Generic Algorithms for Subset Sum

December 23, 2010

The Subset-Sum Problem. The subset sum problem is a famous NP-hard problem which has often been used in the construction of cryptosystems. An instance of this problem consists of a list of n positive integers $(a_1, a_2, \ldots, a_n)$ and an integer S. The solution is given by $(\epsilon_1, \ldots, \epsilon_n) \in \{0,1\}^n$ such that

$$\sum_{i=1}^{n} \epsilon_i a_i = S$$

That is, one must find a subset of the a_i's which sums to S.

The density of the problem is defined as

$$d = \frac{n}{\log(\max_i a_i)}$$

There exist efficient lattice-based algorithms for the problem if the density is either low, $d < 0.94$, or high, $d > 1$. However, for the case where d is close to 1, until recently the best algorithm run in time $O(n2^{n/2})$ using $O(n2^{n/4})$ bits of memory. This algorithm, by Richard Schroeppel and Adi Shamir, dates back to 1979.

Very recently Howgrave-Graham Joux [1] gave a new algorithm improving the running time to $O(2^{0.3113n})$.[1]

Goal. In this project the student will read and report on [1] and maybe some of the related literature. A possible more challenging topic is the following: the algorithm from [1] is only shown to work for "most" instances of the problem (i.e. it's not a worst case algorithm.), give a nice classification of the "bad" instances and an explicit bound on their density.

Supervision. Krzysztof Pietrzak, pietrzak@cwi.nl

References

[1] Nick Howgrave-Graham and Antoine Joux: New Generic Algorithms for Hard Knapsacks. In *EUROCRYPT*, 2010.

[1] A nice exposition of this algorithm is given on Lipton's blog
`http://rjlipton.wordpress.com/2010/02/05/a-2010-algorithm-for-the-knapsack-problem/`