A FULLY HOMOMORPHIC ENCRYPTION SCHEME WITH HIGH-PERFORMANCE POLYNOMIAL MULTIPLIERS

[1]B. Soumya Sree, [2]M. Devadas

[1,2]Dept. of Electronics and Communication Engineering

[1,2]Vaagdevi College of Engineering, Warangal, Telangana, India

ABSTRACT:

Completely Homomorphism Encryption is a method that permits calculations on scrambled information without the requirement for decoding, and it gives protection in different applications, for example, security safeguarding distributed computing. In this article, we present two equipment models streamlined for speeding up the encryption and decoding activities of the Fan-Vercauteren homomorphic encryption plot with elite execution polynomial multipliers.

We propose two streamlined equipment models for speeding up the encryption and decoding activities of the Fan-Vercauteren homomorphic encryption plot using elite execution polynomial multipliers in this article.

Our models will be used in a product code sign gas pedal system, in which the encryption and unscrambling chores will be offloaded to an FPGA device, while the remaining functions in the BFV plot will be done in programs running on a standard personal computer. Our gas pedal technology, in particular, has been modified to speed up the Straightforward Encoded Number juggling Library, which was produced by the Cryptography Exploration Gathering at Microsoft Exploration.

The equipment elements of the suggested system center around the XILINX VIRTEX7 FPGA gadget, which chats with its product portion by use of a fringe part interconnect express connection.

We will carry out our ideas independently for plaintext and figure text, concentrating on 1024-degree equations with 8-cycle and 32-digit coefficients. When compared to their unadulterated programming executions, the suggested structure achieves roughly 12 and 7 percent speedups, respectively, including I/O operations for the offloaded encrypting and decoding workloads.

Index Terms—Fan-Vercauteren (FV), FPGA, hardware, number theoretic transform, Simple Encrypted , arithmetic Library.

1. INTRODUCTION

Completely Homomorphic Encryption is the name given to any encryption scheme that allows number-crunching and consistent computations to be performed directly on the text. This characteristic deals with the secure management of sensitive data, which is a critical and as of now unfulfilled interest in distributed computing applications. Since its most memorable presentation, the possibility of homomorphism encryption has received wide consideration in the writing, and various homomorphism encryption plans have been presented. Although theoretically sound, homomorphism encryption plans are not exactly fit to be sent for practical applications due to execution limitations of PC structures.

Applications in view of current homomorphism encryption plans, which require productive executions of computationally costly numerical tasks, can be significant degrees more slow than customary programming applications that work on plain text information. for programming executions of homomorphism encryption , single center and multi center central processor exhibitions are basic. For single-center execution, recurrence of the processor straightforwardly influences execution, which can't be expanded considerably with contemporary innovation any further. Likewise, since central processor is expected to give great execution to a different arrangement of utilizations, equipment or potentially design enhancements for a computer chip focusing on just Homomorphism Encryption applications are not possible. Computer chip makers increment the exhibition of a processor with a multi center methodology. In any case, the quantity of centers that can be remembered for a multi center design is restricted because of costly single-center executions.

While Single Center Execution Of A Universally Useful Computer Processor Targets Consecutive Calculations, Multi Center Models Are More Reasonable For Equal Calculations. Most Homomorphism Encryption Plans Include A Blend Of Inherently Sequential And Exceptionally Parallelizable Calculations That Will At Last Perform Best On Heterogeneous Models Which Alludes To The Utilization Of Various Handling Centers To Expand Execution. In This Article, We Propose Such A Heterogeneous Gas Pedal Structure Highlighting A FPGA Center And A Computer Chip To Work On The Presentation Of Homomorphism Encryption Plans On A Framework Level.

Traditional crypto framework, for example, High level Encryption Standard doesn't have homomorphic property that permits number-crunching calculations to be performed Straight forwardly on figure text without decoding it. Then again, homomorphic encryption plans, for example, permit homomorphic tasks straightforwardly on the scrambled information and in this manner empower security protecting handling of data, particularly with regards to distributed computing where by security is a squeezing concern.

Also, HE conspires are evidently sluggish; thus, the case for speed increase is a lot more grounded than customary cryptosystems. With an always expanding interest for security in distributed computing applications, speed increase of homomorphic encryption plans is now a significant examination region.

The GPU execution is contrasted and the SEAL execution. The SEAL group as of late reported exceptionally productive Seal PIR, which is a confidential data recovery device that permits a client to download a component from an information base put away by a far off server without uncovering which component is downloaded . Our gas pedal system, while offloading exceptionally parallelizable encryption and unscrambling activities completely on the FPGA center, leaves the remainder of tasks of SEAL unblemished in programming.

By Conveying Our Edge Work, Any Cloud Design Using Seal For Homomorphism Encryption Applications Can Work On Its Exhibition By Using A Fpga Gadget Close To The Computer Processor,

Without Carrying Out The Whole Homomorphism Encryption Library In The FPGA. Our commitment in this article is recorded as follows.

1) We examine the iterative and the four-step Cooley Calculations for number hypothetical change activity and plan two novel, exceptionally parallelized equipment structures in light of these calculations.

We assess the consequences of our FPGA executions of the two equipment models concerning time and region and look at them against comparative works in the writing. We show that our equipment structures involving the original secluded multiplier calculation for any NTT-accommodating prime modulus, presented in our fundamental work give tantamount time execution to those involving exceptional primes in the writing.

We propose a gas pedal system, including a superior presentation FPGA gadget, associated with a host central processor. The structure interfaces the computer processor and the FPGA by means of a quick fringe part interconnect express association, accomplishing a 32-Gb/half-duplex I/O speed. The structure is utilized to speed up encryption and decoding tasks of SEAL. Each time an encode or decode capability is summoned via SEAL, the calculation is offloaded to the FPGA gadget through the PCIe association.

Our plan uses a development focusing on 128-digit security level. Counting the time spent on I/O,the latencies of the offloaded encryption and decoding tasks are worked on by 12× and 7×,respectively, contrasted with their unadulterated programming executions on SEAL running on an Inteli97900X computer processor. As the structure gives a straightforward Despite the fact that the proposed unscrambling tasks of the BFV homomorphic encryption plot in our accommodation, it tends to be productively utilized to speed up different tasks in homomorphic applications.

To be sure, one of the main commitments of our work is to propose a superior presentation polynomial multiplier that can speed up the duplication of two exceptionally enormous degree polynomials, which is the computational container neck of all homomorphic tasks in the BFV plot or different plans.

Cryptography is the area of constructing crypto graphic systems. Cryptology consists of two branches:

Cryptanalysis is the area of breaking cryptographic systems.

Cryptography is a field of computer science and mathematics that focuses on techniques for secure communication between two parties(Alice & Bob) while a third- party (Eve1or Mallory2) is present (see Figure 1.1). This is based on methods like encryption, decryption, signing, generating of pseudo random numbers, etc.

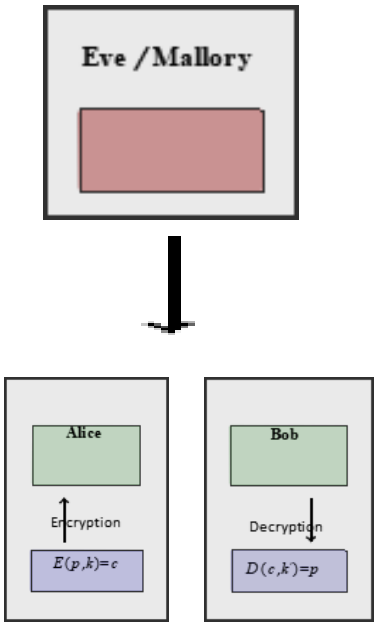


Figure1.1: A Basic Idea for Secure Communication

1. The four ground standards of cryptography are

2. Classification Characterizes as decides that cutoff points access or includes limitation certain data.

3. Information Honesty Deals with the consistency and exactness of information during its whole life-cycle.

Confirmation Affirms the reality of a property of a datum that is professed to be valid by some element.

4. Non-Disavowal Guarantees the powerlessness of a creator of an assertion resp. a snippet of data to deny it.

5. Presently a days there are in everyday two unique plans: From one perspective ,there are symmetric plans, where both, Alic eand Weave, need to have a similar key to scramble their correspondence.

For this, they need to at first safely trade the key. Then again, since Diffie and Hellman's key trade thought from 1976 (see likewise Alice and Sway both have a private and a public key.The public key can be imparted to anybody, so Weave can utilize it to encode a directive for Alice. In any case, just Alice, with the relating private key, can unscramble the scrambled message from Sway

There is the security of the structure itself, based on mathematics.

There is a normalization cycle for cryptosystems in light of hypothetical examination in math and intricacy hypothesis.

Then we have the execution of the designs in gadgets, for example SSL, TLS in your internet browser or GPG for marked resp. scrambled messages.

These executions shouldn't veer from the hypothetical norms.

MODES OF CIPHERS

For ciphers we have, in general, four different categories:

1. Symmetric and asymmetric ciphers (see Definition2.20),and

2. Stream and block ciphers.

In the following we often assume binary representation of symbols, i.e. we are working with bits in Z/2Z. All of what we are doing can be easily generalized to other representations and other alphabets.

2. HOMOMORPHIC ENCRYPTION

2.1 BLOCK CIPHERS

It follows that the number of permutations of the form DESk2◦DESk1 is much larger than the number of permutations of type DESK.

Until now we always considered that our plaintexts have the same size as the key. Clearly, in general, one wants to encrypt longer documents or texts. For this problem there are several different modes one can apply block ciphers.

2.2 MODES OF BLOCK CIPHERS

Let us assume in this section that Σ = Z/2Z, block size is n @>0 and the key spaces§ § ′=are the same. We switch between representations of plain texts: For example let n = 3, then we can identify all natural numbers between 0 and 7.So we can represent 0 binary as 000 or (0,0, 0) (Z/2Z)3,or 5 as 101or (1, 0,1).

We further assume that there is some magic that randomly resp. pseudo and only and uniformly distributed chooses a key.

Assume we have a plain text p of arbitrary but finite length. We divide p into blocks of length n. If the length of p is not divisible by #n then we add some random symbol sat the end of p. In the end we receive a representation p = (p1, . . . , pm) where all pi are plaintext blocks of length n. Each plain text block pi is encrypted to a corresponding cipher text block ci using a given key k.

There are two main categories of ciphers in terms of key handling: If κ is feasible then§ and need to be kept secret and the cipher is called § symmetric. Otherwise the cipher is called asymmetric.

We also call a cryptosystem symmetric resp. asymmetric if its corresponding cipher is symmetric resp. asymmetric. blocks of length n. If the length of p is not divisible by #n then we add some random symbol sat the end of p. In the end we receive a representation p = (p1, . . . , pm) where all pi are plaintext blocks of length n. Each plain text block pi is encrypted to a corresponding cipher An asymmetric cryptosystem is also called a public key cryptosystem as § can be made public without weakening the secrecy of the “private” key set § for keys.

This can be an arbitrary counter, usually one uses an increment counter: So for each block of the plain text or cipher text the counter is incremented by There are now various possible ways to combine then once with the counter.

Homomorphic Encryption is a leading edge new innovation which can empower private distributed storage and calculation arrangements, and numerous applications were portrayed in the writing over the most recent couple of years. However, before Homomorphic Encryption can be taken on in clinical, wellbeing, and monetary areas to safeguard information and patient and buyer security, it should be normalized, no doubt by different normalization bodies and government offices. A significant piece of normalization is expansive settlement on security levels for shifting boundary sets. Albeit broad examination and seat stamping has been finished in the exploration local area to lay out the establishments for this work, it is elusive all the data in a single spot, alongside substantial boundary suggestions for applications and sending.

This study is an attempt to capture (at least a portion of) the aggregate knowledge about the currently known state of security of these plans, to determine the plans, and to prescribe a wide range of limits to be used for

homomorphic encryption at various security levels. To produce these border proposals, we depict known attacks and their estimated running times. We also highlight additional features of these encryption plans that make them useful in a variety of applications and situations.

It is typical that future work outdate and expand this Homomorphic Encryption Standard will employ the accompanying numbering show:

We show a few viewpoints in the supplement that are not determined in this report and are expected to be covered by future records.

2.3 SECTION PROPERTIES:

Semantic Security or IND CPA Security: A homomorphic encryption scheme is thought to be secure if no adversary has a benefit in speculating (with a better than half chance) whether a given code message is an encryption of both of two similarly reasonable specific messages. This assumes that encryption will be randomized so that two distinct encryptions of a similar message do not appear to be identical.

Assume a client computes the key tuple by running the boundary and key-age calculations. A adversary is anticipated to have the limits, the assessment key EK, a public key PK(only in the public-key plan) and can receive encryptions of communications of its choice.

The enemy is then provided an encryption of one of two messages of its choice, registered by the aforesaid encryption computation, without recognizing which message the encryption.

The security of HE then ensures that the adversary cannot figure out which message the encryption compares to with critical benefit better than a half chance. This captures the fact that no information about the messages is revealed in the code message.

Minimization: The conservativeness property of a homomorphic encryption plot assures that homomorphic process on the code texts don't grow the length of the code texts. That is, any evaluator can play out an erratic upheld rundown of assessment capability calls and get a code text in the code text space (that doesn't rely upon the intricacy of the assessed capabilities).

Successful decoding: Profitable unscrambling attribute implies that the homomorphic encrypting plot typically ensures that the decoding runtime doesn't rely upon the capability which was assessed on the code texts

2.4 KEY ASSESSMENT

Assume a server has a corpus of code texts scrambled under a mystery key SK, and the client who claims SK is aware that SK may have been compromised.

It is appealing for an encryption strategy to have the associated key development property. Allow the client to generate another mystery key SK' to replace SK, another assessment key EK', and a change key TK so that: the server, given only TK and EK', can completely replace all code texts in the corpus with new code texts that (1) can be decoded using SK' and (2) fulfill semantic security in any event, for a foe who holds SK.

Any suitably homomorphic encryption scheme meets the following key development characteristic. Allow TK to serve as SK's encryption. To be more explicit, TK is a code text that, when unscrambled using secret key SK', provides SK. A server with TK and EK' can modify the wording of a code.

2.5 SIDE CHANNEL ASSAULTS

Side channel assaults take into account enemies who can obtain insufficient information about an encryption plot's mystery key, such as by running timing assaults during the decoding calculation. Versatility against such attacks, also known as leakage flexibility, is an important security requirement for an encryption strategy. That is, it ought to be difficult to abuse semantic security even in presence of side channel assaults. Normally, spillage strength can only withstand limited data spillage concerning the mystery key.

An appealing component of encryption plans in light of immovability of whole number cross section issues, and specifically realized HE conspires.

2.6 CHARACTER BASED ENCRYPTION

It is possible to deliver encoded messages to clients in a personality encrypted plot without knowing either a public key or a secret key, but only the personality of the beneficiary, where the personality can be a legitimate name or an email address.

This is possible as long as a rusted party distributes a few public borders PP and has an expert mystery key MSK. After authenticating herself, for example, by producing a government-issued ID, a client having character X will be given a secret key SKx, which the client can use to decode any code message sent by the personality X.

To scramble message M to character X, all that is required is knowledge of the public bounds PP and X. Personality-based homomorphic encryption is an appealing version of public key homomorphic encryption. GSW modification maintains personality-based homomorphic.

3.BRAKERSKI / FAN-VERCAUTEREN(BFV)

We follow the same notations as the previous section.

BFV.ParamGen(λ,PT,K,B)→Params.

We assume the parameters are instantiated following there commendations outlined in Section 5. Similarly to BGV, the parameters include:

Key-and error-distributions1,2 a ring $R$and its corresponding integer modulus integer modulus for the plaintext. In addition, the BFV parameters also include: Integer T, and L= log Tq. T is the bit decomposition modulus. Integer W=$\lfloor q/p \rfloor$.

3.1 BFV.SecKeygen(Params) -> SK,EK

The secret key $SK$of the encryptions is a random element from the distribution defined as per Section 5. The evaluation key consists of LWE samples encoding the secret a specific fashion. In particular, for = 1,…,L, sample a random aI from R/qR and error eI from $D2$, computeEK=($EK1$,…,EKL).EK=(−(ais+ei)+Tis2,ai),

3.2 BFV.PubKeygen(params)-> SK,PK,EK.

The secret key SK of the encryption scheme is a random element from the distribution. The public key is a random LWE sample with the secret$s$. In particular, it is computed by sampling a random element a from R/qR and an error e from the distribution $D2$ and setting :P= (−(as+e),where all operations are performed over the ring R/qR. The evaluation key is computed as in BFV.SecKeygen.

3.3 BFV.PubEncrypt(PK,M)->C

BFV.Pub. Encrypt first maps the message comes from the message space into an element in the ring R/p.To encrypt a message $R/$ pR, parse the public key as a pair (pk0, pk1). Encryption consists of two LWE samples using a secret where (ppk0,pk1)is these are sample is auxiliary.

3.4 BFV.Decrypt(SK,C)->M

The main in variant of the BFV scheme is that when we interpret the elements of a cipher text C as the coefficients of a polynomial then, C(s)=WM+e for some “small” error e. The message M can be recovered by dividing the polynomial (by W, rounding each coefficient to the nearest integer, and reducing each coefficient modulo

3.5 PROPERTIES SUPPORTED.

The complete BFV scheme supports many features described in Section6, including packed evaluations of circuits and can be extended into a threshold homomorphic encryption scheme. In terms of security, the BFV homomorphic evaluation algorithms can be augmented to provide evaluation privacy.

For details on the implementation of the full BFV scheme, were the reader to [B12],[FV12].

3.5.1 Comparison Between Bgv And Bfv

When implementing HE schemes, there are many choices which can be made to optimize performance for different architectures and different application scenarios. This makes a direct comparison of these schemes quite challenging.

3.5.2 The Gsw Scheme And Boot Strapping

Currently, the most practical homomorphic encryption schemes only allow to perform bounded depth computations. These schemes can be transformed into fully homomorphic ones (capable of arbitrary computations) using a "bootstrapping" technique introduced by Gentry [G09], which essentially consists of a homomorphic evaluation of the decryption algorithm given the encryption of the secret key.

Bootstrapping using the BGV or BFV schemes requires assuming that lattice problems are computationally hard to approximate within factors that grow super polynomials in the lattice dimension n. This is a stronger assumption than the in approximability with in polynomial factors required by standard (non-homomorphic) lattice-based public key encryption.

In [GSW13], Gentry, Sahai and Waters proposed a new homomorphic encryption scheme (still based on lattices) that offers a different set of trade-offs than BGV and BFV. An important feature of this scheme is that it can be used to bootstrap homomorphic encryption based on the assumption that lattice problems are hard to approximate within polynomial factors. Here we briefly describe the GSW encryption and show how both its security and applicability to boot strapping are closely related to LWE encryption, as used by the BGV and BFV schemes. property which we call threshold-HE is desirable. In threshold-HE the key-generation algorithms, encryption and decryption algorithms are replaced by a distributed-key-generation

The (DKG) algorithm, distributed-encryption (DE) and distributed-decryption (DD) algorithms. Both the distributed-key- generation algorithm and the distributed- decryption algorithm are executed via an interactive process among the participating users. The evaluation algorithms EvalAdd, Eval Mult,Eval Mult Const, Eval Add Const and Refresh remain unchanged. Bootstrapping is a very time-consuming operation and improving on its efficiency is still a 25very active research area. So, it may still not be ready for standardization, but it is the next natural step to be considered.

In [GSW13], Gentry, Sahai and Waters proposed a new homomorphic encryption scheme (still based on lattices) that offers a different set of trade-offs than BGV and BFV. An important feature of this scheme is that it can be used to bootstrap homomorphic encryption based on the assumption that lattice problems are hard to approximate within polynomial factors. Here we briefly describe the GSW encryption and show how both its security and applicability to bootstrapping are closely related to LWE encryption, as used by the BGV and BFV schemes. So, future standardization of bootstrapping (possibly based on the GSW scheme) could build on the current standardization effort. The cipher text encrypt the message as ($a+m$, $as+ e$), but this is just a minor variant on LWE encryption, and equivalent to it from a security stand point.) Security rests on the standard LWE assumption, as used also by BGV and BFV, which says that the distribution (A,$A*S+E$)is pseudorandom.

4. RESULT

Encryption and Decoding are the Significant Piece of Gotten correspondence, In homomorphic Encryption Plan Procedure on Chiper Text can be handily finished and without Need of any Unscrambling First.

The Engineering utilized gives improved Results in calculation speed, region, energy and power are of greater Need Result to be Exact i.e.,

By Utilizing ISE Plan SUITE Task Guide, Xilinx 14.7 adaptation in the recreation system the improved Results are Accomplished with Less Circuit Region With Low Power Utilization.

The encryption activity ought to be checked that the given information message is scrambled impeccably or not .on the off chance that any mistakes happen in the encryption activity, similar advances happen.

4.1 SIMULATION PROCEDURE

For clk: forcing a clock value: duty cycle: 50, period : 100. Rst: forcing a value: 1

Give any user input M=Msg= 34

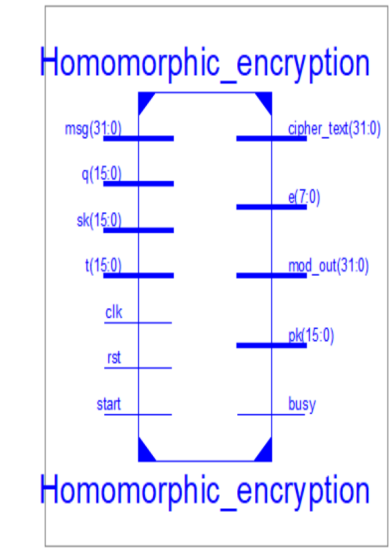
Q = Modulus input= 8

T= poly modulus input= 4

Run simulation and again change rst: forcing a value: 1 Run simulation and again change rst: forcing a value: 0 And run simulation

For the above inputs the outputs for the encryption of the message is shown in the below images.

4.2 RESULTS OF ENCRYPTION OPERATION



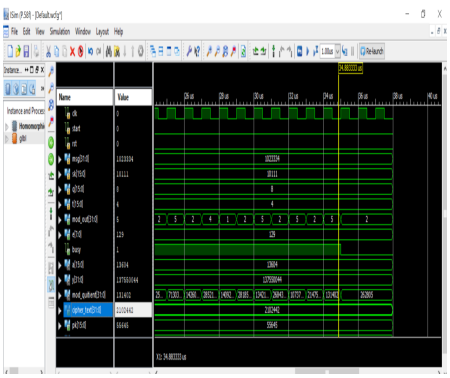


Figure 4.1: Results of Encryption Operation

After performing the encryption the received output message should be decrypted at the output end by performing decryption operation and the outputs of the decryption is shown below.

The outputs of the encryption will be verified and the decryption operation should be performed which is shown in below.

The outputs from the encryption operation should be verified that the given input message is encrypted perfectly or not .if any errors occur in the encryption operation then the same steps should be performed.

4.3 RESULTS OF DECRYPTION OPERATION

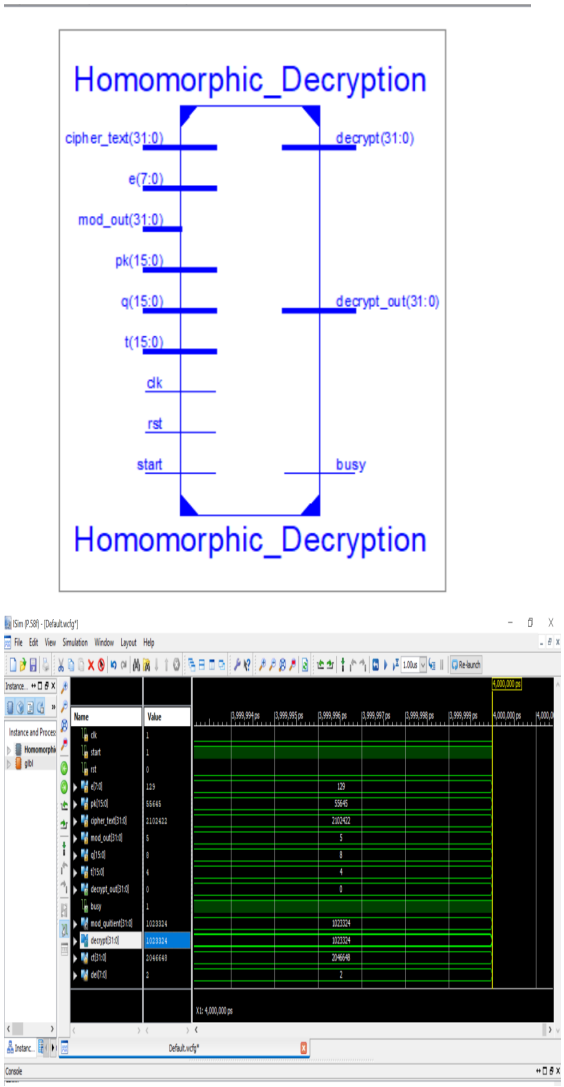


Figure 4.2: Results of Decryption Operation

The comparison table that shows the improved parameters for existing method to theproposed method is given below

```
Asynchronous Control Signals Information:
----------------------------------------
No asynchronous control signals found in this design

Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 1.943ns (Maximum Frequency: 514.646MHz)
   Minimum input arrival time before clock: 27.715ns
   Maximum output required time after clock: 2.150ns
   Maximum combinational path delay: No path found

Timing Details:
```

Figure: Existing delay

Table 4.1: Comparison for Existing Method and the Proposed Method

Parameters	Existing Method	Proposed Method
Clock Frequency	200 MHz	514.64 MHz
LUT	800	526
Slice Registers	726	105
Delay (ns)	5.0	1.943

The comparison table that shows the improved parameters for existing method to the proposed method is given above. According to the above table, the suggested technique has a shorter latency than the present way, which improves the performance of the architectures employed.

5. CONCLUSION AND FUTURE SCOPE

Here this Project We Can Reduce the Power Consumption and Delay by Using Vedic Multiplier and Kogge stone Adder which is Fast in Performance**.** The greater values of frequency will results in the reduction in delay. So the performance of the system will be increased. And also the number of LUT's used in this project will alsobe reduced.

This work can be used Furtherly for 128 bits, 256 bits, can be Implemented by Further Different Multipliers and Fastest Adder to Increase the Performance of the Circuit. Ultimately, given minor adjustments, this accelerator's core logic modules may be utilized to construct ring arithmetic with bigger ring degrees and modulus sizes. We are now working on a new design based on our existing architecture that will lower energy usage using a parallel prefix adder (similar to the Kogge stone Adder), and the findings will be revealed in our future research.

REFERENCES

[1] Albrecht, M. R. (2017). On dual lattice attacks against small-secret LWE and parameter choices in HE lib and SEAL. In J. Coron & J. B. Nielsen (Eds.), EUROCRYPT 2017, parii (Vol.10211, pp. 103–129).Springer, Heidelberg.

[2] Martin R. Albrecht, Robert Fitzpatrick, and Flori an Gopfert: On the Efficacy ofSolving by Reduction to Unique-SVP. In Hyang-Sook Lee and Dong-Guk Han, editors, ICISC 13,volume8565 of LNCS, pages293-310.Springer,
November2014

[3] Martin R. Albrecht, Rachel Player and Sam Scott. On the concrete hardness of Learning with Errors. Journal of Mathematical Cryptology. Volume 9,Issue 3.

[4] Alkim, E.,Ducas,L.,Pöppelmann,T.,& Schwabe, P. (2016).Post-quantum key exchange-A new hope. In T.Holz & S. Savage (Eds.), 25thUSENIX security Alperin-Sheriff,J.,Peikert,C.:Fasterbootstrappingwithpolynomialerror.In:Garay,J.A.,Gennaro,R.(eds.)CRYPTO2014. LNCS, vol.8616, pp. 297–314.

[5] László Babai : On Lovász' lattice reduction and the nearest lattice point problem, Combinatorica, 6(1):1-3,1986.

[6] Becker, A.,Ducas,L., Gama,N., &Laarhoven,T. (2016).New directions innearest neighbor searching with applications to lattice sieving. In R. Krauthgamer (Ed.),

[7] W. Castryck, I. Iliashenko, F. Vercauteren, Provably weak instances of ring-lwe revisited.In:Eurocrypt2016. vol.9665, pp.147–167. Springer(2016)

[8] W. Castryck, I. Iliashenko, F. Vercauteren, On error distributions in ring-based LWE. LMS Journal of Computation and Mathematics 19(A),130–145 (2016) 7.

[9] Y. Chen, P.Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In: Lee D.H., Wang X.(eds) Advances in Cryptology – ASIACRYPT 2011. ASIACRYPT .

[10] Laarhoven, T. (2015). Search problems in cryptography: From fingerprinting to lattice sieving (PhD thesis). Eindhoven University of Technology.

[11] Vadim Lyubashevsky, Chris Peikert, and Oded Regev : A tool kit for ring- LWE cryptography. Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg,2013.