Mathemathics with a Computer
intermediate report

Dmitar Zvonimir Mitev

5 April 2024

1 Introduction

As part of the course "Mathematics with a Computer", I am implementing the functional encryption (FE) scheme for computing inner products by Abdalla et. al. that is based on the decisional Diffie-Hellman (DDH) assumption. All the definitions concerning FE and the scheme itself can be found on [Abd+15]. My goal is to implement all four algorithms from the scheme: Setup, Encrypt, KeyDer and Decrypt. The programming language of choice is C++. It was chosen because of its speed and the possibility of using some more sophisticated external libraries that are not available for C.

2 My work so far

Since in cryptography we are dealing with very big integers and C++ has a fixed-precision arithmetic, I am forced to work with an external library that offers arithmetic for large numbers. There are multiple libraries that accomplish that and out of them, I chose Crypto++ [Cry]. It supports multiple-precision arithmetic and it has a lot of in-built functions. Using it I have written a function that returns a large cryptographically secure random integer which can be found in `random_crypto_secure.cpp` and by that began implementing the Setup algorithm. For the algebraic group in which the operations will take place I chose the group of quadratic residues modulo a safe prime. This group is interesting since it is believed that the DDH assumption holds in this group, hence theoretically speaking, the scheme can be shown to be secure. For proper usage of the scheme I need a function that returns a generator of the aforementioned group. I implemented the function `generatorOfQuadraticResidues` that does exactly that in `scheme.cpp`.

3 Problems so far

The main problems thus far are concerned with C++ and with the Crypto++ library. As a programming language C++ is statically typed, which means

that each time I want to return an additional parameter from a function I have to change a large portion of the function itself (in comparison to Python for example, in which this is trivial). On the other hand, the documentation of the Crypto++ library is pretty limited, as it includes no examples of usage. Hence I am learning almost everything by trial-and-error.

4 Future plans

As mentioned in the beginning, the end-goal is to have a cryptographically secure implementation of the scheme with all four algorithms working properly. Since in the deciphering process one needs to compute a discrete logarithm, the plan is to implement the Baby-step giant-step algorithm which computes the desired discrete logarithm in $O(\sqrt{n})$ time, where n is the number of the elements in the group as opposed to a simple brute force that requires $O(n)$ time.

After implementing everything, I plan on performing tests and determining how large the numbers may be so that the scheme still returns a result in a reasonable time. Additionally, I plan on measuring the time needed for execution of the algorithms. I plan on achieving these two goals by setting different upper bounds for the elements and different dimensions of the plaintext vector (parameters B and ℓ in [Abd+15], respectively) and by computing the average time the algorithms need for execution. In the end, the Decrypt algorithm for parameters $B = 10\,000$ and $\ell = 100$ should take (much) less than 34 seconds.

References

[Abd+15] M. Abdalla et al. "Simple functional encryption scheme for inner products". In: *Public-Key Cryptography – PKC 2015*. Springer. 2015, pp. 733–751.

[Cry] Crypto++ library. `https://www.cryptopp.com/`. Accessed: 2024-05-04.