Tables de codage

une innovation alternative à la cryptographie algorithmique

par Pascal Thoniel, NTX Research

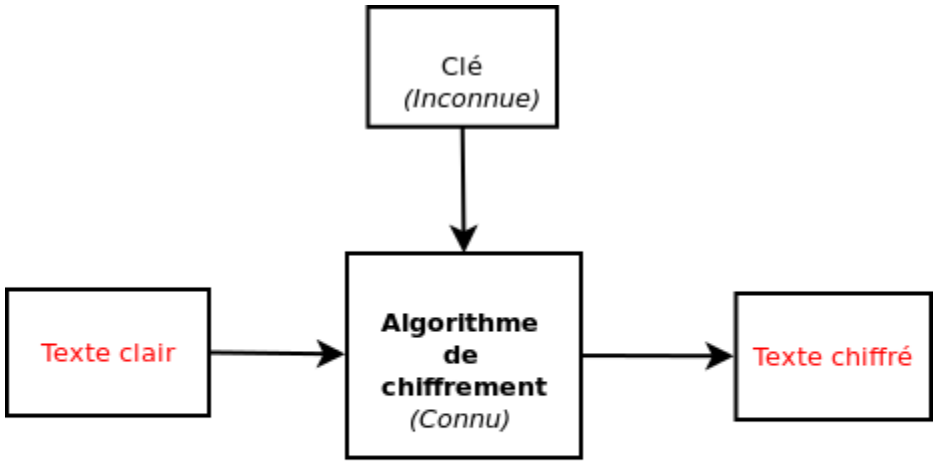
Séminaire sur la Confiance Numérique (Limos)
Clermont-Ferrand
2 juin 2016

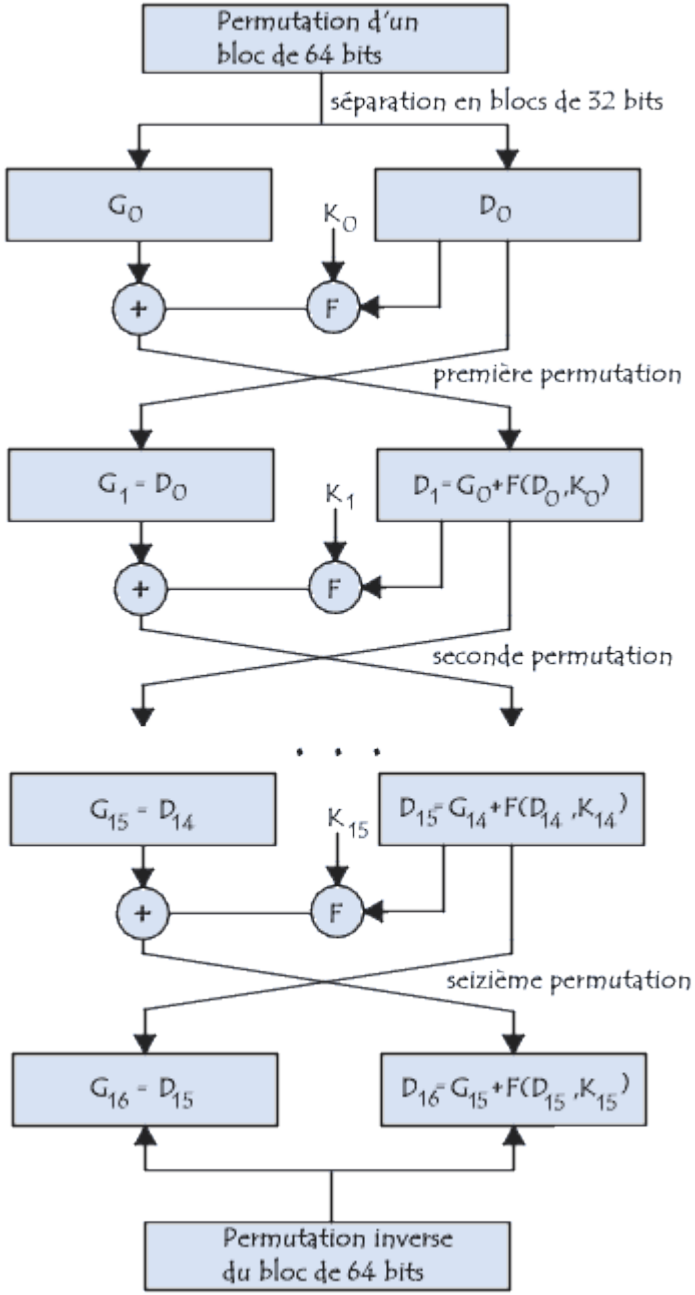
INTRODUCTION

Qu'elle soit symétrique ou asymétrique la cryptographie d'aujourd'hui est algorithmique.

C'est-à-dire que les fonctions cryptographiques utilisées pour assurer la confidentialité (chiffrement), l'authentification et la signature numérique sont des algorithmes, des programmes mathématiques.

La cryptographie algorithmique est non seulement nécessaire mais elle est aussi très efficace. Elle reste donc le socle indispensable de la cybersécurité, et pour longtemps.





Toutefois, la cryptographie algorithmique présente quelques limitations dans son usage.

Dans ces cas d'usage particuliers, une autre voie cryptographique est possible : les tables de codage, fondées sur la production aléatoire de caractères.

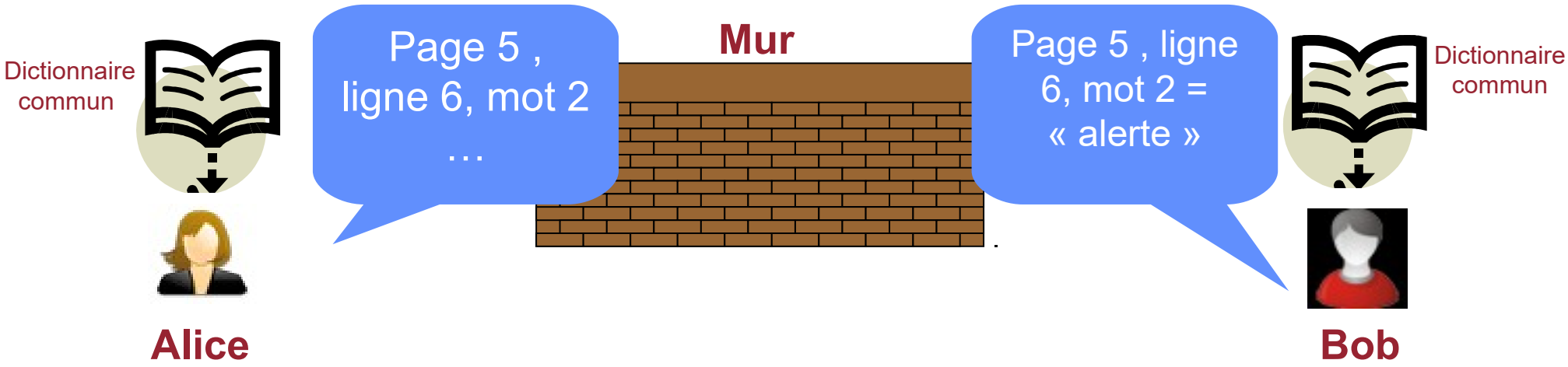
Le plan

Nous parlerons de l'origine historique des tables de codage (pendant la seconde guerre mondiale) et sa transformation récente (1996) pour un usage informatique.

Nous étudierons :
- la création des tables de codage
- leur utilisation pour assurer l'authentification forte des utilisateurs (en protocole défi-réponse)
- leur utilisation pour assurer un chiffrement probabiliste des clés de session
- leurs modes de distribution

Enfin, nous mettrons en lumière les avantages pour la cybersécurité de cette nouvelle voie cryptographique qui pourrait compléter avec élégance la cryptographie algorithmique que nous connaissons tous.

Les tables de codage en action pendant la seconde guerre mondiale



NTX Research va éliminer les faiblesses du dictionnaire **commun** utilisé comme système de chiffrement pendant la seconde guerre mondiale par l'utilisation de dictionnaires **compatibles**

Les tables de codage en mode matrice

Les tables de codages sont des dictionnaires de caractères générés de façon pseudo-aléatoire.

On crée autant de tables que d'utilisateurs…

Par exemple, en base 64 on peut créer 64^65025 matrices !

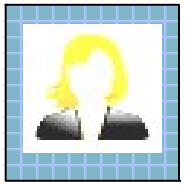
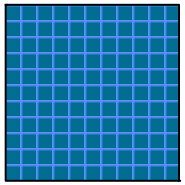


Table origine pour Alice

Elle sera immédiatement effacée du système dès que les tables client et serveur seront distribuées

Table Client pour Alice



Chaque case de la table contient un caractère tiré au hasard.

Une implémentation classique est 255 lignes x 255 colonnes = 65 025 caractères

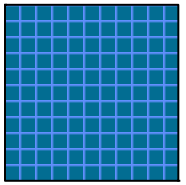


Table Serveur pour Alice

NTX Research invente l'authentification dite « bataille navale » ou « grille d'authentification » en 1996 !

Nous avons très significativement amélioré le système des tables de codage (dictionnaire commun -> compatible) utilisé lors d'un protocole défi-réponse, en mettant à contribution les dernières avancées informatiques et cryptographiques.

En 1996, NTX Research remet les tables de codage à l'honneur en inventant l'authentification matricielle en protocole défi-réponse.

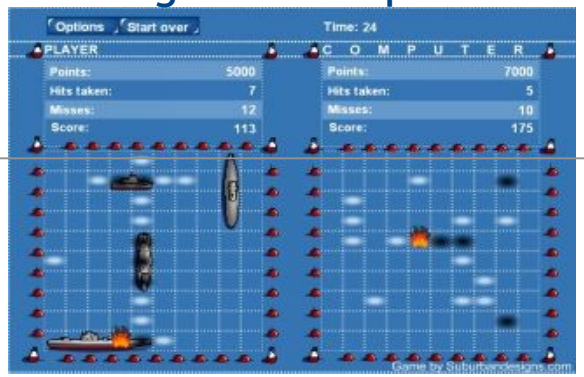
Les tables de codages permettent d'authentifier les utilisateurs en mode défi-réponse suivant le principe aujourd'hui connu sous le nom de « bataille navale ».

Le défi correspond à des coordonnées de la table tirées au hasard et la réponse correspond à la valeur trouvée dans la table aux coordonnées de ce défi.

L'authentification par bataille navale (1)

Processus d'authentification en mode défi-réponse sans code secret :

- Les deux tables de l'utilisateur sont identiques côté client et côté serveur.
- Le défi généré de façon pseudo-aléatoire côté serveur est « B5 ».
- Le défi est envoyé côté client à l'utilisateur.
- La valeur trouvée en coordonnées « B5 » de la table de l'utilisateur stockée dans l'authentifieur est « C ».
- La réponse calculée côté client « C » est envoyée au serveur. C'est un mot de passe à usage unique ou One Time Password (OTP).
- Le serveur trouve en coordonnées « B5 » de la table de l'utilisateur stockée côté serveur la valeur « C ».
- La réponse utilisateur est égale à la réponse serveur : l'utilisateur est authentifié.



L'authentification par bataille navale (2)

Sans code secret

	1	2	**3**	4	**5**	6	7	8	9	10
A	C	M	T	A	J	Z	W	A	G	H
B	S	Y	U	R	**K**	V	X	W	L	K
C	Q	S	Z	A	B	G	F	D	O	U
D	T	U	T	F	D	S	J	G	V	B
E	N	M	P	E	D	O	F	S	B	G
F	Q	F	A	Z	C	X	W	B	N	G
G	F	T	**P**	K	H	M	U	Y	B	H
H	S	Q	B	Z	A	R	F	F	K	S
I	L	N	H	F	G	K	T	J	G	A
J	Q	B	D	F	G	K	T	R	E	K

Table Client pour Alice

Défi : Alice, quelles valeurs en B5 et G3 ?

Processus d'authentification

La réponse est hachée à sens unique avant d'être envoyée au serveur.

Réponse : valeurs en V(B5)= K V(G3)=P

Un défi classique est constitué de 6 couples de coordonnées.

	1	2	**3**	4	**5**	6	7	8	9	10
A	C	M	T	A	J	Z	W	A	G	H
B	S	Y	U	R	**K**	V	X	W	L	K
C	Q	S	Z	A	B	G	F	D	O	U
D	T	U	T	F	D	S	J	G	V	B
E	N	M	P	E	D	O	F	S	B	G
F	Q	F	A	Z	C	X	W	B	N	G
G	F	T	**P**	K	H	M	U	Y	B	H
H	S	Q	B	Z	A	R	F	F	K	S
I	L	N	H	F	G	K	T	J	G	A
J	Q	B	D	F	G	K	T	R	E	K

Table Serveur pour Alice

Réponse côté client Alice : K P
Réponse côté Serveur pour Alice : K P
Les réponses Client et Serveur correspondent
=> Alice est authentifiée

Réponse attendue K P
=
Réponse reçue K P

Par exemple, en base 64 il y aura 64^6 réponses possibles (OTP).

L'authentification par bataille navale (3)

Processus d'authentification en défi-réponse **avec** code secret :

- Pour notre exemple, nous partons du fait que :
  - » Le code secret utilisateur est "+1,+1"
  - » Le code secret serveur est "+2,+2"

- Les deux tables de l'utilisateur sont *compatibles* (issues de la même table d'origine) *mais différentes* : côté client elle a été permutée ou mélangée suivant (+1+1) et côté serveur elle a été permutée ou mélangée suivant (+2+2).

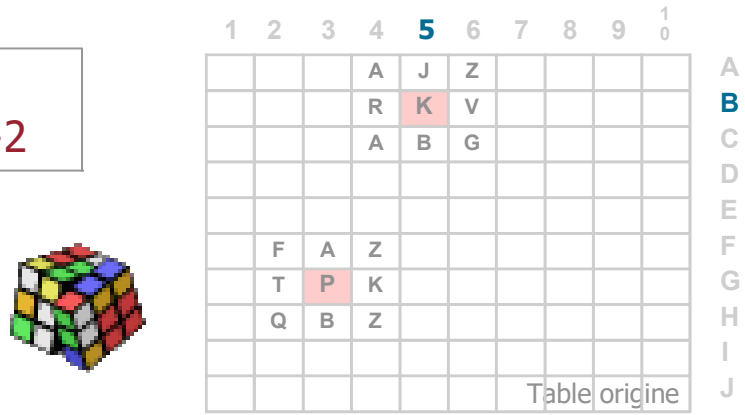
- La table origine n'existe plus, ne reste plus que deux matrices différentes et chargées de manière pseudo-aléatoire.

Une taille classique de code secret est de 5 caractères comme par exemple :
« abc13 » ou « rs234 ».

L'authentification par bataille navale (4)

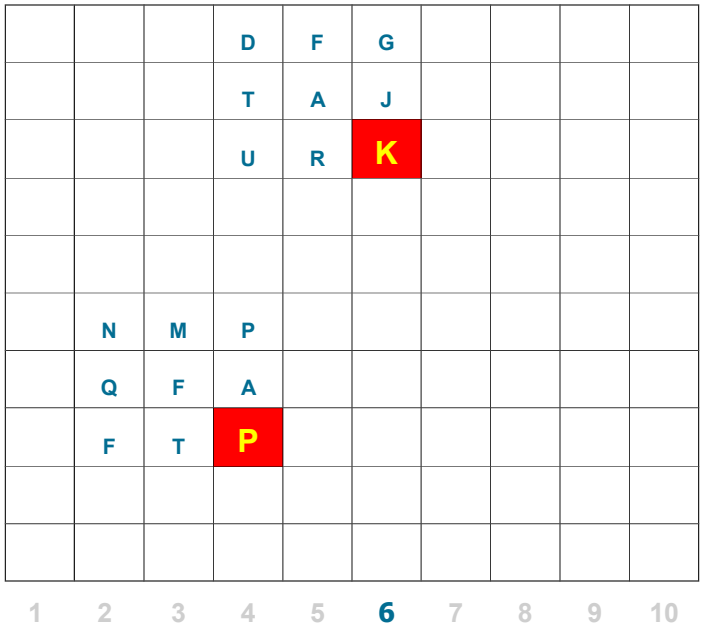
code secret Alice : +1, +1
code secret Serveur : +2, +2

Introduction des codes secrets disymétriques



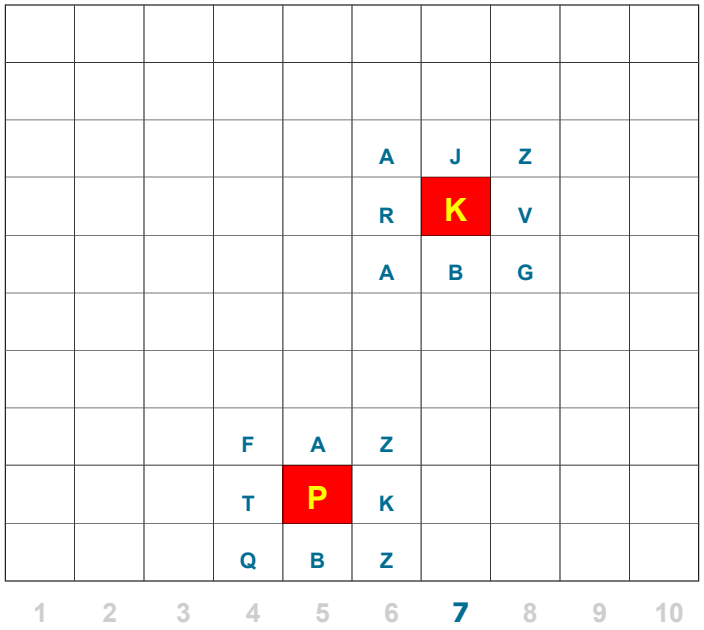
La table client Alice est mélangée avec le code d'Alice (+1,+1)

La table Serveur est mélangée avec le code serveur (+2,+2)



Les tables sont mélangées suivant leur code secret respectif

Exemple de fonction triviale de mélange des caractères des tables.



L'authentification par bataille navale (5)

Le défi est généré de façon pseudo-aléatoire par le serveur : par exemple « B5 ».

Ce défi est envoyé côté client à l'utilisateur :

- On demande à l'utilisateur de saisir son code secret (+1+1)
- L'applet/midlet locale transforme le défi initial « B5 » en « C6 » (B+1 = C, 5+1 = 6)
- La valeur trouvée en coordonnées « C6 » de la table de l'utilisateur stockée dans l'authentifieur est « K ».
- La réponse « K » est envoyée au serveur. C'est le mot de passe à usage unique ou One Time Password (OTP).

C'est maintenant au tour du serveur de lire, dans sa propre table concernant l'utilisateur, la valeur aux bonnes coordonnées :

- Le serveur lit en mémoire vive du serveur le code secret de l'administrateur (+2+2) ce qui transforme le défi initial « B5 » en « D7 » (B+2 = D, 5+2 =7)
- Le serveur trouve en coordonnées « D7 » de la table de l'utilisateur stockée côté serveur la valeur « K ».

La réponse utilisateur est égale à la réponse serveur : l'utilisateur est authentifié.

L'authentification par bataille navale (6)

Ainsi pour le défi suivant : B5 et G3

table client Alice
B5 -> C6 et G3 -> H4
valeur en C6 = K
valeur en H4 = P
réponse = K P

Avec code secret +1+1

+1 +1

	1	2	3	4	5	6	7	8	9	10
A				D	F	G				
B				T	A	J				
C				U	R	K				
D										
E										
F		N	M	P						
G		Q	F	A						
H		F	T	P						
I										
J										

Les réponses Client et Serveur sont identiques. Donc Alice est authentifiée.

+2 +2

table serveur pour Alice
B5 -> D7 et G3 -> I6
valeur en D7 = K
valeur en I6 = P
réponse = K P

Avec code secret +2+2

	1	2	3	4	5	6	7	8	9	10
A										
B										
C						A	J	Z		
D						R	K	V		
E						A	B	G		
F										
G										
H				F	A	Z				
I				T	P	K				
J				Q	B	Z				

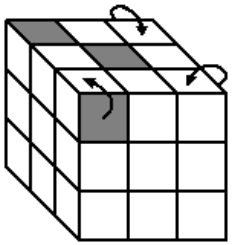
Le changement des codes secrets est possible de part et d'autre.

Comment ?

En recalculant la table origine en décalage négatif, puis en recalant immédiatement avec un nouveau décalage.

Le nouveau code secret est donc créé sans connexion : le serveur ne connaîtra jamais ce nouveau code.

Opération indépendante des deux parties -> processus identique.



Ce qui change tout !

- Les codes secrets ne sont stockés nulle part dans le système d'information
- Les codes secrets ne transitent pas sur le réseau
- Le code secret est mémorisable par l'utilisateur : 5 caractères alphanumériques suffisent (paramétrable)
- Les codes secrets ne sont pas partagés !
- Pas de répudiation possible : le serveur ignore le code secret de l'utilisateur
- Si le support contenant la table est volé, le pirate ne peut pas retrouver le code secret de l'utilisateur - non stocké - et donc le calcul des valeurs dans la table sera faux
- Localement, le pirate ne peut pas tester toutes les combinaisons possibles du code secret sans que le serveur ne le détecte
- Les codes secrets sous contrainte sont indétectables (*under duress*)

XCA : bénéfices économiques et sécurité (1)

- La solution d'authentification XCA fonctionne en mode web, web-mobile et mobile
- XCA est compatible avec tous les systèmes d'exploitation et tous les navigateurs du marché.
- Pas de matériel à installer sur les ordinateurs, les tablettes et les smartphones. 100 % logiciel.
- Pas de logiciel à installer sur les ordinateurs, les tablettes et les smartphones.
- XCA est compatible avec tous les authentifieurs physiques.
- Le niveau de sécurité est ajustable suivant les enjeux et la maturité des consommateurs (moyen, fort, très fort).
- Sécurité renforcée contre l'écoute de ligne : **pseudo masque jetable.**
- Sécurité renforcée contre l'emprunt, la perte et le vol de l'authentifieur : code secret inviolable.

XCA : bénéfices économiques et sécurité (2)

- L'utilisateur/client/acheteur conserve sa démarche habituelle acquise lors d'un achat de proximité avec sa carte bancaire :
  - » Saisie d'un code secret facilement mémorisable.
- L'utilisateur/client/acheteur peut choisir son code secret.
- Ce code n'est pas contraint ni en taille, ni en caractères.
- L'utilisateur/client/acheteur peut changer son code secret à tout moment, « hors ligne » :
  - » Sans le communiquer à qui que ce soit
  - » Sans le stocker nulle part
- L'utilisateur/client/acheteur est maître de la personnalisation de sa sécurité ce qui lui procure un sentiment de confiance.
- L'utilisateur/client/acheteur peut bénéficier d'une option de code d'alerte sous contrainte, totalement indétectable, lorsque les enjeux sont importants.

XCA : autres bénéfices (3)

La solution de sécurité XCA est entièrement paramétrable et ajustable :

- Sur le plan de la sécurité (cryptographie, stéganographie)
- Sur le plan de son interface mobile :
  - » Mode calculette « hors-ligne »
  - » Mode SMS avec un véritable défi-réponse seul garant de la sécurité
  - » Mode Internet Data

XCA est une solution de sécurité entièrement traçable (pas de boîte noire).

XCA est une solution « lego » dont on peut s'approprier la maîtrise.

Application au chiffrement

Crypto-système hybride :

> chiffrement symétrique efficace pour les données (AES, Blowfish, etc.)

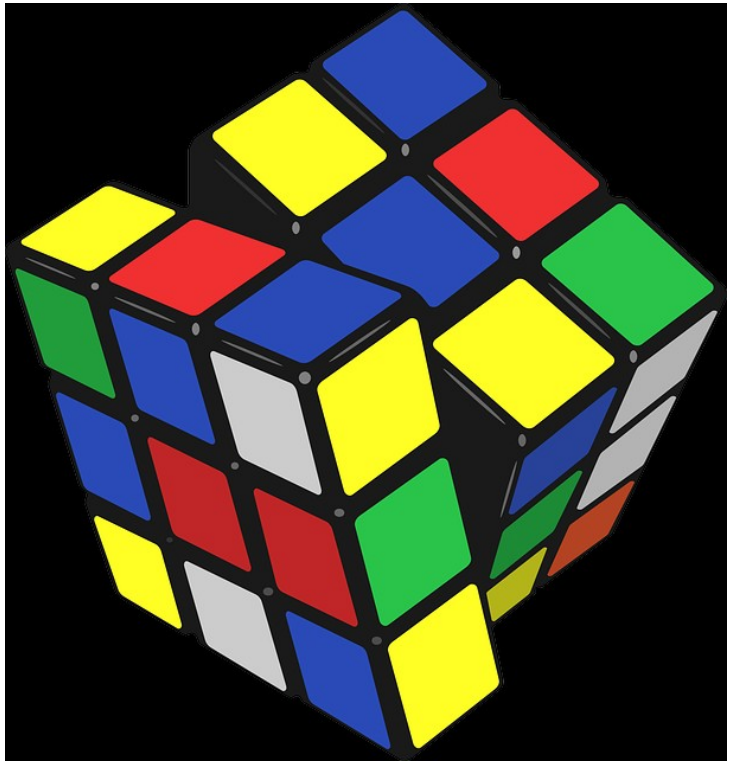
> chiffrement probabiliste des clés de session avec XC

Chiffrement probabiliste :

un même texte en clair donnera un cryptogramme différent d'une fois sur l'autre (avec la même table de codage et le même code secret)

Avec un nuage de points : c'est incassable en ligne !

Comment chiffrer un texte avec un Rubik's Cube ?



Décomposition du texte à chiffrer caractère par caractère (octet par octet)

Application d'une fonction aléatoire de recherche du caractère à chiffrer

Application du code secret Alice au chiffrement

Application du code secret de Bernard au déchiffrement

Chiffrement/déchiffrement en plusieurs rondes

Utilisation possible de plusieurs tables de codage en cascade

A l'épreuve de l'analyse par fréquence de lettres : « abba ».

Le même texte clair chiffré par la même table de codage et le même code secret donne un cryptogramme différent à chaque fois grâce à la fonction de recherche aléatoire.

La table de codage étant aléatoire et le texte clair étant aléatoire dans le cas d'une clé de session alors tout caractère du chiffré est équiprobable si on se place du point de vue de l'écoute de ligne.

Les modes de distribution des matrices

Sur des supports physiques de stockage

En téléchargement individuel après la saisie d'un code d'activation

En génération coté Client puis envoi de la matrice serveur via une liaison https sécurisée par le certificat du serveur

En mode papier : code barre 2D + PRNG

Diffie-Hellman + PRNG

Diffie-Helman matriciel

Stéganographie

Une solution cryptographique à assembler comme un Lego

Jeu de caractères interchangeable

Dimension des matrices : 2, 3, ou même plus en rajoutant un modulo

Taille des matrices (100x100, …, 255x255) → adaptation aux environnements

Algorithme de transposition des matrices (di-symétrie possible)

Longueur et contenu des codes secrets

Système de protection des matrices stockées

Système de mutation périodique ou ponctuel des matrices

En conclusion

Les tables de codage sont une application pratique du « masque jetable » de Vernam et Mauborgne dans l'environnement informatique web et mobile d'aujourd'hui.

La logique des tables de codage est probabiliste et non déterministe contrairement à la cryptographie algorithmique.

Contact

Pascal Thoniel
Fondateur et Directeur R&D
thoniel@ntx-research.com

*

NTX Research SA
111 avenue Victor Hugo
75116 Paris
Poitiers France – Boston USA
ntx-research.com