Summer Project Report - 2022

Implementation of Number Theoretic Transform for AVR Processors

Anmol Shetty

Supervised by
Dr Sasirekha, Prof Madhav Rao

Contents

1 Introduction 3

2 Objectives and Goals 3

3 Results 3

4 Skills learnt through the Project 5

5 Future Work 5

6 Conclusion 5

1 Introduction

This report aims to address the implementation of Number Theoretic Transform on the Arduino Uno for the purpose of building the Polynomial Multiplication module which multiplies two vectors in an efficient manner. Fast implementations of the homomorphic encryption schemes heavily depend on efficient polynomial arithmetic, multiplication of very large degree polynomials over polynomial rings. Number theoretic transform (NTT) accelerates large polynomial multiplication significantly, and therefore, it is the core arithmetic operation in the majority of homomorphic encryption scheme implementations.

2 Objectives and Goals

The objective of this project is to have a hardware implementation of Polynomial Multiplication using a convolution method known as Number Theoretic Transform which are used in Ring Learning with Errors encryption schemes. These schemes are proven to be extremely useful in protecting confidentiality and integrity of billions of connected devices. This module can be further used in Ring LWE Public Key Cryptosystems to add a layer of protection to the processed data. The papers referenced below give us the efficient implementation of the algorithm [1, 2].

3 Results

The outputs have been tested with a Python equivalent of NTT and Inverse NTT with our C code at polynomial sizes of $n = 1024$ and the modulus was set at $q = 12289$. These root values to perform the NTT have been pre-computed for faster computation with the Arduino implementation yielding a computation time of $122ms$ at array size of $n = 64$.

Observation Table		
Size	CPU Time (ms)	Arduino Time (ms)
1	0.82	1
2	0.89	1
4	0.94	4
8	2.67	9
16	2.69	20
32	2.72	49
64	2.81	122

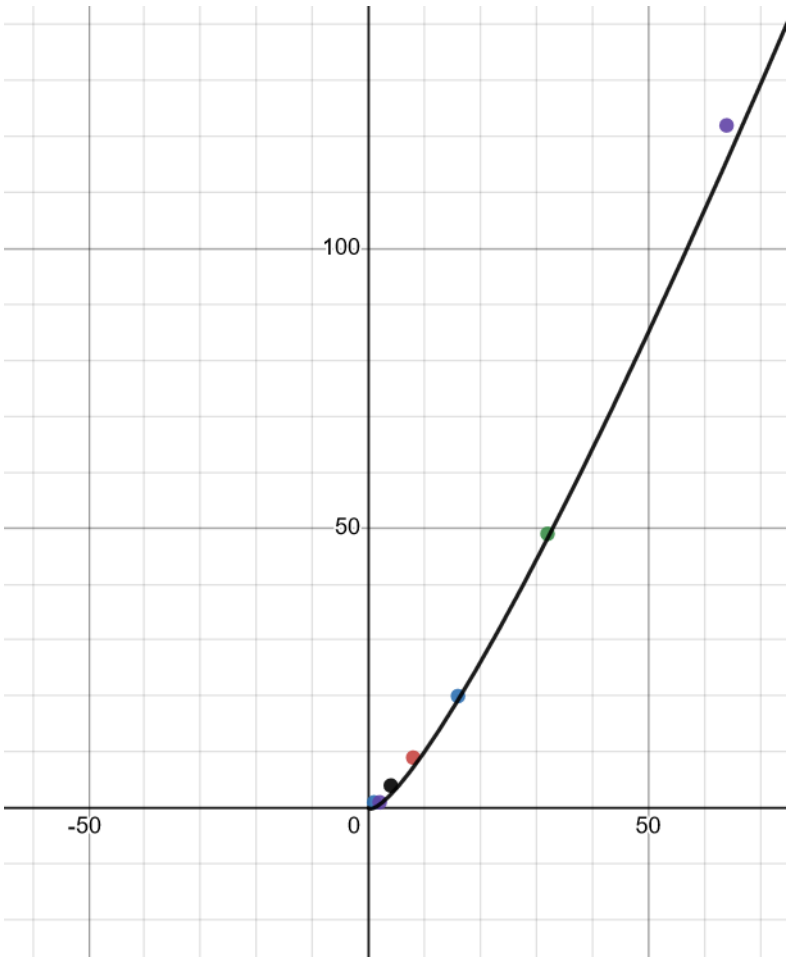

Figure 1: Interpolation of Arduino Time with $f(x) = nlog(n)$

From [1] we see that it indeed satisfies the clause that $f(x) = nlog(n)$ where n is the polynomial

degree. Note: The CPU time has been calculated with 1000 iterations of NTT and Inverse NTT. The average might vary over successful iterations due to the array elements being random in nature.

4 Skills learnt through the Project

- Arduino or any other AVR processors have limited amount of memory. The project emphasizes on the efficient usage of memory to save time. In the Arduino implementation, the root values were pre-computed which cut down the programs' run time.
- Learn about the NTT which is a faster albeit limited version of the DFT(Discrete Fourier Transform) and it's applications in the world of cryptography.
- Learnt about Ring Learning with Errors and how Gaussian Noise can be used to encrypt data along with Homomorphic encryption wherein operations could be performed on encypted data.

5 Future Work

The main challenge faced during this work was to find ways to increase the speed and to allow higher powers of inputs such as $n = 128$ or $n = 256$. Since my algorithm relied on the Iterative version of the NTT, Arduino could only handle array sizes of upto $n = 64$ before returning garbage values for higher values of n. Further work would improve upon this limitation and also possibly allow for higher values of q which currently stands at 12289, which is a prime number that satisfies $k \times 2^m + 1$.

6 Conclusion

We see that NTT has many unique and interesting applications in the world of Public Key Cryptosystems and how it could be used in the wider framework of the post-quantum security systems. The Arduino Uno which uses the AtMega328 AVR Processor is used in this effort to provide for efficient multiplication of two polynomials. The computation times were almost in line with the predicted values.

Future extensions to the project may involve finding hardware specific NTT implementations

optimized for the limited space in the AVR processor. This move could drastically improve the capacity of the multiplier which currently stands at $n = 64$.

References

[1] Z. Liu, H. Seo, S. S. Roy, J. Großschädl, H. Kim, and I. Verbauwhede, "Efficient ring-lwe encryption on 8-bit avr processors," Cryptology ePrint Archive, Paper 2015/410, 2015. [Online]. Available: https://eprint.iacr.org/2015/410

[2] E. C. M. A. Özerk, Ö., "Efficient number theoretic transform implementation on gpu for homomorphic encryption," 2022. [Online]. Available: https://doi.org/10.1007/s11227-021-03980-5