Implementation aspects of information set decoding algorithms

Charles MEYER-HILFIGER
Joint work with: Nicolas SENDRIER

The security of code-based cryptosystems such as McEliece's [11, 2] rest upon the difficulty to distinguish some structured linear code from a random one and from the difficulty of the binary syndrome decoding problem, which consists essentially in finding one solution of the equation $He = s$ for $H \in \mathbb{F}_2^{k \times n}$ with a constraint on the number of ones in the solution vector e. Thus, showing that these problems are hard in average is fundamental to prove that the McEliece cryptosystem is secure in practice. We will focus here on the syndrome decoding problem in the generic case. This problem has been widely studied for many years, is NP-Complete and supposed hard in average, that is, there exist no subexponential generic algorithm to solve it. Among the algorithm used to solve it, the most efficient are information set decoding algorithms. Among them we can cite Prange algorithm [12], Lee-Brickel [8], Stern [14], Dumer [4, 5], and, more recently MMT [9], BJMM [1] and May-Ozerov [10] and its variants [3]. While these algorithm have been studied asymptotically, [6] is the only non asymptotic study we know of of Dumer, MMT and BJMM algorithms.

Each newer algorithm [14, 4, 5, 9, 1] is better asymptotically but comes with a growing polynomial factor. As a result, which algorithm is the best in practice is not clear. Furthermore, the bottleneck asymptotically in all these algorithms is to find the intersection of 2 sets of exponential sizes. Thus when implementing these algorithms two questions arises:

- For cryptographic instances which algorithm is better?
- How to correctly implement the intersection of 2 sets?

When implementing the intersection of 2 sets, two major approaches exists: 1) generating and storing the first data structure and enumerating the second one to search on the fly if each element is present in the first data structure 2) generating and storing the two data structures explicitly and merging them to find their intersection.

A natural approach for 1) is to use hash tables and for 2) to use sorted lists. We used a static array as a hash table and a LSB-radix sorting algorithm to sort our lists.

We implemented Dumer and MMT algorithms in a generic way in C++ and benchmarked them using these two approaches. We used the Shamir-Schroeppel [13, 7] technique when implementing MMT as described in [6] to gain a memory factor.

We show that in Dumer and MMT, 1) and 2) are comparable in performance with a slight advantage for 1) in Dumer and a slight advantage for 2) in MMT. We also show that MMT algorithm becomes practically better than Dumer algorithm starting from small code length challenges [1] (when decoding an error of weight equal to the Gilbert-Varshamov bound).

We try to explain these results considering the theoretical optimal parameters and the practical bests values of these parameters. More precisely, for instances of cryptographic interest, Dumer practical optimal parameters are such that the data structures considered fits in the last level of cache of our processor, resulting in very few cache misses when using hash tables. Whereas in MMT algorithm, the practical optimal parameters are such that the data structure won't fit in the cache at a point, resulting in performance-killing cache misses when using hash tables.

[1] decodingchallenge.org

References

[1] Anja Becker et al. "Decoding Random Binary Linear Codes in $2^{n/20}$: How $1 + 1 = 0$ Improves Information Set Decoding". In: *Advances in Cryptology - EUROCRYPT 2012*. LNCS. Springer, 2012.

[2] Daniel J. Bernstein et al. *Classic McEliece: conservative code-based cryptography*. `https://classic.mceliece.org`. Second round submission to the NIST post-quantum cryptography call. Mar. 2019.

[3] Leif Both and Alexander May. "Decoding Linear Codes with High Error Rate and Its Impact for LPN Security". In: *Post-Quantum Cryptography 2018*. Ed. by Tanja Lange and Rainer Steinwandt. Vol. 10786. LNCS. Fort Lauderdale, FL, USA: Springer, Apr. 2018, pp. 25–46. DOI: `10.1007/978-3-319-79063-3{\_}2`. URL: `https://doi.org/10.1007/978-3-319-79063-3%7B%5C_%7D2`.

[4] Il'ya Dumer. "Two decoding algorithms for linear codes". In: *Probl. Inf. Transm.* 25.1 (1989), pp. 17–23.

[5] Matthieu Finiasz and Nicolas Sendrier. "Security Bounds for the Design of Code-based Cryptosystems". In: *Advances in Cryptology - ASIACRYPT 2009*. Ed. by M. Matsui. Vol. 5912. LNCS. Springer, 2009, pp. 88–105.

[6] Yann Hamdaoui and Nicolas Sendrier. *A Non Asymptotic Analysis of Information Set Decoding*. IACR Cryptology ePrint Archive, Report2013/162. `http://eprint.iacr.org/2013/162`. 2013.

[7] Nicholas Howgrave-Graham and Antoine Joux. "New generic algorithms for hard knapsacks". In: *Advances in Cryptology - EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. LNCS. Sringer, 2010.

[8] Pil J. Lee and Ernest F. Brickell. "An Observation on the Security of McEliece's Public-Key Cryptosystem". In: *Advances in Cryptology - EUROCRYPT'88*. Vol. 330. LNCS. Springer, 1988, pp. 275–280.

[9] Alexander May, Alexander Meurer, and Enrico Thomae. "Decoding random linear codes in $O(2^{0.054n})$". In: *Advances in Cryptology - ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. LNCS. Springer, 2011, pp. 107–124.

[10] Alexander May and Ilya Ozerov. "On Computing Nearest Neighbors with Applications to Decoding of Binary Linear Codes". In: *Advances in Cryptology - EUROCRYPT 2015*. Ed. by E. Oswald and M. Fischlin. Vol. 9056. LNCS. Springer, 2015, pp. 203–228.

[11] Robert J. McEliece. "A Public-Key System Based on Algebraic Coding Theory". In: DSN Progress Report 44. Jet Propulsion Lab, 1978, pp. 114–116.

[12] Eugene Prange. "The use of information sets in decoding cyclic codes". In: *IRE Transactions on Information Theory* 8.5 (1962), pp. 5–9. DOI: `10.1109/TIT.1962.1057777`. URL: `http://dx.doi.org/10.1109/TIT.1962.1057777`.

[13] Richard Schroeppel and Adi Shamir. "A $T = O(2^{n/2})$, $S = O(2^{n/4})$ Algorithm for Certain NP-Complete Problems". In: *SIAM J. Comput.* 10.3 (1981), pp. 456–464. DOI: `10.1137/0210033`. URL: `http://dx.doi.org/10.1137/0210033`.

[14] Jacques Stern. "A method for finding codewords of small weight". In: *Coding Theory and Applications*. Ed. by G. D. Cohen and J. Wolfmann. Vol. 388. LNCS. Springer, 1988, pp. 106–113.