Autour des réductions de recherche à décision dans le cas des codes structurés

Maxime Bombar

Dans beaucoup de cryptosystèmes à base de code, la sécurité repose sur le problème de décoder un code aléatoire. On peut en considérer deux versions:

- La version *décisionnelle*, qui demande de distinguer entre un mot de code bruité par une petite erreur, et un élément aléatoire de l'espace ambiant.
- La version *recherche*, qui demande de décoder un mot de code bruité.

A priori, la version recherche est plus difficile que la version décisionnelle. Or, c'est bien souvent sur cette dernière que repose la sécurité des systèmes cryptographiques. On peut notamment citer le système d'Alekhnovich (2003). Il est donc intéressant de chercher une réduction dans l'autre sens *i.e.* étant donné un distingueur entre un mot bruité et un élément aléatoire, est-il possible d'obtenir un véritable décodeur fonctionnant en temps équivalent ? C'est ce qu'on appelle une réduction de *recherche à décision*.

En 1996, Fisher et Stern ont prouvé qu'une telle réduction existait bien dans le cas de codes aléatoires. Leur preuve repose sur l'utilisation d'un théorème de Goldreich et Levin. Cependant, dans le but de réduire la taille des clés, il a été proposé d'utiliser des codes plus structurés (en particulier des codes quasi-cycliques) mais la réduction de Fisher et Stern ne s'étend pas à ce cas là. Par exemple, le cryptosystème HQC (*Hamming Quasi-Cyclic*) qui est un des finalistes alternatifs du 3ème tour de la compétition du NIST repose sur cette version structurée, et le NIST a encouragé à investiguer les relations entre le problème de décision et la variante recherche.

En revanche, dans le monde des réseaux euclidiens, de telles réductions ont été trouvées au cours de la dernière décénie pour des problèmes liés aux réseaux structurés. On pourra en particulier citer les problèmes *Learning With Errors over Rings* et *Learning With Errors over Modules*.

Dans cet exposé je vais introduire un nouveau problème adapté du monde des réseaux euclidens: *Learning With Errors in Function Fields* (nom temporaire), qui s'avère offrir un cadre plus agréable pour faire des réductions, et je vais apporter une réponse positive dans le cas de certains codes structurés.

Il s'agit d'un travail en commun avec Alain Couvreur et Thomas Debris–Alazard.