

---

PRÉ-RAPPORT

---

# Outil de lutte contre l'usurpation et le vol de photos privées

---

Victorine Cassé  
Fabio Gadegbeku  
Thomas Perel  
Nathan Stchepinsky

Télécom SudParis  
1<sup>ère</sup> année

16 février 2022



IP PARIS

# Table des matières

<b>I</b>	<b>Le projet</b>	<b>2</b>
I.	Introduction . . . . .	3
II.	Cahier des charges . . . . .	4
<b>II</b>	<b>Gestion de projet</b>	<b>6</b>
I.	Plan de charge . . . . .	7
II.	Planning prévisionnel . . . . .	7

# Première partie

## Le projet

## I. Introduction

La vie privée et la protection de la propriété intellectuelle sont des enjeux au coeur des récentes innovations. Devenus incontournables, ces deux piliers fondamentaux nécessitent l'adaptation de nos usages et de nos technologies qui sont parfois peu respectueux du droit à la vie privée.

Nous proposons donc dans ce rapport le détail de la protection à deux fléaux émergents : le vol de photos privées (*et leur diffusion sur les réseaux sociaux*) et le partage non-autorisé d'images protégées par le droit d'auteur. Afin de lutter contre ces deux dérives nous souhaitons mettre à disposition des particuliers et des propriétaires de photos, un logiciel capable de signer et filigraner (*de manière invisible macroscopiquement et irrémédiablement*) les dites images. En parallèle, nous proposerons aux réseaux sociaux une deuxième fonctionnalité servant à vérifier la signature, ou non, d'une image. Ces derniers pourraient alors en interdire la publication d'images ou la capture d'écran si la personne n'en est pas propriétaire.

## II. Cahier des charges

Le coeur de notre projet est la création d'un logiciel de signature et de vérification de signature d'images. Il devra donc contenir les fonctionnalités suivantes :

- Signer une image pour un unique auteur.
- Vérifier la signature d'une image et de la légitimité de l'utilisateur.

Ces fonctionnalités seront premièrement implantées en `python` et disponibles via un terminal. En second temps, elles seront accessibles via une interface graphique.

Les contraintes techniques sont les suivantes :

- Langage de programmation utilisé : **python**
- La photo signée devra être produite en **PNG**. *Le logiciel devra donc être muni d'un outil de conversion.*
- La signature sera incorporée à l'image afin d'en éviter la suppression
- L'interface (terminal) de l'application devra être conforme à la maquette suivante :

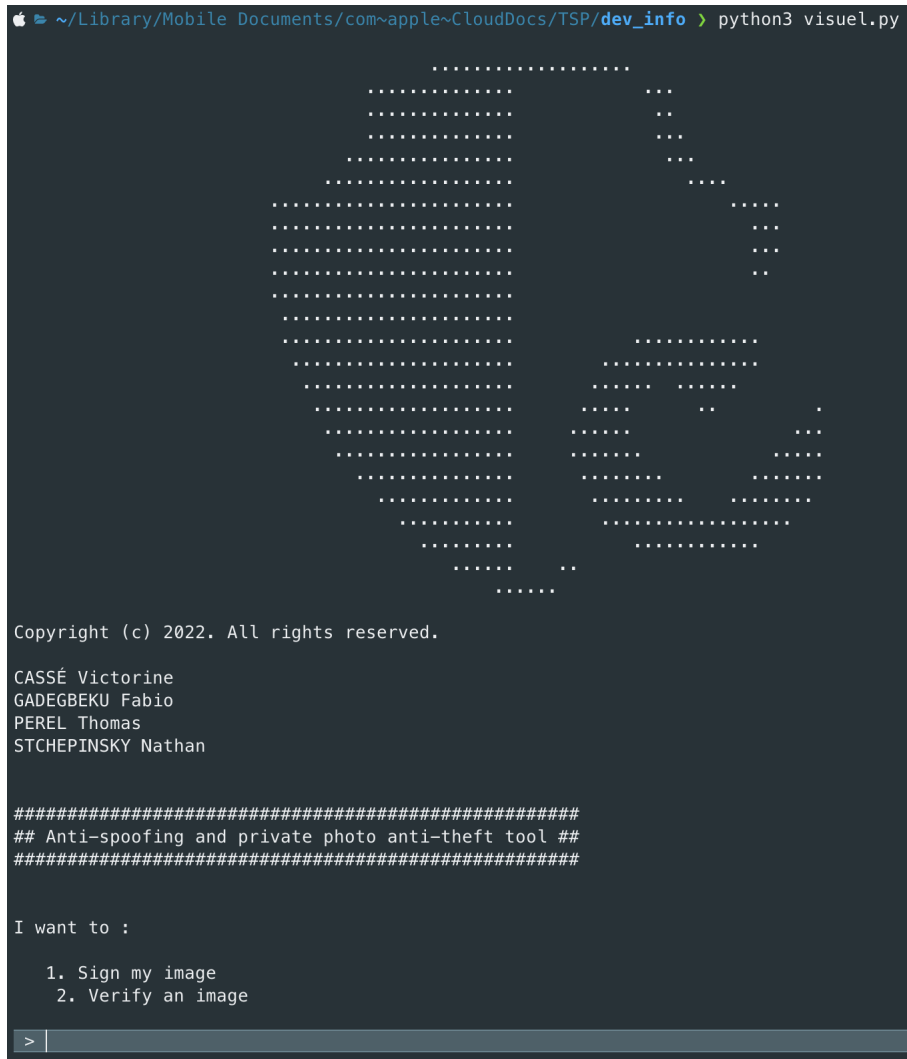


FIGURE 1 – Maquette de l'interface de l'application (version terminal)

De même, dans un temps ultérieur, nous envisagerons la création d'une interface web, proposant les même fonctionnalités que l'application en local. Cette interface exécutera ensuite notre logiciel python hébergé sur un serveur Apache.

# Deuxième partie

## Gestion de projet

## I. Plan de charge

Plan de charges prévisionnel							Suivi d'activités (Charge consommée)					
Description de l'activité	Charge (en %)	Charge (en h)	Charge (en h/Participant)				Charge en %	Charge en h	Charge en h / Participant			
			Victorine	Fabio	Thomas	Nathan			Victorine	Fabio	Thomas	Nathan
<b>Total</b>	<b>100,00%</b>	<b>220</b>	<b>55</b>	<b>55</b>	<b>55</b>	<b>55</b>	<b>10,50%</b>	<b>200</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>6</b>
<b>Gestion de projet</b>												
Réunion de lancement		4	1	1	1	1		1	1	1	1	1
Planning prévisionnel et suivi d'activité		4	1	1	1	1		1	1	1	1	1
Réunion de suivi		24	6	6	6	6		1	1	1	1	1
Rédaction		13	4	4	1	4		0	0	0	1	1
Github		9	1	1	4	3		0	0	0	0	0
<b>Spécification</b>												
Définition des fonctionnalités		8	2	2	2	2		2	2	2	2	2
<b>Conception préliminaire</b>												
Énumération des méthodes des méthodes stéganographiques		12	3	3	3	3						
Tests avec différents formats/taille des images		8	2	2	2	2						
Choix de la méthode stéganographique		4	1	1	1	1						
Choix de la méthode de chiffrement		4	1	1	1	1						
<b>Conception détaillée</b>												
Définition des classes		4	1	1	1	1						
Définition des méthodes		8	2	2	2	2						
Définitions des tests unitaires		8	2	2	2	2						
Auto-formation		14	4	5	5	0						
Maquettage des interfaces		6	2	2	2	0						
<b>Codage</b>												
Codage des classes		5	2	1	1	1						
Codage des méthodes		35	8	8	8	11						
Codage des tests unitaires		14	3	3	3	5						
<b>Intégration</b>												
Intégration des modules		4	1	1	1	1						
Tests d'intégration		16	4	4	4	4						
<b>Soutenance</b>												
Préparation de la soutenance		12	3	3	3	3						
Soutenance		4	1	1	1	1						

FIGURE 1 – Plan de charges

## II. Planning prévisionnel

### Février

- **Spécification** : Définition des fonctionnalités en accord avec le cahier des charges pré-établi.
- **Conception préliminaire** : Énumération la plus exhaustive possible des différentes méthodes stéganographiques. Documentation sur leurs avantages et vulnérabilités.
- **Conception détaillée** : Auto-formation
- **Conception préliminaire** : Compte tenu des différentes méthodes énumérées, détail des contraintes appliquées aux formats ainsi qu'aux tailles des images.
- **Conception préliminaire** : Choix de la méthode stéganographique la plus adaptée. Ce sera celle retenue pour le projet



## Mars

- ▶ **Conception préliminaire** : *Choix de la méthode de chiffrement en accord avec la méthode stéganographique retenue*
- ▶ **Conception détaillée** : *Auto-formation*
- ▶ **Conception détaillée** : *Définition des classes*
- ▶ **Conception détaillée** : *Définition des méthodes*
- ▶ **Conception détaillée** : *Définition des test unitaires*
- ▶ **Codage** : *Codage des classes*
- ▶ **Codage** : *Codage des méthodes*
- ▶ **Codage** : *Codage des tests unitaires (ébauche)*

## Avril

- ▶ **Codage** : *Codage des classes*
- ▶ **Codage** : *Codage des méthodes*
- ▶ **Codage** : *Codage des tests unitaires*
- ▶ **Intégration** : *Intégration de l'application*

## Mai

- ▶ **Intégration** : *Tests d'intégration*
- ▶ **Soutenance** : *Préparation de la soutenance*
- ▶ **Soutenance** : *Passage de la soutenance*