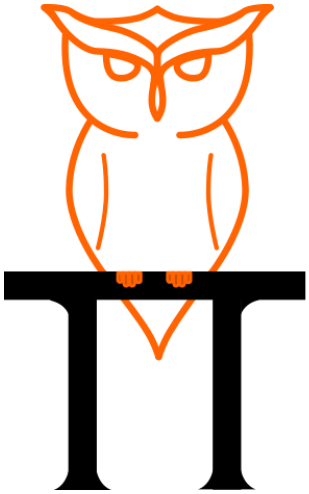

TransCrypt

Sommaire :

Les indices (x) font référence à des parties de la documentation (voir dossier).

1. Présentation générale

> Le chiffrement **TransCrypt**

Un peu d'histoire...

Le besoin de **préserver nos communications sécurisées** est présent depuis des millénaires. En réponse à ce besoin, les Hommes ont développé des techniques plus ou moins avancées, dans deux catégories : stéganographie (message caché) et **cryptologie** (message incompréhensible aux non-initiés). De nos jours, dans ce dernier domaine, sont principalement utilisés des algorithmes à clés publiques (ou asymétriques) et algorithmes à clés privées (ou symétriques).

Faiblesses des systèmes actuels

En premier lieu, les **systèmes actuels de cryptologie grand public** ne peuvent se prévaloir d'autre que le **pseudo-aléatoire informatique**. Ils sont donc empiriquement linéaires, prévisibles et n'assurent pas la confidentialité des échanges.

Ils sont également **empiriques** (les tailles de clé en nombre de bits augmentent régulièrement) en fonction des technologies actuelles et donc au contexte évolutif de la loi de Moore de la rapidité des microprocesseurs : ce qui est confidentiel un jour ne l'est plus par la suite.

Pourquoi un n-ième algorithme de chiffrement ?

La **deuxième meilleure solution pour un chiffrement** –outre des organisations dispendieuses- **consiste à générer un vrai aléa** (ou *True number generator,* TRNG en opposition au pseudo-aléatoire) **à distance sur un ordinateur**, un moyen n'autorisant a priori que le pseudo-aléatoire. En effet **le nombre Pi de par sa propriété de nombre transcendant** permet de **générer, sans coûts supplémentaires** liés au transport physique ou à la génération physique, **une suite de nombres sans aucune répétition** (donc avec exactement le même niveau d'entropie que le TRNG) **et à l'infini**. Cet algorithme est synonyme de **solution grand public gratuite**

Une découverte récente sur Pi

Jusqu'au 19 septembre 1995, il était impossible de mettre en œuvre cet algorithme car pour calculer le *n-ième* chiffre après la virgule du nombre Pi, il fallait avoir calculé au préalable le chiffre n-1. Seuls des supercalculateurs le faisaient et les temps de calculs étaient inexploitables. Mais cela est devenu possible depuis la **formule BBP** (ou formule de Bailey-Borwein-Plouffe) **qui permet de calculer le *n*-ième chiffre après la virgule du nombre Pi sans avoir à en calculer les précédents**, et en utilisant très peu de mémoire et de temps.

Nous vous proposons ici un nouvel algorithme de chiffrement, **TransCrypt** (contraction de *Transcendental Crypting*) (6), fondé sur cette découverte**,** et permettant la confidentialité des conversations.

> Application dans une messagerie : **TransOwl**

Une application parmi les plus évidentes de Transcrypt est son utilisation dans un **système de messagerie sécurisée, que nous avons baptisée TransOwl**. Nous en avons donc implémenté une, axée sur la confidentialité, qui en prime d'un chiffrement bout-à-bout avec notre algorithme TransCrypt, n'utilise que des sessions temporaires pour permettre de ne conserver aucune information sur le serveur.

2. Organisation du travail

> Notre équipe

Le projet **TransCrypt** est porté par une équipe de 4 lycéens en classe de première générale. Informés de l'existence du concours **Les Trophées NSI**, ils décident d'un commun accord de participer à l'édition 2022.

> Rôles de chacun (répartition des tâches)

- **Alex-Pauline :** Inventrice du chiffrement/déchiffrement TransCrypt (6) et rédactrice de son implémentation en Python (PYTHON) ;
- **Clémence :** Coordinatrice de l'équipe, traductrice en JavaScript de TransCrypt (JS) et rédactrice des documents suivants : présentation, documentation et résumé du projet ;
- **Daniel :** Responsable du Frontend et du responsive design (HTML / CSS / JS), concepteur de l'interface graphique, modélisateur du logo et du schéma de l'architecture de TransOwl (2) ;
- **Gabriel :** Responsable du Backend, concepteur du serveur, chargé des requêtes ajax (PYTHON/ JS : JQUERY).

La conception du logo (1) et la vidéo résultent d'un travail collaboratif de toute l'équipe.

> Mode de travail

Un **fonctionnement hybride** –mi-"présentiel", mi-"distanciel"- sur deux semaines a été choisi par les membres de ce projet. Les activités nécessitant l'implication simultanée de l'ensemble du groupe, comme le tournage de la vidéo, ont été réalisées dans leur établissement sur leur temps libre. Les parties laissant davantage d'autonomie individuelle ont, elles, pu être faites séparément, avec communication virtuelle et partage de code fréquents via les logiciels et applications Pearltrees et Whatsapp. (4)

Suivant les besoins, des groupes ont pu être formés au sein de l'équipe. En particulier, Daniel et Gabriel, en charge de la messagerie, se sont entraidés pour le bon fonctionnement de celle-ci. Gabriel et Clémence ont également travaillé ensemble afin de permettre l'insertion du chiffrement TransCrypt dans les communications de la messagerie. De manière générale, **chaque membre de l'équipe en a conseillé un autre à divers moments**.

3. Les étapes du projet

Le projet **TransCrypt** peut être partitionné selon les étapes chronologiques suivantes :
1/ Invention & implémentation du chiffrement
2/ En simultané : interface graphique et communication client/serveur de la messagerie

3/ Création du logo TransCrypt
4/ Rédaction du résumé du projet
5/ Film et montage de la vidéo explicative.
6/ Rédaction du présent document, de la documentation & relecture attentive.

4. Fonctionnement et opérationnalité

> Un projet opérationnel

Le **chiffrement ainsi que la messagerie associée sont opérationnels**. L'objectif initial est donc rempli et le projet considéré comme fini. Nous avons cependant des idées d'amélioration de la messagerie (décrites au paragraphe 5.1).

> Protocoles mis en œuvre pour s'assurer du bon fonctionnement et de la facilité d'utilisation de la messagerie

Tous les scripts ont fait l'objet d'une **relecture attentive** par l'ensemble de l'équipe. Le **travail de groupe** a ainsi permis la **création de tests plus variés**, qu'un tiers seul n'aurait peut-être pas pu trouver. De plus, **nous avons** nous-mêmes, et d'autres camarades de classe, **utilisé la messagerie** pour en repérer les éventuelles faiblesses lors d'un emploi fréquent.

> Difficultés rencontrées et solutions apportées

Concernant l'interface graphique, le **responsive design** en javascript et en CSS nous a posé quelques difficultés mineures, résolues à force de **persévérance**.

Nous avons également eu des problèmes avec les **requêtes client-serveur**, solutionnés par l'utilisation d'**appels AJAX en jQuery** (ensemble de méthodes permettant une communication asynchrone entre le navigateur et le serveur).

Au niveau du **chiffrement/déchiffrement**, il nous a fallu utiliser différentes approches. Nous avions à l'origine implémenté l'algorithme de celui-ci avec Python. Pour une intégration plus facile et propre dans la messagerie, nous avons ensuite décidé de le coder avec Javascript. **Javascript ne traitant pas les entiers de la même manière que le Python**, il nous est paru nécessaire de nous servir ou de la propriété BigInt de ce langage, ou d'importer une bibliothèque spécialisée. Ces deux solutions ont leurs avantages et inconvénients respectifs : BigInt est facile d'utilisation mais ne permet pas des opérations à temps constant, et n'est donc pas spécialement recommandé pour la cryptographie. D'un autre côté, importer une bibliothèque extérieure est souvent davantage synonyme de problèmes, nécessite un ou plusieurs téléchargements supplémentaires, sans compter les droits d'auteur. Notre préférence s'est donc portée sur **BigInt**. Aspirant à une meilleure sécurité et efficacité, nous avons alors décidé d'utiliser un **double chiffrement**, plutôt qu'un simple, **avec 2 clés de 4 chiffres**.

5. Ouverture

> Idées d'améliorations

Le chiffrement TransCrypt est un produit fini. Concernant sa preuve de concept, TransOwl, plusieurs **améliorations** peuvent être apportées dans un futur proche, afin d'en **faciliter l'utilisation** :

- Création de groupes de conversation;
- Adapter au nombre de clients connectés l'intervalle entre deux requêtes getMessage (requêtes vérifiant si des messages ont été reçus sur le serveur), afin de ne pas surcharger le serveur;
- Etablissement de profils "privés" ou "publics", au choix de l'utilisateur. Chaque utilisateur pourrait alors rechercher et correspondre avec tous les utilisateurs "publics" connectés au serveur;
- Diffuser cette messagerie à grande échelle (WAN pour *Wide Area Network*).

En complément, passer par GitHub à la place de Pearltrees pour le partage de code pourrait faciliter nos actions futures.

> Stratégie de diffusion

Afin de **toucher un large public**, plusieurs **stratégies de diffusion** pourront être mises en place :

- **A l'échelle du lycée** : partage de la messagerie, débats ouverts engagés sur la cryptographie et TransCrypt. Promouvoir le projet via les différents clubs (informatique & algorithmique) susceptibles d'être directement concernés avant de le propulser au travers des groupes de conversation communs à chaque niveau.
- **A toutes les échelles :** Proposer des **approches pédagogiques** et souligner les points suivants :
    - **Valorisation de la confidentialité** des messages envoyés, alors que les nouvelles générations se tournent de plus en plus vers ce type d'applications (voir la croissance de Télégram).
    - Mise en avant de la **transparence du projet**, ainsi que de son **origine française.** Ce projet porte en effet une dimension forte quant à la **souveraineté nationale** qu'il permettrait –Historiquement, conceptions et implémentations des chiffrements appartiennent toutes deux à des entreprises étrangères, notamment nord-américaines et a fortiori, à leurs états. Le projet est également synonyme d'une solution gratuite et grand public.

> Analyse critique du résultat

Nous avons été assez **efficaces** lors de la réalisation de ce projet. Nous avons finalement conçu un chiffrement novateur et développé un exemple d'implémentation fonctionnel. Cette mise en pratique d'une idée scientifique en un produit a été une riche expérience de travail. Ce projet aura également été pour nous l'occasion d'appréhender la méthode agile qui pourra potentiellement nous être nécessaire dans nos perspectives professionnelles.

Les logiciels, langages de programmation et modules utilisés se sont dans la plupart des cas révélés être adaptés. Nous aurions néanmoins pu dès le départ implémenter le chiffrement en javascript et non en python, afin de pouvoir directement l'utiliser dans la messagerie. Les fonctionnalités techniques du projet correspondent bien à nos attentes et ne nous ont pas déçues.