Selected Areas in Cryptology - Part 1: Post-quantum cryptography

Exercise sheet 3, 25 February 2024

1. Write the (one pass) identification protocol using MCE as a hardness assumption (described in slides 32-34) in more detail. Specifically, you need to write (on paper, in Magma or in SageMath) the following algorithms

   (a) Commitment
   Alice's data: $(\mathbf{C}_0^{(1)}, \ldots, \mathbf{C}_0^{(k)}), (\mathbf{C}_1^{(1)}, \ldots, \mathbf{C}_1^{(k)}), \mathbf{A}, \mathbf{B}, \mathbf{T}$
   Input: none
   Output: $(\tilde{\mathbf{C}}_0^{(1)}, \ldots, \tilde{\mathbf{C}}_0^{(k)})$

   (b) Response
   Alice's data: $(\mathbf{C}_0^{(1)}, \ldots, \mathbf{C}_0^{(k)}), (\mathbf{C}_1^{(1)}, \ldots, \mathbf{C}_1^{(k)}), \mathbf{A}, \mathbf{B}, \mathbf{T}$, $(\tilde{\mathbf{C}}_0^{(1)}, \ldots, \tilde{\mathbf{C}}_0^{(k)})$, $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{T}}$
   Input: $b \in \{0, 1\}$
   Output: $\mathbf{A}_{\text{resp}}, \mathbf{B}_{\text{resp}}, \mathbf{T}_{\text{resp}}$.

   Hint: To sample a random element from $\mathrm{GL}_m(q)$ write

   (a) $\mathbf{A} \xleftarrow{\$} \mathrm{GL}_m(q)$ on paper ;

   (b) `A := Random(GL(m, FiniteField(q)));` in Magma ;

   (c) `A = GL(m, GF(q)).random_element()` in SageMath ;

2. What is the size of the public key and the signature in a Fiat-Shamir digital signature scheme using MCE as a hardness assumption, with parameters $q = 5$ (one byte), $m = n = k = 13$, and Fiat-Shamir security parameter $\lambda = 64$ (**Recall:** this means that the soundness error of the protocol should be 2^{-64}) ?

3. We modify the signature scheme from Exercise 2 to use the soundness amplification technique: we have $s = 256$ end codes in the public key.

   (a) How many iterations of the underlying identification protocol do we need to do, to get to a soundness error of 2^{-64}?

(b) What is the size of the public key and the signature in this case?

(c) **Bonus:** Comparing the answer to b) in this exercise to the answer to Exercise 2, we observe that, with the soundness amplification technique, we obtain smaller signatures at the expense of increasing the size of the public key. Can you find a value for the parameter s where the public key and the signature are balanced (their sizes are closest to each other)?

Hint: Once you understand how we calculate the sizes for a fixed s, you can either (1) write this in a script and loop through increasing values of s until it hits the smallest difference in the sizes, or (2) do the computation on paper by using a dichotomy approach: try for some value of s between 1 and 256; if the signature is too big, try for a (is it *bigger* or *smaller*?) value for s and vice versa if the public key is too big.

4. Write down the equations for the algebraic attack on MCE described in the lecture (the "better" modelisation).

   (a) What is the degree of the system and what is the number of equations and variables?

5. **Bonus:** We consider the combinatorial attack on MCE with parameters $n = m = k$ over $\mathbb{F}_q$. We have chosen a target rank r, and we know that the probability that a codeword is of this rank r is $q^{-\frac{k}{3}}$.

   (a) What is the complexity of the first part of the combinatorial attack, corresponding to the algorithm in SampleSet? **Hint:** Answer the following questions:

      i. How many times do we *expect* to pick a random codeword before we find a codeword of rank r?

      ii. There exist (approximately) how many codewords of rank r in total?

      iii. How many codewords of rank r do we need to find, aka. how big is the list that we need to build?

   Since the goal of this part of the attack is to find *enough* (as per the birthday paradox) elements of rank r, the complexity is given by the number of times we need to draw a random codeword before we fill the list with the required number of codewords.

(b) What is the complexity of the second part of the attack (corresponding to the iteration of the **for all** loop in the algorithm in CollisionFind)? To answer this question, we will denote by C_{FF} the cost of one call to the FindFunction, aka the cost of solving the corresponding system of equations. Hence, if we find that for the attack we need to call the FindFunction α times, then we conclude that the complexity is $\mathcal{O}(\alpha\, C_{FF})$.

Hint to check your answer: With the target rank r chosen here, you should obtain the **same** complexity for both parts of the algorithm.

6. Why is the MCE problem without a change of basis easy? **Hint:** Think about what the main goal is in the combinatorial attack.