AL RASHEED Hamze
BOUVIER Rémi
M2-INFO

Info002 : Cryptologie
Le challenge du logo ANSSI

Table des matières

1. Contexte et présentation

Le challenge du logo ANSSI est un challenge publié par l'ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information), un service Français crée en juillet 2009, rattaché au secrétariat général de la Défense et de la Sécurité Nationale, ayant pour but d'assister le Premier ministre dans l'exercice de ses responsabilités en matière de défense et de sécurité nationale. Le logo de l'ANSSI est dévoilé pour la première fois le 7 octobre 2011 par Patrick Pailloux lors des Assises de la sécurité. Il est ensuite publié sur le site de l'ANSSI le 3 février 2012 et contient le challenge dès sa publication. Ce logo est aujourd'hui encore le même qu'à l'époque. Avec la publication de ce logo, une phrase est également publiée : "Les curieux apprécieront les fonds d'écran qui ont également été réalisés. " Cette phrase est en fait un moyen de faire comprendre que le logo cache un défi à relever, qui a été imaginé pendant une vingtaine de semaines par des experts de l'ANSSI.

Quelques jours après la publication, certaines personnes remarquent que dans les fonds d'écran dont parlent cette phrase se cache un challenge de sécurité et la nouvelle commence à se répandre sur internet et les réseaux sociaux. Plusieurs personnes commencent donc à s'intéresser à la recherche de solution pour résoudre ce challenge. Il s'avère qu'il fait en fait appel à trois principaux types de compétences :

- La stéganographie, l'art de la dissimulation : faire passer inaperçu un message dans un autre message
- Le chiffrement
- La rétro-ingénierie : analyse d'un système dans le but d'en créer une représentation à un plus haut niveau d'abstraction.

Certains morceaux de solutions commencent à être partagés un peu partout, notamment par des chercheurs. Un site est alors créé : http://anssi.santo.fr/. Il regroupe toutes les solutions trouvées par les différentes personnes s'intéressant à ce challenge et devient le site de référence. Deux des énigmes du challenge sont plutôt complexes. On observe peu d'avancées sur ces énigmes et l'engouement pour le challenge diminue donc progressivement au fur et à mesure du temps et plutôt rapidement (plus vraiment de personnes publiant de nouvelles solutions à partir de fin Février 2012, soit 1 mois seulement après le début du challenge). En octobre 2012, l'ANSSI communique pour la première fois sur le challenge en annonçant que "le challenge glissé dans [leur] logo n'est toujours pas résolu". L'ANSSI dévoile pour la première et seule fois qu'il y a bien un challenge dans leur logo, mais cela n'a pas de réel impact et ne relance pas l'engouement autour de sa résolution.

Il aura finalement fallu 2 années pour qu'un ingénieur en Recherche et Développement, Pierre Bienaimé, propose une solution complète à ce challenge et la publie sur son blog. Dans la suite de ce document, nous allons revenir sur l'ensemble des étapes qu'il a franchies afin de résoudre ce challenge. Nous commencerons par les premières constatations qui ont été faites et nous verrons ensuite pour chaque partie du problème, quels ont été les défis rencontrés et comment ils ont été résolus.

2. Premières constatations

a. Ce qui saute aux yeux

Figure 1 : Image du challenge ANSSI

Le challenge est une image (*figure 1*) au format PNG qui représente le logo de l'ANSSI sur fond noir. En l'examinant de plus près (*figure 2*), on remarque que le cercle intérieur est agrémenté de chaîne de caractères suspectes, séparées en 3 blocs.

Figure 2 : Zoom sur le cercle intérieur du logo

Le premier bloc est composé de caractères hexadécimaux. Pour comprendre de quoi il s'agit, il suffit de compter chaque octet, et comme ils sont presque uniformément distribués, on en conclut que les données sont certainement chiffrées.

Le deuxième bloc est constitué des trois mots « AUTH : DE9C9C : PCA ». Le logo est l'œuvre de Pierre Capillon, la chaîne PCA corresponde donc à ses initiales. Quant à DE9C9C55, cette chaîne permit à quelqu'un de remonter jusqu'au CV de Pierre Capillon. On suppose que la chaîne est un hash et comme il fait 4 octets, on pense à CRC32.

Le troisième bloc est celui qui permet d'avancer. On reconnaît un encodage base64. Une fois la chaîne décodée, cela produit la phrase suivante : « Le sourire de la Joconde cachait bien des mystères… ». On apprend que la technique utilisée par Léonard de Vinci pour créer l'effet vaporeux de son sourire s'appelle le ***sfumato.*** Le principe est de superposer plusieurs fines couche de peinture de différentes couleurs. On peut alors imaginer qu'en jouant avec les couleurs du fond d'écran, il sera possible de révéler un contenu secret.

b. Découverte des couches

Il est possible d'ouvrir le fichier PNG avec un éditeur d'images et de pousser la luminosité et le contraste au maximum (figure 3).

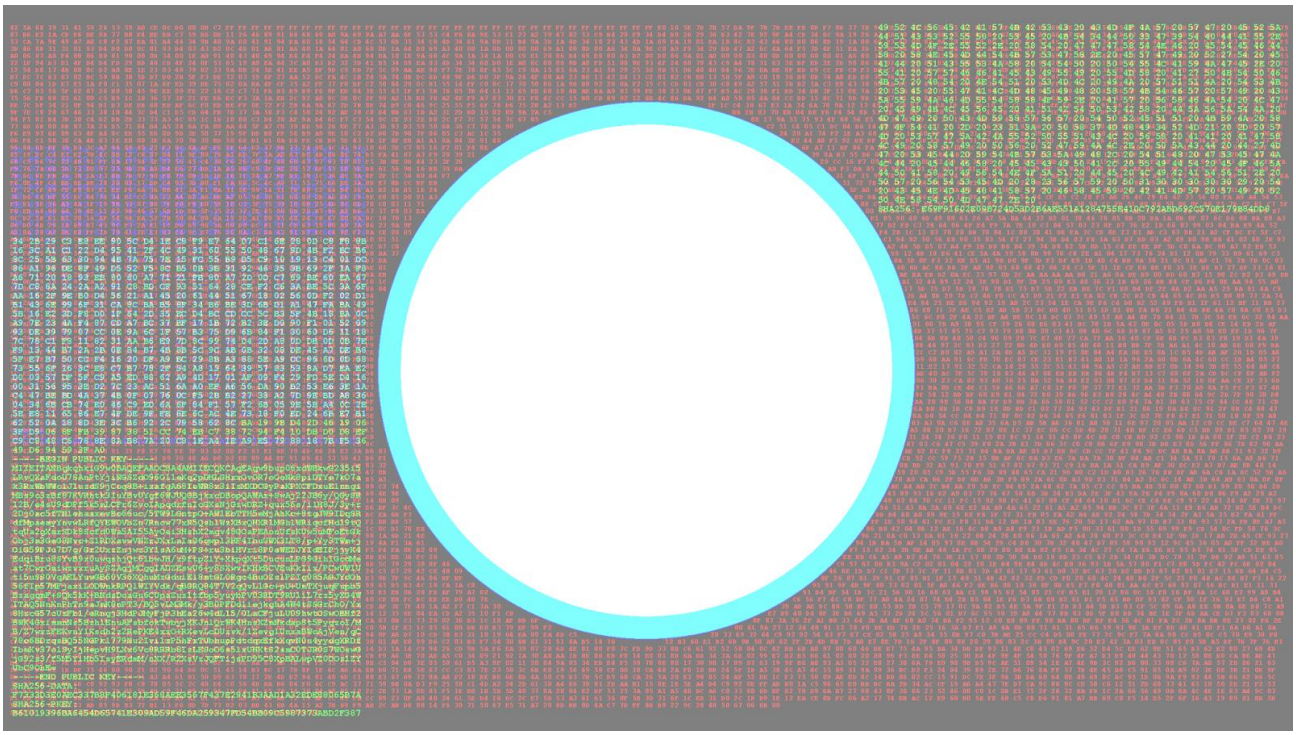

Figure 3 : Découverte des différentes couches

Des caractères hexadécimaux rouges, verts et bleus apparaissent, à l'exception d'un bloc encodé en base64 qui porte la mention BEGIN PUBLIC KEY. Il est possible de distinguer quatre parties : un bloc vert en haut à droite, un double-bloc vert à gauche, un bloc bleu à gauche et un bloc rouge qui couvre toute la surface de l'image. La couche bleu clair que l'on aperçoit sur la gauche n'existe pas réellement. Il s'agit simplement de caractères de la couche verte et de la couche bleue qui sont superposés. Il y a donc de la stéganographie dans l'image.

On peut maintenant recréer une image en ne conservant que les bits de poids faible des composantes de chaque pixel. Cela permet de constater visuellement qu'une image contient de la stéganographie. Dans le cas du logo, l'image générée (figure 4) à partir des deux bits de poids faible dévoile directement les couches.

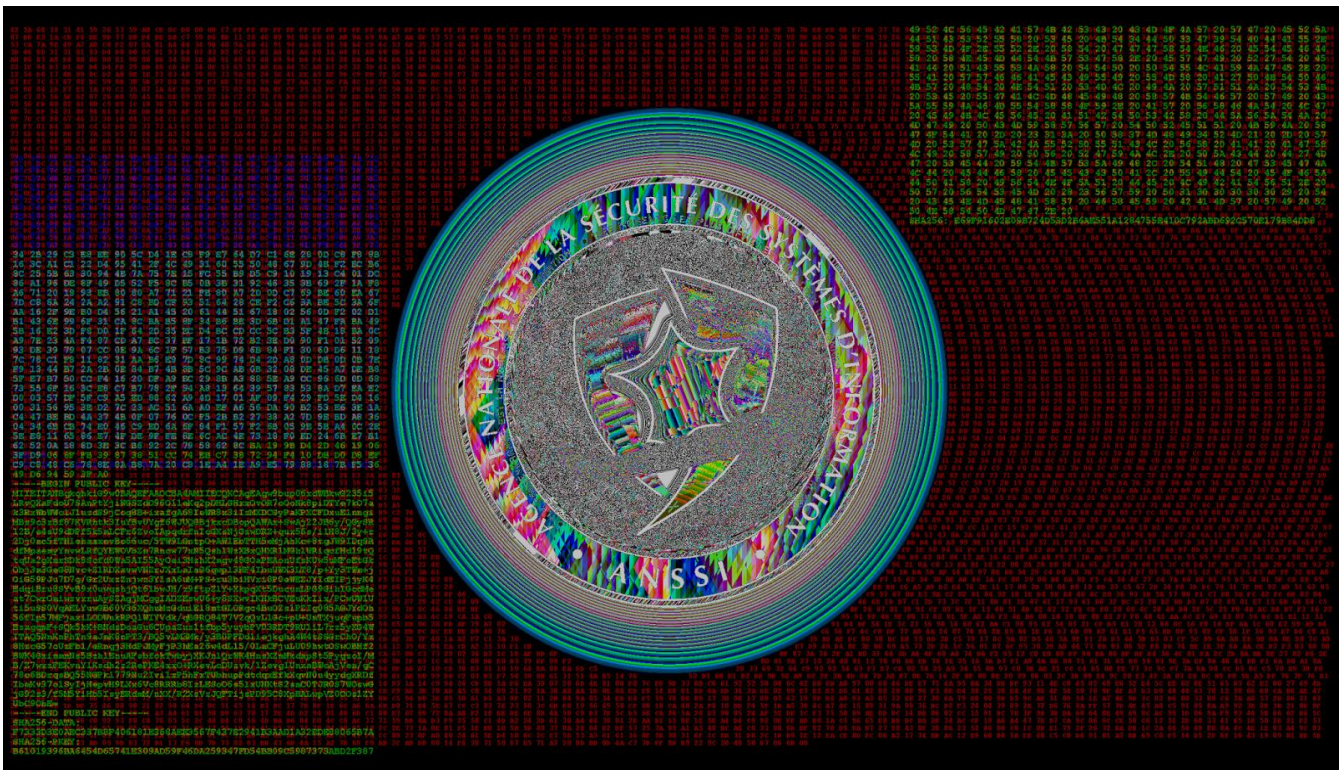

Figure 4 : Image LSB

L'avantage de cette représentation est qu'elle révèle des contours qui sont presque invisibles dans le fond d'écran d'origine. En examinant attentivement le cercle intérieur, deux citations se dessinent : « La persévérance est la noblesse de l'obstination » (Adrien Decourcelle) et « Les moments les plus difficiles sont ceux qui donnent le plus de satisfaction » (Claude Lelouch).

c. OCR : Logiciel de reconnaissance de caractères

On utilise à présent un logiciel de reconnaissance de caractères afin d'extraire les données de chaque couche. Le but de cet OCR est simplement de gagner du temps, car il est possible (mais horriblement rébarbatif) de réaliser cette extraction à la main. Pour cette étape, les concepteurs du challenge sont plutôt cléments puisqu'ils fournissent les "sha256" de chaque bloc, ce qui permet que l'OCR fasse correctement son travail. Après avoir isolé les couches, il convient d'appliquer quelque pré-traitement pour augmenter le taux de réussite de l'OCR (conversion en noir et blanc, découpage propre des zones de texte…). Chacune des images / couleurs semble contenir une énigme différente à résoudre (Figure 5).

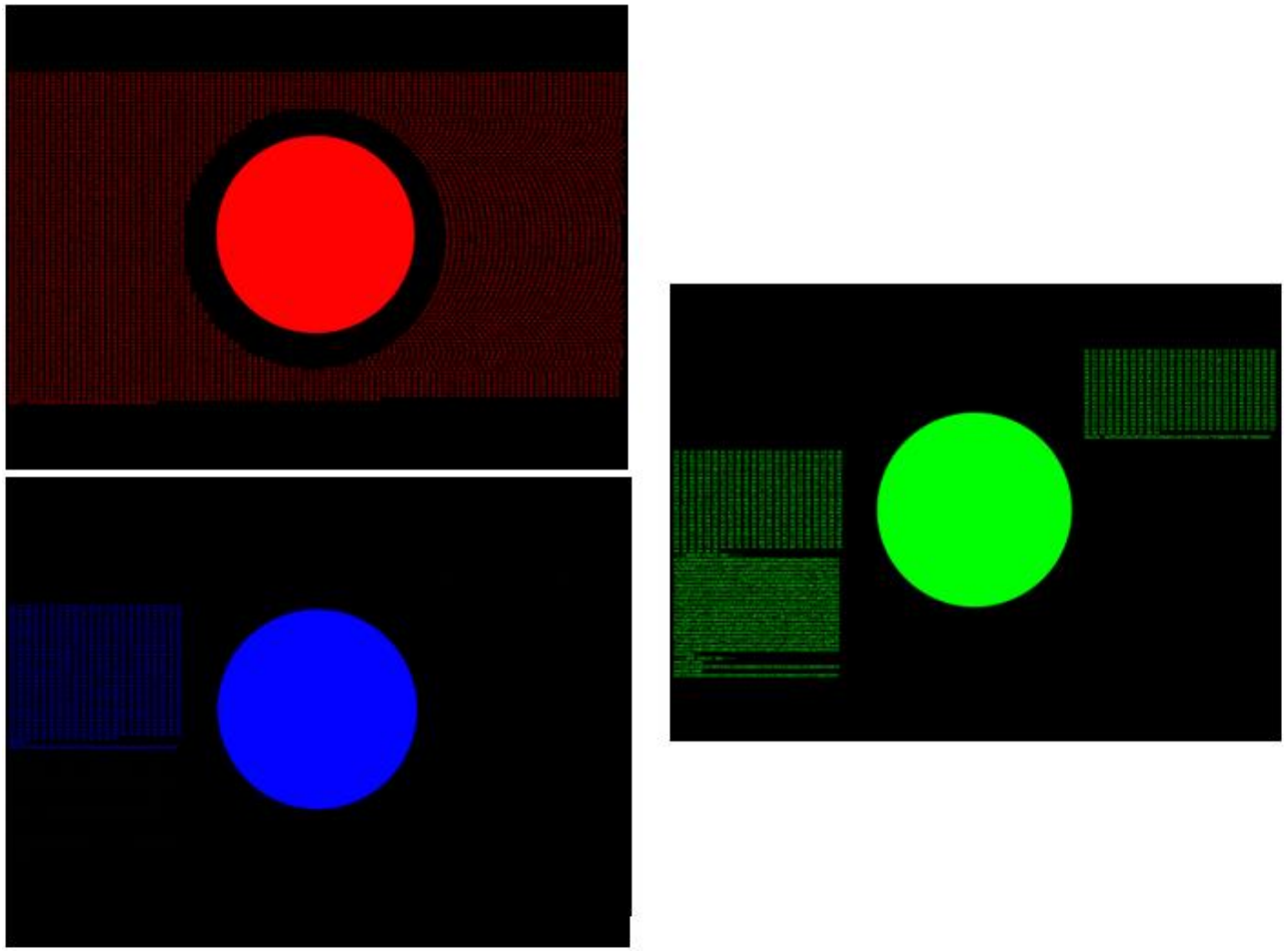

Figure 5 : Images après avoir appliqué l'OCR

3. Résolution des énigmes de chaque couleur

a. Vert (droite)

L'énigme verte donne le texte ci-dessous :

```
IRLVEBAWKBSC CMOJW WG ERZDQCSRUX SE KT4DP3G9T@DAU.YSMO.UR. XT
GGGXTNF ETEFDX XNEMDTKWSGX. EWGI R'T EAD QCUSJX TTP PTULAYJGE.
UA WWFFAECIUI UMX A'PKTPFKW HT NTQ SML IJ WQQJ TSK SE
UGALMHEIH XWKTFW WI CZUYJFMUTXXOY. AW VXFJT LG EIKLEVE
AQBTPSBX DZVZTJ MGI PCMYXWVW TPREQQ KYJ XGOTA - #1: PX7MHI4RM!
- WM SWGZBJURPUQCL VX AA AGXLI XWI PV RGYJL. PZCD D'MG SED
YTKWSZIH, TQH GSEGJLD EDFX EECCPA, UIDT EOFZDPAX IVTNOZQ DE
LIBATVQ. PW VTSEM (#VWY P100000) T CENMEHAXW FXEY BAMW WI
RPNXTPMGG.
```

Les mots du message ne veulent rien dire, mais la ponctuation à l'air d'être positionnée à des endroits plutôt cohérents. Le challenge étant organisé par l'ANSSI, les chercheurs ayant travaillé sur le challenge ont commencé par considérer que le texte était en langue Française. Pour déterminer quel type de chiffrement a été utilisé, il est pertinent de faire une analyse fréquentielle du message vert. On obtient les résultats suivants :

```
>>> char_frequency(green)
Counter({'t': 7.775, 'e': 7.239, 'w': 6.434, 'x': 6.434, 'g':
5.362, 'a': 5.094, 'p': 4.826, 'i': 4.558, 'm': 4.558, 'd':
4.021, 's': 4.021, 'u': 3.753, 'c': 3.217, 'j': 3.217, 'q':
3.217, 'f': 2.949, 'l': 2.949, 'v': 2.949, 'k': 2.681, 'r':
2.681, 'y': 2.681, 'z': 2.413, 'b': 1.877, 'h': 1.877, 'o':
1.609, 'n': 1.609})

>>> FRENCH_FREQUENCY
Counter({'e': 17.124, 'a': 8.122, 's': 7.948, 'i': 7.58, 't':
7.244, 'n': 7.095, 'r': 6.553, 'u': 6.369, 'l': 5.456, 'o':
5.387, 'd': 3.669, 'c': 3.345, 'p': 3.021, 'm': 2.968, 'v':
1.628, 'q': 1.362, 'f': 1.066, 'b': 0.901, 'g': 0.866, 'h':
0.737, 'j': 0.545, 'x': 0.387, 'y': 0.308, 'z': 0.136, 'w':
0.114, 'k': 0.049})
```

Si la fréquence des lettres du texte est proche de la fréquence des lettres moyenne dans la langue française, alors le chiffrement est une simple permutation. Si l'on retrouve la même courbe de répartition mais avec des lettres différentes, alors il s'agit d'un chiffrement par substitution monoalphabétique. Ici, on ne retrouve aucun de ces 2 cas. La fréquence des lettres est lissée et les extrêmes ont disparu : on peut remarquer l'effet d'un chiffrement par substitution polyalphabétique si l'on a l'habitude de travailler avec les chiffrements. Il en existe plusieurs, mais les personnes travaillant sur le challenge ont trouvé pertinent de commencer par réfléchir sur le chiffre de Vigenère, étant le chiffrement par substitution polyalphabétique le plus connu et le moins complexe.

On rappelle rapidement le principe du chiffre de Vigenère : choisir une clé et l'utiliser de manière cyclique pour chiffrer un texte. Cette technique de chiffrement est donc vulnérable aux attaques par analyse fréquentielle, mais on peut également deviner juste un morceau du texte clair afin de recalculer directement la clé. Le texte chiffré contient une chaîne qui peut être facilement devinable. KT4DP3G9T@DAU.YSMO.UR ressemble en effet à une adresse mail. Les adresses mail du personnel de l'ANSSI utilisent le domaine ssi.gouv.fr, ce qui donne 9 lettres de la clé : LIMSESTPA. Les chercheurs travaillant sur le challenge ont trouvé que le mot ressemblait à un mot faisant partie du vocabulaire de la cryptologie : PALIMPESTE (manuscrit sur lequel on a fait disparaître les inscriptions pour pouvoir y écrire de nouveau). Quand Pierre Bienaimé essaye de déchiffrer le texte avec cette clé, il obtient bien un premier mot existant, "transmission" mais le reste du texte n'est pas correctement déchiffré. Il découvre que c'est en fait une variante du chiffre de Vigenère qui a été utilisée où une lettre de la clé est consommée quand on passe sur un caractère qui n'est pas une lettre. Le texte clair, déchiffré avec cette variante du chiffre de Vigenère et la clé PALIMPESTE est donc :

```
"TRANSMISSION RECUE EN PROVENANCE DE CH4LL3N9E@SSI.GOUV.FR. LE CONTENU
SEMBLE INTERESSANT. TOUT N'A PAS ENCORE ETE DECHIFFRE. IL SEMBLERAIT QUE
L'ECHANGE DE CLE AIT EU LIEU PAR DE MULTIPLES MOYENS DE COMMUNICATION.
LE DEBUT DU MESSAGE SEMBLAIT DONNER UNE PREMIERE PARTIE SUR TROIS - #1:
AX7EVT4NU! - LE DECHIFFREMENT DE LA SUITE EST EN COURS. POUR L'UN DES
MESSAGES, LES CALCULS SONT LANCES, CELA POURRAIT PRENDRE LA SEMAINE. LE
RESTE (#REF A100000) A NECESSITE BIEN PLUS DE REFLEXION."
```

Ce message contient plusieurs informations, qui ne sont pas forcément claires pour l'instant. La partie importante est qu'il contient le premier secret, **AX7EVT4NU!** et qu'on sait qu'il y en aura trois au total. Il donne également quelques indices sur l'obtention des deux autres secrets.

b. Bleu

Si on essaie de décoder la partie bleue, on obtient un texte qui a l'air d'être chiffré. Comme pour la partie précédente, on voit ce que donne une analyse fréquentielle sur ce texte. On obtient une fréquence des lettres proche de celle de la langue française : il s'agit donc probablement d'une permutation des lettres. Il faut donc trouver une méthode pour retrouver le bon ordre des lettres. Les algorithmes usuels (lire un caractère sur deux, trois, un par ligne…) ne donnent pas de résultats. Les retours à la ligne n'ont pas de sens, la permutation ne s'applique pas qu'aux lettres. On remarque aussi que les permutations ont l'air d'être plutôt locales : un caractère chiffré ne se retrouve pas trop éloigné de sa position de départ.

L'ingénieur Pierre Bienaimé décide de revenir sur les indices récupérés précédemment et notamment sur l'indice suivant : "LE RESTE (#REF A100000) A NECESSITE BIEN PLUS DE REFLEXION." En cherchant la chaîne A100000 sur Google, on tombe sur OEIS, l'encyclopédie en ligne des suites de nombres entiers. Le lien nous renvoie vers la séquence d'entiers suivante : [3, 6, 4, 8, 10, 5, 5, 7]. On a donc un chiffrement par permutation et une suite d'entiers. Pierre Bienaimé remarque la présence de deux guillemets, d'un tiret et des lettres majuscules **L, J, M et H** dans les trois dernières lignes du texte. Il fait la supposition qu'il s'agit d'une citation (il y a déjà eu d'autres citations dans les autres parties du challenge). Il cherche donc un auteur ayant un prénom composé contenant les initiales J, M et H. Il tombe sur José-Maria de Heredia, un poète Français (ce nom peut bien être reconstitué grâce aux lettres des 3 dernières lignes du message chiffré). Il reste donc quelques lettres sur ces 3 dernières lignes qui permettent de retrouver le titre du texte original : "Les conquérants".

Si on compare les caractères du texte chiffré et du texte original, les caractères qui diffèrent sont "-#3", ce qui veut dire que le secret 3 est un des mots du poème. Même avec le texte chiffré, la séquence d'entiers et le texte clair, Pierre Bienaimé n'arrive pas à retrouver l'algorithme de permutation correspondant. Néanmoins, du fait que les secrets 1 et 2 font chacun 10 caractères, il en déduit que le secret 3 fait lui aussi 10 caractères. Il n'y a qu'un seul mot de 10 lettres se trouvant proche de "#3" dans ce poème. Il suppose donc que le secret 3 est le mot "**caravelles**." Il a donc réussi à déterminer le secret 3 sans parvenir à trouver l'algorithme de permutation utilisé pour chiffrer le poème.

c. Rouge

La partie rouge est une archive bz2. Une fois décompressée, elle nous donne 256 kio de données non identifiées. Comme d'habitude, pour se faire une idée de ce que l'on manipule, on compte les octets :

```
>>> char_frequency(red, False)
Counter({'\x00': 95.582, '\xbd': 0.403, '\xff': 0.311, '>':
0.126, '\x99': 0.119, '\xe7': 0.111, '\xdf': 0.101, '2': 0.074,
'\xdc': 0.071, '\x07': 0.07, ...
```

Le fichier est composé à 95% d'octets nuls, 0.4% de 0xbd et 0.3% de 0xff. Les autres octets sont présents en faible quantité. Ces proportions sont relativement cohérentes avec la thèse d'une forme binaire inconnue, car pour beaucoup d'entre eux, 0x00 et 0xff sont les deux octets les plus fréquents. La commande string ne relève rien de très excitant, à part ce qui pourrait être appelé un alphabet :

```
$ strings red
zyxwvutsrqponmlkjihgfedcba`_^]\\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<
;:9876543210/.-,+*
```

Cette liste d'alphabet est à l'envers, on tente alors, en vain, de lire le fichier en partant de la fin. Finalement, l'opération correcte qui va remettre cet alphabet dans le bon sens est un NOT sut tout le fichier. Le résultat est le suivant :

```
$ file red.not
'Gameboy ROM: "R", [ROM+MBC1+RAM], ROM: 2Mbit, RAM: 128Kbit'
```

Le résultat nous parle d'une ROM de gameboy. On remarque d'ailleurs que cette cartouche ne contient pas que de la ROM, mais également de la RAM et un MBC (Memory Bank Controller). On lance cette ROM dans un émulateur Game Boy, une fois le jeu lancé, l'écran affiche « Action ? » et reste bloqué. En entrant un Konami code (« Haut, Haut, Bas, Bas, Gauche, Droite, Gauche, Droite, B, A »), cela débloque un écran secret. Un clavier virtuel apparaît et on a la possibilité de rentrer trois mots, puis la chaîne "Enjoy" s'affiche à l'écran accompagnée de 32 octets sous forme hexadécimale (*figure 6*), on comprend rapidement que le jeu n'est pas une énigme, mais plutôt une interface dans laquelle il faudra valider les trois secrets des autres épreuves. En tentant plusieurs dizaines de combinaisons, on s'aperçoit que la sortie ressemble à un mélange des trois mots passés en entrée. Par exemple, en entrant ABC, abc et 123, la sortie commence par 0x41 0x61 0x31, c'est-à-dire les codes ASCII de la 1ère lettre de chaque secret. La suite est 0x32 0x42 0x62 (2-B-b), la 2ème lettre des secrets mais cette fois dans le désordre. La fin est 0x70 0x20 0x50 (p-espace-P), des

caractères qui ne font pas partie des mots en entrée. Le mélange varie en fonction des entrées d'une manière qui ne semble pas logique. Pour comprendre l'algorithme, il va falloir examiner le code assembleur z80 de la ROM.

Figure 6 : Test de l'algorithme de mélange de la ROM Game Boy

Nous avons le secret 1 ainsi qu'un bon candidat pour le secret 3. On peut appliquer le reverse de la ROM pour trouver 9 routines de mélange qui prennent en entrée un caractère de chaque secret et génèrent trois caractères en sortie. La plupart de ces routines ne font que permuter les caractères, mais d'autres font des XOR. Voici la sortie des 9 routines en question :

1. `c1, c2, c3`
2. `c1, c3, c2`
3. `c2, c1, c3`
4. `c2, c3, c1`
5. `c3, c1, c2`
6. `c3, c2, c1`
7. `c1^c2, c2^c3, c1^c3`
8. `c1^c3, c1^c2, c2^c3`
9. `c1^c2^0x42, c2^c3^0x42, c1^c3^0x42`

La seconde difficulté a été de déterminer comment est choisie la routine de mélange qui va s'appliquer à chaque triplet de lettre. C'est ici que la puce MBC et le concept de " Bank Switching" entrent en jeu, car le jeu utilise ce hack matériel pour accéder à plus de mémoire. Le code des routines est situé dans 9 banques différentes. Pour changer la banque courante, il faut écrire une valeur dans la ROM sur la plage d'adresse [2000-3FFF]. Cette tentative d'écriture est interprétée par la puce MBC qui va charger la banque correspondante de la cartouche. En plaçant un breakpoint en écriture sur cette plage, on peut remonter jusqu'au code qui sélectionne le numéro de routine à appliquer. Ce dernier est assez horrible puisqu'en assembleur z80, il n'y a pas de modulo : cette opération a été implémentée ici avec toutes sortes de bit-shifts. Le numéro de routine est le résultat du calcul :

(((c1 + c2 + c3) & 0xff) % 9) + 1
En n'utilisant toujours le triplet de l'étape N-1 (pour le premier triplet, c'est toujours la routine 1 qui est sélectionnée).

4. Vert (Gauche)

Cette partie est la partie la plus difficile du challenge et celle qui a posé le plus de problèmes aux personnes ayant essayé de le résoudre. En effet, le bloc vert gauche est une clé publique RSA accompagnée de données qui ont a priori été chiffrées avec. La clé RSA contient un module N de 4096 bits. Le record de factorisation RSA a été établi sur un N de 768 bits au moment de la parution de l'article. Il n'est donc techniquement pas possible de casser la clé publique pour retrouver la clé privée et qu'il va falloir chercher un autre moyen. Dans l'indice de l'énigme verte (droite), il est évoqué des calculs qui vont prendre une semaine, il y a donc peut-être un élément exploitable dans une des étapes de la génération de la clé publique mais il n'apparaît pas évident au premier abord.

De nombreuses choses sont alors proposées pour essayer de résoudre ce challenge, mais aucune n'aboutit à un résultat concret. Pierre Bienaimé, l'auteur de l'article, revient sur le challenge plusieurs mois après sa publication. Il regarde les solutions proposées et essaye d'en trouver d'autres, sans succès. Désespéré, il envoie un e-mail à l'adresse CH4LL3N9E@ssi.gouv.fr trouvée grâce à la première énigme en espérant recevoir un indice. Il reçoit une réponse sous forme de poème, accompagné d'un entier de 2049 bits. Le début de chaque vers du poème forme un acrostiche "Pollard P Moins Un", qui est un algorithme de factorisation d'entiers. L'auteur de l'article essaye donc de créer une implémentation de cet algorithme.

Dans une clé publique RSA, le module N est le produit de 2 grands nombres premiers P et Q. Factoriser N nous permet donc de pouvoir recalculer la clé privée. L'algorithme P-1 de Pollard a pour but de trouver un très grand multiple de P-1 (ou Q-1). Une fois trouvé, on peut retrouver P en appliquant le petit théorème de Fermat et en calculant un pgcd. Afin de trouver un multiple de P-1, on utilise la notion de friabilité : si P-1 est friable, il n'a que des facteurs premiers inférieurs à une borne B), et factorielle B est probablement un grand multiple de P-1. Le calcul de factorielle B peut néanmoins s'avérer coûteux. Il est possible d'optimiser le calcul en utilisant uniquement des nombres primaires (nombre sous la forme nombre premier puissance entier positif). Le code final de Pierre Bienaimé fait 10 lignes de Python et il lui a fallu 4 heures pour calculer N.

```
from sage.all import *
N = 5346...[4096 bits]
M = prime_powers(2**30)
i = count = 0
a = 2
while True:
      a = pow(a, M[i], N)
      i += 1
      count += 1
      if count == 100000:
      count = 0
      r = gcd(a-1, N)
      if r != 1:
            break
print r
```

Une fois ce code lancé et la clé privée recalculée, on peut lire le bloc chiffré suivant :

```
“ACCUSEZ RECEPTION DU MESSAGE CHIFFRE A ch4ll3n9e@ssi.gouv.fr
ENVOYEZ SON EMPREINTE POUR ACQUITTEMENT.
CODES ATTENDUS POUR VOTRE LIVRAISON.
- #2: yh%Jc/!23B -”
```

Le message n’est une fois de plus pas directement compréhensible, mais il contient la valeur du secret 2.

5. Solution du challenge : ADFGVX

Le secret 1 est **AX7EVT4NU!**, le deuxième est **yh%Jc/!23B** et le dernier est **caravelles**. Lorsqu’on les rentre dans le Game Boy, la sortie n'a rien d'exceptionnel :

```
AychXa\x12WEMif5\x15 /TeX\x15MlN23UeBs!\x00\x00
```

En passant en revue le challenge, on constate que la seule chose qui n'a pas encore été utilisée est la chaîne chiffrée du tout début, affichée autour du logo. De plus, le premier message parle d'un échange de clé. Peut-être que la mystérieuse chaîne que nous venons de récupérer est la clé qui va permettre de déchiffrer ces données. Elle fait 32 octets de long et l'algorithme le plus connu qui utilise des clés de 256 bits est AES. Bingo. Le déchiffrement AES-CBC du bloc de données génère une archive gzip datée du 26 octobre 2011. Nous connaissons donc avec exactitude le moment où le challenge a été inséré dans le logo.

```
$ file data
data: gzip compressed data, from Unix, last modified: Wed Oct 26
15:55:10 2011
```

Une fois l'archive décompressée, nous obtenons le fichier texte suivant :

```
FGAXAXAXFFFAFVAAVDFAGAXFXFAFAGDXGGXAGXFDXGAGXGAXGXAGXVFVXXAGXDDA
XGGAAFDGGAFFXGGXXDFAXGXAXVAGXGGDFAGGGXVAXVFXGVFFGGAXDGAXFDVGGA
```

Il s'agit d'un chiffre ADFGVX, un système de chiffrement utilisé par les Allemands pendant la première guerre mondiale. Il est relativement robuste, car il combine un chiffrement par substitution (utilisant une table aléatoire) et un chiffrement par permutation (utilisant un mot de passe). Il peut normalement être cassé en mélangeant analyse fréquentielle et force brute, mais c'est difficile sur un échantillon aussi court.
Cette chaîne s'avère être un exemple qui existe réellement. Ce message porte le nom de Radiogramme de la Victoire et a joué un rôle notable lors de la première guerre mondiale. Il a été déchiffré en 1918 par le français Georges Painvin. Le texte clair allemand est *« Munitionierung beschleunigen Punkt soweit nicht eingesehen auch bei Tag »* et sa traduction française est *« hâtez l'approvisionnement en munitions, le faire même de jour tant qu'on n'est pas vu »*. Ce message, le Radiogramme de la Victoire indique en fait que le challenge est terminé et donc que toutes les différentes étapes ont bien été résolues.