# RFC 2777 : Publicly Verifiable Nomcom Random Selection

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 décembre 2009

Date de publication du RFC : Février 2000

https://www.bortzmeyer.org/2777.html

---

Ce curieux RFC répond à une question apparemment paradoxale : comment tirer au hasard de manière publiquement vérifiable ? Par exemple, les membres d'un des comités de l'IETF, le Nomcom, sont choisis au hasard. Comment permettre à tous les participants à l'IETF de vérifier que le tirage était bien aléatoire, alors que l'opération se fait sur l'Internet, sans qu'on puisse vérifier le dé ou la roulette ?

Une solution traditionnelle est de désigner une autorité arbitraire, considérée comme digne de confiance, et dont on n'a pas le droit de remettre l'honnêteté en doute. C'est ce que font toutes les loteries qui se déroulent « sous contrôle d'huissier ». Une telle méthode, basée sur la confiance aveugle en une entité particulière, ne colle pas avec la culture IETF.

Une autre méthode est proposée par notre RFC 2777 [1]. Utiliser un certain nombre de sources d'entropie, **publiquement** consultables, et les passer à travers une fonction de hachage qui garantira la dispersion des résultats, rendant ainsi très difficile d'influencer ceux-ci. Et chacun pourra faire tourner l'algorithme chez lui, vérifiant ainsi que le résultat est honnête.

La section 1 du RFC rappelle le problème original, la désignation du Nomcom ("*Nominating Committee*", cf. RFC 2727) de l'IETF. Mais, depuis, la méthode de ce RFC a été utilisée dans bien d'autres cas.

La section 2 décrit les étapes à suivre :
— Identifier les candidats possibles (le RFC décrit le cas du Nomcom mais cette étape existe pour toute désignation),

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc2777.txt

— Publication de l'algorithme **et** des sources d'entropie utilisées, pour permettre la future vérification. Cette étape est indispensable car la possibilité de tous les participants de contrôler eux-même l'honnêteté du processus est vitale. (À noter que c'est justement cette possibilité qui est retirée au citoyen dans le cas de mesures anti-démocratiques comme le vote électronique <https://www.bortzmeyer.org/non-au-vote-electronique.html>.)
— Faire tourner l'algorithme et publier le résultat.

Ça, c'était le principe général. Maintenant, le gros problème est évidemment le choix de bonnes sources aléatoires. Toute la section 3 est vouée à cette question. La source doit produire un résultat qui ne peut pas être facilement influencé et qui doit être accessible publiquement. (La section 6 détaille les problèmes de sécurité.)

La section 3.1 analyse les sources possibles et en identifie trois :

— Les loteries sérieusement organisées, notamment celles faites par l'État. Notez qu'une loterie particulière peut être truquée mais, d'une part, l'algorithme repose sur plusieurs sources, d'autre part les gens qui sont prêts à truquer une loterie le feront plutôt pour des gains financiers, pas pour perturber l'IETF.
— Les résultats financiers comme le prix de vente d'une action à la clôture de la Bourse ou le volume d'actions échangée en une journée. Ce ne sont pas des nombres aléatoires mais leur valeur dépend des actes de beaucoup d'opérateurs séparés, en pratique, le résultat exact sera imprévisible.
— Les événements sportifs mais ils posent pas mal de problèmes pratiques comme les risques d'annulation ou de retard.

Attention à ne pas mettre les plus mauvaises sources en dernier, car, avec la connaissance des autres sources, publiées avant, un attaquant qui peut influencer la dernière source aurait un pouvoir excessif.

On pourrait penser qu'il faut multiplier le nombre de sources, pour limiter le pouvoir de nuisance d'une source manipulée. Mais le RFC note que le mieux est l'ennemi du bien : chaque source supplémentaire est également une source de problème, par exemple si elle ne produit aucun résultat (évenement annulé, par exemple). Il recommande environ trois sources.

Aucun des types de source cités plus haut ne produit une distribution uniforme. C'est une des raisons pour lesquelles on doit les faire passer à travers une fonction de hachage (cf. RFC 4086) pour obtenir cette uniformité. Une autre raison est que, pour certaines sources, leur valeur approximative peut être influencée (le cours en Bourse d'une action peut grimper ou descendre selon la volonté d'un gros opérateur) mais pas la totalité des chiffres. Le hachage empêchera donc ledit influenceur d'obtenir le résultat qu'il désire (section 3.2).

La section 3.3 calcule la quantité d'entropie nécessaire pour sélectionner P vainqueurs parmi N possibilités. Par exemple, pour 100 candidats et 10 places, il faut 44 bits d'entropie. (À noter que la fonction de hachage MD5, utilisée par notre RFC en suivant le RFC 1321 ne fournit « que » 128 bits d'entropie.)

Un algorithme détaillé est ensuite présenté en section 4. Les valeurs obtenues des diverses sources sont transformées en chaînes de caractères (en les séparant par des barres obliques). Ce sont ces chaînes qui sont ensuite condensées par MD5. Un exemple complet figure dans la section 5. Notez la précision avec laquelle les sources sont définies !

Une mise en œuvre de référence figure en section 7. Je l'ai rassemblée dans une archive tar (en ligne sur `https://www.bortzmeyer.org/files/rfc2777.tar`). On la compile ainsi :

```
% make
cc -c -o rfc2777.o rfc2777.c
cc -c -o MD5.o MD5.c
cc -lm -o rfc2777 rfc2777.o MD5.o
rfc2777.o: In function 'main':
rfc2777.c:(.text+0x16a): warning: the 'gets' function is dangerous and should not be used.
```

et on l'utilise ainsi (sur l'exemple de la section 5) :

```
% ./rfc2777
Type size of pool:
(or 'exit' to exit) 25
Type number of items to be selected:
(or 'exit' to exit) 10
Approximately 21.7 bits of entropy needed.

Type #1 randomness or 'end' followed by new line.
Up to 16 integers or the word 'float' followed by up
to 16 x.y format reals.
9 18 26 34 41 45
9 18 26 34 41 45
Type #2 randomness or 'end' followed by new line.
Up to 16 integers or the word 'float' followed by up
to 16 x.y format reals.
2 5 12 8 10
2 5 8 10 12
Type #3 randomness or 'end' followed by new line.
Up to 16 integers or the word 'float' followed by up
to 16 x.y format reals.
9319
9319
Type #4 randomness or 'end' followed by new line.
Up to 16 integers or the word 'float' followed by up
to 16 x.y format reals.
float 13.6875
13.6875

Type #5 randomness or 'end' followed by new line.
Up to 16 integers or the word 'float' followed by up
to 16 x.y format reals.
end
Key is:
 9.18.26.34.41.45./2.5.8.10.12./9319./13.6875/
index        hex value of MD5        div  selected
 1  746612D0A75D2A2A39C0A957CF825F8D  25  -> 12 <-
 2  95E31A4429ED5AAF7377A15A8E10CD9D  24  ->  6 <-
 3  AFB2B3FD30E82AD6DC35B4D2F1CFC77A  23  ->  8 <-
 4  06821016C2A2EA14A6452F4A769ED1CC  22  ->  3 <-
 5  94DA30E11CA7F9D05C66D0FD3C75D6F7  21  ->  2 <-
 6  2FAE3964D5B1DEDD33FDA80F4B8EF45E  20  -> 24 <-
 7  F1E7AB6753A773EFE46393515FDA8AF8  19  -> 11 <-
 8  700B81738E07DECB4470879BEC6E0286  18  -> 19 <-
 9  1F23F8F8F8E5638A29D332BC418E0689  17  -> 15 <-
10  61A789BA86BF412B550A5A05E821E0ED  16  -> 22 <-

Done, type any character to exit.
```

On note l'affichage de la clé (les valeurs séparées par des barres obliques) et celui du nombre de bits d'entropie nécessaires.

Outre les exemples de sélection du Nomcom qu'on trouve dans le RFC, cette méthode a été utilisé pour sélectionner le préfixe `xn--` des IDN. Voici la description de la méthode <http://www.ietf.org/mail-archive/web-old/ietf-announce-old/current/msg22384.html> et l'annonce du résultat <http://www.ietf.org/mail-archive/web-old/ietf-announce-old/current/msg22565.html>.

Un autre exemple intéressant sera l'utilisation de ce RFC 2777 pour sélectionner les /8 <http://blog.icann.org/2009/09/selecting-which-8-to-allocate-to-an-rir/> à donner aux RIR lors de l'épuisement final des adresses IPv4 <https://www.bortzmeyer.org/epuisement-adresses-ipv4.html>.