# NTP, Autokey et groupe de serveurs (IFF/GQ).

**Laurent Archambault <archi.laurent@gmail.com>**

# NTP, Autokey et groupe de serveurs (IFF/GQ).

par Laurent Archambault

# Table des matières

# Contexte

Le but de ce document n'est pas de décrire avec précision les échanges, et les flux nécessaires pour une authentification via « Autokey » dans un contexte de réseau NTP. Je vais m'efforcer d'expliquer comment installer cette authentification. Les mécanismes de sécurisation sont loin d'être simples, de plus ils sont coûteux en temps pour expliquer les détails.

L'ensemble des commandes, exemples... est applicable pour une distribution Gnu/Linux exclusivement, en architecture X86 pour les options de compilation.

Cette documentation s'adresse à des administrateurs possédant des bases sur le service NTP. Le cas échéant, la lecture du document suivant (du même auteur) devrait permettre une meilleure approche : http://archi.laurent.perso.neuf.fr/Doc_reseauNTP.html

# Chapitre 1. L'authentification via `autokey`.

## La nécessité d'authentifier.

Vous n'êtes pas obliger de réaliser une authentification des flux, néanmoins renseignez-vous sur le « Clogging Attack/Clogging Vulnerability ». Vous allez vous apercevoir qu'un réseau (NTP) peut-être fragile. Il est nécessaire d'authentifier une ou plusieurs relations NTP dans les situations suivantes, il peut en exister d'autres :

- Pour le broadcast (fortement conseillé).
- Pour les relations symétriques (client/serveur).
- Pour éviter les « clogging attack / Clogging Vulnerability ».
- Pour sécuriser son serveur vis-à-vis de requêtes des clients.

## Le choix des schémas.

Il existe 5 schémas NTP possibles, dont 2 presque spécifiques pour le broadcast (PC et MV).

- TC est dédié aux certificats générés par un PKI (autres que « ntp-keygen » et « Openssl »).
- IFF peut travailler avec des certificats créés par un PKI ou avec « ntp-keygen ». C'est le schéma le plus utilisé, et celui que j'ai choisi pour cette documentation.
- GQ est dédié aux certificats uniquement générés par « ntp-keygen ». Concernant les paramètres pour "IFF" et "GQ" seules les options pour "ntp-keygen" changent.

## Pourquoi faire un ou des groupes.

Un groupe "NTP" consiste en la présence de plusieurs serveurs appartenant à une ou plusieurs STRATES. Ce même groupe possède également des clients(serveurs) appartenant à ce même groupe. Tous les clients doivent avoir des noms de machines différents, c'est un impératif. L'option "-H" ne doit pas être utilisée pour le certificat, contrairement à l'option "-i" suivi du nom du groupe concerné. L'identification d'un « client » est réalisée par une clé spécifique. Un client ne peut s'identifier dans un groupe autre que le sien. Le certificat et son mot de passe (non privé) doivent être diffusés aux abonnés du groupe sans limites. Sur ce point, la transmission du mot de passe doit être sécurisée, c'est la clé de tout. La différence se faisant ensuite par les clés d'identifications en particulier.

Le groupe apporte également l'avantage d'une gestion des clés grandement simplifiée. L'autorité de certification (TA) ne fabrique plus une clé par client, mais une clé pour un groupe.

# Chapitre 2. Le logiciel NTP.

## L'installation.

Le logiciel pour le NTP est la version « ntp-dev-4.2.5p249-RC » , cette version et les suivantes sont disponibles sur le site de « ntp.org » :

http://www.ntp.org/downloads.html.

C'est effectivement une version dite instable, néanmoins je n'ai rencontré aucun problème avec cette version (presque...). Ce choix s'est basé sur diverses modifications du code concernant l'authentification via « autokey », c'est aussi la future version.

## La compilation.

La version de NTP utilisé est en développement, il n'existe aucun package quelconque pour une distribution. La compilation est donc une étape indispensable, elle permet également une adaptation du logiciel en fonction de vos besoins.

Pour la gestion des clés, NTP utilise Openssl. Vous devez installer ce produit avant de procéder à la compilation de NTP. Vous avez 2 possibilités. Soit installer la version de votre système via ce type de commande **apt-get install openssl openssl-dev**, ou par compilation. J'ai choisi ce choix, et compte tenu des problèmes de sécurité que rencontre parfois ce programme (pourtant hautement fiable), sa mise à jour par cette méthode est presque devenue une nécessité pour un administrateur.

Voici les options que j'ai sélectionnées :

```
#! /bin/sh
make clean dep ;
./configure --host=i686-pc-linux-gnu \
--prefix=/usr \
--mandir=/usr/share/man \
--infodir=/usr/share/info \
--datadir=/usr/share \
--sysconfdir=/etc/ \
--localstatedir=/var \
--with-libtool \
--disable-linux-caps \
--build=i686-pc-linux-gnu \
--enable-debugging \
make && make install
```

*Bien noter que la directive **--prefix** cible le répertoire **/usr**. Les binaires seront donc installés dans **/usr/bin** et / **usr/sbin**. C'est mon choix; néanmoins il est de bon ton d'installer dans le répertoire (pratiquement définie à cet usage) **/usr/local**.*

Une fois installé il reste à configurer le serveur. Pour mémoire ce document ne traite pas ce thème, vous pouvez vous référer à ce lien pour y parvenir :

http://archi.laurent.perso.neuf.fr/Doc_reseauNTP.html (du même auteur).

## Le fichier `leap.seconds`.

Ce fichier est marqué dans toute les documentations comme impératives pour une utilisation de « autokey ». Pour mémoire ce fichier recense les diverses modifications entre le temps TAI(atomique) et UTC(Terre). Pour installer ce fichier sur chaque serveur, il suffit de récupérer le fichier au travers de ce lien :

ftp://time.nist.gov/pub/.

Le fichier à récupérer doit ressembler à celui là, car il peut évoluer : `leap-seconds.3427142400`.

Ce fichier doit être installé dans le répertoire /etc/ntp (là ou est présent "ntp.conf"), et relié par un lien.

```
ln -sf leap-seconds.3427142400 ntpkey_leap
```

Voilà ce n'est pas plus compliqué que çà, et pour avoir vécu cette situation, ce fichier est indispensable.

# Chapitre 3. Les certificats.

## ntp-keygen

Le logiciel « ntp-keygen » est inclus dans les sources de NTP, voici les options possibles :

```
# ntp-keygen -h
Using OpenSSL version 90807f
ntp-keygen: illegal option -- h
ntp-keygen (ntp) - Create a NTP host key - Ver. 4.2.5p242-RC
USAGE:  ntp-keygen [ flag [val] | --name[{=| }val] ]...
  Flg Arg Option-Name    Description
   -c Str certificate    certificate scheme
   -d no  debug-level    Increase output debug message level
   -D Str set-debug-level Set the output debug message level
   -e no  id-key         Write IFF or GQ identity keys
   -G no  gq-params      Generate GQ parameters and keys
   -H no  host-key       generate RSA host key
   -I no  iffkey         generate IFF parameters
   -i Str issuer-name    set issuer name
   -M no  md5key         generate MD5 keys
   -m Num modulus        modulus
   -P no  pvt-cert       generate PC private certificate
   -p Str pvt-passwd     output private password
   -q Str get-pvt-passwd input private password
   -S Str sign-key       generate sign key (RSA or DSA)
   -s Str subject-name   set subject name
   -T no  trusted-cert   trusted certificate (TC scheme)
   -V Num mv-params      generate num MV parameters
   -v Num mv-keys        update num MV keys
      opt version        Output version information and exit
   -? no  help           Display usage information and exit
   -! no  more-help      Extended usage information passed thru pager
   - opt save-opts      Save the option state to a config file
   - Str load-opts      Load options from a config file

Options are specified by doubled hyphens and their name
or by a single hyphen and the flag character.

please send bug reports to:  http://bugs.ntp.org, bugs(AT)ntp.org
```

Ce programme génère les certificats et les clés qui sont utilisées par le protocole NTP en version 4. Les différents fichiers qui sont créent interviennent pour le chiffrement symétrique, on peut trouver :

- des clés MD5,
- des clés de signature
- des certificats (X509 Version 3 - format PEM),
- des clés d'identité,
- des "clés" d'identités du schéma utilisé,
- des clés de session (cookies),
- des mots de passe MD5.

Toutes ces données sont prises en compte par « ntp-keygen » sous couvert de « Openssl ». Il reste compatible avec NTP dans sa version 3. Néanmoins, un PKI peut générer les certificats sans toutefois utiliser des extensions spécifiques à NTP comme l'identité. En conséquence, ce type d'infrastructure ne peut être conseillé, ou alors en mode dégradé. De plus la vérification du certificat auprès de ce PKI, complique l'acheminement des paquets.

L'encodage pour le certificat est fixé à 512 bits, cette imposition est le choix des développeurs de NTP. Les flux entre serveurs doivent être sécurisés, sans pour cela prendre une place trop importante au niveau de la charge du serveur.

## Avertissement

Lors de mes essais j'ai placé les certificats, les clés... dans le répertoire `/etc/ntp/`. Libre à vous de les placer ailleurs, de créer des sous répertoires, attention aux paramètres à modifier dans le fichier `ntp.conf`

Bien prendre en compte que « ntp-keygen » génère les fichiers dans le répertoire de travail, (là ou vous êtes).

# Chapitre 4. Les serveurs NTP.

Mon réseau et les différents exemples qui vont arriver seront basés sur 2 machines (hélas). L'une d'elle s'appelle « serveur » et appartient au groupe « GR1 (+GR2) ». L'autre machine porte le nom de « portable », du groupe « GR1 ». Ses 2 serveurs possèdent les mêmes caractéristiques NTP. Voici d'autres données pour la suite des certificats :

```
serveur     serveur.archi.amt   192.168.1.11    GR1 et GR2, et autorité de certification
portable    portable.archi.amt  192.168.1.203   GR1, et autorité de certification (GR1).
```

Ne cherchez pas à comprendre pourquoi le serveur NTP s'appelle « portable », et le client NTP « serveur »...

# Les mots de passe et les options.

- ***"-p"***

  Représente le mot de passe "privé" pour lire et chiffrer un certificat. Si cette option n'est pas renseignée, Le mot de passe par défaut est le nom de la machine (sans le domaine). Ce mot de passe est réservé à un usage local, il ne doit pas être communiqué.

- ***"-q"***

  Ce mot de passe "public" permet d'extraire les données cryptées envoyées par un serveur, pour le schéma IFF ou GQ. Ce mot de passe permet d'encoder(chiffrer) des données et de lire d'autres données transmises par un client(serveur). Si cette option n'est pas demandée, le mot de passe par défaut est le nom de la machine (sans le domaine). C'est ce mot de passe qui est à transmettre en même temps que le certificat au groupe (GR1).

# Création des certificats

## Pour le serveur NTP ("portable")

Voici la commande à exécuter pour la création d'un certificat « auto signé, valable pour un groupe (GR1) » :

- L'option **-p**, renseigne le mot de passe privé.
- L'option **-T**, permet de signer le certificat.
- L'option **-I**, permet de choisir le schéma « IFF ».

```
ntp-keygen -p private -T -I -i GR1
Using OpenSSL version 90807f
Using host portable.archi.amt group GR1
Generating RSA keys (512 bits)...
RSA 0 1 7       1 11 24    3 1 2
Generating new host file and link
ntpkey_host_portable.archi.amt->ntpkey_RSAhost_portable.archi.amt.3467897340
Using host key as sign key
Generating IFF keys (256 bits)...
IFF 0 408 416   1 49 160 2 1 2    3 1 4
Confirm g^(q - b) g^b = 1 mod p: yes
Confirm g^k = g^(k + b r) g^(q - b) r: yes
Generating new iffkey file and link
ntpkey_iffkey_GR1->ntpkey_IFFkey_GR1.3467897340
Generating new certificate GR1 RSA-MD5
X509v3 Basic Constraints: critical,CA:TRUE
X509v3 Key Usage: digitalSignature,keyCertSign
X509v3 Extended Key Usage: trustRoot
Generating new cert file and link
ntpkey_cert_portable.archi.amt->ntpkey_RSA-MD5cert_portable.archi.amt.3467897340
```

Le fichier « ntpkey_cert_portable.archi.amt » est au format X509 version 3 (.PEM), pour information.

```
lrwxrwxrwx  1 root    root       48 2009-11-22 17:49 ntpkey_cert_portable.archi.amt -> n
lrwxrwxrwx  1 root    root       44 2009-11-22 17:49 ntpkey_host_portable.archi.amt -> n
lrwxrwxrwx  1 root    root       28 2009-11-22 17:49 ntpkey_iffkey_GR1 -> ntpkey_IFFkey_
-rw-r--r--  1 root    root      530 2009-11-22 17:49 ntpkey_IFFkey_GR1.3467897340
```

```
-rw-r--r--  1 root    root      631 2009-11-22 17:49 ntpkey_RSAhost_portable.archi.amt.3
-rw-r--r--  1 root    root      576 2009-11-22 17:49 ntpkey_RSA-MD5cert_portable.archi.a
```

L'ensemble de ses fichiers a été généré par la commande précédente. Il n'y a rien à faire par la suite, les liens sont réalisés...

La commande suivante est chargée de la création de la clé privée qui sera par la suite diffusée aux différents clients d'un groupe. Le mot de passe *client* est nécessaire pour décrypter cette même clé, l'autre mot de passe est nécessaire pour l'encodage.

```
ntp-keygen -q client -p private -e > temporaire
Using OpenSSL version 90807f
Using host portable.archi.amt group GR1
Using host key ntpkey_RSAhost_portable.archi.amt.3467897340
Using host key as sign key
Using IFF keys ntpkey_IFFkey_GR1.3467897340
Writing IFF parameters ntpkey_iffpar_GR1.3467897340 to stdout
Writing IFF keys ntpkey_iffkey_GR1.3467897340 to stdout
```

Voici à quoi ressemble ce type de fichier.

```
cat temporaire
# ntpkey_iffpar_GR1.3467897340
# Sun Nov 22 17:49:00 2009

-----BEGIN DSA PRIVATE KEY-----
MIHkAgEAAkEApsnlbe+X6ygkey1cyVw2DXVfS5wTdWJ3prdeG0WkODbPXxJDPD6f
9KFwU9ibYyrkAwRsQ+ohEju+TL2uFzu7nwIVALJpNFlXwXC5axYsvXvbsE4AUf5L
AkAeqwej0ocyIPYBavKGGaOBRtnkxqbJ9tu217NPIleo1FgZxceFfY1nAk6Nnx35
G4wfxe/zIfAcK3WTwZK5z0yEAkAgSyGb1mU+I3CMNGvL5adb4enzUaoUwsDjAq0F
fDgFrALUs5+BTpERetf0Q0UnuC1nAYmCbo7v9Susm1b4Gd2tAgEB
-----END DSA PRIVATE KEY-----
```

Ce fichier doit être transmis aux clients du groupe GR1. Il faut également et impérativement transmettre le mot de passe (« client »). La sécurisation des flux est assurée par le certificat, mais aussi par le mot de passe (« client »), le certificat peut donc être diffusé librement, par exemple sur un serveur HTTP, FTP...

Avec un mode de diffusion comme FTP ou HTTP, le client peut même créer un script pour la récupération du certificat (TA) automatiquement.

A ce stade ce serveur est prêt à fonctionner, vous pouvez lancer le service NTP et attendre...

### Avertissement

L'authentification est un processus assez long entre serveurs, et parfois variable en plus. Comptez plusieurs minutes (environ 5 min) avant de vérifier vos statuts de connexions.

## Pour le client (serveur NTP tout de même)

La procédure est simple, voici la commande à réaliser : **ntp-keygen -H -p client**, ou « client », représente le mot de passe « générique » du groupe (GR1).

# Modification sur le fichier `ntp.conf`

### Avertissement

L'authentification par groupe est souvent conseillé pour une connexion symétrique. A l'heure où j'écris et ne sachant pourquoi l'instruction **peer serveurX autokey** provoque une erreur de type « unsupsported_identity_type ». Néanmoins la connexion symétrique est activée.

en conséquence, cette instruction ne sera pas utilisée pour le moment, peut-être est dû à la version « instable » de NTP.

## Pour le « serveur » ("portable" / 192.168.1.203) et TA.

```
serveur 192.168.1.11   # pas obligatoire
driftfile /etc/ntp/ntp.drift
```

```
crypto pw private ident GR1
keysdir /etc/ntp/                    # ou sont les clés...
restrict 192.168.1.11 mask 255.255.255.0 nomodify       # mon autre srv
```

### Note

A noter que cette machine ne fait pas de « peer », elle est considérée comme ma référence. Cette même machine dite de « référence »ne fait que diffuser, en conséquence elle ne demande aucune authentification pour les requêtes(flux) entrantes. Cette situation est à inverser par contre vis-à-vis des clients de ce serveur(!).

## Pour le « client » ("serveur" / 192.168.1.11).

```
server 192.168.1.203 iburst autokey
peer 192.168.1.203 autokey # vers portable (provoque des erreurs, MAIS FONCTIONNE ! (?)
driftfile /etc/ntp/ntp.drift
crypto pw client ident GR1
keysdir /etc/ntp/
restrict 192.168.1.203 mask 255.255.255.0 notrust  # l'oblige a "venir" authentifié
```

### Note

Le paramètre « notrust » demande à ce que le serveur s'authentifie pour toutes les requêtes NTP.

## La partie commune :

Concerne l'ensemble des fichiers et paramètres pour les historiques, dont la partie authentification.

```
statsdir /var/log/ntpstats/
# Statistiques desirees
statistics loopstats peerstats sysstats cryptostats
#
filegen loopstats file loopstats type week enable
filegen peerstats file peerstats type week enable
filegen sysstats file sysstats type week enable
filegen cryptostats file cryptostats type week enable
```

# Les contrôles.

## Le mode « DEBUG » de ntpd

**`/etc/init.d/ntp stop ; /usr/sbin/ntpd -c /etc/ntp/ntp.conf -D2`**

La valeur « -Dx », ou « x » représente un chiffre de 1 à 5. Plus le chiffre est grand, et plus les logs sont présents, à vous de voir.

```
session_key: 192.168.1.11 > 192.168.1.203 e32550b5 90cf473b hash a5a52626 life 2
receive: at 1142 192.168.1.203 --192.168.1.11 mode 1 keyid e32550b5 len 68 auth 1
poll_update: at 1142 192.168.1.11 poll 7 burst 0 retry 0 head 0 early 2 next 40
```

La valeur *`auth 1`* est la bonne valeur.

## Par les fichiers d'historiques.

Le fichier `cryptostats` :

```
55160 75609.284 0.0.0.0 signature update ts 3468171609
55160 75609.284 192.168.1.203 sign serveur.archi.amt GR1 0x4 md5WithRSAEncryption (8) f
55160 75738.640 192.168.1.203 assoc 3775 3775 host GR1 md5WithRSAEncryption
55160 75740.640 0.0.0.0 signature update ts 3468171740
55160 75740.640 192.168.1.203 cert GR1 GR1 0x5 md5WithRSAEncryption (8) fs 3468153591
55160 75772.647 192.168.1.203 iff GR1 fs 3468153591
55160 75834.649 192.168.1.203 cook 2fa21542 ts 3468171834 fs 3468171667
55160 75896.648 0.0.0.0 signature update ts 3468171896
```

Tout est présent : la signature, le certificat, le groupe, le schéma et le cookie...

## Par les programmes « ntpq / ntpdc »

```
watch -n5 ntpq -c as
ind assid status  conf reach auth condition  last_event cnt
===========================================================
  1 20452  8011   yes    no  none    reject    mobilize  1
  2 20453  963a   yes   yes  none  sys.peer    sys_peer  3
  3 20454  f424   yes   yes   ok  candidate   reachable  2
  4 20455  8043   yes    no  none    reject unreachable  4
```

Et pour connaître qui se cache derrière le chiffre « 20454 », il suffit de valider la commande suivante. Il s'agit bien de « serveur.archi.amt » du group « GR1 », et authentifié (« auth »).

La colonne « condition » mérite d'exposer les différents états qu'elle peut retourner, à l'exception de "sys.peer" :

- candidate : le serveur est considéré comme une bonne source.
- outlyer : la qualité n'est pas suffisamment bonne.
- falseticker : les données sont de mauvaises qualités (NTP).
- reject : les flux sont rejetés ! (temporaire dans ma situation).

```
ntpq
 ntpq> pstatus 20454
 associd=20454 status=f43a conf, authenb, auth, reach, sel_candidate, 3 events, sys_peer
 srcadr=portable.archi.amt, srcport=123, dstadr=192.168.1.11, dstport=123,
 leap=00, stratum=3, precision=-20, rootdelay=67.993, rootdisp=48.798,
 refid=81.19.16.225,
 reftime=ceb89d87.c51dbd30  Thu, Nov 26 2009  7:24:07.769,
 rec=ceb89e4e.50c6730f  Thu, Nov 26 2009  7:27:26.315, reach=377,
 unreach=0, hmode=3, pmode=4, hpoll=7, ppoll=7, headway=138, flash=00 ok,
 keyid=3927447286, offset=1.931, delay=1.526, dispersion=5.404,
 jitter=0.898, xleave=0.044,
 filtdelay=     5.55    1.79    1.53    1.55    3.36    1.53    1.53    1.54,
 filtoffset=   -0.07    1.73    1.78    1.68    2.62    1.93    2.30    2.88,
 filtdisp=      0.00    2.03    3.05    4.08    5.07    6.09    7.11    8.15,
 host="GR1", flags=0x87f21, signature="md5WithRSAEncryption"
```

# Durée et renouvellement des certificats/clés.

## Les certificats (partie serveur)

La durée maximale est de 365 jours par défaut, il est conseillé de renouveler celui-ci bien avant cette date, par sécurité.

Voici la commande à utiliser qui reste identique à la commande initiale (cf à la section intitulée « Les certificats (partie serveur) »). Cette action nécessite de relancer le service NTP juste après.

```
ntp-keygen -T -p private -i GR1

Using OpenSSL version 90807f
Using host portable.archi.amt group GR1
Generating RSA keys (512 bits)...
RSA 0 2 252     1 11 24    3 1 2
Generating new host file and link
ntpkey_host_portable.archi.amt->ntpkey_RSAhost_portable.archi.amt.3468637201
Using host key as sign key
Generating new certificate GR1 RSA-MD5
X509v3 Basic Constraints: critical,CA:TRUE
X509v3 Key Usage: digitalSignature,keyCertSign
X509v3 Extended Key Usage: trustRoot
Generating new cert file and link
ntpkey_cert_portable.archi.amt->ntpkey_RSA-MD5cert_portable.archi.amt.3468637201
```

C'est la même commande que la commande initiale sans l'option « -I ». Ceci fonctionne très bien, le groupe est reconnu, vous pouvez également activer ce paramètre sans problème. Le résultat est le même.

# Pour le client.

La durée maximale des clés est de 30 jours, vous devez donc (hélas) relancer votre serveur une fois par mois. Voici la commande nécessaire pour cette mise à jour, qui nécessite de relancer le service NTP juste après.

```
ntp-keygen -p client
```

```
Using OpenSSL version 90807f
Using host serveur.archi.amt group serveur.archi.amt
Using host key ntpkey_RSAhost_serveur.archi.amt.3468430652

Using host key as sign key
Generating new certificate serveur.archi.amt RSA-MD5
X509v3 Basic Constraints: critical,CA:TRUE
X509v3 Key Usage: digitalSignature,keyCertSign
Generating new cert file and link
ntpkey_cert_serveur.archi.amt->ntpkey_RSA-MD5cert_serveur.archi.amt.3468506505
```

## Le contrôle d'un certificat.

Voici une commande qui peut aider à retrouver les dates de validités d'un certificat. Et pour avoir essayé aussi, vous pouvez intégrer dans Firefox ce fichier, il sera reconnu comme certificat, même si effectivement une fois importé il ne sert à rien.

```
openssl x509 -startdate -enddate -in ntpkey_cert_portable.archi.amt
```

```
notBefore=Nov 21 14:06:49 2009 GMT
notAfter= Nov 21 14:06:49 2010 GMT
-----BEGIN CERTIFICATE-----
MIIBQzCB7qADAgECAgTOsnJ5MA0GCSqGSIb3DQEBBAUAMA4xDDAKBgNVBAMTA0dS
MTAeFw0wOTExMjExNDA2NDlaFw0xMDExMjExNDA2NDlaMA4xDDAKBgNVBAMTA0dS
MTBaMA0GCSqGSIb3DQEBAQUAA0kAMEYCQQDCesW4bHDyntmGexrfQXnbUijEu5VZ
81er05ADxEESH2VFGDrAoR2CnMPWbmoSk5sM9zPCBn2E5bKDUIfkCI+dAgEDozYw
NDAPBgNVHRMBAf8EBTADAQH/MAsGA1UdDwQEAwIChDAUBgNVHSUEDTALBgkrBgEF
BQcwAQswDQYJKoZIhvcNAQEEBQADQQC7DysKFjH+pwbv7mI97gWWFnrtcvJGNFJq
 YU6WXiL8SuO2+sebuDPL2NvAYOuZQU2//VZz+6PHjl+D+GHPG+qp
 -----END CERTIFICATE-----
```

### Avertissement

Bien faire attention c'est l'heure en GMT, çà peut tout changer...

# Chapitre 5. Conclusion

Ce document est lié à la sortie de la version stable en version 4.2.5 de ntp. En effet l'option *`peer serveurX autokey`*, semble provoquer des messages d'erreurs. Néanmoins, tous les autres paramètres fonctionnent très bien, ceci concerne également le « broadcast autokey ». Mes différents essais ont montré que les nouvelles versions du logiciel « ntp », même en développement restent compatibles avec les versions antérieures. La dernière version que j'ai testée est la « ntp-dev-4.2.5p249-RC ». Je suis avec intérêt cette évolution, envers justement la fonction « autokey »

à suivre.

# Annexe A. Le fichier : « ntp.conf »

## Exemple pour le serveur (portable).

Ce fichier représente le serveur (TA) « portable », les connexions authentifiées sont toutes effectuées entre le client et ce même serveur. Ceci dans le but de réduire la charge CPU de ce serveur, c'est aussi pour moi ma référence de STRATE « basse » sur mon humble réseau. Il est sécurisé donc, et ne demande pas que des connexions authentifiées (serveurs internet obligent aussi).

```
#
# Fichier de configuration pour ntpd
# Nom de ce fichier   = /etc/ntp/ntp.conf
# Repertoire des cles = /etc/ntp/keys (rep)
#
# ////////////// SERVEURS \\\\\\\\\\\\\\
# Declaration des serveurs classiques
# server krishna.via.ecp.fr prefer    # fiable -> 138.195.130.71 - strate 2
# server 195.83.66.158 iburst prefer  # IP de ntp.univ-poitiers.fr
# server 145.238.203.10  # poitiers uni
server 193.55.167.1  # ntp.ensma.fr 2/2
server 81.19.16.225  # ntp1.adviseao.net
server 80.74.64.1  # ns1.pulsation.fr
server 192.168.1.11  # mon autre srv
# peer 192.168.1.11 autokey
#
# déclaration/activation de mon serveur en strate 3
#  avec auth uniquement pour le peer.
server 127.127.1.0 # local horloge
fudge 127.127.1.0 stratum 2 refid Port2
# fudge 127.127.1.0 stratum 2 refid Port2
#
# ////////////// DRIFT \\\\\\\\\\\\\\\
# Emplacement du fichier DRIFT
# par défaut : driftfile /var/lib/ntp/ntp.drift
# désarmais ici avec les siens :
driftfile /etc/ntp/ntp.drift
#
# ////////// GESTION DES CLES \\\\\\\\\\
#
crypto pw private ident GR1
keysdir /etc/ntp/
# randfile /dev/urandon     # franchement pas obligatoire
#
#////////////// BROADCAST/MANYCAST/MULTICAST \\\\\
# Pas de broadcast ni de crypto ...
#
# broadcast 192.168.1.255 autokey
# broadcastdelay 0.0008
# broadcastclient   # broadcast vis a vis d'un autre réseau(mode client!)
#
# enable bclient    # Active le broadcast client (pour un serveur)
#
# manycastserver 239.1.1.1 autokey maxpoll 12 ttl 1
# ttl 1 = pas de routeur chez moi ...
# manycastclient 239.1.1.1 maxpoll 12 ttl 1 autokey
#
# multicastclient     # Ecoute sur 224.0.1.1
#
#////////////// RESTRICTIONS \\\\\\\\\\
# Les restrictions d'accès :
# Ajouter vos réseaux dans les restrictions ...
# La limitation globale :
#
# restrict default limited noserve kod
restrict default limited noserve kod nopeer nomodify notrap
#
```

```
# pour le serveur qui a X IP :
restrict 80.74.64.1 noquery nomodify
restrict 80.74.64.2 noquery nomodify
restrict 81.19.16.225 noquery nomodify # ntp1.adviseao.net
restrict 195.83.66.158 noquery nomodify # univ-poitiers.fr
restrict 193.55.167.2 noquery nomodify # ensma
restrict 193.55.167.1 noquery nomodify  # ensma
# pour le serveur :
restrict 138.195.130.71 noquery nomodify
# pour mon horloge locale et service locale
restrict 127.0.0.1 mask 255.0.0.0
# pour mon reseau locale
restrict 192.168.1.11 mask 255.255.255.0 nomodify                    # mon autre srv
#restrict 192.168.1.11 mask 255.255.255.0 nomodify notrust  # mon autre srv
restrict 192.168.1.13 mask 255.255.255.0 nomodify           # mon autre srv
restrict 192.168.1.203 mask 255.255.255.0 nmodify          # mon autre srv
# pour mon reseau locale
restrict 10.0.2.15 mask 255.255.255.0
# pour le reste de mon reseau ... soit plus grand chose, mais bon...
restrict 192.168.1.0 mask 255.255.255.0 nomodify nopeer kod version limited notrap
# pour manycast ... en test
restrict 239.1.1.1 nomodify nopeer kod
restrict 10.0.2.0 nomodify nopeer kod
#
#////////// DIVERS - STATISTIQUES - LOG \\\\\\\\\\
#Active les stats... stats/monitor/ntp = par défaut
# Active l'authentification. (broadcast...)
#
# enable auth
#
# Fichier des traces de ntpd
logfile /var/log/ntpd.log
#
# Repertoire contenant les fichiers de stats
statsdir /var/log/ntpstats/
#
# # Statistiques desirees
statistics loopstats peerstats sysstats cryptostats
#
filegen loopstats file loopstats type week enable
filegen peerstats file peerstats type week enable
filegen sysstats file sysstats type week enable
filegen cryptostats file cryptostats type week enable
#
# A activer/modifier si vous avez une source (strate 0) :
# statistics loopstats peerstats clockstats sysstats
# filegen clockstats file clockstats type week enable
# ------------------ FIN -------------------
```

# Annexe B. Divers

## Les « flags » Autokey

Les valeurs suivantes sont visibles lors de la commande : **ntpq pstatus id**, exemple « flags=0x87f21 ». En partant de droite à gauche, ce drapeau est décomposable comme ceci :

- 1 = authentification activée
- 20 = IFF reconnu
- F = (lié au schéma (?))
- 7 = 0x1000 + 0x2000 + 0x4000 = Autokey + Certificat + leapseconds sont correctes.

### Avertissement

Nombreuse sont les demandes qui posent la question et le reste soit « 0x8 », et je n'ai toujours pas la réponse...un mystère.

```
>  */ The following bits are set by the CRYPTO_ASSOC message from
>  * the server and are not modified by the client.
>  */
> #define CRYPTO_FLAG_ENAB  0x0001 /* crypto enable */
> #define CRYPTO_FLAG_TAI   0x0002 /* leapseconds table */
> #define CRYPTO_FLAG_PRIV  0x0010 /* PC identity scheme */
> #define CRYPTO_FLAG_IFF   0x0020 /* IFF identity scheme */
> #define CRYPTO_FLAG_GQ    0x0040 /* GQ identity scheme */
> #define CRYPTO_FLAG_MV    0x0080 /* MV identity scheme */
> #define CRYPTO_FLAG_MASK  0x00f0 /* identity scheme mask */
> #define CRYPTO_FLAG_VALID 0x0100 /* public key verified */
> #define CRYPTO_FLAG_VRFY  0x0200 /* identity verified */
> #define CRYPTO_FLAG_PROV  0x0400 /* signature verified */
> #define CRYPTO_FLAG_AGREE 0x0800 /* cookie verifed */
> #define CRYPTO_FLAG_AUTO  0x1000 /* autokey verified */
> #define CRYPTO_FLAG_SIGN  0x2000 /* certificate signed */
> #define CRYPTO_FLAG_LEAP  0x4000 /* leapseconds table verified */
```

# Annexe C. Autres liens

- NTP server misuse and abuse (GB): http://en.wikipedia.org/wiki/NTP_server_misuse_and_abuse#Common_NTP_client_problems
- Le Draft IETF sur Autokey (GB): http://tools.ietf.org/html/draft-ietf-ntp-autokey-07#section-4
- Le réseau / protocole NTP: http://archi.laurent.perso.neuf.fr/Doc_reseauNTP.html
- Comment configurer Autokey (BGB):http://support.ntp.org/bin/view/Support/ConfiguringAutokey