Le VPN

Table des matières

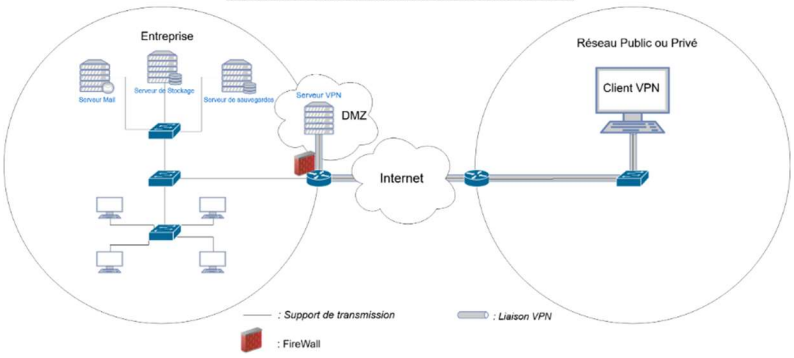
1. Qu'est-ce que le VPN

Le VPN signifie « Virtual Private Network », en français « Réseau Virtuel Privé ». Comme son nom l'indique. Il permet de créer des réseaux virtuels. Le principe du VPN, est de créer un réseau sécurisé auquel on peut se connecter, peu importe l'endroit où l'on se trouve. Le VPN est un protocole qui permet d'échanger des données dans un tunnel chiffré. Les VPN sont utilisés pour différents contextes :

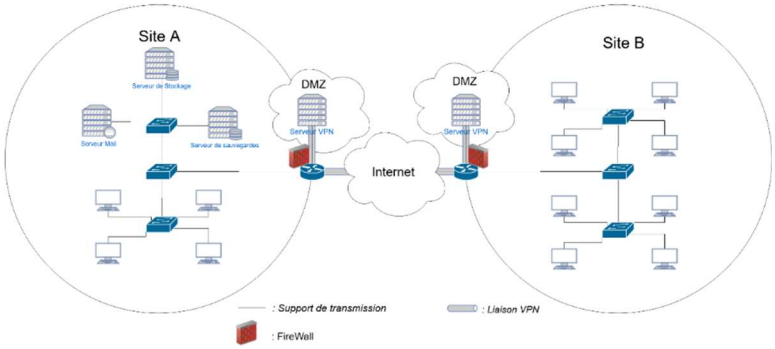
En entreprise, les VPN sont utilisés pour relier des membres du personnel/clients, qui travaillent chez eux ou bien en déplacement. Comme dit précédemment le VPN permet de faire abstraction de la géolocalisation. De ce fait on n'a pas besoin d'être connecter directement au réseau de l'entreprise pour utiliser ses ressources. L'une des autres utilisations du VPN, permet de lier plusieurs sites d'une entreprise entres eux.

Voici les différents schémas réseaux simplifiés, des différentes utilisations d'un VPN :

Connexion entre un client et un serveur VPN



Connexion inter-sites avec 1 serveur VPN par site



Il existe aussi des VPN grand publique, qui permettent aux particuliers de se connecter à un VPN afin de sécuriser les échanges. Certes l'utilisation d'un VPN permet de chiffrer les échanges jusqu'au serveur VPN, mais si on se connecte sur un VPN, notre trafic ne peut pas être déchiffré mais l'administrateur VPN lui, peut savoir ce qui passe par le VPN. Il est préférable ne pas prendre les offres VPN les moins chers. Il est plus rentable de mettre sois même en place un VPN, plutôt que de prendre des abonnements pour l'accès. Les abonnements permettent de ne pas administrer le VPN.

2. Pourquoi utiliser un VPN

Le VPN permet de créer un réseau virtuel privé. On pourrait se dire « Oui le VPN n'est pas très utile, pourquoi devrais t'on en utiliser un ? ». L'un des plus grands avantages du VPN est de pouvoir relier des hôtes/réseau à différents lieux géographiques, en chiffrant les données et flux transmit à l'aide d'un tunnel chiffré, ce qui rend en théorie les données indéchiffrables.

Le VPN peut être utilisé pour chiffrer une connexion, qui de base ne supporte pas le chiffrement, ou encore, si l'on se connecte sur un réseau public type « wifi gratuit », on ne sait pas, s'il n'y a pas une personne qui écoute le réseau, ou bien un « man in the middle » qui intercepte le trafic. De plus le chiffrement de données et flux permet donc de chiffrer des protocoles qui ne sont pas supportés par le chiffrement, comme le Telnet, http, ftp, etc….

Nous pouvons faire passer tous les flux dans ce tunnel, contrairement au SSH, qui lui est chiffré, mais, qui ne fait pas passer tous les flux.

La différence du VPN et le SSH : le SSH permet de faire passer des flux, cependant des protocoles spéciaux et adapter au tunnel SSH doivent être développés car, ils interviennent sur la couche 3 du modèle OSI. Le SSH est utilisé par plusieurs protocoles comme « le sftp », qui est un serveur ftp passant dans un tunnel SSH, afin de chiffrer les échanges.

Le VPN intervient au niveau de la couche 2 du modèle « OSI » qui permet d'encapsuler les données et de chiffrer le contenu, car la couche 2 du modèle « OSI » permet de fournir les moyens de transférer les données à d'autres hôtes. C'est pour cela, que l'on peut faire passer tous les flux, car c'est en dessous des couches des autres protocoles.

De plus, quand on fait passer les informations dans un VPN chiffré, notre « FAI » sait que le contenu est chiffré il va en destination du serveur VPN, il n'a aucune idée des destinataires auxquels nos informations vont être envoyé. On peut utiliser un VPN pour contourner la censure ou les restrictions « des FAI ».

3. Quels protocoles sont utilisés pour les VPN

Les technologies VPN utilisées pour encapsuler les données et informations, qui transitent dans le tunnel les plus utilisées et les plus connues sont :

- ***VPN SSL:***

Aussi appelé « clientless », n'a pas besoin d'un logiciel client, seul un navigateur internet compatible avec l'ouverture des sessions HTTPS SSL/TLS est suffisant. Contrairement au VPN IPsec, le tunnel VPN SSL ne permet pas le transport de différents protocoles de communication.

- ***VPN IPsec:***

L'installation d'un logiciel « agent » est nécessaire, afin d'établir un tunnel vers un serveur VPN. Un Tunnel VPN IPsec permet de véhiculer différents protocoles de communication tels que Telnet, RDP, SMB, SMTP, IMAP, etc…

- ***OpenVPN:***

Comme son nom l'indique, OpenVPN est un protocole VPN open source qui utilise « Secure Socket Layer (SSL) » pour créer une authentification pour une connexion Internet cryptée. Dans

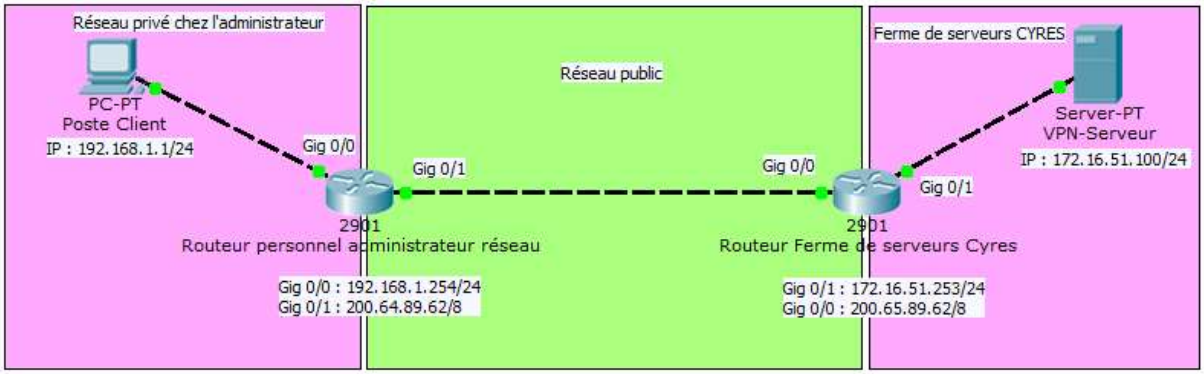
l'ensemble, le protocole OpenVPN offre l'une des meilleures combinaisons de performances et de sécurités, il peut être utilisé pour contourner facilement les pares feux ainsi que les restrictions des FAI.

Pour ce qui est des protocoles de transport OpenVPN peut utiliser l'UDP ou le TCP.

4. Prérequis

Pour l'installation d'OpenVPN, voici la configuration requise pour l'installation :

Schéma réseau :



5. Mise en place d'un serveur OpenVPN

a. Installation des paquets

OpenVPN n'étant pas installé directement sur la distribution linux, nous devons installer les paquets suivants :

```
apt-get install easy-rsa openvpn
```

easy-rsa : autorité de certification pour la connexion au VPN

openvpn : paquet d'installation du serveur et client OpenVPN

b. Générer le certificat et la clé de l'Autorité de Certification maître

Pour créer le certificat et la clé d'autorité de certification, nous devons faire une copie des fichiers de génération, pour cela il faut exécuter ces commandes.

```
mkdir /certificats/
```

Créer le dossier « certificats » à la racine

```
chmod 777 /certificats/
```

On change les droits du dossier

```
cp -r /usr/share/easy-rsa/* /certificats/
```

On copie les fichiers qui nous permettent de créer nos certificats, pour créer les autorisations de connexion des utilisateurs

Une fois fait, nous devons modifier un fichier du nom « vars », pour créer notre certificat.

```
cd /certificats/
```

On se déplace dans le dossier « /certificats/ »

```
nano vars
```

On édite le fichier vars

```
# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Fort-Funston"
export KEY_EMAIL="me@myhost.mydomain"
export KEY_OU="MyOrganizationalUnit"
```

```
# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="FR"
export KEY_PROVINCE="CENTRE-VAL-DE-LOIRE"
export KEY_CITY="TOURS"
export KEY_ORG="Paul Louis Courier"
export KEY_EMAIL="me@myhost.mydomain"
export KEY_OU="BTS SIO"
```

Voici à peu près la configuration que vous devez avoir

Si le fichier « openssl.cnf » n'existe pas, il faut le renommer.

```
mv openssl-1.0.0.cnf openssl.cnf
```

Cette commande doit être effectuée, uniquement si le fichier « openssl.cnf » n'existe pas.

Maintenant que la modification a été effectuée, nous allons initialiser les variables.

```
. ./vars
```

Le « point/espace/point » n'est pas une erreur de saisir.

Nous devons effacer, tous les certificats existants.

```
./clean-all
```

Il est recommandé d'effacer le dossier « /certificats/key/ », en cas de certificats déjà existants.

Nous allons créer notre certificat et la clé de l'Autorité de Certification Maitre

```
./build-ca
```

Aucune information n'a besoin d'être saisie, sauf pour « Common Name » qui doit être le nom de la machine.

Maintenant, nous allons créer une clé pour le serveur.

```
./build-key-server serveur-vpn
```

Pendant la génération du certificat et de la clé, il nous faut mettre dans le champ « Common Name », le nom hostname de la machine. Il nous ait demandé de saisir un mot de passe(Obligatoire) et un nom d'entreprise(Facultatif).

```
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'FR'
stateOrProvinceName   :PRINTABLE:'CENTRE-VAL-DE-LOIRE'
localityName          :PRINTABLE:'TOURS'
organizationName      :PRINTABLE:'Paul Louis Courier'
organizationalUnitName:PRINTABLE:'BTS SIO'
commonName            :PRINTABLE:'serveur-vpn'
name                  :PRINTABLE:'EasyRSA'
emailAddress          :IA5STRING:'me@myhost.mydomain'
Certificate is to be certified until Nov  9 22:24:09 2027 GMT (3650 days)
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
```

Ne pas oublier de saisir « y » et de valider, deux fois sans quoi le certificat et la clé ne seront pas créé.

c. Création des certificats et des clés clients

Pour créer des clés et des certificats et clients, c'est exactement la même méthode à utiliser, la commande est juste un peu différente.

```
./build-key utilisateur1
```

Attention, l'argument « utilisateur1 » doit être remplacé par le nom de l'utilisateur et doit être unique.

Il nous ait demandé, de saisir des informations comme l'étape précédente, il faut juste saisir dans « Commun Name » le nom de l'utilisateur. Pour mon cas le nom de la clé et du certificat son identique au nom de mon utilisateur. Tout comme l'étape précédente, il nous ait demandé de saisir un mot de passe(Obligatoire) et un nom d'entreprise(Facultatif).

d. Les paramètres Diffie Hellman doivent être générés pour le serveur OpenVPN

Nous devons générer les paramètres pour le serveur OpenVPN.

```
./build-dh
```

La génération peut prendre un peu de temps.

Voici le contenu et l'explication du dossier « /certificats/keys/ ».

Nom de fichier	Utilisation	Utilité	Secret
ca.crt	Serveur et tous les clients	Certificat racine CA	Non
ca.key	Clé signant la machine seulement	Clé racine CA	OUI
dh2048.pem	Serveur seulement	Paramètres Diffie Hellman	Non
serveur-vpn.crt	Serveur seulement	Certificat serveur	Non
serveur-vpn.key	Serveur seulement	Clé serveur	Oui
utilisateur1.crt	Utilisateur 1 seulement	Certificat Utilisateur 1	Non
utilisateur1.key	Utilisateur 1 seulement	Clé Utilisateur 1	Oui

Peut varier selon le nom donner au clé et certificat créer.

Une fois la génération terminée, nous devons copier les fichiers suivant, dans le dossier d'OpenVPN.

```
cp keys/dh2048.pem keys/ca.crt keys/serveur-vpn.crt keys/ serveur-vpn.key /etc/openvpn/
```

e. Configuration du VPN coté serveurs

Pour configurer le serveur OpenVPN, il faut créer un fichier de configuration et ajouter les spécifications de notre serveur, pour cela il faut faire ces commandes :

```
cd /etc/openvpn/
nano /etc/openvpn/server.conf
```

```
port 1194

proto tcp

dev tun

ca /etc/openvpn/ca.crt

cert /etc/openvpn/serveur-vpn.crt
```

```
key /etc/openvpn/serveur-vpn.key

dh /etc/openvpn/dh2048.pem

server 10.0.8.0 255.255.255.0

client-to-client

keepalive 10 120

cipher AES-256-CBC

persist-key

persist-tun

status /etc/openvpn/openvpn-startus.log

verb 3

push "route 10.0.8.0 255.255.255.0";

push "route 172.16.5x.0 255.255.255.0";

push "redirect-gateway def1 bypass-dhcp";
```

Configuration du fichier de config /etc/openvpn/server.conf

Une fois notre serveur configuré avant de le lancer, nous devons mettre en place la translation d'adresse sur le serveur VPN. Pour cela, nous devons exécuter ces commandes.

```
nano /etc/sysctl.conf
```

```
net.ipv4.ip_forward=1
```

Fichier de configuration "/etc/sysctrl.conf " , il suffie d'enlever le "#" devant la ligne

Ensuite, nous allons donc mettre une règle nat pour la translation d'adresse entre le VPN et le réseau 172.16.53.0/24. Pour cela, exécuter cette commande.

```
iptables -t nat -A POSTROUTING -s 10.0.8.0/24 -o Votre_interface_réseau -j MASQUERADE
iptables-save -c
```

Met en place le NAT et sauvegarde l'iptable

Une fois fait, redémarrer votre serveur

```
service openvpn restart
```

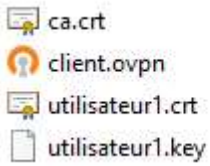
f. Configuration du VPN côté client

Pour ce qui est du côté client, il faut aller sur le site « OpenVPN » télécharger le client de « OpenVPN » qui se nomme « OpenVPN GUI ».

Nous devons copier le fichier « client.ovpn » qui se trouve dans le dossier « C:\Program Files\OpenVPN\sample-config ».

Puis le copier dans ce dossier « C:\Users\yohan\OpenVPN\config\ », qui est le dossier de configuration utilisateur, pour chaque utilisateur.

Ensuite, nous devons copier les clés et certificat généré, pour ma part j'ai utilisé un serveur samba, dans le dossier « C:\Users\votre_utilisateur\OpenVPN\config\ » au même emplacement que le fichier de configuration VPN client.



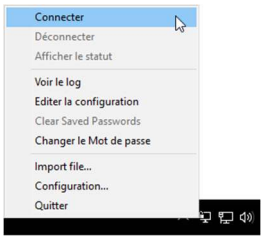
Pour établir la connexion, il faut modifier ces lignes dans le fichier de configuration.

```
client

dev tun

proto tcp

remote 200.65.89.62 1194

resolv-retry infinite

nobind

persist-key

persist-tun

ca ca.crt

cert utilisateur1.crt

key utilisateur1.key

remote-cert-tls server

cipher AES-256-CBC

verb 3
```

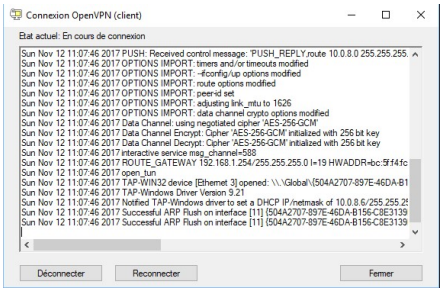
Fichier client.ovpn

g. Connexion au VPN

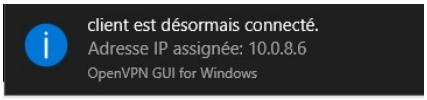
Pour se connecter au VPN, il faut fait clic droit sur l'icône « OpenVPN Gui » dans la barre des tâches, puis cliquez sur « connecter ».



Une fenêtre de log va apparaitre, avec les informations du client et de la connexion.



Une fois connecté, un message va apparaitre et nous informer de notre adresse IP, dans le réseau VPN en 10.0.8.0/24.



Cette icône  signifie que l'on est connecté. Si cette icône  apparait, le client essai de joindre le serveur VPN. Celui-là  signifie que l'on n'est pas connecté.

Pour ping les autres hôtes du réseau 172.16.5x.0, vous devez ajouter une route statique sur le routeur de la ferme de serveur : 10.0.8.0/24 vers 172.16.5x.100.

h. Conclusion

Si nous pouvons ping les serveurs et les autres clients, cela veut dire que le VPN, est fonctionnel. Si ce n'est pas le cas, vérifier que les passerelles sont bien les bonnes et que les interfaces du routeur et le NAT sont bien configurées sur chacun des serveurs.

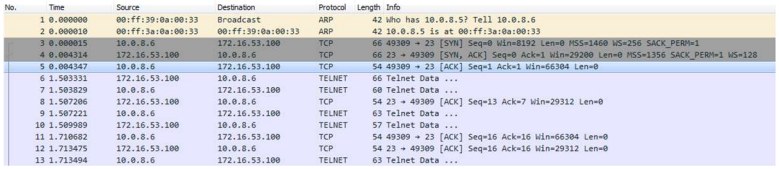
6. Observation des flux VPN

Une fois votre serveur OpenVPN fonctionnel, et si vous arrivez à vous connecter, on peut à ce moment observer les flux qui passent.

Pour ce test, nous allons utiliser un serveur Telnet, pour l'installer exécuter cette commande sur le serveur VPN et sur votre machine docker (En principe il est déjà installé).

```
apt-get install telnetd
```

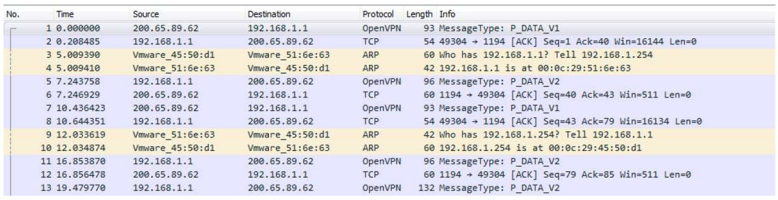
Connecter-vous à votre serveur openVPN en Telnet et sniffer les cartes réseau voici ce que l'on obtient :



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:ff:39:0a:00:33	Broadcast	ARP	42	Who has 10.0.8.5? Tell 10.0.8.6
2	0.000010	00:ff:3a:0a:00:33	00:ff:39:0a:00:33	ARP	42	10.0.8.5 is at 00:ff:3a:0a:00:33
3	0.000015	10.0.8.6	172.16.53.100	TCP	66	49309 → 23 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
4	0.004314	172.16.53.100	10.0.8.6	TCP	66	23 → 49309 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1356 SACK_PERM=1 WS=128
5	0.004347	10.0.8.6	172.16.53.100	TCP	54	49309 → 23 [ACK] Seq=1 Ack=1 Win=66304 Len=0
6	1.503331	172.16.53.100	10.0.8.6	TELNET	66	Telnet Data ...
7	1.503829	10.0.8.6	172.16.53.100	TELNET	60	Telnet Data ...
8	1.507206	172.16.53.100	10.0.8.6	TCP	54	23 → 49309 [ACK] Seq=13 Ack=7 Win=29312 Len=0
9	1.507221	10.0.8.6	172.16.53.100	TELNET	63	Telnet Data ...
10	1.509989	172.16.53.100	10.0.8.6	TELNET	57	Telnet Data ...
11	1.710682	10.0.8.6	172.16.53.100	TCP	54	49309 → 23 [ACK] Seq=16 Ack=16 Win=66304 Len=0
12	1.713475	172.16.53.100	10.0.8.6	TCP	54	23 → 49309 [ACK] Seq=16 Ack=16 Win=29312 Len=0
13	1.713494	10.0.8.6	172.16.53.100	TELNET	63	Telnet Data ...

Flux telnet à destination serveur OpenVPN

Contenue de l'échange telnet



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	200.65.89.62	192.168.1.1	OpenVPN	93	MessageType: P_DATA_V1
2	0.208485	192.168.1.1	200.65.89.62	TCP	54	49304 → 1194 [ACK] Seq=1 Ack=40 Win=16144 Len=0
3	5.009390	Vmware_45:50:d1	Vmware_51:6e:63	ARP	60	Who has 192.168.1.1? Tell 192.168.1.254
4	5.009410	Vmware_51:6e:63	Vmware_45:50:d1	ARP	42	192.168.1.1 is at 00:0c:29:51:6e:63
5	7.243758	192.168.1.1	200.65.89.62	OpenVPN	96	MessageType: P_DATA_V2
6	7.246929	200.65.89.62	192.168.1.1	TCP	60	1194 → 49304 [ACK] Seq=40 Ack=43 Win=511 Len=0
7	10.436423	200.65.89.62	192.168.1.1	OpenVPN	93	MessageType: P_DATA_V1
8	10.644351	192.168.1.1	200.65.89.62	TCP	54	49304 → 1194 [ACK] Seq=43 Ack=79 Win=16134 Len=0
9	12.033619	Vmware_51:6e:63	Vmware_45:50:d1	ARP	42	Who has 192.168.1.254? Tell 192.168.1.1
10	12.034874	Vmware_45:50:d1	Vmware_51:6e:63	ARP	60	192.168.1.254 is at 00:0c:29:45:50:d1
11	16.853870	192.168.1.1	200.65.89.62	OpenVPN	96	MessageType: P_DATA_V2
12	16.856478	200.65.89.62	192.168.1.1	TCP	60	1194 → 49304 [ACK] Seq=79 Ack=85 Win=511 Len=0
13	19.479770	192.168.1.1	200.65.89.62	OpenVPN	132	MessageType: P_DATA_V2

Les flux VPN envoyer sur le réseau

```
.%0.....m..."..R.....;..g.d
..,.........(H......f.!1.
.0......]B.....9........QW..%0.........l.n.;..3c.n.}s...@....B..D.
(H......g
..j=.-p.0..e
.Ny...m.M . '.y7H..LH......h..c...y[..Z.g.N.}
%k7.....~.A.CG.wm..kZ..T1Y.*.Fa3gf...H...L<...rU....I0.........#A.J...
lv6...f=....+m.x...Y*.+....#_..S...
4.:gCO.."......u..hB..@H......i.B-.]......o.M..Jl./.&     h.X.?.S.f
.........mx'e.....s.1.y.I0......A...+e.......s...lu.x.F."...2......
.........i.8.../....`._.5."G=..FH......j.i.......3T|.4....`
G.           .....W>.`.TX..L.;(..R.....;79k._....~.~.=0...
...`..N.,.i.m..Yp....!<...
```

Quand on affiche les échanges, ils sont cryptés