Tresses et Cryptographie

Flore Le Roux

Année 2020-2021, Lycée Lakanal

Table des matières

Introduction

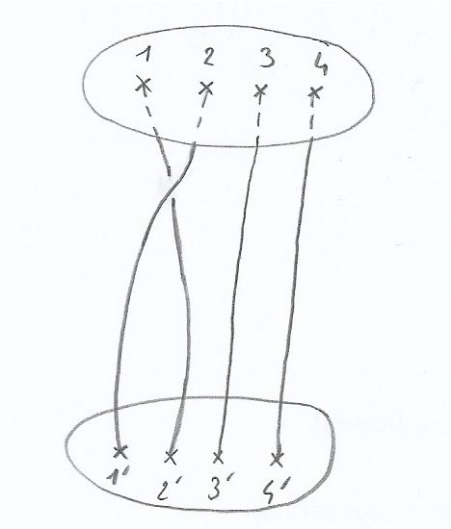
Les tresses de la vie de tous les jours peuvent être modélisées de manière mathématique et, munies d'un produit, elles forment une structure de groupe. Le but de cette étude est d'expliquer pourquoi le groupe des tresses est un cadre propice à la mise en place de la technique de cryptographie à clés publiques.

De manière générale, la cryptographie à clés publiques repose sur le principe suivant. Alice et Bob veulent créer une clé K de cryptage. Alice choisit une clé secrète a et publie une clé publique p_a obtenue par opérations sur a. Bob fait de même avec b et p_b. Ensuite, Alice et Bob obtiennent K par opérations sur a et p_b pour Alice et sur b et p_a pour Bob. Une condition nécessaire au bon fonctionnement impose que Bob et Alice puissent trouver la même clé K à partir de a et p_b d'un côté et de b et p_a de l'autre. Il faut aussi que p_a et p_b soient simples à trouver à partir de a et b, mais que a et b soient impossible à retrouver à partir de p_a et p_b. On parle de fonction à sens unique.

1 Présentation du groupe des tresses

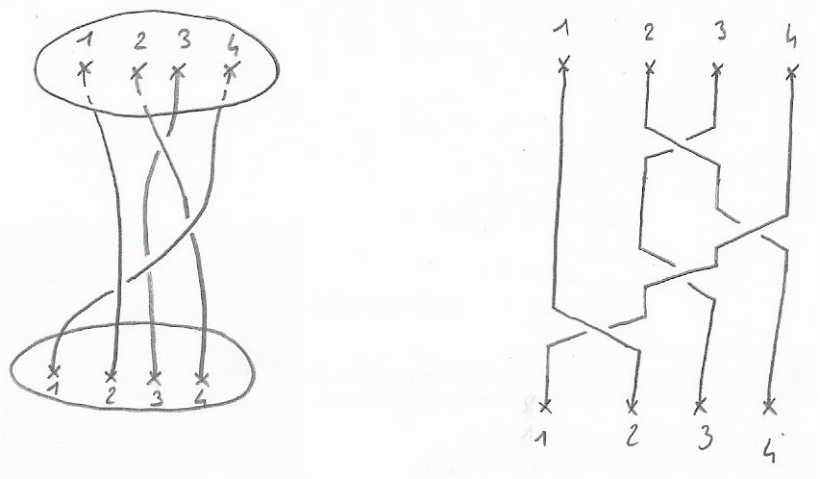
1.1 Modélisation géométrique et définition algébrique du groupe des tresses

Une tresse peut être représentée mathématiquement par l'union de n brins de l'espace joignant n points numérotés d'un plan à n autres points numérotés dans le même sens d'un autre plan sans intersection entre les brins. Par commodité, on donne un nom aux tresses en nommant les croisements entre leurs brins : pour tout i de $[\![1, n-1]\!]$, on note σ_i le croisement entre les brins i et $i+1$ lorsque le brin $i+1$ passe au-dessus du brin i et σ_i^{-1} le même croisement quand le brin i passe au-dessus du brin $i+1$.



Le croisement σ_1 pour une tresse à quatre brins

Pour faciliter la compréhension et la représentation des tresses géométriques, on utilise les diagrammes de tresses. Il s'agit de la projection dans le plan d'une tresse. Pour chaque croisement, on représente en trait continu le brin qui passe au-dessus et en trait discontinu celui du dessous.



Une tresse et son diagramme

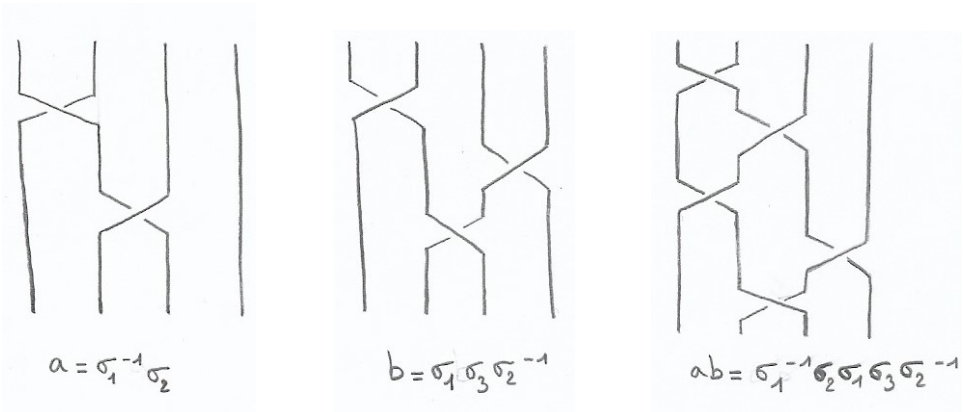
Par la suite, on représentera les tresses par leur diagramme.

Ensuite, on peut classifier les tresses grâce à la notion d'isotopie.

Définition. *Une tresse géométrique est isotope à une autre si on peut passer de la première à la deuxième par une déformation continue des brins de la première.*

On peut vérifier facilement qu'il s'agit d'une relation d'équivalence. On considère alors que deux tresses géométriques isotopes sont en fait la même tresse.

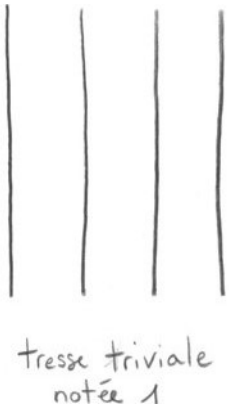
Par ailleurs, on peut concaténer les tresses.



Deux tresses et leur concaténation

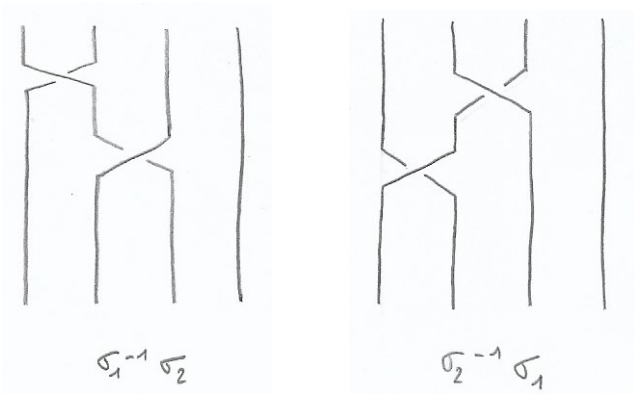
Cette concaténation permet de donner à l'ensemble des tresses une structure de groupe et l'on parle alors de produit.

La tresse triviale, c'est à dire l'élément neutre du groupe, est représentée par la tresse sans croisement entre les brins.



Représentation géométrique de l'élément neutre

Le symétrique d'un élément est représenté par son image dans un miroir et s'obtient donc en symétrisant la tresse par rapport au plan contenant les points d'arrivée de ses brins.



Une tresse et son symétrique

On peut à présent donner une définition algébrique du groupe des tresses. Chaque croisement entre deux brins d'une tresse géométrique représente un générateur du groupe des tresses algébriques. On obtient alors une définition de groupe par générateurs et relations :

Définition. *Le groupe des tresses, noté B_n, est le groupe engendré par $n-1$ générateurs $\sigma_1, \cdots, \sigma_{n-1}$ et les relations :*

$$(1)\ \sigma_i\ \sigma_j = \sigma_j\ \sigma_i\ \forall\ (i,j) \in [\![1, n-1]\!]^2 \text{ tel que } |i-j| \geq 2$$

$$(2)\ \sigma_i\ \sigma_j\ \sigma_i = \sigma_j\ \sigma_i\ \sigma_j\ \forall\ (i,j) \in [\![1, n-1]\!]^2 \text{ tel que } |i-j| = 1.$$

Cela signifie que l'on peut calculer dans le groupe des tresses uniquement avec ces deux relations et les relations normales entre un générateur et son inverse. Comme le groupe est noté de manière multiplicative, l'élément neutre est noté 1. Un produit de générateurs et d'inverses de générateurs est appelé un mot.

1.2 Explication des relations du groupe des tresses

Essayons de mieux comprendre les relations en voyant ce qu'elles donnent géométriquement.

La première relation

$$(1)\ \sigma_i\ \sigma_j = \sigma_j\ \sigma_i\ \forall\ (i,j) \in [\![1, n-1]\!]^2 \text{ tel que } |i-j| \geq 2$$

vient du fait que géométriquement, ces deux croisements n'impliquent pas les mêmes brins et donc n'ont pas d'impact l'un sur l'autre.

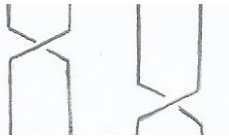


Illustration de la première relation

Comme on est passé d'une tresse à l'autre par déformation continue, les deux tresses sont isotopes.

La seconde relation s'illustre comme suit :

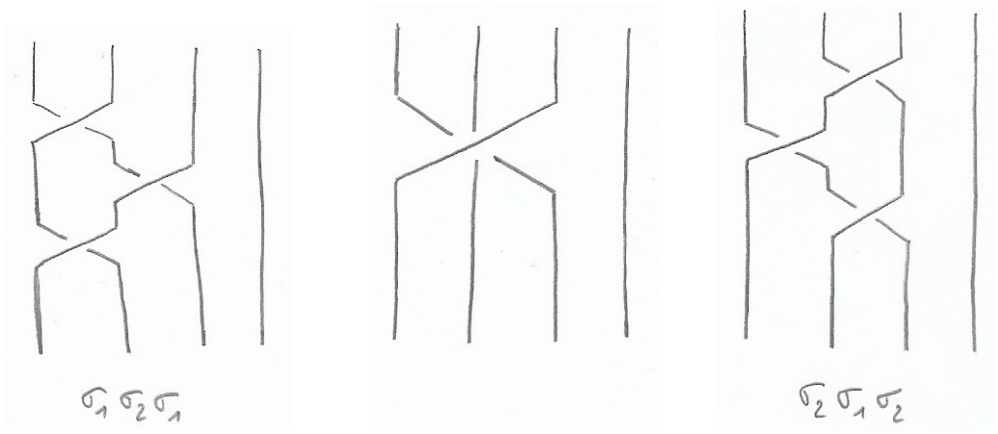


Illustration de la seconde relation

Le dessin du milieu représente la transition entre la tresse de gauche et celle de droite. Le brin du milieu passe entre les deux autres brins. C'est donc le brin de gauche qui est le plus en dessous. On voit qu'en décalant le brin du milieu vers la droite depuis sa position sur la tresse de gauche, on passe par l'état du milieu pour arriver finalement à la tresse de droite. Ces deux tresses sont isotopes, puisque l'on est passé de l'une à l'autre par une déformation continue des brins. En fait, on voit à travers ces deux exemples que l'égalité algébrique correspond à l'isotopie géométrique.

Voyons maintenant un exemple de calcul dans le groupe des tresses pour appréhender les différentes relations algébriquement.

Essayons de simplifier l'expression :

$$a = \sigma_3\ \sigma_1\ \sigma_2\ \sigma_3\ \sigma_2^{-1}\ \sigma_3^{-1}\ \sigma_2^{-1}\ \sigma_1^{-1}.$$

En utilisant la première relation, on obtient :

$$a = \sigma_1\ \sigma_3\ \sigma_2\ \sigma_3\ \sigma_2^{-1}\ \sigma_3^{-1}\ \sigma_2^{-1}\ \sigma_1^{-1}.$$

On peut alors utiliser la deuxième relation :

$$a = \sigma_1\ \sigma_2\ \sigma_3\ \sigma_2\ \sigma_2^{-1}\ \sigma_3^{-1}\ \sigma_2^{-1}\ \sigma_1^{-1}.$$

Tous les générateurs se simplifient alors avec leur inverse et on obtient :

$$a = 1.$$

Cet exemple montre qu'il n'est pas évident de savoir si deux tresses sont égales d'un point de vue algébrique, puisqu'on est parti d'un mot de longueur 8 pour arriver au mot trivial. Géométriquement, la notion d'isotopie n'est pas plus simple, comme on a pu le voir en illustrant la deuxième relation du groupe des tresses. Le fait de savoir

déterminer si deux éléments d'un groupe sont égaux est appelé communément problème du mot et pourrait gêner l'utilisation des tresses en cryptographie. En effet, on doit pouvoir manipuler les tresses sans ambiguité pour ne pas compromettre le cryptage et décryptage des messages. Cependant, dans le groupe des tresses, le problème du mot est résolu, c'est-à-dire qu'on dispose d'algorithmes de complexité au plus polynomiale permettant de nous dire si deux tresses sont identique ou non.

1.3 Commutativité et tresses

Déterminons si B_n est abélien ou non. Pour cela, on admet le lemme suivant.

Lemme. *Soient $s_1, \cdots, s_{n-1}$ des éléments d'un groupe G qui satisfont les relations précédentes :*

$$(1)\ s_i\ s_j = s_j\ s_i \text{ pour } |i-j| \geq 2$$

$$(2)\ s_i\ s_j\ s_i = s_j\ s_i\ s_j \text{ pour } |i-j| = 1.$$

Alors il existe un unique morphisme $f : B_n \longrightarrow G$ tel que :

$$s_1 = f(\sigma_1), \quad \cdots, \quad s_{n-1} = f(\sigma_{n-1}).$$

On applique ce lemme à certains éléments du groupe symétrique.
Soient $s_1 = (1\ \ 2), \cdots, s_{n-1} = (n-1\ \ n)$. Alors on a :

$$(1)\ s_i\ s_j = s_j\ s_i \text{ pour } |i-j| \geq 2.$$

En effet, deux transpositions à supports disjoints commutent. De plus, on a :

$$(2)\ s_i\ s_j\ s_i = (i\ \ i+2) = s_j\ s_i\ s_j \text{ pour } |i-j| = 1.$$

Corollaire. *Il existe un unique morphisme :*

$$\pi : B_n \longrightarrow S_n \quad \textit{tel que} \quad \pi(\sigma_1) = s_1, \ \cdots, \ \pi(\sigma_{n-1}) = s_{n-1}.$$

Ce morphisme est surjectif car les transpositions $(i\ \ i+1)$ engendrent le groupe symétrique.

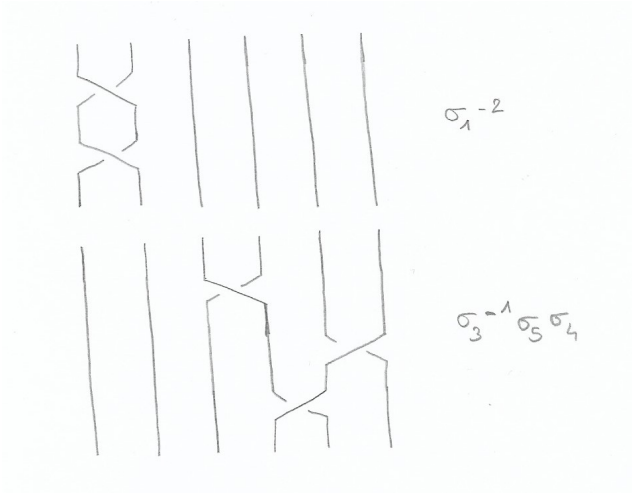
Corollaire. *Pour $n \geq 3$, B_n n'est pas abélien.*

Démonstration. *Pour $n \geq 3$, S_n n'est pas abélien car :*

$$s_1\ s_2 = (1\ \ 2) \circ (2\ \ 3) = (1\ \ 3\ \ 2) \neq (1\ \ 2\ \ 3) = (2\ \ 3) \circ (1\ \ 2) = s_2\ s_1.$$

Comme π est surjectif, $\exists\ (\sigma_i, \sigma_j) \in {B_n}^2$ tel que $s_1 = \pi(\sigma_i)$ et $s_2 = \pi(\sigma_j)$.
Ainsi $\pi(\sigma_i)\ \pi(\sigma_j) \neq \pi(\sigma_j)\ \pi(\sigma_i)$.
Donc $\pi(\sigma_i\ \sigma_j) \neq \pi(\sigma_j\ \sigma_i)$.
Et finalement $\sigma_i\ \sigma_j \neq \sigma_j\ \sigma_i$.

Cependant, on peut tout de même trouver des catégories de tresses qui commutent entre elles. Par exemple, notons DB_r le sous-groupe de B_{l+r} engendré par les générateurs $\sigma_{l+1}, \cdots, \sigma_{l+r-1}$, c'est-à-dire le sous-groupe de B_n où seuls les r derniers brins peuvent faire des croisements. On note encore B_l le sous-groupe de B_n engendré par $\sigma_1, \cdots, \sigma_{l-1}$. Alors la première relation du groupe des tresses nous dit que les éléments de DB_r et de B_l commutent entre eux. En effet, les générateurs $\sigma_1, \cdots, \sigma_{l-1}$ commutent avec les générateurs $\sigma_{l+1}, \cdots, \sigma_{l+r-1}$.



Deux tresses qui commutent : l'une appartient à B_2, l'autre à DB_4

2 Cryptographie avec des tresses

2.1 Problème de recherche du conjuguant et création des clés

On a vu en introduction que la cryptographie à clés publiques a besoin d'une fonction à sens unique. Dans le groupe des tresses, il y a un problème non résoluble en temps polynomial qui est propice à la création d'une telle fonction. C'est le problème de recherche du conjuguant. On considère deux tresses a et b. On dit qu'on conjugue a par b lorsqu'on effectue le calcul bab^{-1}. La tresse $c = bab^{-1}$ est le conjugué de a par b. Le calcul de la tresse c est très facile à effectuer car le produit de deux tresses est simplement leur concaténation. Cependant, retrouver b, le conjuguant, à partir de a et c est compliqué à cause des simplifications dues aux relations du groupe des tresses. On peut donc bien créer une fonction à sens unique, la fonction qui, à une tresse, associe son conjugué par une tresse fixée.

On construit les clés d'Alice et Bob à l'aide de cette fonction. Alice et Bob choisissent d'abord une tresse commune dans B_{l+r}, que l'on note x. Alice choisit ensuite une tresse a dans B_l et Bob, une tresse b dans DB_r. Alice calcule après $p_a = axa^{-1}$ et Bob, $p_b = bxb^{-1}$. Ils envoient alors cette tresse chacun à l'autre, et crée la clé qui va permettre de crypter et décrypter des messages, Alice en conjuguant p_b par a et Bob en conjuguant p_a par b. Ainsi, la clé K vaut $abxb^{-1}a^{-1}$, car a et b commutent.

2.2 Principe de l'algorithme de cryptage et décryptage

Après avoir créé la clé K, Alice et Bob sont prêts à s'échanger des messages cryptés. Supposons par exemple que Bob veuille envoyer un message à Alice. Pour cela, il traduit d'abord ce message en un message M écrit en binaire, par exemple avec le code ASCII. Il calcule ensuite l'image de sa clé par une fonction de hachage H, c'est-à-dire qu'il génère un nombre binaire, de taille fixée, représentant la clé K. Il peut alors crypter son message en posant : $C = H(K) \oplus M$, où $\oplus$ est l'addition en base deux. Il finit le cryptage en envoyant son message crypté C à Alice.

Alice peut ensuite décrypter le message. Pour cela, il lui suffit d'ajouter à C le nombre $H(K)$. En effet, en base deux, la somme d'un nombre avec lui même vaut toujours 0 puisque c'est une nombre pair en base dix. On a ainsi : $H(K) \oplus C = H(K) \oplus H(K) \oplus M = M$. Alice obtient le message décrypté, qu'elle n'a plus qu'à repasser en littéraire pour récupérer le mot initial.

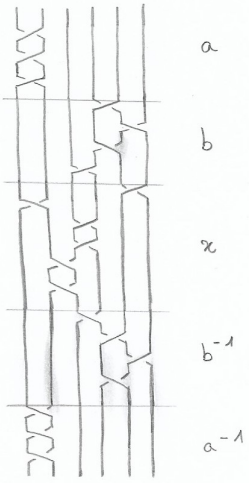
2.3 Exemples

On peut maintenant voir quelques exemples de cryptage et décryptage par cette méthode à l'aide d'un algorithme codé en Python. On utilise le code ASCII, accessible

directement dans Python, pour coder les messages en binaire, chaque lettre étant codée sur 8 bits. On utilise également une fonction de hachage du module hashlib de Python. Cette dernière renvoie la clé sur 256 caractères et on découpe donc le message à crypter en sous-message de taille 256 pour que la somme en base 2 fonctionnne correctement, en ajoutant des espaces au message si besoin.

Dans le premier exemple, on crypte un message et on affiche ensuite le version littéraire du message crypté pour vérifier l'action du programme. On décrypte ensuite le message et on récupère bien le message initial, aux espaces ajoutées à la fin près.

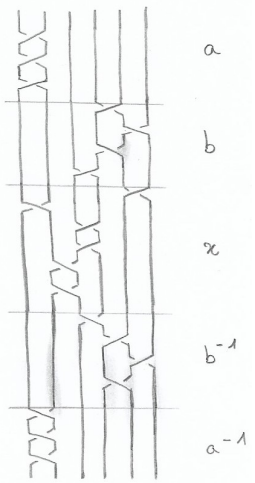
```
>>> x=[5,-1,3,3,-2,-2]

>>> a=[1,1,1]

>>> b=[4,-5,-4,3]

>>> C=cryptage("Je fais un test, donc ça sert à tester",x,a,b)

>>> C
['000110011010110001110110011110111110011110111011001001110110111010111011101111010101110110
1111101101101010001101001000100100111111100011001001110000110111111011011001001010011001101 0
0000110001111101011101111100011011111111101010110011100100100000011001110001', '001001111010111
0000100101011010011111000111010000001110100011011101110111011110011010111011110101010011111111
0111100111011001000011111000110110001101000010101101010010011110100110101011110111000011101 0
1111101101011100101010100011100111100100101100011001110001']

>>> binaire_a_latin(concatene(C))
'\x19¬v{ç»\¯n»½]¾U\x8d"Oã\'\rûd¦h1ë¾7ú³\x92\x06q\'¬%iã\xa0tnîó]ê\x9fPvCãcBµ\'¦¯pëír¨ç\x92Æq'

>>> decryptage(C,x,a,b)
'Je fais un test, donc ça sert à tester                                                                '
```



Un premier exemple et la clé utilisée

Dans le deuxième exemple, on considère un texte un peu plus long (le début de l'article *Wikipédia* sur les groupes). On observe que l'algorithme fonctionne bien.

```
>>> x=[5,-1,3,3,-2,-2]

>>> a=[1,1,1]

>>> b=[4,-5,-4,3]

>>> C=cryptage("En mathématiques, un groupe est une des structures algébriques fondamentales
de l'algèbre générale. C'est un ensemble muni d'une loi de composition interne associative a
dmettant un élément neutre et, pour chaque élément de l'ensemble, un élément symétrique. La
structure de groupe est commune à de nombreux ensembles de nombres, par exemple les nombres
entiers relatifs, munis de la loi d'addition. Mais cette structure se retrouve aussi dans de
nombreux autres domaines, notamment en algèbre, ce qui en fait une notion centrale des mathé
matiques modernes.",x,a,b)

>>> decryptage(C,x,a,b)
"En mathématiques, un groupe est une des structures algébriques fondamentales de l'algèbre g
énérale. C'est un ensemble muni d'une loi de composition interne associative admettant un él
ément neutre et, pour chaque élément de l'ensemble, un élément symétrique. La structure de g
roupe est commune à de nombreux ensembles de nombres, par exemple les nombres entiers relati
fs, munis de la loi d'addition. Mais cette structure se retrouve aussi dans de nombreux autr
es domaines, notamment en algèbre, ce qui en fait une notion centrale des mathématiques mode
rnes.                "
```



Un deuxième exemple et la clé utilisée

Conclusion

On voit donc que, en théorie, on peut bien utiliser les tresses mathématiques pour crypter. Cependant, dans la pratique, cette méthode n'est pas utilisée car elle demande des efforts technologiques et de formation trop importants.

Références

[1] C. Kassel, V. Turaev, *Braid Groups*, Springer, 2008.

[2] J. Riou, X. Caruso, *Groupe des tresses d'Artin*, Mémoire de magistère, `https://www.math.u-psud.fr/~riou/doc/tresses.pdf`.

[3] C. Milliet, *Groupe de tresses et cryptographie*, Rapport de stage, `http://math.univ-lyon1.fr/~milliet/rapportstagetresses.pdf`.

[4] Article *Wikipédia* : Problème du mot, `https://fr.wikipedia.org/wiki/Probl%C3%A8me_du_mot`.

[5] L. Paris, *Les tresses : de la topologie à la cryptographie*, `http://images.math.cnrs.fr/Les-tresses-de-la-topologie-a-la-cryptographie.html`.