle apparence pour ouvrir/monter un volume. Amélioration des animations.

Design : Amélioration de l'apparence de GostCrypt via de meilleurs contrastes ou choix de formes/couleurs.

Fonctionnalités : La majeure partie des fonctionnalités de GostCrypt est reliée (par exemple : Mount options, Change volume password, Keyfile generator, Volume histo, etc.). Les Keyfiles (fichiers utilisés en tant que mot de passe) sont gérés par l'interface.

Notification : Implémentation du système de notifications et gestion de leur affichage dynamique.

Expérience utilisateur : Diverses améliorations dans toute l'interface pour simplifier, diminuer le nombre de clics, automatiquement placer le curseur dans une zone de texte à remplir. Affichage graphique si l'utilisateur est en mode majuscules pour éviter les mauvaises saisies de mot de passe.

Site web : Réalisation d'un site web statique grâce à un framework dynamique (Jekyll). Création de la charte graphique, du design, et des différentes pages. L'entièreté du site a été codé directement par nos soins.

Quentin Varo :

Contrairement à mes confrères je ne travaille sur le projet GostCrypt que depuis Septembre 2017 en tant qu'étudiant Espoir Recherche et membre du PST. Avant de m'assigner une tache j'ai donc dû en premier temps me mettre à jour sur les travaux accomplis et sur le code effectué sur le projet. Dans les premières semaines, j'ai secondé mes collègues pour bien comprendre le fonctionnement du logiciel GostCrypt et des modifications que nous devons apporter au cours de l'année. J'ai ainsi travaillé avec Antoine sur la réécriture du code de la partie Volume et sur la documentation du code. De plus j'ai aidé William à relier l'interface avec la partie Core à l'aide de signaux. J'ai aussi réalisé une fiche pour comparer les différentes licences logicielles pour nous aider à choisir la nouvelle licence pour GostCrypt. Aussi, j'ai réalisé une fiche d'installation pour les éventuels futurs repreneurs du projet (voir Annexes). Cette fiche leur permettra de rapidement, étape par étape, installer leur environnement de développement de GostCrypt.

Depuis novembre 2017 mon rôle principal est de travailler sur l'analyse et le renforcement du code de GostCrypt. Pour faire ces analyses je travaille avec des outils d'analyse de code, CppCheck et Flawfinder pour les outils d'analyse statique, et Valgrind et American Fuzzy Lop pour les outils d'analyse dynamique. J'ai pu corriger les erreurs trouvées par CppCheck et rédiger un rapport d'erreur recensant toutes les erreurs rencontrées, le code posant problème et une solution de correction de code associée. J'ai aussi travaillé sur l'outil Valgrind qui se divise en deux outils sur Qt Creator, Memcheck pour vérifier la mémoire et Callgrind qui est un profileur de fonction. Comme Valgrind est un outil d'analyse dynamique j'ai dû tester GostCrypt en ligne de commande et tester chaque commande avec les différentes options proposées. A ce propos j'ai pu

m'apercevoir que certaines options contenaient des erreurs d'implémentation que j'ai pu corriger.

De plus j'ai aussi travaillé avec Antoine dans l'implémentation de nouvelles fonctions dans le projet. J'ai ainsi pu implémenter les fonctions de benchmark qui permettront aux utilisateurs de comparer la vitesse de chiffrement/déchiffrement des différents algorithmes disponibles dans le logiciel.

Enfin, puisque le projet après les échéances académique, je compte faire les analyses avec Flawfinder et American Fuzzy Lop qui devraient prendre moins de temps car le code aura déjà été analysé avec un outil d'analyse statique (CppCheck) et un autre dynamique (Valgrind).

Antoine Hébert :

Je suis membre du projet GostCrypt depuis début 2017 en tant que membre du PST de troisième année puis de quatrième année et d'étudiant espoir-recherche. Louis et moi nous sommes occupés les premiers mois de l'analyse du code source de l'ancien GostCrypt afin de comprendre son architecture globale et la raison des choix de conception qui avaient été faits. J'ai ensuite proposé une nouvelle architecture logicielle pour la nouvelle version de GostCrypt, qui s'inspire du fonctionnement de la version précédente en résolvant un certain nombre de problèmes de conception (reste de code non utilisé, programmation par exception, dépendance circulaire). Cette architecture a évolué grâce aux discussions avec les autres membres de l'équipe. Je me suis ensuite occupé du développement des modules Core puis Fuse et Volume de Gostcrypt.

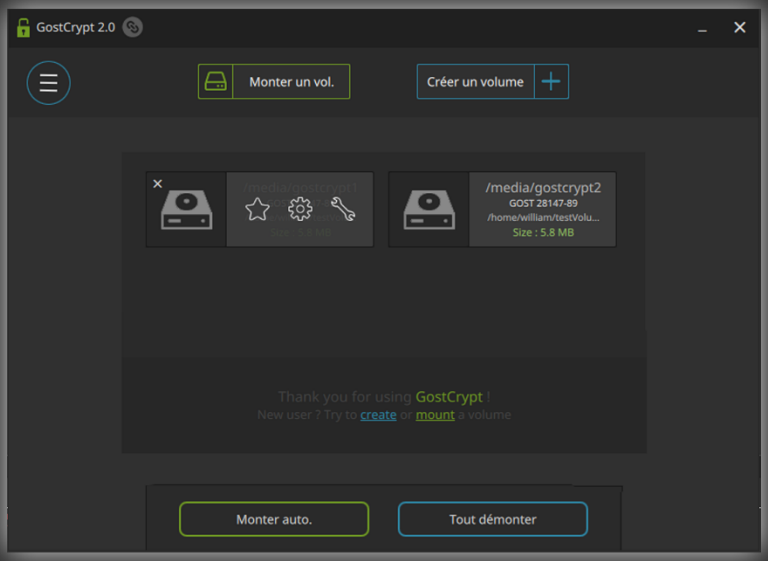
Grâce à ma bonne connaissance du fonctionnement de GostCrypt, j'ai aussi pu assister les autres membres de l'équipe lors de leur développement, les aider à déterminer les tâches à effectuer et à synchroniser leur travail. Au mois d'août, j'ai mis en place la plateforme de développement collaboratif GitLab, afin de mieux travailler ensemble même à distance et de plus facilement gérer le logiciel de gestion de version Git que nous utilisions déjà. Depuis septembre, j'ai continué le développement des derniers modules qui n'avaient pas encore été réécrits, notamment le module Fuse sur lequel j'ai travaillé seul, et le module volume sur lequel j'ai travaillé avec Louis. Depuis janvier, je me suis consacré à la relecture complète du code source pour m'assurer de sa cohérence, corriger les éventuelles erreurs ou oubli et finir de le documenter. Cela m'a permis de fixer le code sources des modules Core, Fuse et UI qui ne devrait plus avoir à changer beaucoup. Je prévois de continuer dans les prochains mois avec le module Volume que je n'ai pas eu le temps de traiter.

#8 - AFFICHE DU PROJET

PROJET SCIENTIFIQUE ET TECHNIQUE DE 4E ANNÉE
2017-2018

LOGICIEL DE CHIFFREMENT DE VOLUME OPEN SOURCE
SUIVI PAR ÉRIC FILIOL

GOSTCRYPT 2.0



NOUVELLE ARCHITECTURE LOGICIELLE
RÉÉCRITURE COMPLÈTE EN LICENCE OPEN SOURCE
IMPLÉMENTATION D'ALGORITHMES MUTANTS INNOVANTS
NOUVELLES FONCTIONNALITÉS

GOSTCRYPT
ENCRYPTION
SOLUTION

esiea
ECOLE D'INGENIEURS
DU MONDE NUMERIQUE

HÉBERT ANTOINE, LARDIER WILLIAM, VARO QUENTIN

#9 - TIMELINE (SEMESTRE 1)

Réunion de lancement.

Antoine : Finalisation du module Core (implémentation des dernières fonctionnalités, documentation et tests).

Louis : Développement de la CLI (Command Line Interface) et intégration.

Quentin : Formation au fonctionnement de GostCrypt.

William : Reliage de la nouvelle interface au Core (monter un volume, wizard).

Septembre

Octobre

Réunion intermédiaire.

Antoine : Liaison des nouvelles fonctionnalités du Core à l'interface graphique (côté C++). Développement du module Fuse et intégration avec le module Core.

Louis : Développement de la CLI et intégration (suite).

Quentin : Découverte et prise en main de CppCheck, Valgrind et FlawFinder. Suite de la formaion au fonctionnement de GostCrypt (partie interface).

William : Améliorations sur le design de l'interface, suite du reliage avec le Core, ajout du système de notifications. Début de la création du nouveau Wizard.

Réunion intermédiaire.

Antoine : Développement du nouveau module Volume par modifications successives de l'ancien code (permet de tester au fur et à mesure). Suppression du module Platform.

Quentin : Correction des alertes générées par CppCheck. Analyse des quelques fuites mémoires détectées et correction.

William : Suite du développement du nouveau wizard, diverses améliorations de l'interface et de ses fonctionnalités. Suite du reliage du code, correction de quelques bugs dans GostCrypt.

Novembre

Décembre

Réunion intermédiaire.

Antoine Tests du logiciel, correction de bugs. Rédaction du rapport. Amélioration esthétique du code source. Documentation des modules Fuse et Volume (en cours).

Quentin : Fin de la correction avec CppCheck, rédaction du rapport de correction, du rapport intermédiaire. Début des tests avec FlawFinder, et reliage (côté Core) de certaines nouvcelles fonctionnalités avec l'interface (en cours).

William : Fin du reliage des fonctionnalités au Core, ajout d'améliorations graphiques, prise en charge des utilisateurs daltoniens, amélioration du système de notification et nettoyage du code inutilisé. Rédaction du rapport.

Janvier

Réunion intermédiaire.

Antoine : Relecture du module Core, correction du code.

Quentin : Implémentation des fonctions de Benchmark dans le Core.

William : Amélioration dans l'interface, reliage des dernières fonctionnalités et optimisation du code. Ajout des thèmes pour l'interface QML.

Février

.

Antoine Relecture du module Fuse et documentation exhaustive des modules Core et Fuse.

Quentin : : Tests et finalisation de la partie Command Line.

William : Prise en main de Jekyll, puis développement du site web de GostCrypt (design, maquette, base code puis les différentes pages).

Mars

Réunion intermédiaire.

Antoine : Migration du projet sur un nouveau serveur plus performant, tests et correction des bugs dans les fonctionnalités, finalisation de la documentation.

Quentin : Analyse avec Valgrind (Détection de fuite mémoire).

William : Fin de développement du site de Gostcrypt, correction des divers bugs présent dans les fonctionnalités.

#10 - ANNEXES

Fiche Annexe 1 : *Fonctionnement bas niveau du QML*

CHAPITRE 1 : LE QML, gestion mémoire côté C++/QML

Il existe 2 versions de QML :

- La première (**QML1**) est ancienne, et permettait juste d'afficher des QWidget C++ dans une fenêtre, donc cela simplifiait surtout le code.
- La seconde utilisée pour GostCrypt (**QML2**) utilise un moteur de rendu avec le GPU (un moteur web), et tous les objets sont dérivés de 2 classes : QObject si c'est un objet non-visuel, et **QQuickItem** (qui hérite de QObject) si c'est un objet visuel.

Une interface QML est donc en mémoire représentée comme un **arbre de QObjects**, alloués sur le tas du processus en cours : chaque objet a un parent. Quand un objet QML est détruit, il détruira aussi récursivement tous ses enfants, comme pour Qt C++.

Maintenant, il existe 2 façons de créer des objets QML :

- **La façon statique** : c'est ce qu'on met par exemple dans un fichier *main.qml*. Il ne bougera pas.
- **La façon dynamique** : on charge l'objet depuis un "Loader" (c'est le cas des sous-fenêtres pour par exemple monter un volume).

Il existe également 2 façons pour QML de stocker en mémoire les infos d'un objet :

- Les propriétés **normales** des objets QML (width, height, color, ...) sont stockées sur le tas du processus courant dans des QQmlVMEVariant.
- Toutes les **variables** sont quant à elles allouées comme des QV4::Values dans un QV4::Array qui se trouve sur le tas du JavaScript.

Donc quand on détruit un objet QML, les propriétés sur le tas du processus courant sont supprimées, mais les variables JS de l'objet sont simplement dissociées, et seront supprimées plus tard par le **Garbage Collector** (**GC**) du Javascript.

Cas particulier : il est possible de stocker des objets QML dans des variables JS (utile pour un système d'affichage dynamique), et dans ce cas si l'objet est supprimé, la variable sera bien nettoyée en même temps que le QObject, sans passer par le GC.

Le **Garbage Collector** va se charger des objets QML :

- Qui ont le flag QQmlEngine::JavaScriptOwnerShip (dans ce cas tout est sur le JS heap),
- Qui n'ont pas de parent,
- Qui n'ont plus de référence javascript s'il s'agit d'un objet créé avec du JS.

Dans le cas d'un objet QML chargé depuis un Loader (donc dynamiquement), il suffit de changer la source du Loader pour automatiquement libérer l'objet qml qui était chargé

(et on revient ensuite à ce qui a été dit plus haut) : "An item loaded with a Loader element may be released by resetting the "source" or "sourceComponent" property of the Loader, while other items may be explicitly released by calling destroy() on them."

Dans le cas de GostCrypt, c'est ce qu'il se passe quand on ferme la sous-fenêtre : la source est réinitialisée et cela permet de libérer l'objet pour qu'il ne reste pas inutilement en mémoire.

CHAPITRE 2 : Gestion de la mémoire côté JS

Intéressons-nous maintenant à comment est gérée la mémoire par JavaScript, et comment fonctionne le GC sous QML.

Lors de la création d'un volume, les informations sont stockées dans une variable JS (de type ***variant***, qui correspond au **QVariant** en C++) ensuite envoyée à C++ via un signal.

QML utilise le moteur V4. C'est **QV4::MemoryManager** qui gère la mémoire.

L'allocation se fait sur demande sur le tas (heap) JS, et des "**buckets**" (espace mémoire) sont alloués. Ces blocs mémoire ont une taille arrondie à une puissance de 2 (32, 64, ..., 512 bytes). Globalement, la mémoire se compose d'une liste de liste de buckets : "Segregated fits allocation". Plus précisément, les buckets sont regroupés dans des **chunks** qui sont alloués sur demande.

Ce système est robuste contre la fragmentation externe, c'est-à-dire l'impossibilité de fusionner des blocs libres car ils ne sont pas contigus, mais il est possible qu'il y ait de la fragmentation interne (allocation d'un bloc beaucoup trop large, et donc gaspillage de mémoire).

Ces buckets ne sont pas alloués via un malloc, mais via **mmap** (**POSIX**) ou **VirtualAlloc** (**Windows**), **sauf si l'allocation dépasse 512 bytes**, auquel cas c'est malloc/free qui sont appelés.

Les chunks sont donc divisés en items de taille n. Ces items sont mis sur la « **freelist** » qui **indique pour un bucket donné s'il est libre ou pas**.

Quand cette « freelist » est vide (c'est à dire qu'il n'y a plus de place), soit un nouveau chunk est alloué, soit le Garbage Collector est invoqué.

Note : « A newly allocated chunk is commited memory »

Propriété intéressante, lorsqu'un chunk est alloué, le prochain aura une taille deux fois supérieure. Si le premier chunk fait 64KB (valeur réelle), le second fera 128, etc., le tout étant plafonné à 2MB. Cette propriété induit un fort risque de gâchis de la mémoire. Il est possible de modifier la taille minimale et maximale d'un chunk avec les macros : QV4_MM_MAXBLOCK_SHIFT et QV4_MM_MAX_CHUNK_SIZE

Comment le GC fonctionne ?

Il est invoqué lors d'une allocation (si la freelist est pleine), ou manuellement via C++ ou JS. Il tourne dans le thread principal, et est donc bloquant pour l'application. On trouve son implémentation dans **QV4::MemoryManager**.

1. **Phase 1**

Chaque objet/variable accessible par le GC possède un bit de marquage. S'il est à 0, c'est que le GC peut libérer la mémoire de cet objet. Depuis Qt5,2, le GC est de type « PRECISE », c'est à dire qu'il sait directement les objets qu'il doit considérer. (Avant, c'était un « CONSERVATIVE » GC, il devait vérifier qu'il s'agissait d'une adresse mémoire valide avant de libérer la mémoire).

2. **Phase 2**

Le GC va balayer la mémoire. Chaque objet marqué aura sa marque retirée, et chaque objet sans marque sera détruit, « nulled » (mis à 0), et le bucket associé sera ajouté à la freelist. Si un chunk entier est libéré (et ce depuis Qt5.5), l'espace mémoire du chunk est redonné à l'OS. Lors de la fermeture du programme, un balayage est fait sans prendre en compte les marques (tout est détruit et mis à 0).

Problèmes et recommandations connues liées au GC :

- Lorsque l'on stocke une chaîne de caractère (QV4::String), elle est en réalité stockée sur le tas C++ comme QStringData*. **Le GC ne va alors jamais libérer cet espace mémoire.**
- Lorsque l'on « unload » un gros objet QML, il est utile d'appeler le GC manuellement pour s'assurer qu'un tour de QEventLoop s'est fait une fois avant la libération mémoire (dans le cas où des opérations sont faites lors de la suppression de l'objet QML). On peut même lancer malloc_trim(0) afin de redonner à l'OS un éventuel chunk libéré entièrement.

CHAPITRE 3 : Memory profiling dans QT

L'IDE Qt propose un « profiler » de mémoire, et dans ce dernier, il y en a un pour le JavaScript. Il est possible également d'utiliser QV4_MM_STATS en mettant sa valeur à 1 dans le code. Cependant ils ne permettent pas de voir la partie QML.

Pour cela il est aussi possible d'utiliser Valgrind couplé à massif ou massif-visualize pour visualiser l'utilisation de la mémoire (les objets QML sont visibles).

Finalement :

- Profil de la mémoire de l'application : méthode spécifique à l'OS
- Profil de la mémoire JS : Qv4_MM_STATS, QtCreator
- Utilisation mémoire par QML : Valgrind

Fiche Annexe 2 : How to Install the environment for GostCrypt Development step by step

1) Install a Linux OS, for example Linux Mint is good to start with.

2) Go on the Internet at the URL: https://info.qt.io/download-qt-for-application-development and download the open source package of Qt.

3) Open a terminal and go where the package has been downloaded and change the rule to make it executable.

For example:

cd Downloads/

chmod +x qt-unified-linux-x64-3.0.1-online.run

4) Execute the file of Qt in sudo to instal it.

For example:

sudo ./qt-unified-linux-x64-3.0.1-online.run

5) Now the installation will start, just be careful to include « sources » on your version of Qt and include the installation of Qt Creator too.

6) Qt is installed you can run Qt Creator on your panel but you have to include build-essential and some libraries: fuse, opengl, blkid and libblkid.

For example:

sudo apt-get install build-essential

sudo apt-get install libfuse-dev

sudo apt-get install libgl-dev

sudo apt-get install libblkid-dev

sudo apt-get install blkid-dev

7) Now you can open Qt Creator and run the GostCrypt.

9) To Install Doxygen go on the Internet at the URL: and download the last version of Doxygen. Open a terminal, change the rule to make it executable and put it at Qt/Tools/QtCreator/lib/qtcreator/plugins/.

For example:

chmod +x libDoxygen-0.4.6-qtc4.3.x-x86_64.so

mv libDoxygen-0.4.6-qtc4.3.x-x86_64.so Qt/Tools/QtCreator/lib/qtcreator/plugins/

8) If GostCrypt is your project within the CVO lab, you can register on the git at https://git.legendia.fr/.

9) First of all, generate your public RSA key.

ssh-keygen

10) Copy your public key and go on the Internet at the URL: https://git.legendia.fr/

Go on your private place and in the parameters past your public key on the tab « SSH ».

11) Now open a terminal and clone the project on your Desktop, choose the place where you want to clone it and enter:

git clone gogs@git.legendia.fr:hantoine/gostcrypt.git

Fiche annexe 3 : Tableau comparatif des licences

License	copyleft	copyfree	sublicensing	GPL compatibility	secret modification	French law
ceCILLv2	yes	no	no	yes	no	yes
CeCILL-C v2	no	no	no	yes	no	yes
CeCILL-B v2	no	no	-	no	-	yes
Original BSD	no	no	-	no	-	no
Modified BSD	no	yes	-	yes	yes	no
EUPL	yes	no	yes	no	-	no
GPLv3	yes	no	no	yes	no	no
LGPL	no	no	no	yes	-	no
Apache v2	no	no	yes	GPL v3 only	-	no
MIT	no	yes	yes	yes	yes	no

Compte rendu de réunion n°1

Projet : GostCrypt
Objet de la réunion :
Date – horaire et lieu : Le 3 octobre 2017, à 16h30 au sein du laboratoire CVO.
Présents (trié dans l'ordre des sociétés, des fonctions, des noms) : Louis Béclair, Antoine Hébert, William Lardier, Quentin Varo.
Absents : Néant

Contenu de la réunion :
Partie documentation : Le code de la partie Core est fini d'être réécrite, il faut documenter cette partie.
Actions à mener : Documenter la partie Core et réaliser un graphe dynamique à l'aide de l'outil Zscope pour décrire la structure de notre code source.

Partie cryptologie : Réflexion sur les algorithmes de chiffrement et de hachage à conserver et à supprimer.
Actions à mener : On enlève GOST 64 bits et on conserve Grasshopper avec modification de clé à tous les blocs. Puis après tout ceci, voir avec M.Filiol pour intégrer le nouvel algorithme de chiffrement chinois SM4.

Partie analyse de code : Vérifier qu'il ne reste pas du code inutile.
Actions à mener : Faire des analyses à l'aide de Cppcheck pour qu'il détecte les fonctions, les structures et les variables inutilisées dans le code et les supprimer.

Partie licence : Réfléchir à quelle licence nous allons utiliser pour Gostcrypt car elle ne doit pas être bloquante.
Actions à mener : Comparer les licences CeCill, Free BSD, GNU 3 et Apache et réaliser une fiche pour présenter ces différentes licences et mettre en évidences leurs particularités.

Décisions prises et récapitulatif des actions à mener : Finir de documenter la partie Core. Réaliser la description de notre code source à l'aide d'un graphe dynamique. Retirer GOST 64 bits et conserver Grasshopper avec modification de clé à tous les blocs. Commencer l'analyse du code à l'aide de l'outil CppCheck et corriger les erreurs trouvées.
Préparer une fiche pour comparer les différentes licences afin de choisir la plus appropriée pour GostCrypt.

Horaire de fin de réunion : 17h05
Prochaine réunion : Une prochaine réunion sera fixée lorsque le projet aura suffisamment avancé (d'un commun accord avec M.Filiol).

Compte rendu de réunion n°2

Projet : GostCrypt

Objet de la réunion : Effectuer un retour sur l'avancement du projet et poser les questions prévues afin de résoudre certains problèmes.

Date - horaire et lieu : Le 15 novembre 2017, à 14h30 au sein du laboratoire CVO.

Présents (trié dans l'ordre des sociétés, des fonctions, des noms) : Louis Béclair, Antoine Hébert, William Lardier, Quentin Varo.

Absents : Néant

Contenu de la réunion :

Partie cryptologie : on ne garde que le GOST Grasshopper, avec deux modes de fonctionnement. Action à mener : cartographier les procédures crypto, et les documenter.

Partie documentation : Proposition de réaliser un nouveau site pour GostCrypt.
Actions à mener : réaliser un site statique, créer une documentation présentant sous forme de graphe les fonctionnalités de GostCrypt pour faciliter l'accès aux informations pour l'utilisateur. Réaliser un manuel utilisateur.

Partie analyse de code : Présentation des premiers résultats de CppCheck.
Actions à mener : continuer l'analyse du code avec Valgrind, American Furry Lop.

Partie licence : Présentation de la licence CeCill.
Actions à mener : ajouter des headers CeCill dans tous les fichiers du projet.

Partie interface : Préparation des fichiers de traduction. Comment est détectée la langue ? Gérer le daltonisme.
Actions à mener : continuer de relier les fonctionnalités et se renseigner sur comment Qt détecte la langue d'un ordinateur. Tester le rendu visuel de l'interface suivant le type de daltonisme.

Décisions prises et récapitulatif des actions à mener : Utilisation de la licence CeCill, mise à disposition de fichiers de traduction dès qu'ils seront prêt pour traduire GostCrypt dans un maximum de langages, retirer l'algorithme GOST-89, réaliser une nouvelle forme de documentation web.
Horaire de fin de réunion : 15h10
Prochaine réunion : Lorsque le projet aura suffisamment avancé (d'un commun accord avec M. Filiol)

Compte rendu de réunion n°3

Projet : GostCrypt

Objet de la réunion : Faire part de l'avancement du projet sur le module Volume, l'interface et l'audit du code

Date - horaire et lieu : Le 12 décembre 2017, à 16h00 au sein du laboratoire CVO.

Présents (trié dans l'ordre des sociétés, des fonctions, des noms) : Louis Béclair, Antoine Hébert, William Lardier, Quentin Varo.

Absents : Néant

Contenu de la réunion :

Partie module Volume : Antoine et Louis ont terminé de recoder le module Volume. Il est fonctionnel et a été testé avec la nouvelle interface.
Actions à mener : Documenter ce module.

Partie Audit : L'analyse statique du code avec CppCheck est terminée, un rapport a été rédigé présentant par type les différents problèmes rencontrés ainsi que leur résolution.
Actions à mener : Continuer l'analyse du code avec FlawFinder et Valgrind dans un premier temps.

Partie interface : Présentation de la nouvelle fonctionnalité des notifications. De nombreuses fonctionnalités ont été reliées à l'interface et les fenêtres associées ont été codées.
Actions à mener : Finir de relier les dernières fonctionnalités et continuer d'améliorer l'interface.

Partie licence : Il n'y a pas de header CeCill intéressant sur internet pour le projet.
Actions à mener : Trouver un header que l'on puisse utiliser. [Mise à jour suite à la réunion : nous avons trouvé un en-tête que nous pourrons utiliser]

Partie documentation : Début d'étude sur Zscope.
Actions à mener : développer la nouvelle documentation sous forme de graphe.

Décisions prises et récapitulatif des actions à mener : Documenter le module Volume. Ajouter les fonctionnalités manquantes de GostCrypt. Continuer l'audit du code avec les outils d'analyse. Commencer la documentation sous forme de graphe.
Horaire de fin de réunion : 16h30
Prochaine réunion : Lorsque le projet aura suffisamment avancé (d'un commun accord avec M. Filiol)

Compte rendu de réunion n°4

Projet : GostCrypt
Objet de la réunion : Retour des vacances de noël, avancées sur le projet
Date - horaire et lieu : Le 26 janvier 2018, à 15h00 au sein du laboratoire CVO.
Présents (trié dans l'ordre des sociétés, des fonctions, des noms) : Antoine Hébert, William Lardier, Quentin Varo.

Absents : Néant (Louis Béclair mais pas membre du PST)

Contenu de la réunion :

Partie Analyse de code : Pas vraiment d'avancement. Action à mener : continuer avec les analyseurs dynamiques.

Partie site internet : Début du développement du nouveau site pour GostCrypt. Utilisation du framework Jekyll pour générer des pages statiques. Création du thème graphique.
Actions à mener : Finir les Template et remplir le site des textes. Commencer ensuite la documentation sous forme de graphe dans ce dernier.

Partie améliorations : Préparer la seconde phase su développement du logiciel : enrichissement des fonctionnalités. On ne garde que Grasshopper, et on abandonne SMS4 pour cause de probables failles. Il serait intéressant d'améliorer la vitesse de Grasshopper. Mais il est déjà développé en multithread, multiplateforme. Faire attention aux deadlocks.
Actions à mener : Tester de remplir 1Go de RAM avec Grasshopper pour voir sa vitesse. Nouveaux algorithmes : DFC : se documenter + tester + multithread + trouver les sources.
Nouveauté : Grasshopper mutant
Protocole U2F pour l'authentification. Utilisation d'une clé et d'un bouton pour authentifier l'utilisateur : faut-il le rendre obligatoire ? Notion d'argent (clé usb de clé payante). Implémenter ce protocole dans GostCrypt.

Partie documentation : Pas d'avancement.
Actions à mener : Documenter la partie crypto.

Conférence en Russie du 19 au 23 mars

Décision et récapitulatif des actions à mener : Utilisation de la licence CeCill, mise à disposition de fichiers de traduction dès qu'ils seront prêt pour traduire GostCrypt dans un maximum de langages, retirer l'algorithme GOST-89, réaliser une nouvelle forme de documentation web.
Horaire de fin de réunion : 15h10
Prochaine réunion : Lorsque le projet aura suffisamment avancé (d'un commun accord avec M. Filiol)

GostCrypt : Liste des commits (en date du 13/01)

2018-01-13 16:30:58 Rename RandomNumberGenerator to RandomGenerator
2018-01-13 16:30:25 Move GetFileSystemTypePlatformNative from header file to cpp file
2017-12-12 15:40:48 Rename RandomNumberGenerator to RandomGenerator
2017-12-12 15:39:04 Improve exceptionHandling in RandomNumberGenerator
2017-12-12 15:28:45 Move VolumeParams constructor to cpp file
2017-12-12 15:20:32 Fix hidden volume creation offset + Add incorrectparameter exception when not using 100% of the primary volume
2017-12-12 15:11:27 fix XTS mode graphic
2017-12-12 11:13:35 add the final report of CppCheck Warnings
2017-12-11 21:37:12 Remove all remaining use of std::wstring
2017-12-11 21:30:45 Replace std::wstring by QString in EncryptionAlgorithm::getName()
2017-12-11 21:17:39 Fix graphical glitch with drag'n'drop
2017-12-11 20:45:38 Fix tests
2017-12-11 20:30:40 Fix compilation errors after merging
2017-12-11 19:41:57 Solve deadlock in EncryptionThreadPool
2017-12-11 01:08:00 Fix #62 Make requestId in Exception actually work by cloning the exception to add the requestId before throwing it again
2017-12-10 23:53:22 Remove almost not used class Volume::Version
2017-12-10 23:50:21 Fix #61 : Finish developing CoreBase::restoreBackup (restore from backup header file)
2017-12-10 23:37:32 Displaying exceptions via notifications #33
2017-12-10 20:46:24 Develop CoreBase::restoreBackup when using internal backup
2017-12-10 18:03:34 - Removed unused menu items and dialogs - Renamed the "FirstGI" struct to "UI" - Binded the "Clear volume history" menu option
2017-12-10 02:10:14 Prepare for CoreBase::restoreBackup
2017-12-10 01:59:19 Develop CoreBase::backupHeader
2017-12-10 01:59:01 Fix compilation errors and errors with the argument useBackupHeader not used at the right place in argument list
2017-12-10 00:54:55 Remove unused arguments in Volume::Volume::Open method
2017-12-09 23:37:23 Prepare methods for (Restore|Backup)Header
2017-12-09 23:09:56 Fix indentation EncryptionThreadPool.cpp
2017-12-07 22:20:41 Correct requests dump file name
2017-12-07 21:41:11 Fix #14
2017-12-07 16:16:11 Give the same name to the argument at the declaration and the definition.
2017-12-07 14:57:09 Give the same name to the argument at the declaration and the definition.
2017-12-07 14:53:38 Give the same name to the argument at the declaration and the definition.
2017-12-07 14:21:48 precedent commit was uncomplete
2017-12-07 13:26:26 HMAC functions are now prefixed with HMAC_. fixing #11
2017-12-07 12:58:22 Fixing RandomNumberGenerator and adding support for getrandom. closing #45.
2017-12-06 23:57:42 Remove Platform module ! :D :D :D
2017-12-06 23:09:51 Remove NewCore which had been renamed to Core
2017-12-06 23:09:04 Indentation correction in MountFilesystemManager.cpp
2017-12-06 19:44:45 Cleaning unused files
2017-12-06 19:37:01 Added PROGRESS UPDATE in the createRandomFile function
2017-12-06 18:48:35 Mounting options are functional
2017-12-06 17:10:14 HUGE CHANGES. Buffer
2017-12-06 01:07:00 Partial : binded wizard with the core (Direct Hidden creation part) + Added sliders to choose the inner/outer volume relative size + Improved wizard variable management & display
2017-12-05 23:14:58 Delete CoreRequest.cpp.orig
2017-12-06 00:05:45 Move changeVolumePassword to CoreBase
2017-12-05 23:52:03 Add filesystemtype in exception FailMountFilesystem to help debugging #59
2017-12-05 23:35:09 Add ProgressTracking within creation volume sub methods as an example #60
2017-12-05 19:59:39 Close #57 Develop CoreRoot::changeVolumePassword
2017-12-05 16:54:13 Added task update when creating a volume
2017-12-05 16:44:46 Add comments in Volume.h

2017-12-05 16:44:16 Fix valgrind error request dump file now closed correctly
2017-12-05 16:43:02 Solve Uninitialized data in CoreRequest (requestId)
2017-12-05 16:28:16 Some Linux desktops allow by right-clicking on the taskbar to display a full-screen window of fixed size: this commit makes changes so that gostcrypt displays correctly regardless of the size of the window.
2017-12-05 16:26:56 Partial: binding the wizard with the core. It is now possible to create a volume. Only the Whirlpool hash works. The volumes created by the wizard cause an exception when you mount them (idem cli)
2017-12-05 13:29:33 Partial: binding wizard with core
2017-12-05 00:25:20 Adapt for change in Keyfilelist (Qlist instead of std::list)
2017-12-05 00:24:50 Clean VolumePassword
2017-12-05 00:21:25 Replace std::list by QList for KeyFileList
2017-12-05 00:20:55 Remove unnecessary code in VolumePassword
2017-12-04 17:34:30 replace every typedef gst_word
2017-12-04 17:31:08 include HAVE_FSTATAT
2017-12-04 16:31:20 remove a useless function
2017-12-04 16:19:23 remove the function readlink_at
2017-12-04 15:17:21 remove the union of the structure WorkItem
2017-12-04 15:15:33 make the functions of Test static
2017-12-04 15:14:39 make the functions of deserialization and serialization static
2017-12-04 15:09:54 make the functions of EncryptBuffer and DecryptBuffer static
2017-12-04 14:32:33 Remove Endian from Memory.h
2017-12-04 10:53:49 Finish Exception Handling + Debug
2017-12-04 10:53:17 Close #27 Develop System Logging in FuseService
2017-12-04 10:45:37 reorganize the source code and move all files from Common/ and Crypto/ to Volume/Crypto + Remove GfMul.[ch]
2017-12-04 10:07:46 remove the function readlink_at
2017-12-04 10:06:58 remove functions mul_x and mul_x64
2017-12-04 10:05:38 remove functions loopcxt_get_backing_devno
2017-12-03 22:12:02 Finish handling exceptions
2017-12-02 18:35:07 Binding notification for 'Generate Keyfile' action
2017-12-02 18:34:32 Final : dynamic subwindow height
2017-12-02 17:39:45 Close #26
2017-12-01 16:21:25 remove some useless functions
2017-12-01 16:07:50 Binding Generate Keyfile subwindow
2017-12-01 15:41:35 remove the functions sysfs_get_devname
2017-12-01 15:39:11 remove the function crc32_selftests
2017-12-01 15:37:35 remove the functions loopcxt_find_by_backing_file
2017-12-01 15:36:04 remove the functions TestCiphers
2017-12-01 15:34:53 remove the functions Gf128MulBy64Tab
2017-12-01 15:33:56 remove the functions GetKey
2017-12-01 14:52:29 Using SecureTextField QML component in the wizard
2017-12-01 14:44:17 Fixed the problem that prevented opening a volume mounted on the IU. Display when the keyboard is in uppercase mode when entering a password Updating translation files
2017-12-01 13:27:02 remove the exception e after the throw
2017-11-30 18:18:41 remove the variable KeySwapped
2017-11-30 18:17:42 initialize the variables in the constructor of VolumeLayout
2017-11-30 18:17:08 initialize the variables in the constructor of Volume
2017-11-30 18:15:54 check the member of the construcor
2017-11-30 18:14:12 remove the variable ch
2017-11-30 17:27:35 remove the structure EncryptionAlgorithm
2017-11-30 17:26:41 remove some useless functions
2017-11-30 17:26:28 remove some useless functions
2017-11-30 16:45:23 remove functions get_pkcs5_prf_name and get_pkcs5_iteration_count
2017-11-30 16:44:30 remove functions is_power_of_2 and get_hostname_max
2017-11-30 16:43:33 remove functions crc32int and crc32_selftest
2017-11-30 16:42:18 remove the function canonicalize_path_restricted
2017-11-30 16:40:55 remove functions is_blkdev
2017-11-30 16:16:44 remove functions ValidateDataParameters and ValidateDigestParameters
2017-11-30 16:16:00 remove the function TestAll
2017-11-30 16:15:15 remove the function SetHeader

2017-11-30 16:08:04 remove Endian.c and Edian.h because they are useless
2017-11-30 16:02:00 remove functions GfMul128
2017-11-30 15:44:40 remove functions GetEncryptionMode
2017-11-30 15:31:06 remove functions SetSize
2017-11-30 15:23:59 remove functions GetMaxBlockSize
2017-11-30 15:22:36 remove functions GetHash
2017-11-30 15:08:34 remove functions GetAvailableModes
2017-11-30 14:43:10 remove the functions DecryptBlock
2017-11-30 14:19:38 remove functions CheckProbability and IsEmpty
2017-11-29 18:06:37 Reorganize Exceptions with GostCryptException + Add requestId in all exceptions
2017-11-29 18:03:19 remove the const at the end of the declaration of the function an put a static at the beginning of it.
2017-11-29 18:00:55 put a this before the variable preserveTimestamps
2017-11-29 17:59:49 initialzation of the variables in the constructor VolumeFile
2017-11-29 17:41:05 Managing the Keyfiles when mounting/creating a gostcrypt volume
2017-11-29 17:28:23 Remove the File Password.h because it is useless
2017-11-29 17:20:32 Remove the declaration of the structure UINT64_STRUCT and replace every initialisation of this structure by quint64.
2017-11-29 15:09:22 initialize the variable isDevice in the contructor MountVolumeRequest in CoreRequest.cpp
2017-11-29 14:31:11 Remove PLatform/FilePath and Volume/VolumePath replaced by QFileInfo and new methods in VolumeFile #16 #48
2017-11-29 14:23:33 reduce the scope of the variable workItem in EncryptionThreadPool.cpp
2017-11-29 14:19:47 reduce the scope of the variable workItem in EncryptionThreadPool.cpp
2017-11-29 13:40:27 check if mountpoint is defined before accessing to it
2017-11-29 13:21:01 Replace remaining std::list by QList
2017-11-29 12:00:06 test to pass the function DecryptBufferXTS8Byte and EncryptionBufferXTS8Byte in static but it is better to keep const
2017-11-29 11:57:18 make the constructor SecureTextField explicit in SecureTextField.h
2017-11-29 11:54:52 initialization of the pointer mGI to nullptr in GrphicInterface.h
2017-11-29 00:34:03 Deleted unnecessary files + Fix compilation warnings + Add missing header file from common in Volume project
2017-11-28 23:15:41 Remove Platform/File. We now use VolumeFile only for Volume file access (kept for pread and pwrite more efficient for Random access
2017-11-28 18:29:44 Replace sharedValue by QAtomicInteger + Remove Platform.h + Replace std::list by QList
2017-11-28 18:04:50 Precise the informations sent by ParseException as a QString
2017-11-28 16:59:37 Debug from new-volume
2017-11-28 16:57:07 VolumeLayouts are now separed in different files. Every file equals one class now. CLOSING #7
2017-11-28 16:54:57 make the constructors ProgressTrackingParameters explicit in CoreRequest.h
2017-11-28 16:40:29 reduce the scope of the variable tmp in Stribog.c
2017-11-28 16:36:28 VolumeLayoutV1 is now removed completely. closing #20
2017-11-28 16:32:49 reduce the scope of the variables sum and add_bytes in GostHash.c
2017-11-28 16:30:09 Cipher.cpp should be CipherAlgorithm.cpp
2017-11-28 16:24:14 Renaming Hash functions to VolumeHash##name. #7
2017-11-28 15:36:40 Add debug feature imported from new-volume
2017-11-28 15:33:20 Stribog
2017-11-28 15:29:00 Remove n1=0 and n2 = 0 at the beginning of the functions gost_encrypt & gost_decrypt in GostCipher.c
2017-11-28 15:14:19 clarify Calculation in loopdev.c
2017-11-28 15:01:04 reduce the scope of the variable p in canonicalize.c
2017-11-28 14:48:28 remove the function blkdev_get_geometry in blkdev.c because it is never used
2017-11-28 14:42:22 remove the structure hd_geometry because it is never used
2017-11-28 14:37:11 Restore Header Volume
2017-11-28 14:20:47 replace the postfix operators ++ in prefix in CoreBase.cpp
2017-11-28 14:17:04 merged Pkcs5 with VolumeHash and removed Pkcs5. close #11.
2017-11-28 13:15:08 Adding confirmation message for backup header volume
2017-11-28 11:52:02 Backup volume header
2017-11-28 10:00:49 Partial : Backup Volume Header
2017-11-28 08:51:17 correction of the overlapping animation in the homeframe

2017-11-27 23:16:13 Develop debug features: Service Handler now export a dumpFile allowing to replay the child processes in debug mode
2017-11-27 23:15:13 Debug FuseService
2017-11-27 23:14:11 Add a small test volume (password: a)
2017-11-27 16:22:20 make the constructor Service explicit in Service.h
2017-11-27 16:20:36 make the constructor ProgressTrackingParameters explicit in CoreRequest.h
2017-11-27 16:09:59 make the constructor VolumeHeader explicit in VolumeHeader.h
2017-11-27 15:58:15 make the constructor Keyfile explicit in Keyfile.h
2017-11-27 15:46:33 make the constructor VolumePassword explicit in VolumePassword.h
2017-11-27 15:30:48 Ciphers are now separed in multiple files. #7
2017-11-27 15:29:21 put the variable Data in an Initializer for VolumePath (line 28 & 29) in Volume.h
2017-11-27 15:27:38 remove the structures Hash and Cipher in Crypto.h because they are never used
2017-11-27 14:36:16 Remove Platformtest
2017-11-27 14:34:30 Changed CipherList to CipherAlgorithmList (#7) and solved a compilation error with #define <typeinfo>
2017-11-27 14:20:43 put the variable Data in an Initializer for VolumePath (line 28 & 29) in Volume.h
2017-11-27 12:42:30 Possibility to send signals when clicking on a button in the menu that does not open a subwindow
2017-11-27 11:52:59 change the cstyleCast in C++stylecast at line 45 in Thread.h
2017-11-27 10:46:04 make the constructor SystemException explicit in SystemException.h
2017-11-27 10:39:12 make the constructor Exception explicit in Exception.h
2017-11-27 10:30:49 make the constructor Serializer explicit in Serializer.h
2017-11-26 23:35:43 New welcome window for Gostcrypt The display has been improved to make it more attractive
2017-11-26 02:53:16 Remove unnecessary files in Platform
2017-11-25 23:44:00 Finish implementing new EncryptionThreadPool using Qt technos Fixed #23
2017-11-25 19:22:37 Remove many files from platform and fix the code in order not to use it anymore. WARNING: Nothing had been tested
2017-11-25 18:08:44 Fix return statement missing in Volume::Volume::getVolumeInformation
2017-11-25 18:00:47 Remove many files from platform and fix the code in order not to use it anymore. WARNING: Nothing had been tested
2017-11-24 16:29:05 Fix mountPoint crash on starting gostcrypt if a volume was badly mounted
2017-11-24 16:20:16 GST_EXCEPTION_DECL explicit function
2017-11-24 16:18:04 Fixing explicit problems
2017-11-24 15:56:49 parameter keyfileIdFilter passed by reference in SecurityToken.cpp
2017-11-24 14:57:10 make these constructors explicits: SecurityTokenKeyfilePath
2017-11-24 14:22:43 make the constructor ScopeLock explicit in Mutex.h
2017-11-24 14:16:56 make the constructors Buffer & SecureBuffer explicit in Buffer.h
2017-11-24 14:01:23 make the constructors ContainerForward & ContainerReverse explicit in ForEach.h
2017-11-24 13:56:32 Replace shared_ptr by QSharedPointer
2017-11-24 13:52:29 make the functions SharedPtr::operator== && SharedPtr::operator=! const in SharedPtr.h
2017-11-23 18:09:22 EncryptionModeLRW now deleted too. closing #12
2017-11-23 18:07:03 EncryptionModeCBC is now deleted. #12
2017-11-23 18:02:49 fix #13. Cipher is now renamed to CipherAlgorithm
2017-11-23 17:50:17 EncryptionAlgorithms are now renamed in a better way and separated in different files. #7
2017-11-23 15:43:41 solving compilation errors : File missing
2017-11-23 14:51:18 Removing goto statements fixing #8.
2017-11-16 19:24:00 Add Volume namespace + Fix #18
2017-11-16 17:14:47 Fix #4. Merging the two Volume::Open function since one was useless.
2017-11-16 16:45:28 Removing a few references to LayoutV1 in Volume::Open(). Ref #20
2017-11-16 16:08:35 Adapt .gitignore in build/ for new module names
2017-11-16 15:50:59 Remove unnecessary code used for SystemEncryption. Refs: #19
2017-11-15 18:04:07 Fix menu element size
2017-11-15 18:00:25 Now displaying the notification 5 seconds before it disapear when the GUI is receiving percent updates
2017-11-15 17:21:19 Subwindow 'test vectors'
2017-11-14 18:38:30 Rename NewCore to Core and NewFuseService to FuseService
2017-11-14 17:28:36 Write doc for NewFuseService
2017-11-14 16:38:41 find a false positive warning in GfMul.h (struct used in GfMul.c)
2017-11-14 16:26:43 reduce the scope variable 'value' in Endian.c and 'xx' in GfMul.c
2017-11-14 15:35:47 merge and qdatastream

2017-11-13 12:20:27 license
2017-11-12 19:56:29 About subwindow
2017-11-12 16:31:44 Subwindow Preferences
2017-11-12 01:24:17 Managing user favorite volumes with options
2017-10-30 16:41:33 Fix Core communication not working after first response received
2017-10-30 16:35:35 Fix GostCrypt GUI not launched in debug mode
2017-10-28 21:52:30 Final : Ladies and Gentlemen
2017-10-28 21:24:21 Partial : new wizard
2017-10-28 20:00:45 Partial: new wizard
2017-10-28 18:02:22 Partial : new wizard
2017-10-28 01:01:06 Partial : creating the new wizard
2017-10-27 18:42:35 Partial : improving text
2017-10-27 17:48:41 Autofocus with sudo window
2017-10-27 17:38:50 Autofocus the password textfield
2017-10-27 16:24:14 Press enter to validate the mount volume form
2017-10-27 15:50:27 Now using macro to correctly defining signals
2017-10-27 13:46:58 Fix service errorChannel displayed
2017-10-27 12:32:23 Fix Volume mounted not displayed
2017-10-25 16:32:18 testing new platform
2017-10-24 23:39:47 Partial Add Volume namespace
2017-10-24 18:01:34 Adding a description for all the commands when calling 'GostCrypt --help'
2017-10-24 17:37:52 Task system : finished !
2017-10-24 17:35:44 Few text improves for comand line interface
2017-10-24 16:50:50 Solving a bug where the help was not showed
2017-10-24 15:12:33 Fix exception in MountVolume not leaving in a clean state
2017-10-24 15:11:59 Partial: Improve Progress tracking
2017-10-24 15:06:20 Partial : binding tasks with core
2017-10-20 14:20:51 Fix ProgressUpdateResponse not frwarded properly between processes
2017-10-14 15:31:11 Fixing tooltip issue
2017-10-14 15:11:47 Partial (notification system): creating the signals/slot to bind GI with Core and QML
2017-10-13 00:38:59 Fix signal sendProgressUpdate missing & VolumeALreadyMounted exception always thrown
2017-10-13 00:31:18 Manual Merge with new-fuse-driver
2017-10-13 00:16:22 Notification system creation
2017-10-12 23:00:48 Creating the notification system
2017-10-12 10:50:18 adding the progress function system
2017-10-10 20:07:10 Add macros to help setting progress tracking
2017-10-10 16:17:10 Partial: ProgressTracking developed in Core
2017-10-05 15:34:30 bind device hosts
2017-10-05 10:40:06 Create default construtor for MountVolumeRequest TODO: Create default constructor for all requests
2017-10-04 01:23:55 Fix project path in build.sh
2017-10-04 00:46:49 Rename source dirs (remove kamelcase for quicker navigation)
2017-10-04 00:45:13 Remove GostCrypt_ prefix in sub-project's names
2017-10-03 22:59:17 Fix compilation warnings in NewCore/loopdevlib/
2017-10-03 22:18:13 Create art dedicated directory
2017-10-03 22:16:24 Fix paths in build.sh & runtest.sh
2017-10-03 22:13:41 Rename src dirs
2017-10-03 22:12:01 Reorganiza build dir
2017-10-03 22:04:16 Cleaning unnecessary old files + Reorganization
2017-10-03 21:47:17 Create gitignore for test logs
2017-10-03 21:46:35 Add cleaning in runtests.sh
2017-10-03 21:44:02 Add root check in runtests.sh
2017-10-03 21:40:52 Clean test results files
2017-10-03 21:40:01 Fix remaning absolute path in runtests.sh
2017-10-03 21:39:26 Fix option for mounting mountForGroup not working
2017-10-03 21:38:53 Fix build steps not in order
2017-10-03 19:17:32 Adding test script
2017-10-03 19:12:58 adding scripts directory
2017-10-03 18:59:10 Change Core to asynchronous + clean unecessary includes of old Core

2017-09-27 16:26:32 Clean old files coming from GostCryptV1
2017-09-27 16:24:43 Clean old files coming from GostCryptV1
2017-09-26 19:05:10 Fix build steps
2017-09-26 17:08:16 Documentation for new developer dev environement
2017-09-26 17:03:14 Bug correction. Added -u and -g for user and group ownership when mounting a volume (WORKING)
2017-09-26 16:22:31 Fix Volume creation with GUI
2017-09-26 16:16:46 Change .pro
2017-09-26 15:25:10 Correct build steps
2017-09-26 14:25:55 wizard fix
2017-09-26 14:18:23 Comments CoreBase.h
2017-09-26 11:44:00 fixing some errors
2017-09-26 11:36:42 Remove references to previous Core in .pro + Change position of DEBUG_CORESERVICE_HANDLER define
2017-09-26 00:49:22 binding the wizard
2017-09-25 23:14:22 binding the wizard
2017-09-23 19:59:13 Binding (now the new-core is binded in the same way as the old one)
2017-09-23 19:11:16 Add missing newline character at the end of exception messages
2017-09-23 19:09:24 binding
2017-09-23 19:08:43 Add exception IncorrectVolumePassword
2017-09-23 18:29:14 binding
2017-09-23 18:08:53 Fix CLI not adapted to changement made in Core for the GUI
2017-09-23 18:08:53 binding
2017-09-23 18:08:53 Add list of volumePath succesfully dismounted in DismountVolumeResponse
2017-09-23 18:08:53 binding
2017-09-23 18:08:36 Add list of volumePath succesfully dismounted in DismountVolumeResponse
2017-09-23 18:07:25 binding
2017-09-23 18:07:19 Add list of volumePath succesfully dismounted in DismountVolumeResponse
2017-09-23 18:06:18 binding
2017-09-23 18:06:18 Binding
2017-09-23 18:06:18 Add list of volumePath succesfully dismounted in DismountVolumeResponse
2017-09-23 18:06:18 Binding the new core
2017-09-23 18:06:18 New core and new UI binding (not finished yet)
2017-09-23 18:06:18 First branch commit
2017-09-23 18:06:02 Fix CLI not sinishing when argument missing
2017-09-23 17:58:00 binding
2017-09-23 17:57:06 Fix CLI not adapted to changement made in Core for the GUI
2017-09-23 17:48:18 binding
2017-09-23 17:47:12 Add list of volumePath succesfully dismounted in DismountVolumeResponse
2017-09-23 17:43:21 binding
2017-09-23 17:43:21 Binding
2017-09-23 17:43:21 Add list of volumePath succesfully dismounted in DismountVolumeResponse
2017-09-23 17:43:21 Binding the new core
2017-09-23 17:43:21 New core and new UI binding (not finished yet)
2017-09-23 17:43:21 First branch commit
2017-09-23 17:25:39 Fix CLI not working
2017-09-23 16:09:43 binding
2017-09-23 16:09:43 Binding
2017-09-23 16:09:43 Add list of volumePath succesfully dismounted in DismountVolumeResponse
2017-09-23 16:09:43 Binding the new core
2017-09-23 16:09:43 New core and new UI binding (not finished yet)
2017-09-23 16:09:43 First branch commit
2017-09-23 14:55:52 binding
2017-09-22 15:53:13 Binding
2017-09-22 13:24:12 Add list of volumePath succesfully dismounted in DismountVolumeResponse
2017-09-18 20:40:22 Binding the new core
2017-09-15 00:35:45 New core and new UI binding (not finished yet)
2017-09-13 23:37:52 debug
2017-09-13 23:31:20 Add feature: new params when mounting: mountForUser and mountForGroup

2017-09-13 17:51:53 forgot what i did
2017-09-13 17:35:38 dismount now cleans all the bad mounted volumes.
2017-09-13 16:13:04 bug soling in parser parseSize()
2017-09-13 15:19:58 replacing QStrings with chars for debian stretch
2017-09-13 15:14:05 transforming char to Qstring...
2017-09-13 15:11:57 transforming int to char for compilation under debian stretch
2017-09-13 15:02:53 Bug solved. All testcases are now working.
2017-09-11 20:13:00 working
2017-09-09 22:25:09 First branch commit
2017-09-02 13:57:27 Rename all CoreParams to CoreRequest (Need to execute qmake and recompile whole project)
2017-09-02 13:45:08 Simplify Exception transfer by including exception in ExceptionResponse object
2017-08-14 16:24:28 Fix serialization issues causing exception not to propagate to main process
2017-08-14 15:50:05 Fix serialization issues
2017-08-14 01:06:41 Remove getFilesystemTypeSupported which was wrong and became pointless
2017-08-14 01:05:56 Develop getEncryptionAlgorithms
2017-08-14 00:39:53 Now using blkidlib to automatically detect filesystem type of the volume before mounting
2017-08-13 23:33:33 Communication with worker process now working with all requests
2017-08-13 18:53:57 Exception handling across processes done :D
2017-08-12 21:30:49 Debug inter-process communication
2017-08-12 00:38:31 Improve exception handling when the worker process crash (was detected as IncorrectPassword before)
2017-08-11 23:31:47 Fix warnings / Add IncorrectSudoException / Fix some serialization issues (still issues left to solve)
2017-08-11 17:28:50 Changing the CLI architecture. Pushing only to commit with Antoine's changes to have a working build.
2017-08-11 17:23:35 Integrate as into main project
2017-08-11 17:22:59 Changing the CLI architecture. Pushing only to commit with Antoine's changes to have a working build.
2017-08-11 15:41:07 adding createkeyfiles
2017-08-11 15:24:52 Improve as to support exceptions
2017-08-11 14:38:18 Improve mountVolume to support all filesystemTypes
2017-08-11 13:16:59 bug solving
2017-08-11 10:19:51 Change dismountVolume to allow to dismount all volumes
2017-08-11 01:40:11 Finish final version of as test project
2017-08-10 16:53:29 adding list hashs and list algorithms
2017-08-10 15:44:56 bug corrections
2017-08-09 18:13:57 createkeyfile function added.
2017-08-09 17:50:20 Cleaning code
2017-08-08 17:52:38 mounting
2017-08-08 15:01:34 Remove no more necessary convertion from QSharedPointer to shared_ptr
2017-08-08 14:58:09 CHANGING ALL SHARED_PTR TO QSHARED_PTR
2017-08-08 12:04:10 Transforme as tets project into Request/response communication
2017-08-08 01:03:33 Transformation of as test project to asynchrone
2017-08-07 18:38:59 Fix some errors
2017-08-07 17:49:09 Progres made in CoreUser / Need to think about new method
2017-08-03 18:57:07 Improve Exception handling && start CoreUser
2017-08-03 18:32:58 import test project
2017-08-03 16:15:00 DismountVolume Done !
2017-08-03 13:32:54 working on volumecreator
2017-08-03 02:16:11 CoreRoot::mountVolume Done
2017-08-02 23:37:08 Debug many things
2017-08-02 17:35:31 Fix compilation issue (undefined references) due to bad libraries order when linking
2017-08-02 17:24:12 Remove duplicated files in Platform subdir
2017-08-02 17:16:35 Add missing files in Platform subdir
2017-08-02 16:54:21 Finish MountFilesystemManager and create all necessary Exceptions
2017-08-02 15:05:14 working on createvolume
2017-08-02 13:44:41 Start Mountmanager and loopdevicemanager
2017-08-01 18:19:48 async
2017-08-01 14:14:15 working on the creator
2017-08-01 14:12:55 fix

2017-08-01 13:49:00 Change QFileInfo to QSharedPointer<QFileInfo> in CoreBase
2017-08-01 11:47:30 Correct compilation errors
2017-07-31 14:09:11 working on creator
2017-07-30 23:20:46 progress
2017-07-28 10:56:57 IsVolumeMounted Done + VolumeALreadyMounted and MissingParam Exceptions Done
2017-07-28 10:06:45 Remove some warnings
2017-07-28 09:47:13 GetFileSystemsTypesSupported Done
2017-07-27 16:18:50 comments
2017-07-27 16:17:35 Improve CoreException && Debug getHost Devices
2017-07-27 11:53:57 Fixing 'password' in openVolume
2017-07-26 17:04:54 adding getsupportedfilesystems() to corebase.h
2017-07-26 16:51:12 Replacing std::cout with qStdout()
2017-07-26 16:26:30 removing filesystemtype enumeration
2017-07-26 14:12:50 Finish GetMountedVolumes method
2017-07-26 11:50:18 working on the parser.
2017-07-26 11:50:18 Finish getMountedVolumes method to be tested
2017-07-26 11:38:39 Volumes favorites in QSettings
2017-07-26 10:30:26 Add Core project to meta-project to easily access revious Core
2017-07-26 10:29:58 getMountedFilesystems done
2017-07-26 10:26:09 cleaning CoreParams.cpp and CoreResponse.cpp from useless namespaces definitions
2017-07-26 10:18:25 improving enums with QString list for indexof
2017-07-26 09:56:13 Work on getMountedFileSystems
2017-07-25 22:42:47 Adding shortchut button in each volume item
2017-07-25 17:51:18 Solved bug when debugging the GUI
2017-07-25 17:43:12 prevent crash
2017-07-25 17:18:12 bug solving
2017-07-25 17:05:28 Add CoreException and change operator as friend functions
2017-07-25 15:00:38 Dynamic translation
2017-07-25 14:45:44 adding cmdUserInterface
2017-07-25 14:10:30 fusion part2
2017-07-25 13:59:54 Created Core files
2017-07-24 18:07:30 Bug resolution
2017-07-24 16:12:24 Improving the submenu
2017-07-24 09:50:57 Bug solved : UI.depends = UI
2017-07-21 18:05:14 Finish CoreBase
2017-07-21 17:10:35 Adding the SubMenu and filling
2017-07-21 16:16:44 adding parser for command-line interface
2017-07-21 16:00:39 Creation of NewCore Poject and Params and Response structures
2017-07-21 12:37:45 Choice of device
2017-07-21 11:37:15 Getting the device list from C++ into QML openDialog
2017-07-20 15:46:59 clean
2017-07-20 15:38:44 Creation of the branch
2017-07-20 15:32:08 Adding the random mouse position generator
2017-07-20 14:52:22 Wizard dynamic text & error message is now closing faster
2017-07-20 14:12:14 Improving wizard implementation & documentation
2017-07-19 16:15:07 Adding the translation files (to be filled with QtLinguist)
2017-07-19 15:53:14 Fixing password verification (wizard) and the 'format' button
2017-07-19 15:25:05 Fix file missing and previous bug
2017-07-19 15:11:36 Remove no more used gitignore
2017-07-19 15:09:55 MountVolume is no longer called twice
2017-07-19 15:09:55 Work in progress
2017-07-19 14:03:23 Fix static librairies not relinked after updated with new build system
2017-07-18 16:42:26 Update .gitignore
2017-07-18 16:27:21 Update Contributors list
2017-07-18 16:25:42 Clean unnecessary files
2017-07-18 16:07:02 Debug new build system

2017-07-18 15:03:09 Rename and improve pro file for UI
2017-07-18 14:16:35 Clean unecessary files
2017-07-18 14:13:22 New Build system using qmake for remaining libraries
2017-07-18 10:26:09 MountVolume is no longer called twice
2017-07-18 09:56:43 Add description of old Gostcrypt CLI
2017-07-18 09:56:43 Remove unused files
2017-07-18 09:56:43 Move method in pipe
2017-07-18 09:56:43 Patching the sudo dialog
2017-07-18 09:56:43 Add volume correctly when mounted
2017-07-18 09:56:43 Fix bug in CoreLinux::AttachFileToLoopDevice
2017-07-18 09:56:43 Rearagnge source code in GraphicUserInterface::receiveMount
2017-07-18 09:56:43 Fix static library not linked again after updated during compilation
2017-07-18 09:56:43 Change file rights
2017-07-18 09:56:43 Attempt to fix volume not diplayed after successfully mounted
2017-07-18 09:56:43 Add comments in VolumeInfo.h about VirtualDevice property
2017-07-18 09:56:43 Remove check for already mounted Volume in GraphicUserInterface::receiveMount since it's done later in GostCrypt::Core->MountVolume
2017-07-18 09:56:43 Add comments in CoreUnix.cpp for method GetMountedVolumes
2017-07-17 15:05:36 Add description of old Gostcrypt CLI
2017-07-17 14:21:06 Remove unused files
2017-07-17 14:17:24 Move method in pipe
2017-07-17 12:11:35 Patching the sudo dialog
2017-07-17 11:55:29 Add volume correctly when mounted
2017-07-17 11:06:24 Work in progress
2017-07-13 14:12:21 Fix bug in CoreLinux::AttachFileToLoopDevice
2017-07-13 11:45:27 Rearagnge source code in GraphicUserInterface::receiveMount
2017-07-13 11:03:21 Fix static library not linked again after updated during compilation
2017-07-13 11:01:10 Change file rights
2017-07-12 20:49:01 Attempt to fix volume not diplayed after successfully mounted
2017-07-12 20:48:04 Add comments in VolumeInfo.h about VirtualDevice property
2017-07-12 20:46:50 Remove check for already mounted Volume in GraphicUserInterface::receiveMount since it's done later in GostCrypt::Core->MountVolume
2017-07-12 20:45:15 Add comments in CoreUnix.cpp for method GetMountedVolumes
2017-07-12 13:55:51 Solve Fuse Bad mount point issue :D
2017-07-11 13:49:39 qml error ?
2017-07-11 13:36:38 working on volume creation
2017-07-10 16:16:19 Reset source_linux
2017-07-10 13:52:50 fast modif in GSOpenVolume.qml
2017-07-05 01:36:56 Improving the subwindow
2017-07-04 03:27:19 Adding a C++ class that manages the volume creation wizard. The new folder (UI/Wizard) contains all the wizard's frames. Improving the subwindow. Filling the first three pages of the wizard
2017-07-03 19:27:07 Adding the volume tools button menu (with signals)
2017-07-03 18:20:40 Adding frames and filling the VolumeFrame.qml file
2017-07-03 16:23:30 The menu is now functional and each frame has its .qml file
2017-07-03 02:18:03 - Reorganization of the QML interface architecture: the main.qml file loads the content from another file - Improved interface: frameless window (+ C++ class that manages the dragging of the window)
2017-07-02 00:14:30 Fixing an index problem in the addVolumePath function
2017-07-01 23:54:55 Drag and drop management to quickly build a volume
2017-07-01 20:29:30 Improvement of the sudo window and the custom ComboBox
2017-07-01 19:28:15 Using a system file (QSettings) to save user preferences
2017-06-30 17:15:00 Add comments and try to debug Fuse
2017-06-30 01:51:17 test new git server
2017-06-28 16:24:18 Adding ERIC FILLIOL's scex41 files to try out Rust.
2017-06-28 15:07:07 Fixing GostCrypt issues
2017-06-28 15:00:33 Update file descriptions
2017-06-27 17:21:10 Cleaning
2017-06-27 17:12:23 Remove unused VolumeAlert files
2017-06-26 12:52:30 Add wallpaper for project presentation
2017-06-26 12:40:35 Improving QML architexcure and comments

2017-06-26 11:33:42 modifs salon
2017-06-14 23:56:38 Adding menu (logo and buttons)
2017-06-14 18:31:40 Debug sudo
2017-06-14 18:14:05 Unshow the volumes on click on the dismount all button
2017-06-14 18:12:30 Debug Red coloration of Password
2017-06-14 17:21:41 Binding & password clear/hide
2017-06-14 17:20:52 Correct bug when volume password is incorrect
2017-06-14 17:02:53 Adding change_password link. (not yet linked but now it exists)
2017-06-14 17:01:39 Debug sudo subWindow not closed
2017-06-14 16:00:31 Update .gitignore files
2017-06-14 16:00:02 Rename old interface dir
2017-06-14 15:56:34 Delete old QML interface project dir
2017-06-14 15:55:02 Comment CoreService class
2017-06-14 15:50:52 Fix last librairies compilation warning about unused u flag when linking librairies
2017-06-14 15:48:46 Add note about what to look at about Rust language
2017-06-14 15:25:14 debugdebug
2017-06-14 15:22:56 Dismount button
2017-06-14 14:56:50 Fixing path
2017-06-14 14:45:21 debugging..
2017-06-14 14:39:03 Fixing font issues and open volume path
2017-06-14 13:36:39 committing to get new changes
2017-06-14 13:34:30 Fix issue
2017-06-14 13:32:30 Binding the VolumeInfo GostCrypt object with the QML application
2017-06-14 11:27:40 adding debug links for volume creation.
2017-06-13 18:04:29 Adding the VolumeItem Component
2017-06-13 14:33:47 Correct binding + implementation of sudo password subwindow
2017-06-13 13:30:21 forgot to include .h for VolumeCreationOptions. Bug fixes.
2017-06-13 13:17:24 Added signal for volume mounting (not tested).
2017-06-13 10:49:19 uncommenting
2017-06-13 01:40:07 Binding the QML interface with the GostCrypt library
2017-06-12 20:08:54 Attempt to connect QML to GraphicUserInterface
2017-06-12 17:26:22 integrating the new interface
2017-06-12 16:25:39 Cleaning the makefile from all the dependencies
2017-06-08 17:18:34 Corrected many bugs while privilege escalation (Not Working yet)
2017-06-08 16:56:39 linking mounting and umounting with the old interface
2017-06-08 11:17:51 modified Gostcrypt.pro to handle multiple versions of QtCreator
2017-06-08 11:03:49 adding c++11 option for compilation in .pro for QtCreator.
2017-06-08 10:50:01 Add project user file in gitignore
2017-06-07 15:32:00 ++
2017-06-07 15:31:33 Debug broken pipe error when opening volume due to CoreService::Stop called to early.
2017-06-06 15:49:06 Solve linking error with crypto inline methods
2017-06-06 15:00:14 Solve linking problems due to commented code
2017-06-06 14:49:19 Creation of .gitignore files for INterface QML project
2017-06-06 14:47:14 Change Makefile for debug compilation by default
2017-06-06 14:46:02 Comment inline function declaration in GrasshooperCipher.h for compatibility with GCC5+
2017-05-12 17:33:24 Debug attempt
2017-05-12 17:33:01 Adding the folder that contains the GostCrypt QML version (UI)
2017-05-06 18:19:30 Change logo to transparent
2017-04-18 22:23:54 Ajout du fichier logo en hd
2017-03-29 02:24:59 Developpement du slot receiveMount + fonction d'initialisation de Core
2017-03-29 01:15:03 Adding a dismount button
2017-03-28 13:16:15 Fix issue
2017-03-28 12:08:44 Adding signal to create a new volume
2017-03-19 20:57:34 Reorder attribute in Window.h to fix warnings
2017-03-19 19:44:13 Fix pro file by removing not compiling wizard files
2017-03-19 18:29:09 Changement du nom du dossier pour Qt
2017-03-19 18:28:32 Fix aprÃ¨s renommage
2017-03-19 18:18:01 Application des renommages

2017-03-19 15:53:38 Renommage
2017-03-18 22:08:53 adding wizard part
2017-03-18 21:52:44 adding the missing 'delete'
2017-03-18 21:41:16 Adding a qDialog to load a volume file
2017-03-18 21:37:31 Adding a qDialog to load a volume file
2017-03-18 17:21:05 Update Makefile for compilation
2017-03-18 17:13:01 Adding new class for slots
2017-03-18 15:31:39 Removing unused files
2017-03-18 15:09:10 Debug volumelist
2017-03-18 14:57:09 Add build dir in gitignore file
2017-03-18 14:53:47 Move QtInterface
2017-03-18 14:52:07 Delet build dir
2017-03-18 14:50:06 Clean Visual Studio Files in Qt project
2017-03-18 14:47:07 Delete user project file
2017-03-18 14:46:45 Clean binary path for Qt project
2017-03-18 14:32:08 Ajout projet avec QT
2017-03-18 14:30:32 .gitignore windows
2017-03-12 12:21:43 Adding the initial QT project (new UI)
2017-03-07 12:01:14 Importation du projet Visual Paradigm
2017-02-28 09:56:39 moving sources to sources_windows
2017-02-28 00:13:57 Supression de l'executable et ajout dans le fichier gitignore
2017-02-28 00:08:01 Correction de l'erreur narrowing lors du chargement des xml utilisation de wstring au lieu de string
2017-02-27 23:35:20 Modification du makefile pour compilation sur ubuntu
2017-02-27 22:11:34 AmÃ©lioration des fixs Bug non rÃ©solu (fuse: bad mount point `': Aucun fichier ou dossier de ce type)
2017-02-27 21:34:21 Importation des source linux aprÃ¨s correction des erreurs qui empechaient la compilation
2017-02-07 15:27:09 Changement des droits
2017-02-07 11:18:37 Importation

Rapport CppCheck

Ce rapport présente l'ensemble des erreurs reportées après l'audit CppCheck.

Resolved & verified:

Problem of « unusedStructMenber » :

/gostcrypt2_src/Common/Crypto.h:116:unusedStructMember:struct member 'Cipher::Id' is never used.'
'/gostcrypt2_src/Common/Crypto.h:117:unusedStructMember:struct member 'Cipher::Name' is never used.'
'/gostcrypt2_src/Common/Crypto.h:118:unusedStructMember:struct member 'Cipher::BlockSize' is never used.'
'/gostcrypt2_src/Common/Crypto.h:119:unusedStructMember:struct member 'Cipher::KeySize' is never used.'
'/gostcrypt2_src/Common/Crypto.h:120:unusedStructMember:struct member 'Cipher::KeyScheduleSize' is never used.'
Remove the structure Cipher because it is never used.

'/gostcrypt2_src/Common/Crypto.h:132:unusedStructMember:struct member 'Hash::Id' is never used.'
'/gostcrypt2_src/Common/Crypto.h:133:unusedStructMember:struct member 'Hash::Name' is never used.'
'/gostcrypt2_src/Common/Crypto.h:134:unusedStructMember:struct member 'Hash::Deprecated' is never used.'
'/gostcrypt2_src/Common/Crypto.h:135:unusedStructMember:struct member 'Hash::SystemEncryption' is never used.'
Remove the structure Hash because it is never used

'/gostcrypt2_src/Core/loopdevlib/blkdev.h:89:unusedStructMember:struct member 'hd_geometry::heads' is never used.'
'/gostcrypt2_src/Core/loopdevlib/blkdev.h:90:unusedStructMember:struct member 'hd_geometry::sectors' is never used.'
'/gostcrypt2_src/Core/loopdevlib/blkdev.h:91:unusedStructMember:struct member 'hd_geometry::cylinders' is never used.'
'/gostcrypt2_src/Core/loopdevlib/blkdev.h:92:unusedStructMember:struct member 'hd_geometry::start' is never used.'
To resolve these warnings remove the struct hd_geometry because it is never used.

'/gostcrypt2_src/Common/Gstdefs.h:88:unusedStructMember:struct member 'UINT64_STRUCT::LowPart' is never used.'
'/gostcrypt2_src/Common/Gstdefs.h:89:unusedStructMember:struct member 'UINT64_STRUCT::HighPart' is never used.'
Remove the declaration of the structure UINT64_STRUCT and replace every initialisation of this structure by quint64.

'/gostcrypt2_src/Common/Password.h:35:unusedStructMember:struct member 'Password::Length' is never used.'
'/gostcrypt2_src/Common/Password.h:36:unusedStructMember:struct member 'Password::Text' is never used.'
'/gostcrypt2_src/Common/Password.h:37:unusedStructMember:struct member 'Password::Pad' is never used.'
Remove the File Password.h because it is useless, and actually it was used by unused functions.

'/gostcrypt2_src/Common/GfMul.h:47:unusedStructMember:struct member 'GfCtx8k::gf_t8k' is never used.'
'/gostcrypt2_src/Common/GfMul.h:52:unusedStructMember:struct member 'GfCtx4k64::gf_t4k' is never used.'
'/gostcrypt2_src/Common/GfMul.h:58:unusedStructMember:struct member 'GfCtx::gf_t128' is never used.'
'/gostcrypt2_src/Common/GfMul.h:59:unusedStructMember:struct member 'GfCtx::gf_t64' is never used.'
Remove the GfMul.c and GfMul.h files because there are useless.

Problem of « useInitializationList » :

'Volume/VolumePath.h:14:useInitializationList:Variable 'Data' is assigned in constructor body. Consider performing initialization in initialization list.'
'Volume/VolumePath.h:15:useInitializationList:Variable 'Data' is assigned in constructor body. Consider performing initialization in initialization list.'

```
class VolumePath
    {
    public:
        VolumePath () { }
    VolumePath (const wstring &path) {Data = path;}
    VolumePath (const FilesystemPath &path) {Data = path;}
```

```
...
}
```

use an Initializer list to remove these warnings.

Problem of « noExplicitConstructor » :

```
'Volume/VolumePassword.h:26:noExplicitConstructor:Class 'VolumePassword' has a constructor with 1 argument that is not explicit.'
  class VolumePassword
      {
      public:
            VolumePassword ();
            VolumePassword (const quint8 *password, size_t size);
            VolumePassword (const char *password, size_t size);
            VolumePassword (const wchar_t *password, size_t charCount);
               VolumePassword (const std::wstring &password);
            VolumePassword (const VolumePassword &password) { Set (password); }
            virtual ~VolumePassword ();
...
}
```

Warning can be removed by specifying "explicit":CONSTRUCTOR

```
'Volume/Keyfile.h:27:noExplicitConstructor:Class 'Keyfile' has a constructor with 1 argument that is not explicit.'
   class Keyfile //inherit from QFile
      {
      public:
      Keyfile (const FilePath &path){ (void)path; }
      virtual ~Keyfile () { }


...
}
```

Warning can be removed by specifying "explicit":CONSTRUCTOR

```
'Volume/VolumeHeader.h:49:noExplicitConstructor:Class 'VolumeHeader' has a constructor with 1 argument that is not explicit.'
class VolumeHeader
      {
      public:
            VolumeHeader (quint32 HeaderSize);
            virtual ~VolumeHeader ();
...
}
```

Warning can be removed by specifying "explicit":CONSTRUCTOR

```
'/gostcrypt2_src/Core/CoreRequest.h:33:noExplicitConstructor:Struct 'ProgressTrackingParameters' has a constructor with 1 argument that
is not explicit.'
struct ProgressTrackingParameters
{
   ProgressTrackingParameters(quint32 requestId) : requestId(requestId), start(0), end(1) {}
   ProgressTrackingParameters() : start(0), end(1) {}
   ProgressTrackingParameters(ProgressTrackingParameters &parent, qreal subStart, qreal subEnd) :
requestId(parent.requestId), start(parent.end*subStart+parent.start*(1-subStart)),
end(parent.end*subEnd+parent.start*(1-subEnd)) {}

...

}
```

Warning can be removed by specifying "explicit":CONSTRUCTOR

```
'/gostcrypt2_src/Core/Service.h:28:noExplicitConstructor:Class 'Service' has a constructor with 1 argument that is not explicit.'
    class Service : public QObject
            {
                Q_OBJECT
            public:
                Service(QString serviceName);
...

}
```

Warning can be removed by specifying "explicit":CONSTRUCTOR

```
'Core/CoreRequest.h:33:noExplicitConstructor:Struct 'ProgressTrackingParameters' has a constructor with 1 argument that is not explicit.'
struct ProgressTrackingParameters
{
   explicit ProgressTrackingParameters(quint32 requestId) : requestId(requestId), start(0), end(1) {}
    ProgressTrackingParameters() : start(0), end(1) {}
    ProgressTrackingParameters(ProgressTrackingParameters &parent, qreal subStart, qreal subEnd) :
requestId(parent.requestId), start(parent.end*subStart+parent.start*(1-subStart)),
end(parent.end*subEnd+parent.start*(1-subEnd)) {}


   quint32 requestId;
   qreal start;
   qreal end;
   DEC_SERIALIZABLE(ProgressTrackingParameters);
};
```

Warning can be removed by specifying "explicit":CONSTRUCTOR

```
'/gostcrypt2_src/UI/SecureTextField.h:10:noExplicitConstructor:Class 'SecureTextField' has a constructor with 1 argument that is not explicit.'
class SecureTextField : public QObject
{
   Q_OBJECT
public:
   SecureTextField(QObject *parent = 0) { (void)parent; }
...

}
```

Warning can be removed by specifying "explicit":CONSTRUCTOR

Problem of « postfixOperator » :

```
'/gostcrypt2_src/Core/CoreBase.cpp:50:postfixOperator:Prefer prefix ++/-- operators for non-primitive types.'
'/gostcrypt2_src/Core/CoreBase.cpp:71:postfixOperator:Prefer prefix ++/-- operators for non-primitive types.'
'/gostcrypt2_src/Core/CoreBase.cpp:244:postfixOperator:Prefer prefix ++/-- operators for non-primitive types.'
'/gostcrypt2_src/Core/CoreBase.cpp:257:postfixOperator:Prefer prefix ++/-- operators for non-primitive types.'
 for(GostCrypt::Volume::EncryptionAlgorithmList::iterator algorithm = algorithms.begin(); algorithm != algorithms.end();
algorithm++)
            {
                        ...

            }

...

for (GostCrypt::Volume::Pkcs5KdfList::iterator pkcs = pkcss.begin(); pkcs != pkcss.end(); pkcs++)
    {
                        ...
            }
```

```
...
 for (GostCrypt::Volume::EncryptionAlgorithmList::iterator ea = eas.begin(); ea != eas.end(); ea++)
		{
			...

		}

....
    for (GostCrypt::Volume::Pkcs5KdfList::iterator pkcs = pkcss.begin(); pkcs != pkcss.end(); pkcs++)
		{
			...
		}
```

To resolve these warnings you have to pass the operator ++/-- in prefix and not in postfix (++« it »).

Problem of « uninitMemberVar » :

'/gostcrypt2_src/Core/CoreRequest.cpp:258:uninitMemberVar:Member variable 'MountVolumeRequest::isDevice' is not initialized in the constructor.'

```
MountVolumeRequest::MountVolumeRequest()
{
   this->doMount = true;
   this->fileSystemType = "vfat";
   this->preserveTimestamps = false;
   this->protection = Volume::VolumeProtection::Enum::None;
   this->sharedAccessAllowed = false;
   this->useBackupHeaders = false;
   this->forVolumeCreation = false;
}
```

To resolve this warning you have to initialized IsDevice (*this*->isDevice = *false*;) in the constructor

'/gostcrypt2_src/UI/GraphicInterface.h:52:uninitMemberVar:Member variable 'MyGuiApplication::mGI' is not initialized in the constructor.'

```
class MyGuiApplication : public QApplication {
Q_OBJECT
public:
   MyGuiApplication(int& argc, char** argv) : QApplication(argc, argv) {}
   bool notify(QObject* receiver, QEvent* event);
   void setGI(GraphicInterface* gi) { mGI = gi; }
...
}
```

To resolve this warning we have to initialized the pointer mGI to nullptr

'Volume/VolumeFile.h:38:uninitMemberVar:Member variable 'VolumeFile::preserveTimestamps' is not initialized in the constructor.'
'Volume/VolumeFile.h:38:uninitMemberVar:Member variable 'VolumeFile::FileHandle' is not initialized in the constructor.'
'Volume/VolumeFile.h:38:uninitMemberVar:Member variable 'VolumeFile::AccTime' is not initialized in the constructor.'
'Volume/VolumeFile.h:38:uninitMemberVar:Member variable 'VolumeFile::ModTime' is not initialized in the constructor.'

To resolve this warning you have to initialized these variable at the declaration of the constuctor.

Problem of « variableScope » :

'/gostcrypt2_src/Core/loopdevlib/canonicalize.c:76:variableScope:The scope of the variable 'p' can be reduced.'

```
char *canonicalize_path_restricted(const char *path)
{
   char *canonical, char *p = NULL ;
      int errsv;
```

```
    uid_t euid;
    gid_t egid;

    if (!path || !*path)
        return NULL;


  ...
    canonical = realpath(path, NULL);
    if (canonical) {
    p = strrchr(canonical, '/');
        if (p && strncmp(p, "/dm-", 4) == 0 && isdigit(*(p + 4))) {
            char *dm = canonicalize_dm_name(p + 1);
            if (dm) {
                free(canonical);
                canonical = dm;
            }
        }
                ...
}
```

To resolve this warning create the variable in child's scope (when we need it).

'/gostcrypt2_src/Crypto/GostHash.c:156:variableScope:The scope of the variable 'sum' can be reduced.'

```
static void add_blocks (quint8 *T, quint8 *F, gst_dword len)
{
    gst_dword i;
    gst_word carry = 0;
    gst_word sum;
    for (i = 0; i < len; i++)
    {
        sum = (gst_word)T[i] + (gst_word)F[i] + carry;
        T[i] = (quint8)sum & 0xFF;
        carry = sum >> 8;
    }
}
```

To resolve this warning create the variable in child's scope (when we need it).

'/gostcrypt2_src/Crypto/GostHash.c:264:variableScope:The scope of the variable 'add_bytes' can be reduced.'

```
void GOSTHASH_add (quint8 *block, gst_udword len, gost_hash_ctx *ctx)
{
    gst_udword add_bytes;

    quint8 *curptr = block;
    quint8 *barrier = block + (len - 32); //In order that curptr += 32 won't overshoot len.

    if (ctx->left) //There are unsigned chars left from the last GOSTHASH_add
    {
        add_bytes = (32 - ctx->left) > len ? len : (32 - ctx->left);
        copy_blocks(ctx->remainder + (quint8)ctx->left, block, (gst_dword)add_bytes);
...
}
```

To resolve this warning create the variable in child's scope (when we need it).

'/gostcrypt2_src/Crypto/Stribog.c:630:variableScope:The scope of the variable 'tmp' can be reduced.'

```
static void Add512 (quint8 *dest, const quint8 *a, const quint8 *b)
{
    quint8 carry = 0;
          gst_dword i;
          quint8 tmp ;
```

```
    for (i = 63; i >= 0; i--)
    {
            tmp = a[i] + b[i] + carry;
        carry = (quint8)((gst_uword)(a[i] + b[i]) >> 8);
        dest[i] = tmp;
    }
}
```

To resolve this warning create the variable in child's scope (when we need it).

'/gostcrypt2_src/Volume/EncryptionThreadPool.cpp:28:variableScope:The scope of the variable 'workItem' can be reduced.'

```
void EncryptionThreadPool::DoWork (WorkType::Enum type, const EncryptionMode *encryptionMode, quint8 *data,
quint64 startUnitNo, quint64 unitCount, size_t sectorSize)
    {
        size_t fragmentCount;
        size_t unitsPerFragment;
        size_t remainder;

    quint8 *fragmentData;
    quint64 fragmentStartUnitNo;

        WorkItem *workItem ;
...
while (fragmentCount-- > 0)
            {
                workItem = &WorkItemQueue[EnqueuePosition++];

                if (EnqueuePosition >= QueueSize)
                    EnqueuePosition = 0;


...
}
```

To resolve this warning create the variable in child's scope (when we need it).

'/gostcrypt2_src/Volume/EncryptionThreadPool.cpp:212:variableScope:The scope of the variable 'workItem' can be reduced.'

```
 void EncryptionThread::run()
   {
        try
        {
                        EncryptionThreadPool::WorkItem *workItem;
              while (!EncryptionThreadPool::StopPending)
            {
                {
            QMutexLocker lock (&EncryptionThreadPool::DequeueMutex);

                        workItem =
&EncryptionThreadPool::WorkItemQueue[EncryptionThreadPool::DequeuePosition++];


...
}
```

To resolve this warning create the variable in child's scope (when we need it).

Problem of « redundantAssignment » :

'/gostcrypt2_src/Crypto/GostCipher.c:70:redundantAssignment:Variable 'n1' is reassigned a value before the old one has been used.'
'/gostcrypt2_src/Crypto/GostCipher.c:71:redundantAssignment:Variable 'n2' is reassigned a value before the old one has been used.'
'/gostcrypt2_src/Crypto/GostCipher.c:127:redundantAssignment:Variable 'n1' is reassigned a value before the old one has been used.'
'/gostcrypt2_src/Crypto/GostCipher.c:128:redundantAssignment:Variable 'n2' is reassigned a value before the old one has been used.'
Remove n1=0 and n2 = 0 at the beginning of the functions **gost_encrypt & gost_decrypt**

because they are useless.

Problem of « unusedFunction » :

'/gostcrypt2_src/Core/loopdevlib/blkdev.c:279:unusedFunction:The function 'blkdev_get_geometry' is never used.
To resolve this warning remove the function.

'/gostcrypt2_src/Volume/VolumePassword.cpp:52:unusedFunction:The function 'CheckPortability' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Volume/VolumePassword.h:38:unusedFunction:The function 'IsEmpty' is never used.'
To resolve this warning remove the fonction in the header file.

'/gostcrypt2_src/Volume/CipherAlgorithm.cpp:26:unusedFunction:The function 'DecryptBlock' is never used.'
'/gostcrypt2_src/Volume/CipherAlgorithm.cpp:66:unusedFunction:The function 'GetAvailableCiphers' is never used.'
'/gostcrypt2_src/Volume/CipherAlgorithm.cpp:90:unusedFunction:The function 'StoreCipherKey' is never used.'
'/gostcrypt2_src/Volume/CipherAlgorithm.cpp:96:unusedFunction:The function 'RestoreCipherKey' is never used.'
'/gostcrypt2_src/Volume/CipherAlgorithm.cpp:102:unusedFunction:The function 'IsKeySwapped' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Volume/CipherAlgorithm.h:31:unusedFunction:The function 'EnableHwSupport' is never used.'
'/gostcrypt2_src/Volume/CipherAlgorithm.h:36:unusedFunction:The function 'GetKey' is never used.'
'/gostcrypt2_src/Volume/CipherAlgorithm.h:41:unusedFunction:The function 'IsHwSupportAvailable' is never used.'
'/gostcrypt2_src/Volume/CipherAlgorithm.h:42:unusedFunction:The function 'IsHwSupportEnabled' is never used.'
To resolve this warning remove the fonction in the header file.

'/gostcrypt2_src/Volume/EncryptionMode.cpp:36:unusedFunction:The function 'GetAvailableModes' is never used.'
'/gostcrypt2_src/Volume/EncryptionMode.cpp:51:unusedFunction:The function 'ValidateParameters' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Volume/EncryptionMode.h:39:unusedFunction:The function 'GetSectorOffset' is never used.'
'/gostcrypt2_src/Volume/EncryptionMode.h:40:unusedFunction:The function 'IsKeySet' is never used.'
'/gostcrypt2_src/Volume/EncryptionMode.h:43:unusedFunction:The function 'SetSectorOffset' is never used.
To resolve this warning remove the fonction in the header file.

'/gostcrypt2_src/Volume/EncryptionAlgorithm.cpp:98:unusedFunction:The function 'GetMaxBlockSize' is never used.'
'/gostcrypt2_src/Volume/EncryptionAlgorithm.cpp:110:unusedFunction:The function 'GetMinBlockSize' is never used.'
'/gostcrypt2_src/Volume/EncryptionAlgorithm.cpp:214:unusedFunction:The function 'ValidateState' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Volume/EncryptionAlgorithm.h:37:unusedFunction:The function 'GetCiphers' is never used.'
To resolve this warning remove the fonction in the header file.

'/gostcrypt2_src/Core/RandomNumberGenerator.cpp:112:unusedFunction:The function 'GetHash' is never used.'
'/gostcrypt2_src/Core/RandomNumberGenerator.cpp:137:unusedFunction:The function 'SetHash' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/RandomNumberGenerator.h:26:unusedFunction:The function 'GetDataFast' is never used.'
'/gostcrypt2_src/Core/RandomNumberGenerator.h:28:unusedFunction:The function 'IsEnrichedByUser' is never used.'
'/gostcrypt2_src/Core/RandomNumberGenerator.h:31:unusedFunction:The function 'SetEnrichedByUserStatus' is never used.'
To resolve this warning remove the fonction in the header file.

'/gostcrypt2_src/Volume/VolumeHeader.cpp:341:unusedFunction:The function 'SetSize' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Volume/VolumeHeader.h:55:unusedFunction:The function 'GetEncryptedAreaLength' is never used.'
'/gostcrypt2_src/Volume/VolumeHeader.h:57:unusedFunction:The function 'GetFlags' is never used.'
'/gostcrypt2_src/Volume/VolumeHeader.h:59:unusedFunction:The function 'GetHiddenVolumeDataSize' is never used.'
'/gostcrypt2_src/Volume/VolumeHeader.h:62:unusedFunction:The function 'GetRequiredMinProgramVersion' is never used.'
'/gostcrypt2_src/Volume/VolumeHeader.h:68:unusedFunction:The function 'GetHeaderVersion' is never used.'
To resolve this warning remove the fonction in the header file.

'/gostcrypt2_src/Volume/Volume.cpp:60:unusedFunction:The function 'GetEncryptionMode' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Volume/Volume.h:37:unusedFunction:The function 'GetHostSize' is never used.'
'/gostcrypt2_src/Volume/Volume.h:38:unusedFunction:The function 'GetLayout' is never used.'
'/gostcrypt2_src/Volume/Volume.h:40:unusedFunction:The function 'GetProtectionType' is never used.'
'/gostcrypt2_src/Volume/Volume.h:45:unusedFunction:The function 'GetTopWriteOffset' is never used.'
'/gostcrypt2_src/Volume/Volume.h:46:unusedFunction:The function 'GetTotalDataRead' is never used.'
'/gostcrypt2_src/Volume/Volume.h:47:unusedFunction:The function 'GetTotalDataWritten' is never used.'
'/gostcrypt2_src/Volume/Volume.h:50:unusedFunction:The function 'IsHiddenVolumeProtectionTriggered' is never used.'
'/gostcrypt2_src/Volume/Volume.h:51:unusedFunction:The function 'IsInSystemEncryptionScope' is never used.'
To resolve this warning remove the fonction in the header file.

'/gostcrypt2_src/Common/GfMul.c:378:unusedFunction:The function 'GfMul128' is never used.'
'/gostcrypt2_src/Common/GfMul.c:425:unusedFunction:The function 'GfMul128Tab' is never used.'
'/gostcrypt2_src/Common/GfMul.c:823:unusedFunction:The function 'GfMulSelfTest' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Common/Endian.c:23:unusedFunction:The function 'MirrorBytes16' is never used.'
'/gostcrypt2_src/Common/Endian.c:29:unusedFunction:The function 'MirrorBytes32' is never used.'
'/gostcrypt2_src/Common/Endian.c:38:unusedFunction:The function 'MirrorBytes64' is never used.'
'/gostcrypt2_src/Common/Endian.c:52:unusedFunction:The function 'LongReverse' is never used.'
To resolve this warning remove Endian.c and Edian.h because they are useless.

'/gostcrypt2_src/Volume/VolumeLayout.h:41:unusedFunction:The function 'SetHeader' is never used.'
To resolve this warning remove the fonction in the header file.

'/gostcrypt2_src/Volume/EncryptionTest.cpp:29:unusedFunction:The function 'TestAll' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Volume/VolumeHash.cpp:30:unusedFunction:The function 'ValidateDataParameters' is never used.'
'/gostcrypt2_src/Volume/VolumeHash.cpp:36:unusedFunction:The function 'ValidateDigestParameters' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/blkdev.c:43:unusedFunction:The function 'is_blkdev' is never used.'
'/gostcrypt2_src/Core/loopdevlib/blkdev.c:240:unusedFunction:The function 'blkdev_is_misaligned' is never used.'
'/gostcrypt2_src/Core/loopdevlib/blkdev.c:257:unusedFunction:The function 'blkdev_is_cdrom' is never used.'
'/gostcrypt2_src/Core/loopdevlib/blkdev.c:274:unusedFunction:The function 'blkdev_scsi_type_to_name' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/canonicalize.c:74:unusedFunction:The function 'canonicalize_path_restricted' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/CoreException.h:39:unusedFunction:The function 'getLine' is never used.'
'/gostcrypt2_src/Core/CoreException.h:40:unusedFunction:The function 'getFilename' is never used.'
'/gostcrypt2_src/Core/CoreException.h:41:unusedFunction:The function 'getFonction' is never used.'
'/gostcrypt2_src/Core/CoreException.h:42:unusedFunction:The function 'clone' is never used.'
Has been already remove

'/gostcrypt2_src/Common/Crc.c:74:unusedFunction:The function 'crc32int' is never used.'

78

'/gostcrypt2_src/Common/Crc.c:91:unusedFunction:The function 'crc32_selftests' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/c.h:196:unusedFunction:The function 'is_power_of_2' is never used.'
'/gostcrypt2_src/Core/loopdevlib/c.h:241:unusedFunction:The function 'get_hostname_max' is never used.'
To resolve this warning remove the fonction in the header file.

'/gostcrypt2_src/Common/Pkcs5.c:439:unusedFunction:The function 'get_pkcs5_prf_name' is never used.'
'/gostcrypt2_src/Common/Pkcs5.c:451:unusedFunction:The function 'get_pkcs5_iteration_count' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/loopdev.c:130:unusedFunction:The function 'loopcxt_has_device' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:300:unusedFunction:The function 'loopcxt_set_fd' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:743:unusedFunction:The function 'loopcxt_get_sizelimit' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:772:unusedFunction:The function 'loopcxt_get_encrypt_type' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:794:unusedFunction:The function 'loopcxt_get_crypt_name' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:884:unusedFunction:The function 'loopcxt_is_partscan' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:927:unusedFunction:The function 'loopcxt_is_readonly' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:1009:unusedFunction:The function 'loopcxt_set_offset' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:1022:unusedFunction:The function 'loopcxt_set_sizelimit' is never used.
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:1317:unusedFunction:The function 'loopcxt_add_device' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:1394:unusedFunction:The function 'loopdev_is_autoclear' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:1412:unusedFunction:The function 'loopdev_get_backing_file' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:1431:unusedFunction:The function 'loopdev_is_used' is never used.
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:1454:unusedFunction:The function 'loopdev_delete' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:1503:unusedFunction:The function 'loopdev_find_by_backing_file' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:1525:unusedFunction:The function 'loopdev_count_by_backing_file' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/sysfs.c:29:unusedFunction:The function 'sysfs_devno_has_attribute' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:110:unusedFunction:The function 'sysfs_devno_to_devpath' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:333:unusedFunction:The function 'sysfs_partno_to_devno' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:384:unusedFunction:The function 'sysfs_read_s64' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:427:unusedFunction:The function 'sysfs_count_dirents' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:441:unusedFunction:The function 'sysfs_count_partitions' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:551:unusedFunction:The function 'sysfs_devno_to_wholedisk' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:699:unusedFunction:The function 'sysfs_scsi_host_strdup_attribute' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:719:unusedFunction:The function 'sysfs_scsi_host_is' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:748:unusedFunction:The function 'sysfs_scsi_has_attribute' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:759:unusedFunction:The function 'sysfs_scsi_path_contains' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Volume/EncryptionMode.h:34:unusedFunction:The function 'GetKey' is never used.'
To resolve this warning remove the fonction in the header file.

'/gostcrypt2_src/Volume/EncryptionMode.cpp:42:unusedFunction:The function 'ValidateParameters' is never used.'
'/gostcrypt2_src/Volume/EncryptionMode.cpp:36:unusedFunction:The function 'ValidateState' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Common/GfMul.c:626:unusedFunction:The function 'Gf128MulBy64Tab' is never used.'
'/gostcrypt2_src/Common/GfMul.c:547:unusedFunction:The function 'Gf128Tab64Init' is never used.'
'/gostcrypt2_src/Common/GfMul.c:655:unusedFunction:The function 'Gf64MulTab' is never used.'
'/gostcrypt2_src/Common/GfMul.c:583:unusedFunction:The function 'Gf64TabInit' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Volume/EncryptionTest.cpp:59:unusedFunction:The function 'TestCiphers' is never used.'
'/gostcrypt2_src/Volume/EncryptionTest.cpp:499:unusedFunction:The function 'TestPkcs5' is never used.'
'/gostcrypt2_src/Volume/EncryptionTest.cpp:310:unusedFunction:The function 'TestXts' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/loopdev.c:1208:unusedFunction:The function 'loopcxt_find_by_backing_file' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:801:unusedFunction:The function 'loopcxt_is_autoclear' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:232:unusedFunction:The function 'loopcxt_strdup_device' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:780:unusedFunction:The function 'loopmod_supports_partscan' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Common/GfMul.c:348:unusedFunction:The function 'mul_bex8' is never used.'
'/gostcrypt2_src/Common/GfMul.c:356:unusedFunction:The function 'mul_bex8_64' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/sysfs.c:352:unusedFunction:The function 'sysfs_get_devname' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:324:unusedFunction:The function 'sysfs_get_slave' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:147:unusedFunction:The function 'sysfs_has_attribute' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:235:unusedFunction:The function 'sysfs_is_partition_dirent' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:373:unusedFunction:The function 'sysfs_scsi_get_hctl' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Common/Crc.c:101:unusedFunction:The function
'crc32_selftests' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Volume/EncryptionModeXTS.h:29:unusedFunction:The function 'GetKey' is never used.'
To resolve this warning remove the fonction in the header file.

'/gostcrypt2_src/Common/GfMul.c:494:unusedFunction:The function 'MirrorBits128' is never used.'
'/gostcrypt2_src/Common/GfMul.c:508:unusedFunction:The function 'MirrorBits64' is never used.'
'/gostcrypt2_src/Common/GfMul.c:428:unusedFunction:The function 'compile_4k_table64' is never used.'
'/gostcrypt2_src/Common/GfMul.c:381:unusedFunction:The function 'compile_8k_table' is never used.'
'/gostcrypt2_src/Common/GfMul.c:135:unusedFunction:The function 'move_block_aligned' is never used.'
'/gostcrypt2_src/Common/GfMul.c:141:unusedFunction:The function 'move_block_aligned64' is never used.'
'/gostcrypt2_src/Common/GfMul.c:146:unusedFunction:The function 'xor_block_aligned' is never used.'
'/gostcrypt2_src/Common/GfMul.c:152:unusedFunction:The function 'xor_block_aligned64' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Volume/Volume.cpp:242:unusedFunction:The function 'ValidateState' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/loopdev.c:764:unusedFunction:The function 'loopcxt_is_used' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/sysfs.c:176:unusedFunction:The function 'sysfs_opendir' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:240:unusedFunction:The function 'sysfs_read_int' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:163:unusedFunction:The function 'sysfs_readlink' is never used.'
'/gostcrypt2_src/Core/loopdevlib/sysfs.c:131:unusedFunction:The function 'sysfs_stat' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/loopdev.c:716:unusedFunction:The function 'loopcxt_get_backing_devno' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:657:unusedFunction:The function 'loopcxt_get_backing_file' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:736:unusedFunction:The function 'loopcxt_get_backing_inode' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Common/GfMul.c:252:unusedFunction:The function 'mul_x' is never used.'
'/gostcrypt2_src/Common/GfMul.c:273:unusedFunction:The function 'mul_x64' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/at.c:86:unusedFunction:The function 'readlink_at' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/sysfs.c:182:unusedFunction:The function 'sysfs_strdup' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

'/gostcrypt2_src/Core/loopdevlib/at.c:80:unusedFunction:The function 'readlink_at' is never used.'
To resolve this warning remove the fonction in the header file and the source file.

Problem of « selfAssignement »:

'/gostcrypt2_src/Volume/VolumeFile.cpp:107:selfAssignment:Redundant assignment of 'preserveTimestamps' to itself.'

```
void VolumeFile::Open (const QSharedPointer<QFileInfo> path, bool readOnly, bool preserveTimestamps)
    {
                ...
                Path = path;
                preserveTimestamps = preserveTimestamps;
        FileIsOpen = true;
    }
```

To resolve this warning put a this→ before the variable.

Problem of « clarifyCalculation » :

'/gostcrypt2_src/Core/loopdevlib/loopdev.c:292:clarifyCalculation:Clarify calculation precedence for '&' and '?'.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.c:295:clarifyCalculation:Clarify calculation precedence for '&' and '?'.'

Put some parenthesis to clarify te calculation

Problem of «functionStatic » :

'/gostcrypt2_src/Volume/VolumeHash.h:51:functionStatic:Technically the member function 'GostCrypt::Volume::VolumeHash::ValidateKeyDerivationParameters' can be static.'

```
namespace Volume {
    class VolumeHash;
  typedef QList < QSharedPointer <VolumeHash> > VolumeHashList;


...

protected:
        SecureBuffer Context;
        bool Deprecated;
            void ValidateKeyDerivationParameters (const BufferPtr &key, const VolumePassword &password, const
ConstBufferPtr &salt, int iterationCount)const;


...

}
```

To resolve this warning remove the const at the end of the declaration of the function an put a static at the beginning of it.

'/gostcrypt2_src/Volume/EncryptionTest.h:22:functionStatic:Technically the member function 'GostCrypt::Volume::EncryptionTest::TestCiphers' can be static.'
'/gostcrypt2_src/Volume/EncryptionTest.h:23:functionStatic:Technically the member function 'GostCrypt::Volume::EncryptionTest::TestPkcs5' can be static.'
'/gostcrypt2_src/Volume/EncryptionTest.h:24:functionStatic:Technically the member function 'GostCrypt::Volume::EncryptionTest::TestXts' can be static.'

Fixed (Indeed make it static is an optimization)

'/gostcrypt2_src/Volume/EncryptionModeXTS.h:38:functionStatic:Technically the member function 'GostCrypt::Volume::EncryptionModeXTS::DecryptBufferXTS8Byte' can be static.'
'/gostcrypt2_src/Volume/EncryptionModeXTS.h:41:functionStatic:Technically the member function 'GostCrypt::Volume::EncryptionModeXTS::EncryptBufferXTS8Byte' can be static.'

Fixed (Indeed make it static is an optimization)

'/gostcrypt2_src/Volume/VolumeHeader.h:66:functionStatic:Technically the member function 'GostCrypt::Volume::VolumeHeader::DeserializeEntry' can be static.'
'/gostcrypt2_src/Volume/VolumeHeader.h:67:functionStatic:Technically the member function 'GostCrypt::Volume::VolumeHeader::DeserializeEntryAt' can be static.'
'/gostcrypt2_src/Volume/VolumeHeader.h:70:functionStatic:Technically the member function 'GostCrypt::Volume::VolumeHeader::SerializeEntry' can be static.'

Fixed (Indeed make it static is an optimization)

Problem of « exceptRethrowCopy » :

'/gostcrypt2_src/Core/CoreBase.cpp:66:exceptRethrowCopy:Throwing a copy of the caught exception instead of rethrowing the original exception.'
'/gostcrypt2_src/Core/CoreBase.cpp:95:exceptRethrowCopy:Throwing a copy of the caught exception instead of rethrowing the original exception.'
'/gostcrypt2_src/Core/CoreBase.cpp:156:exceptRethrowCopy:Throwing a copy of the caught exception instead of rethrowing the original exception.'
'/gostcrypt2_src/Core/CoreBase.cpp:224:exceptRethrowCopy:Throwing a copy of the caught exception instead of rethrowing the original exception.'
'/gostcrypt2_src/Core/CoreBase.cpp:470:exceptRethrowCopy:Throwing a copy of the caught exception instead of rethrowing the original exception.'

To resolve these warnings remove the exception ‘e’ after throw.

'/gostcrypt2_src/Core/CoreRoot.cpp:145:exceptRethrowCopy:Throwing a copy of the caught exception instead of rethrowing the original exception.'
'/gostcrypt2_src/Core/CoreRoot.cpp:173:exceptRethrowCopy:Throwing a copy of the caught exception instead of rethrowing the original exception.'
'/gostcrypt2_src/Core/CoreRoot.cpp:242:exceptRethrowCopy:Throwing a copy of the caught exception instead of rethrowing the original exception.'
'/gostcrypt2_src/Core/CoreRoot.cpp:486:exceptRethrowCopy:Throwing a copy of the caught exception instead of rethrowing the original exception.'
'/gostcrypt2_src/Core/CoreRoot.cpp:500:exceptRethrowCopy:Throwing a copy of the caught exception instead of rethrowing the original exception.'
'/gostcrypt2_src/Core/CoreRoot.cpp:528:exceptRethrowCopy:Throwing a copy of the caught exception instead of rethrowing the original exception.'

To resolve these warnings remove the exception ‘e’ after throw.

Problem of « variableHidingTpedef » :

'/gostcrypt2_src/Crypto/GostCipher.h:30:variableHidingTypedef:The typedef 'quint8' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GostCipher.h:31:variableHidingTypedef:The typedef 'gst_word' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GostCipher.h:32:variableHidingTypedef:The typedef 'gst_dword' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GostCipher.h:33:variableHidingTypedef:The typedef 'gst_uword' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GostCipher.h:34:variableHidingTypedef:The typedef 'gst_udword' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GrasshopperCipher.h:27:variableHidingTypedef:The typedef 'quint8' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GrasshopperCipher.h:28:variableHidingTypedef:The typedef 'gst_word' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GrasshopperCipher.h:29:variableHidingTypedef:The typedef 'gst_dword' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GrasshopperCipher.h:30:variableHidingTypedef:The typedef 'gst_uword' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GrasshopperCipher.h:31:variableHidingTypedef:The typedef 'gst_udword' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/Stribog.h:20:variableHidingTypedef:The typedef 'quint8' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/Stribog.h:21:variableHidingTypedef:The typedef 'gst_word' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/Stribog.h:22:variableHidingTypedef:The typedef 'gst_dword' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/Stribog.h:23:variableHidingTypedef:The typedef 'gst_uword' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/Stribog.h:24:variableHidingTypedef:The typedef 'gst_udword' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GostHash.h:25:variableHidingTypedef:The typedef 'quint8' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GostHash.h:26:variableHidingTypedef:The typedef 'gst_word' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GostHash.h:27:variableHidingTypedef:The typedef 'gst_dword' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GostHash.h:28:variableHidingTypedef:The typedef 'gst_uword' hides a typedef with the same name.'
'/gostcrypt2_src/Crypto/GostHash.h:29:variableHidingTypedef:The typedef 'gst_udword' hides a typedef with the same name.'

Fixed by removing these structure and replaced them with qintXX and quintXX.

Problem of « funcArgNamesDifferent » :

'/gostcrypt2_src/Core/CoreUser.cpp:40:funcArgNamesDifferent:Function 'receiveResponse' argument 1 names different: declaration 'response' definition 'r'.'

Give the same name to the argument at the declaration and the definition.

'/gostcrypt2_src/Core/RandomNumberGenerator.cpp:51:funcArgNamesDifferent:Function 'AddToPool' argument 1 names different: declaration 'buffer' definition 'data'.'

82

Give the same name to the argument at the declaration and the definition.

'/gostcrypt2_src/FuseService/FuseServiceHandler.cpp:22:funcArgNamesDifferent:Function 'receiveResponse' argument 1 names different: declaration 'response' definition 'r'.'

Give the same name to the argument at the declaration and the definition.

'/gostcrypt2_src/UI/GraphicInterface.cpp:192:funcArgNamesDifferent:Function 'printGetMountedVolumes' argument 1 names different: declaration 'r' definition 'response'.'
'/gostcrypt2_src/UI/GraphicInterface.cpp:212:funcArgNamesDifferent:Function 'printDismountVolume' argument 1 names different: declaration 'r' definition 'response'.'
'/gostcrypt2_src/UI/GraphicInterface.cpp:218:funcArgNamesDifferent:Function 'printMountVolume' argument 1 names different: declaration 'r' definition 'response'.'
'/gostcrypt2_src/UI/GraphicInterface.cpp:225:funcArgNamesDifferent:Function 'printCreateVolume' argument 1 names different: declaration 'r' definition 'response'.'
'/gostcrypt2_src/UI/GraphicInterface.cpp:231:funcArgNamesDifferent:Function 'printGetEncryptionAlgorithms' argument 1 names different: declaration 'r' definition 'response'.'
'/gostcrypt2_src/UI/GraphicInterface.cpp:239:funcArgNamesDifferent:Function 'printGetDerivationFunctions' argument 1 names different: declaration 'r' definition 'response'.'
'/gostcrypt2_src/UI/GraphicInterface.cpp:255:funcArgNamesDifferent:Function 'printGetHostDevices' argument 1 names different: declaration 'r' definition 'response'.'
'/gostcrypt2_src/UI/GraphicInterface.cpp:282:funcArgNamesDifferent:Function 'printCreateKeyFile' argument 1 names different: declaration 'r' definition 'response'.'
'/gostcrypt2_src/UI/GraphicInterface.cpp:289:funcArgNamesDifferent:Function 'printChangeVolumePassword' argument 1 names different: declaration 'r' definition 'response'.'

Give the same name to the argument at the declaration and the definition.

'/gostcrypt2_src/UI/Parser.cpp:110:funcArgNamesDifferent:Function 'parseDismount' argument 2 names different: declaration 'options' definition 'volume'.'

Give the same name to the argument at the declaration and the definition

'/gostcrypt2_src/Volume/Crypto/GostHash.c:261:funcArgNamesDifferent:Function 'GOSTHASH_add' argument 1 names different: declaration 'in' definition 'block'.'

Give the same name to the argument at the declaration and the definition

'/gostcrypt2_src/Volume/VolumeException.h:254:selfInitialization:Member variable 'comment' is initialized by itself.'

Change the name of the argument at the declaration

Notified as False Positive :

'/gostcrypt2_src/Core/loopdevlib/loopdev.h:70:unusedStructMember:struct member 'loopdev_iter::proc' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.h:72:unusedStructMember:struct member 'loopdev_iter::ncur' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.h:73:unusedStructMember:struct member 'loopdev_iter::minors' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.h:74:unusedStructMember:struct member 'loopdev_iter::nminors' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.h:75:unusedStructMember:struct member 'loopdev_iter::ct_perm' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.h:76:unusedStructMember:struct member 'loopdev_iter::ct_succ' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.h:78:unusedStructMember:struct member 'loopdev_iter::done' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.h:79:unusedStructMember:struct member 'loopdev_iter::default_check' is never used.'
'/gostcrypt2_src/Core/loopdevlib/loopdev.h:80:unusedStructMember:struct member 'loopdev_iter::flags' is never used.'

False Positive

'/gostcrypt2_src/Crypto/GostCipher.h:38:unusedStructMember:struct member 'gost_s_box::k8' is never used.'
'/gostcrypt2_src/Crypto/GostCipher.h:39:unusedStructMember:struct member 'gost_s_box::k7' is never used.'
'/gostcrypt2_src/Crypto/GostCipher.h:40:unusedStructMember:struct member 'gost_s_box::k6' is never used.'
'/gostcrypt2_src/Crypto/GostCipher.h:41:unusedStructMember:struct member 'gost_s_box::k5' is never used.'
'/gostcrypt2_src/Crypto/GostCipher.h:42:unusedStructMember:struct member 'gost_s_box::k4' is never used.'
'/gostcrypt2_src/Crypto/GostCipher.h:43:unusedStructMember:struct member 'gost_s_box::k3' is never used.'
'/gostcrypt2_src/Crypto/GostCipher.h:44:unusedStructMember:struct member 'gost_s_box::k2' is never used.'
'/gostcrypt2_src/Crypto/GostCipher.h:45:unusedStructMember:struct member 'gost_s_box::k1' is never used.'

False Positive

'/gostcrypt2_src/Crypto/GostHash.h:34:unusedStructMember:struct member 'gosthash_s_box::k8' is never used.'
'/gostcrypt2_src/Crypto/GostHash.h:35:unusedStructMember:struct member 'gosthash_s_box::k7' is never used.'
'/gostcrypt2_src/Crypto/GostHash.h:36:unusedStructMember:struct member 'gosthash_s_box::k6' is never used.'
'/gostcrypt2_src/Crypto/GostHash.h:37:unusedStructMember:struct member 'gosthash_s_box::k5' is never used.'
'/gostcrypt2_src/Crypto/GostHash.h:38:unusedStructMember:struct member 'gosthash_s_box::k4' is never used.'
'/gostcrypt2_src/Crypto/GostHash.h:39:unusedStructMember:struct member 'gosthash_s_box::k3' is never used.'
'/gostcrypt2_src/Crypto/GostHash.h:40:unusedStructMember:struct member 'gosthash_s_box::k2' is never used.'
'/gostcrypt2_src/Crypto/GostHash.h:41:unusedStructMember:struct member 'gosthash_s_box::k1' is never used.'

False Positive

'/gostcrypt2_src/Volume/EncryptionThreadPool.h:63:unusedStructMember:union member 'Anonymous0::Data' is never used.'
'/gostcrypt2_src/Volume/EncryptionThreadPool.h:66:unusedStructMember:union member 'Anonymous0::SectorSize' is never used.'

False Positive

'/gostcrypt2_src/Volume/EncryptionTest.h:28:unusedStructMember:struct member 'XtsTestVector::key1' is never used.'
'/gostcrypt2_src/Volume/EncryptionTest.h:29:unusedStructMember:struct member 'XtsTestVector::key2' is never used.'
'/gostcrypt2_src/Volume/EncryptionTest.h:30:unusedStructMember:struct member 'XtsTestVector::dataUnitNo' is never used.'
'/gostcrypt2_src/Volume/EncryptionTest.h:31:unusedStructMember:struct member 'XtsTestVector::blockNo' is never used.'
'/gostcrypt2_src/Volume/EncryptionTest.h:32:unusedStructMember:struct member 'XtsTestVector::plaintext' is never used.'
'/gostcrypt2_src/Volume/EncryptionTest.h:33:unusedStructMember:struct member 'XtsTestVector::ciphertext' is never used.'

False Positive

'/gostcrypt2_src/Volume/EncryptionThreadPool.cpp:208:unusedFunction:The function 'run' is never used.'

False positive

'/gostcrypt2_src/UI/TranslationApp.cpp:59:unusedFunction:The function 'tr' is never used.'

False positive