

**ADVDISC**

Machine Project 2 Documentation

|  |  |
| --- | --- |
| **Section** | S19 |
| **Team Members** | Amadora, Angelo John  Andres, John Joseph  Fernandez, Ryan Austin  Syfu, Jonah Espiritu |
| **Date Submitted** | December 1, 2015 |

**Table of Contents**

I. Contribution of Each Member / 3

II. Introduction of the Research Area / 4

III. Use of Matrices and Linear Algebra Methods in the Research Area / 5

IV. Design / 9

V. Implementation / 11

VI. Conclusion / 12

VII. References / 13

**I. Contribution of Each Member**

|  |  |  |
| --- | --- | --- |
| **Member** | **Code** | **Other** |
| Andres, John Joseph | ModArith | Testing |
| Amadora, Angelo John | CipherPanel  DecipherPanel  MainPanel  MainWindow | Quality Assurance |
| Fernandez, Ryan Austin | AbstractMatrix  ModularMatrix  Cipher  CofactorExpansion  FileManager  ModArith  Matrix  CipherController  CipherPanel  DecipherPanel  DerivePanel  MainPanel  CipherInputFrame  MainWindow  Driver  AGBLayout  InvalidCipherException  IController | Documentation  Research |
| Syfu, Jonah Espiritu | ModArith  Cipher | Testing |

**II. Introduction**

Linear Algebra is a very useful field of mathematics, especially in the field of computer science, particularly the concept of linear transformations. Linear transformations are used liberally in the field of cryptography.

Cryptography is the study of techniques in securing messages or text by enciphering them using a set algorithm. The original text is referred to as “plaintext” and the enciphered text is referred to a “ciphertext”. Decoding the ciphertext is referred to a “deciphering”. Finding the algorithm used is referred to as “cracking” or “breaking” a cipher.

The algorithm this project aims to implement is the Hill Cipher algorithm. The objective of the project is to implement the enciphering, deciphering, and the breaking of the Hill Cipher.

**III. Use of Matrices and Linear Algebra Methods in the Research Area**

**Enciphering Using the Hill Cipher**

A Hill-n Cipher makes use of an n x n matrix A and a modulus n. The only requirements of the matrix are that its determinant has a modular inverse with respect to the modulus. This means it does not have any common prime factors.

Once an appropriate matrix is found, a plaintext string is enciphered by first transforming it into a matrix. This is done by first representing the string as its numerical representation.

Assume the scheme is the number’s cardinality in the English Alphabet. Take the string BEATRICE. Our modulus is therefore 27 since the values 0 – 26 are being used, 0 representing a whitespace. This would be represented as

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| B | E | A | T | R | I | C | E |
| 2 | 5 | 1 | 20 | 18 | 9 | 3 | 5 |

This is then transformed into vectors in Rn depending on the cipher parameters. Assume a Hill-3 Cipher is being used. The word is transformed into 3-tuple vectors. Any missing characters are pumped with a space.

Let the matrix A be whose determinant is -10, which has no common prime factors with 27 and is thus a valid matrix.

To encipher the message, the vectors created from the message are collated into a single matrix X. The ciphertext comes from the product Y = AX mod 27, where the column vectors of Y are the corresponding vectors of the ciphertext.

Which translates to

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 17 | 14 | 4 | 1 | 20 | 21 | 20 | 13 |
| U | Q | N | D | A | T | U | T | M |

So the corresponding ciphertext for BEATRICE in this cipher scheme is UQNDATUTM.

**Deciphering**

To decipher the ciphertext, first we must get the inverse of the enciphering matrix A. This is done slightly differently than the regular method. First, the adjunct of the matrix is computed. This is simply the transpose of the matrix containing all the matrix’s cofactors.

For the matrix above, the adjunct is

Normally, the inverse is found by multiplying the adjunct by the multiplicative inverse of the determinant, but inverse is now computed by multiplying the adjunct with the multiplicative modular inverse of the determinant. The multiplicative modular inverse of a number modulo m is a number a-1 such that aa-1 = 1 mod m.

The determinant of the original matrix was -10. The multiplicative modular inverse is then 8 since -10(8) = 80 = 1 mod 27.

The inverse is then

So taking the ciphertext UQNDATUTM, translating into

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| U | Q | N | D | A | T | U | T | M |
| 21 | 17 | 14 | 4 | 1 | 20 | 21 | 20 | 13 |

Which is converted into the matrix.

The plaintext is gotten by multiplying A-1Y mod 27.

Which translates to

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 5 | 1 | 20 | 18 | 9 | 3 | 5 |
| B | E | A | T | R | I | C | E |

Which is the original string from the previous chapter.

**Breaking the Cipher**

Given a plaintext “BEATRICE “ and a ciphertext “UQNDATUTM”, the original enciphering matrix could be derived.

Let p1, p2, …, pn be linearly independent plaintext vectors and c1, c2, …, cn be the corresponding ciphertext vectors, the deciphering matrix could be derived by forming first matrices C and P, C having c1T, c2T, …, cnT as row vectors and P having p1T, p2T, …, pnT as row vectors, augmenting P to C, forming [C|P] and row reducing that matrix to [In|(A-1)T]. A-1 is the transpose of the right half of the augmented matrix.

So using the same example, are linearly independent plaintext vectors and are the corresponding ciphertext vectors.

[C|P] is then formed by

Row reducing…

Getting the transpose of the right part of the augmented matrix, we have, which is the exact same deciphering matrix that was computed in the deciphering section of this chapter. The inverse it then the same as the enciphering matrix used in the beginning of the chapter, which is , proving that this algorithm remains consistent.

**IV. Design**

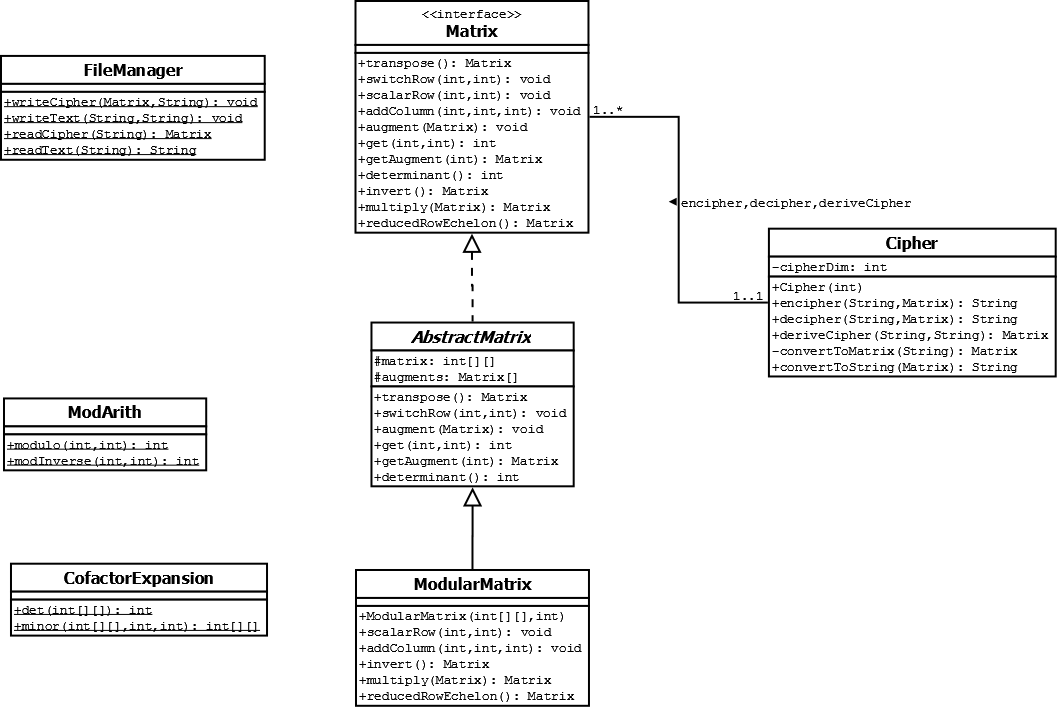
**String Representation**

The way strings are represented uses the American Standard Code for Information Interchange ASCII values of the characters. Not wanting to map to the control characters (ASCII 00h to 1Fh and 7Fh), this gives 95 mappable characters. Including the horizontal tab (09h) and the line feed (0Ah) characters; this gives 97 characters, which makes the modulus for the algorithm discussed above 97, a prime number. Thus, any matrix with a nonzero determinant which is not divisible by 97 would be usable in the Hill cipher algorithm.

The character mapping uses the following function:

**Software Architecture**

The software follows the model view architecture. The only point of interest is the model component.



The model module consists of five parts, namely Matrix, CofactorExpansion, ModArith, Cipher, and FileManager.

The Matrix part handles all matrix operations including transposition, row operations, computing for determinants, creating augmented matrices, multiplication, and reduction to reduced row echelon form. For augmented matrices, the Observer pattern was used. Every time a row operation was performed, the same operations would be performed on the augmented matrices.

Furthermore, the CofactorExpansion class simply handles computation of determinants using the cofactor expansion algorithm, which has a big-O time complexity of O(n!) given an n x n matrix.

In addition to those classes, the ModArith class handles modular arithmetic functions including modulo and finding the modular inverse.

Moreover, the Cipher class handles all Hill Cipher related algorithms, maintaining the dimension as an internal state. The operations include enciphering, deciphering, and cracking the cipher.

Finally, the FileManager class handles all input and output of ciphers, plaintext, and ciphertext.

**V. Implementation**

The Java programming language was used, where Java Swing was used for the Graphical User interface.

No special functions were used aside from the common expression operators, particularly %, which was used to compute the mathematical modulo of numbers for use in the computations using the following formula:

To find modular multiplicative inverses, the following pseudocode was followed

getModInverse(a,m) returns a modular inverse of a

Initialize aInv to 1

while a \* aInv is not congruent to 1 mod m

aInv++

return aInv

For the ModularMatrix, the superclass’ constructor is called and each value is then taken the modulo of relative to 97. All operations on this type of matrix end with this operation to maintain modulus 97.

**VI. Conclusion**

The objective of the project is to implement the enciphering, deciphering, and the breaking of the Hill Cipher. After multiple rounds of testing the final project, it can therefore be concluded that the group was successfully able to fulfill the objective. The software successfully enciphers, deciphers, and cracks the Hill-2 and Hill-3 cipher.

**VII. References**

Anton, H. & Rorres, C. (2010). *Elementary linear algebra: Applications version*. John Wiley & Sons, NJ:

New Jersey.

Cryptography. (n.d.). Retrieved November 5, 2015, In Wikipedia: https://en.wikipedia.org/wiki

/Cryptography