*FERREIRA*
*Tony*
*SISR 1*

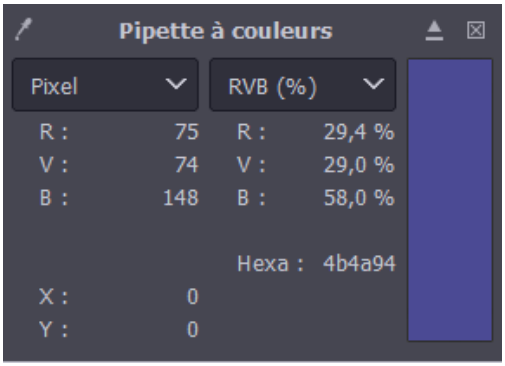
# TP N°1 Images et sécurité informatique / Sténographie :

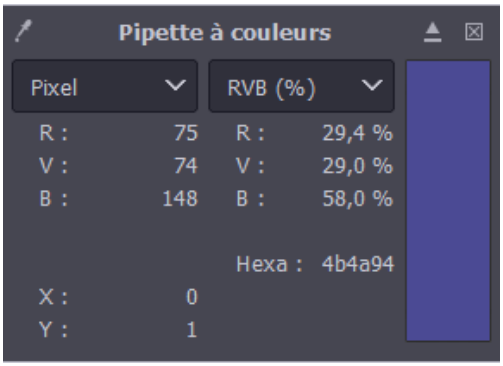
## *1/ Couleur d'un pixel (2pts)*

Le code hexadécimal (HTML) de la couleur du pixel (252, 214) est 581D1A.

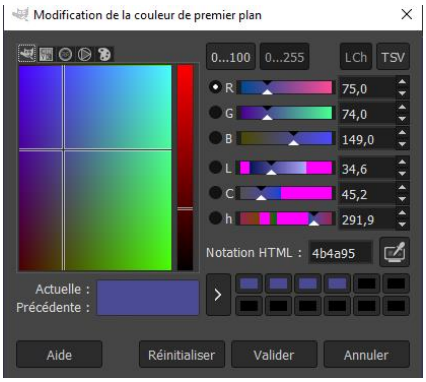
## *2/ Description du procédé stéganographique (4pts)*

1. Effectivement les deux points de coordonnées (0,0) et (0,1) sont exactement de la même couleur.




2. Modification sur l'image de la couleur du pixel (0,0) en ajoutant 1 à la composante bleue.




3. Non je ne vois aucune différence de couleur avec le pixel voisin à l'œil nu, même en zoomant au maximum.

*FERREIRA*
*Tony*
*SISR 1*

# ***3/ Retrouver un message (8pts)***

Les valeurs des composantes bleues des 8 premiers pixels sur la première ligne (ligne d'ordonnée 0) convertis en binaire dans l'ordre :

- (0.0) = 148 = 10010100
- (0.1) = 148 = 10010100
- (0.2) = 148 = 10010100
- (0.3) = 148 = 10010100
- (0.4) = 148 = 10010100
- (0.5) = 149 = 10010101
- (0.6) = 148 = 10010100
- (0.7) = 148 = 10010100

En isolant le bit de poids faible de chaque pixel et en les assemblant ensemble on obtient un nouveau nombre binaire qui est 00000100.

En version décimal / hexadécimal ce nombre binaire est égal à 4.

Le nombre de pixels à analyser sur la ligne d'ordonnée 1 est égal à 8 x /, / étant égal à 4 nous devons analyser 8 x 4 donc 32 pixels pour découvrir le message caché.

Nous allons maintenant utiliser la même méthode utilisée pour découvrir / afin de récolter les 4 codes binaires cacher dans les 32 pixels.

- 01010100
- 01000010
- 00100000
- 00100001

Maintenant que nous avons trouver les 4 codes binaire nous allons utiliser la table de codage ASCII pour révéler le message.

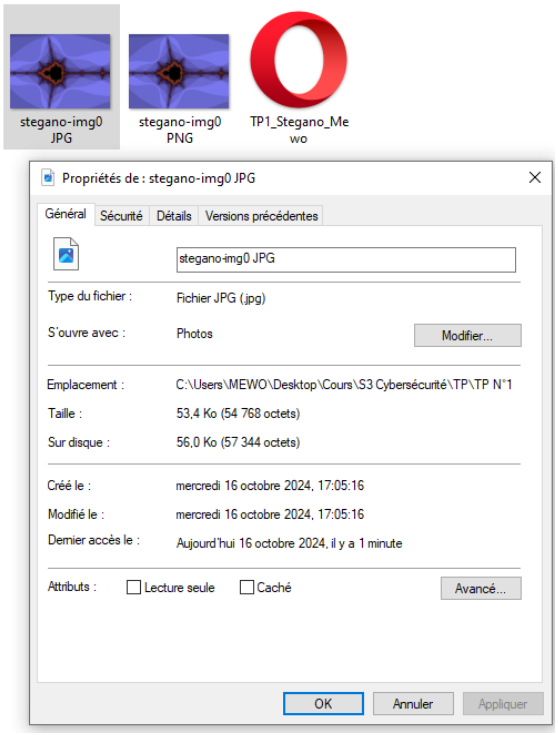
- (0) 101 / 0100 = T
- (0) 100 / 0010 = B
- (0) 010 / 0000 = SP (Space)
- (0) 010 / 0001 = !

Le message caché est donc « TB ! »

*FERREIRA*
*Tony*
*SISR 1*

# ***4/ Choix du format de sauvegarde du fichier (4pts)***

1. J'enregistre la photo stegano-img0.png en format .jpg.

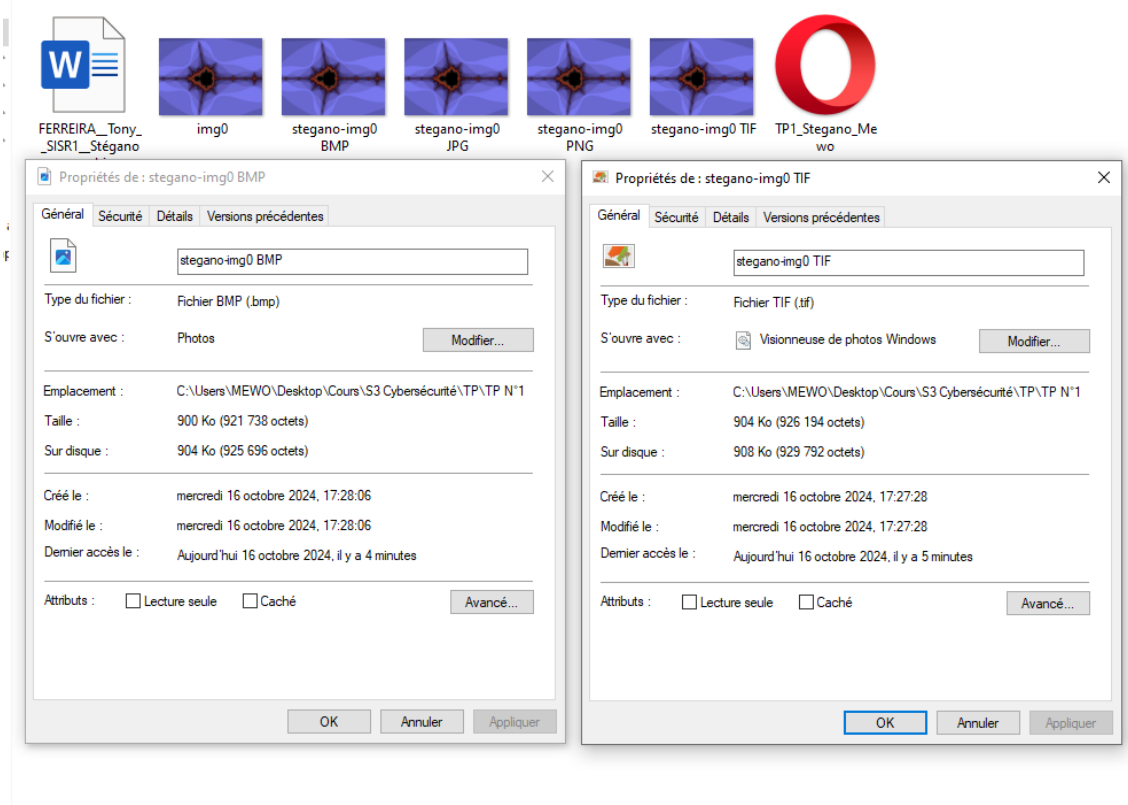



2. J'ai chargé l'image en format .jpg dans le logiciel.
   Je peux constater qu'il est impossible de retrouver le message cacher, la photo n'est plus aussi « précise » qu'avant, on pourrait dire que l'image a été lisser.
   La photo est moins détaillée.

3. L'image en format .jpg est plus grande que celle en version .png, 53,4 Ko contre 47,2 Ko.
   Je trouve cela étonnant sachant que la version .jpg est moins détaillée et moins « lisible » comparé à la version .png.

4. J'ai pu essayer plusieurs autres formats, beaucoup ne fonctionnent pas car ils « effacent » les détails de l'image et nous perdons le message caché dans les bits.
   J'ai trouvé que les formats .tif et .bmp fonctionnent et n'effacent pas les détails des bits en compressant l'image.

*FERREIRA*
*Tony*
*SISR 1*




## ***5/ Vers l'infini et au-delà ! (2pts)***

Les différents types de fichier pouvant cacher des informations :

- Images
- Audio
- Vidéos
- Documents (Word/PDF/…)

Récemment, le groupe cybercriminel TA558 a mené une campagne d'attaques ciblant diverses organisations en Amérique latine, exploitant la stéganographie pour dissimuler du code malveillant dans des fichiers. Nommée SteganoArmor, cette opération utilise des fichiers Word apparemment inoffensifs pour contourner les solutions de sécurité.

TA558 s'appuie sur une vulnérabilité ancienne de Word et Excel (CVE-2017-1182) pour injecter du code malveillant dans des systèmes non mis à jour. Les emails malveillants contiennent des fichiers qui, une fois ouverts, téléchargent automatiquement un script Visual Basic (VBS) depuis un service de stockage en ligne. Ce script récupère ensuite un fichier JPG

cachant une charge utile malveillante, entraînant l'installation de divers malwares, dont AgentTesla, FormBook, et LokiBot.

Les attaques ont principalement ciblé des secteurs industriels et des services publics en Amérique latine, mais ont également touché des entreprises en Russie, Roumanie et Turquie.

Source : https://www.informatiquenews.fr/la-steganographie-fait-son-retour-en-technique-d-attaque-98772