Chat d'Arnold

Cours écrit par jaudi

Sommaire du cours

Introduction

Chers membres de Club Alkindi,
Dans ce nouveau cours, nous allons nous plonger dans les coulisses d'un chiffrement atypique, mais qui a été inventé par un mathématicien renommé.
Je pense que la majeure partie de ce cours peut être accessible à un niveau collégien et lycéen, car j'ai essayé de bien expliciter le chiffrement, sans trop rentrer dans la théorie mathématique sous-jacente.
Ce cours pourra vous être utile pour certaines énigmes du site, déjà parues ou qui arriveront prochainement donc profitez-en !
Bonne lecture,
Jaudi

I – Biographie de Vladimir Igorevitch Arnold

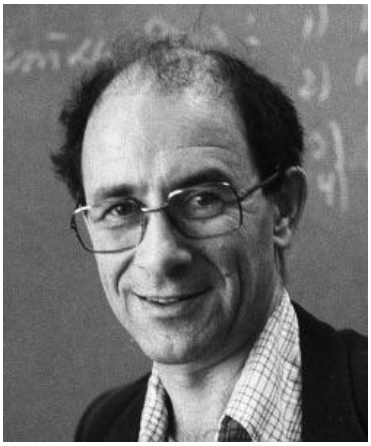

Figure 1 - Portrait de Vladimir Arnold

Vladimir Igorevitch Arnold est né le 12 juin 1937 à Odessa (Ukraine). Issu d'une famille de scientifiques, Arnold est initié très jeune aux mathématiques par l'intermédiaire d'énigmes que son entourage lui propose de résoudre. Il entre à l'Université d'état de Moscou en 1954, qui est connue pour accueillir de grands mathématiciens (Kolmogorov, Markov…).

Sous la direction du célèbre Andreï Kolmogorov, il réalise un coup d'éclat en résolvant, à seulement 19 ans, le treizième problème d'Hilbert, qui résistait aux mathématiciens depuis plus d'un demi-siècle. Il approfondit ensuite ce sujet en soutenant sa thèse en 1961, intitulée « Sur la représentation de fonctions continues de trois variables par la superposition de fonctions continues de deux variables », toujours sous la supervision de son mentor.

Par la suite, il réalisa de nombreux travaux en mathématiques, établissant notamment le théorème de Kolmogorov-Arnold-Moser (KAM). Il est en particulier connu pour ses avancées en topologie et dans la théorie des systèmes dynamiques. Dans ce contexte, il créa l'application du chat d'Arnold, que nous allons étudier dans ce cours.

Il fut ensuite nommé à l'Académie des Sciences de l'Union soviétique. Alors qu'il était nominé pour recevoir la prestigieuse médaille Fields en 1974, le gouvernement soviétique (à cause de son opposition publique à la persécution des dissidents) a empêché sa nomination d'aboutir. Il fut néanmoins récompensé par de nombreux autres prix internationaux de premier plan par la suite (prix Wolf, Crafoord, Shaw…) et obtint un poste de chercheur en 1986 au prestigieux institut Steklov, tout en officiant en parallèle à l'université Paris-Dauphine. Cette grande figure des mathématiques et auteur de nombreux articles, livres et manuels universitaires, décéda en 2010 à Paris.

II – Présentation du chat d'Arnold

II.1 – Notions mathématiques utiles

(i) Le tore, illustré en figure 2, est une surface de révolution générée par un cercle (rouge sur l'image) qui se déplace le long d'un autre cercle (ici rose).

(ii) Une application bijective correspond à une transformation d'un état A vers B, pour laquelle l'application inverse pour revenir à l'état initial (de B vers A) est toujours possible.

(iii) Le modulo (ici noté mod), associé à deux entiers naturels, correspond au reste de la division euclidienne du premier par le deuxième (par exemple 7 mod 3 = 1 car 7=3*2+1 ou encore 12 mod 3 = 0 car =3*4+0). Cette opération, très utilisée en mathématiques et en cryptographie (sur plein d'autres chiffrements) permet d'imposer un maximum à l'intervalle de valeurs.

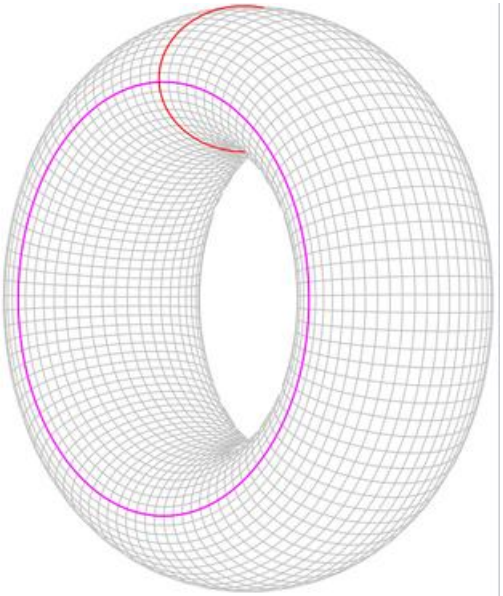

Figure 2 - Génération d'un tore

II.2- Présentation générale du codage

Le chat d'Arnold, appelé « cat map » en anglais, est une application mathématique bijective du tore (i).

Cela signifie que le chat d'Arnold est une fonction qui prend un tore et le transforme un autre tore et, comme elle est bijective (ii), l'opération inverse pour revenir au tore initial est toujours possible. Pour les mathématiciens chevronnés, on pourra remarquer que cette application fait partie des automorphismes toraux hyperboliques.

Cette application a été créée par Arnold lors de ses recherches sur les comportements chaotiques, et son nom spécial de « cat map » vient du fait qu'Arnold s'est servi de l'exemple d'un chat pour l'expliquer.

II.3 – Approche géométrique et visuelle

L'application du chat d'Arnold consiste à déplier un tore pour le mettre dans un carré. Comme illustré en figure 3, le carré initial (« unit square »), suite à l'application de la fonction, va être étiré et cisaillé (« stretch and shear ») pour former un parallélogramme, puis remis dans un carré (« shift back to unit square ») à l'aide de l'opération modulo (iii).

Le modulo est ici utilisé pour remettre le parallélogramme dans le carré, ce qui permet que l'ensemble de départ (un carré) corresponde à l'ensemble d'arrivée (un autre carré).

Sur l'exemple, on observe qu'on peut appliquer plusieurs fois la transformation pour obtenir une image qui semble de plus en plus brouillée. Bien que l'opération de base paraisse plutôt simple, une petite différence dans la position d'un point peut conduire à une énorme différence dans la position du point résultat après un grand nombre d'applications de la transformation, ce qui illustre le comportement chaotique de cette application.

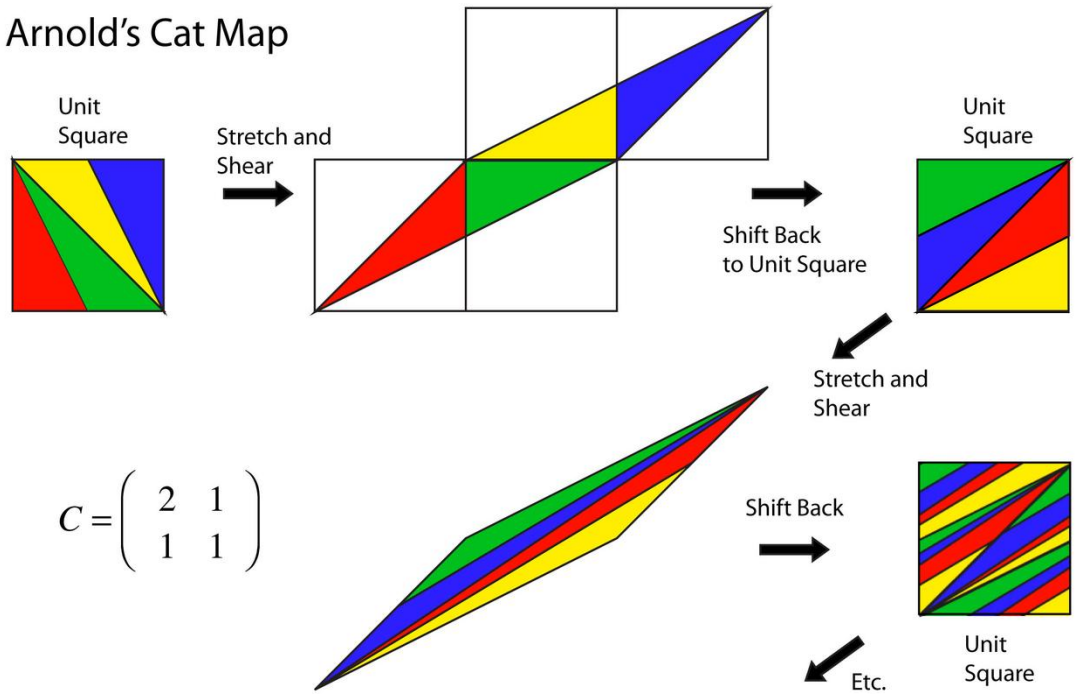

Figure 3 - Etapes du chat d'Arnold

II.4 – Exemple canonique créé par Arnold

Lors de sa présentation de l'application de la « cat map », Arnold utilise la matrice C de la figure 3, qui permet d'effectuer un cisaillement d'une unité vers le haut, puis deux unités à droite. L'application utilisée correspond à un produit matriciel suivi d'un modulo, comme détaillé dans cette formule :

$$\Gamma\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod 1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod 1.$$

On peut définir une fonction de manière très semblable, mais sur un ensemble discontinu de points. En pratique, on prend une image constituée de pixels disjoints, en utilisant des coordonnées entières dans l'intervalle {0, … *N*-1}, N un entier naturel. On pose :

$$\tilde{\Gamma} : \begin{array}{ccc} I_{N,N} & \to & I_{N,N} \\ P\begin{pmatrix} x \\ y \end{pmatrix} & \mapsto & P'\begin{pmatrix} (2x+y) \pmod N \\ (x+y) \pmod N \end{pmatrix} \end{array}$$

Si l'on applique cette transformation Γ̃ plusieurs fois de suite, on retrouve au bout d'un nombre fini d'itérations l'image de départ, comme illustré dans la figure 4

En effet, comme elle consiste à permuter les pixels d'une image et que les permutations des pixels sont en nombre fini, on va forcément tomber dans un cycle et revenir aux pixels initiaux.

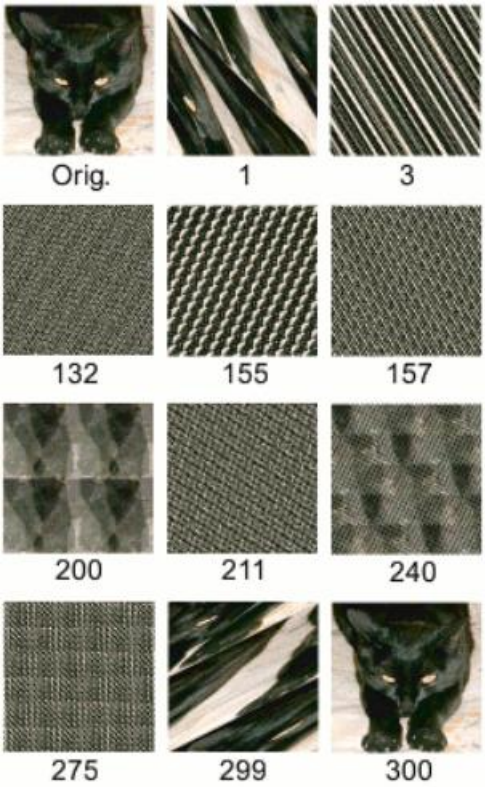

Figure 4 - Retour à l'image de départ

On constate donc un comportement bien différent de cette application Γ̃ par rapport à l'application Γ, car on est passé d'un ensemble continu à un ensemble discontinu de valeurs.

II.5 – Propriétés et déchiffrement du chat d'Arnold

Cette transformation peut être utiliser avec plein d'autres matrices que la matrice C, permettant d'étirer et de cisailler plus ou moins les différents morceaux de l'image.
On considère la matrice M, de dimension 2x2, avec a,b,c et d des entiers relatifs. La matrice M peut être utilisée pour la « cat map » d'Arnold si, et seulement si :

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

(i) Le déterminant (dont la formule est ad-bc) doit valoir 1
(ii) La trace (somme des éléments diagonaux, dont la formule est a+d) doit être, en valeur absolue, strictement supérieure à 2

Ce qui est très intéressant dans cette application est qu'elle permet de totalement brouiller une image et peut donc rendre son contenu invisible à l'œil nu. De plus, elle peut même résister à des adeptes de la méthode car il y a une infinité de matrices possibles permettant de transformer l'image avec le chat d'Arnold.

La période du cycle permettant de retourner à l'image initiale dépend de la taille N de l'image de façon très irrégulière, mais elle présente la propriété d'être majorée par 3N (ce qui limite le nombre d'itérations à effectuer pour les cryptanalystes).

Pour décrypter une image chiffrée par l'application du chat d'Arnold, en connaissant la matrice utilisée, on peut appliquer plein de fois l'application jusqu'à retomber sur l'image en clair, ou alors appliquer la matrice inverse (qui permet de remonter le temps en effectuant les étapes en sens inverse) pour revenir à l'image de départ.

Bibliographie

Pour ceux parmi vous qui sont désireux d'approfondir ce système, voici certains des nombreux articles que j'ai consultés (en juillet 2026) pour écrire ce cours de cryptographie :

Ressources en français :
https://fr.wikipedia.org/wiki/Vladimir_Arnold
https://fr.wikipedia.org/wiki/Chat_d%27Arnold
https://www.bibmath.net/bios/index.php?action=affiche&quoi=arnold
https://elysiatools.com/fr/visualizations/arnold-cat-map

Ressources en anglais :
https://en.wikipedia.org/wiki/Vladimir_Arnold
https://www.jasondavies.com/catmap/
https://galileo-unbound.blog/2019/06/16/vladimir-arnolds-cat-map/