Volume 3, Issue 5, May 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Stand-Alone Java Application for Cryptographic Algorithms

Mrs. Ruchika Gupta[1],
Assistant Professor &
Program Director,
School of IT, Auro University,
Surat, Gujarat, India

Ms. Pallavi sharma[2]
Lecture, Department of Computer Science,
AmitySschool of Engg.& Technology,
Amity university Rajasthan,
Jaipur (RJ), India.

Abstract: ***This paper deals with development of a Java based database application for the Mathematics and Computer Science departments to aid better understanding of cryptographic algorithms. The paper provides a step-by-step animated user interface presentation of how various cryptographic algorithms work. The user interface is designed to give a hands-on experience for students to try out the algorithms by providing various inputs and see how the plain text is converted into ciphered text with a nice visual interpretation. The main motive behind using Java for creating the application is the ease in developing user interfaces with the large library and localization features that the Java programming language provides. The user interface has been localized for French (France), English (US) and Spanish (Mexico and Spain). This application provides work for Vernam Cipher, Vigenère Tableau, and Data Encryption Standard (DES) algorithms.***

Keywords: Cryptography, Vernam Cipher, Vigenère Tableau, DES

1. Introduction:

1.1 Background

Security has been a major issue since the time communication evolved. Everyone needs to have a method to communicate to a known person securely without a third person being able to understand the message. Over the centuries many people have proposed various algorithms or ways to encrypt and decrypt the messages. Some of these algorithms have been weak and easily broken while others have proven to be more rigid. However, most of these algorithms have been broken and there is a constant need for more rigid and powerful algorithms as the world is opening up to the Internet. Some algorithms are simple and easy to understand whereas the more complex algorithms need some kind of visualization to understand Hence, there is a clear necessity for a visualization tool, which acts as a work for various algorithms. This computer-aided work provides step-by-step in-detail analysis of the encryption and decryption process. This tool gives the user a feel of how a message would be encrypted in the backend.

1.2 Limitations of Current Practice

With the help of the visualization tool the student would tend to remember more than he would with trying it out on paper. Pictures are easier to associate and remember than sheets of formulas. Thus the need for a graphical user interface for cryptographic algorithm arises. And this work provides a good solution to some of the algorithms, as it is not possible to provide a work for a large set of available algorithms**.**

2.Theory Of Cryptographic Algorithms Selected For The Work

There are numerous cryptographic algorithms that have been used in the past and are currently being used for encrypting messages. This work gives a step-by-step visual representation for three algorithms: Vernam Cipher, Vigenère Tableau, and Data Encryption Standard (DES).

2.1 Vernam Cipher

The Vernam Cipher is a one-time pad devised by Gilbert Vernam for AT&T. Gilbert Vernam the task of inventing an encryption method the Germans could not break. To use a one-time pad, you need two copies of the 'pad' (also known as the key), which is a block of truly random data at least as long as the message you wish to encode. If the data on the pad is not truly random, the security of the pad is compromised. The one-time pad is unbreakable if used properly. The pad must be composed of truly random data, it must never be used more than once and it must be kept secure. If each key letter in the pad sequence is truly random a cryptanalyst can do no better than try every possible key letter for every *cipher text* message position. Number of possible messages is in the region of 200,000,000,000,000,000,000,000. The *cipher text* can provide no clues as to which one of these possibilities is the real message [1]. .

2.2 Vigenère Tableau

The first well-documented description of a poly-alphabetic cipher was formulated by Leon Battista Alberti around 1467 and used a metal cipher disc to switch between cipher alphabets.

To encrypt, a table of alphabets called a Vigenère square, is used (see Figure 2.1). This table consists of the alphabet written out 26 times in different rows, with each alphabet shifted cyclically to the left compared to the previous and corresponds with the 26 possible Caesar ciphers. Throughout the encryption process, the cipher uses a different alphabet from a certain row. The selection of which alphabet to use depends on a repeating word.

PlainText:	XRAMPISGREAT
Key:	HOUSEHOUSEHO
CipherText:	EFUETPGAJIHH

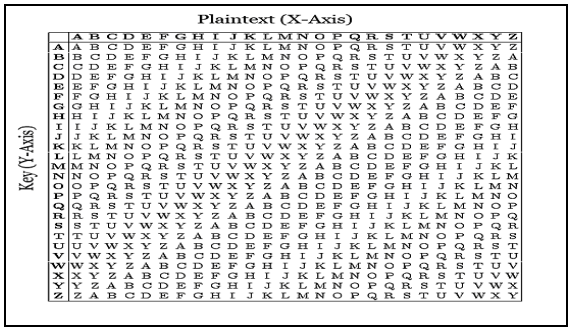

Figure 2.1. Vigenère Tableau

If L is the most commonly found letter in the cipher text, one could estimate that if the original plaintext was in English, E, the most common letter in the English alphabet, could be represented by the L. However, the overall weakness in the Vigenère cipher is due to the relatively short and repeated nature of its key.

2.3 Data Encryption Standard

The Data Encryption Standard (DES) algorithm is a careful and complex combination of two fundamental building blocks of encryption: substitution and transposition. The algorithm derives its strength from repeated application of these two techniques, one on top of the other, for a total of sixteen cycles. The sheer complexity of tracing a single bit through 16 iterations of substitutions and transpositions has so far stopped researchers in the public from identifying more than a handful of general properties of the algorithm [3].

The algorithm leverages the two techniques to conceal information: confusion and diffusion. That is, the algorithm accomplishes two things: ensuring that the output bits have no obvious relationship to the input bits and spreading the effect of one plaintext bit to other bits in the cipher text. Substitution provides the confusion, and transposition provides the diffusion. In general, plaintext is affected by a series of cycles of a substitution and then a permutation.

DES uses only standard arithmetic and logical operations on numbers up to 64 bits long, so it is suitable for implementation in software on most current computers. Although complex, the algorithm is repetitive, making it suitable for implementation on a single-purpose chip.

3. Implementation

3.1 Java- Based Stand- Alone Application

A Java-based application runs any operating system as long as the Java Virtual Machine for that operating system is installed. This work gives an insight to three different algorithms: Vernam Cipher, Vigenère Tableau, and Data Encryption Standard.

The first two of the above-mentioned algorithms have been internationalized for 4 languages:

- en-US (English – United States)
- fr-FR (French – France)
- es-ES (Spanish – Spain)
- es-MX (Spanish – Mexico)

3.1.1 Implementation of Vernam Cipher

The first algorithm is "Vernam Cipher" which is a one-time pad algorithm. The user interface is designed in a very simple manner and is event driven. There are buttons performing different functions such as Encrypting, Decrypting, Next Step, Skip Steps, and New Input. Two processes are explained in this work: Encryption and Decryption.

3.1.2 Implementation of Vigenère Tableau

The second algorithm, Vigenère Tableau, is a Book Cipher. This is a Private Key algorithm where the encryption and decryption need to know the *Key* to understand the message. The interface for this algorithm tries to keep the feel of the *Vernam Cipher* to maintain consistency so that the user finds it easy to use.

The user interface is event driven and has buttons performing different functions such as Encrypting, Decrypting, Next Step, Skip Steps, and New Input. Two processes are explained in this work: Encryption and Decryption.

3.1.3 Implementation of Data Encryption Standard

The third algorithm is Data Encryption Standard (DES). This is also a Private Key algorithm where the encryption and decryption need to know the *Key* to understand the message. The interface for this algorithm tries to keep the feel of the *Vernam Cipher* to maintain consistency so that the user finds it easy to use.

The user interface is event driven and has buttons performing different functions such as Encrypting, Decrypting, Next Step, Skip Steps, Next Iteration, Skip Iterations, Show Tables, Initial Permutation, Permutation Cycles, and Final Permutation. Two processes are explained in this work: Encryption and Decryption.

3.2 Software Internationalization

Internationalization and localization are means of adapting computer software to different languages and regional differences. Internationalization is the process of designing a software application so that the application can be adapted to various languages and regions without any changes involved in the software's business logic. Localization is the process of adapting software for a specific region or language by adding locale-specific components and translating text.

A few years ago, everyone was using English as the display language for the user interface. However when the need for developing software for the world market grew, the application developers realized the need to provide the user interface in the users' native language. This approach of providing the user interface would attract more local users and enhance the business prospective. Hence, the whole idea of "software internationalization" was conceived and most of the software application development languages began supporting this feature.

Localization is needed in the following places.

- Displaying characters for the users' native languages.
- Inputting characters for the users' native languages.
- Handling files written in popular encodings that are used for the users' native languages.
- Using characters from the users' native languages for file names and other items.
- Printing out characters from the users' native languages.
- Displaying messages by the program in the users' native languages.
- Formatting input and output of numbers, dates, money, etc., in a way that obeys customs of the users' native cultures.
- Classifying and sorting characters, in a way that obey customs of the users' native cultures [4].

3.3 Unicode

In most writing systems, keyboard input is converted into character codes, stored in memory, and converted to glyphs in a particular font for display and printing. The collection of characters and character codes form a code set. To represent characters of different languages, a different code set is used.

The application needs to use the resource bundle for the particular locale and get the translated strings by passing the key to the getString method of the resource bundle. We are just using string translations for this application and do not use other functionalities provided by the resource bundles. The resource bundle's properties file for en-US would look like this:

```
user_input = USER INPUT
plain_text = PLAIN TEXT
ciphered_text = CIPHER TEXT
decode = Decode
encode = Encode
new_input = New Input
next_char = Next Char
skip_steps = Skip Steps
create_table = Create Table
numeric_equivalent = NUMERIC EQUIVALENT
random_number = RANDOM NUMBER
sum = SUM
mod = MOD 27
number_to_be_added = NUMBER TO BE ADDED
difference = DIFFERENCE
Title = Vernam Cipher Work
```


The resource bundle's properties file for fr-FR would look like this:
user_input = ENTRÉE
plain_text = TEXTE CLAIR
cipher_key = CHIFFRE CLÉ
ciphered_text = TEXTE CRYPTÉ
decode = Décoder
encode = Encoder
new_input = Nouvelle Entrée
next_char = Char Suivant
skip_steps = Sauter Étapes
create_table = Créer Tableau
numeric_equivalent = NUMÉRIQUE ÉQUIVALENT
random_number = NOMBRE ALÉATOIRE
sum = AJOUTER
mod = MODULO 34
number_to_be_added = NUMBER TO BE ADDED
difference = DIFFÉRENCE
Title = Cours d'instruction de chiffre de Vernam

4. RESULTS (Graphical User Interface)

The GUI for this application is not very complicated. It is very user friendly and well organized and maintains a similar interface for all the algorithms.

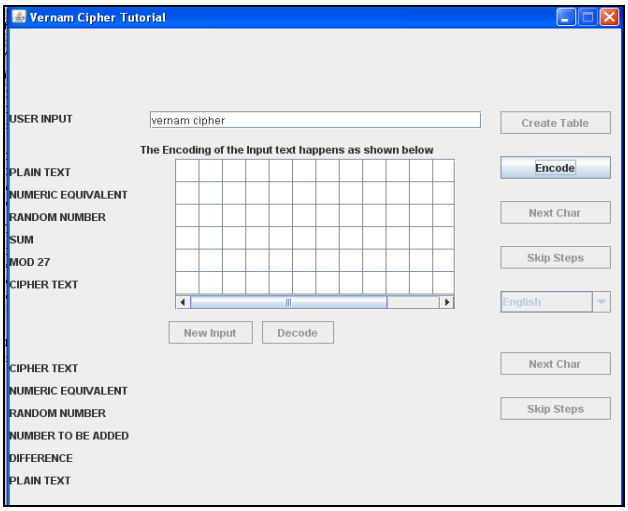

Figure 4.1 Vernam Cipher encoding: English version.

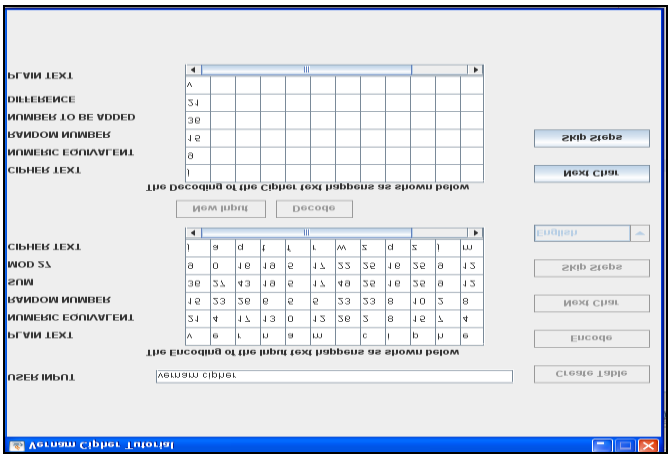

Figure 4.2 Vernam Cipher decoding: English version

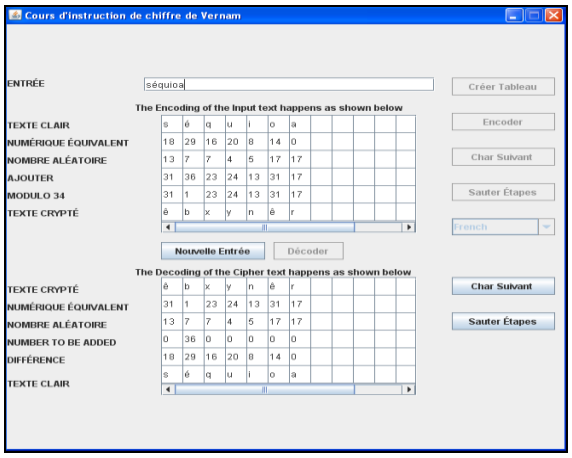

Figure 4.3 Vernam Cipher: French version

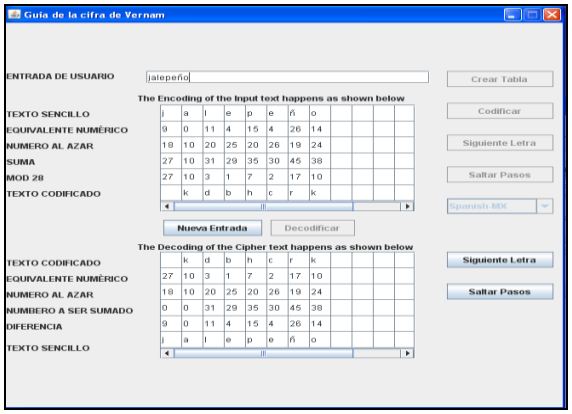

Figure 4.4 Vernam Cipher: Spanish (Mexico) version

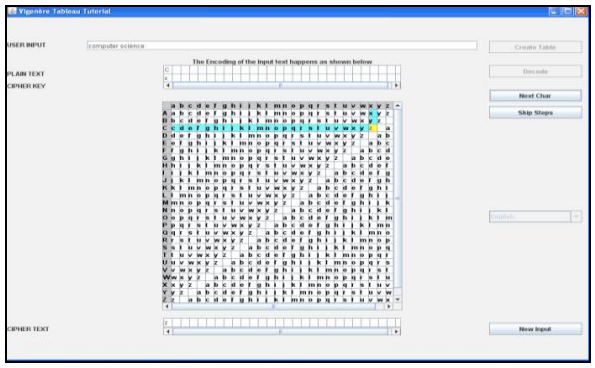

Figure 4.5 Vigenère Tableau encrypting: English version.

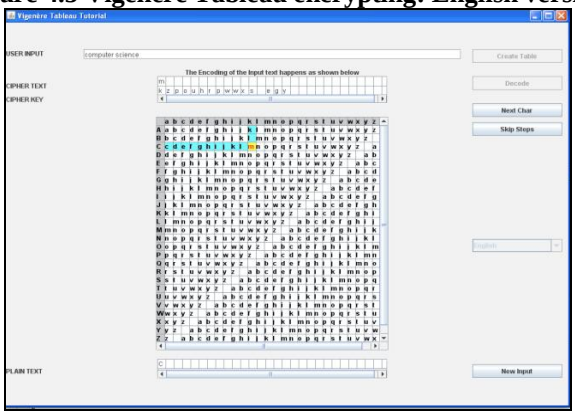

Figure 4.6 Vigenère Tableau decrypting: English version.

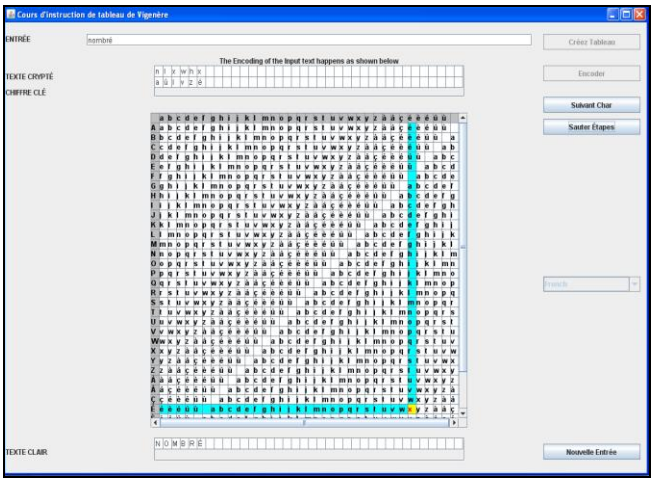

Figure 4.7 Vigenère Tableau: French version.

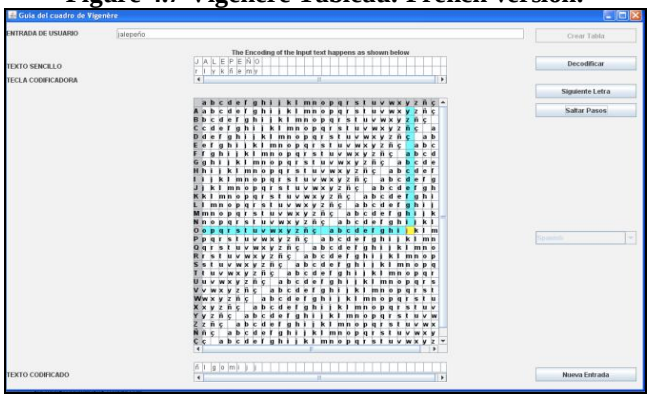

Figure 4.8 Vigenère Tableau: Spanish (international) version.

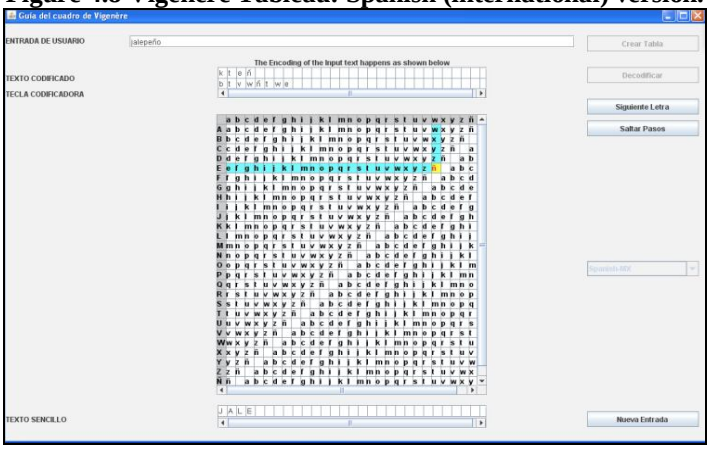

Figure 4.9 Vigenère Tableau: Spanish (Mexico) version.

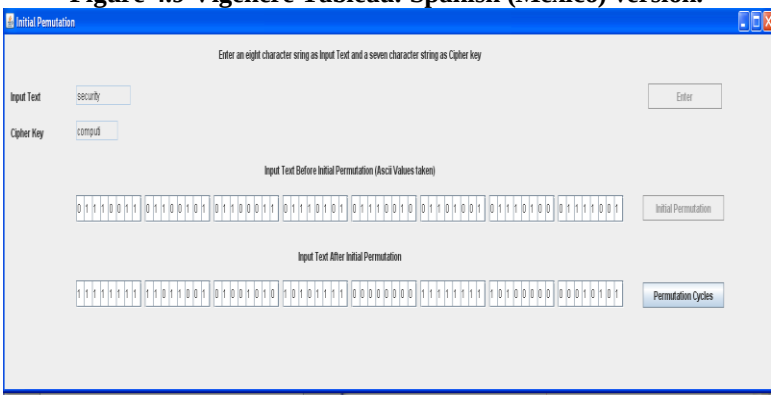

Figure 4.10 DES encryption phase: initial permutation.

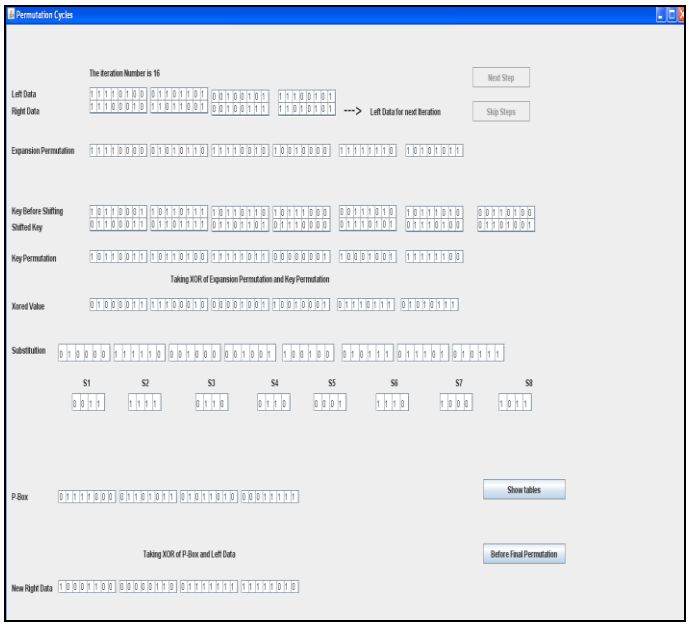

Figure 4.11 DES encryption phase after sixteen permutation cycles.

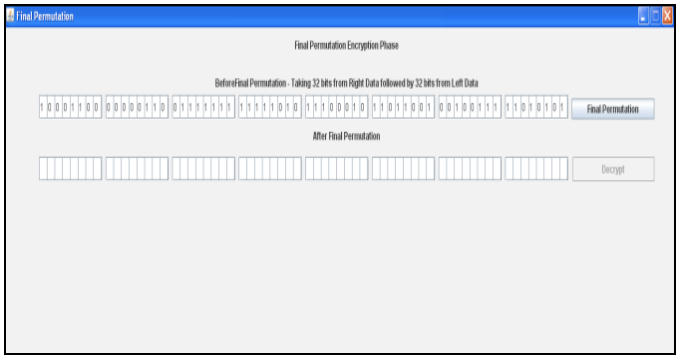

Figure 4.12 DES encryption phase: final permutation.

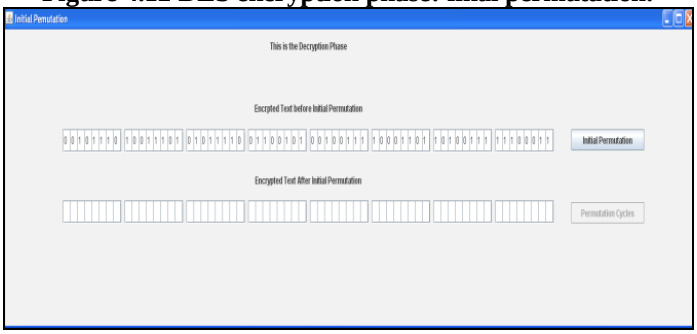

Figure 4.13 DES decryption phase: initial permutation.

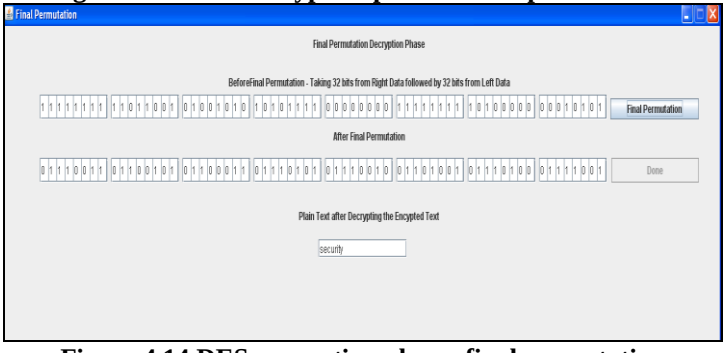

Figure 4.14 DES encryption phase: final permutation

5. Conclusion

The "Stand alone JAVA application for Cryptographic Algorithms" application was an approach to help the students of Mathematics and Computer Science Departments in understanding the cryptographic algorithms in an intuitive manner. The GUI provided in this tool would help them try various inputs and see how the plain text is transformed in each step. This is the first step towards addressing the difficulties the students face and giving them a useful interface to suit their needs. New features can be added to the "Computer Aided Work for Cryptographic Algorithms" application and extended as and when needed.

6. Future Enhancements

This work gives a step-by-step insight into 3 algorithms: Vernam Cipher, Vigenère Tableau, and Data Encryption Standard. In the future, works for algorithms like AES, RSA, and Two-fish can be added that would help the students in easier understanding of the complex algorithms.

Secondly, more languages can be supported for the existing work as well as for any new work to be developed. This work runs as a stand-alone application and could later be converted into a web application so that the user need not download the application and instead run it from the browser.

References

[1] Cryptology and Data Secrecy: The Vernam Cipher. Retrieved November 2008 from the Protechnix Web site http://www.pro-technix.com/information/crypto/pages/vernam_base.html

[2] Cryptology and Data Secrecy: The Vernam Cipher. Retrieved November 2008 from the Protechnix Web site http://www.pro-technix.com/information/crypto/pages/vernam_base.html

[3] Vigenère Tableau. Retrieved November 2008 from the Wikipedia Website http://en.wikipedia.org/wiki/Vigenère_cipher

[4] Pfleeger, C. P. & Pfleeger, S. L. (2003). *Security in Computing*. Upper Saddle River, New Jersey: Prentice Hall.

[5] Kubota, T. (2008). Introduction to I18N. Retrieved November 2008 from the Debian Web site http://www.debian.org/doc/manuals/intro-i18n/ch-intro.en.html

[6] Cryptology and Data Secrecy: The Vernam Cipher. Retrieved November 2008 from the Protechnix Web site http://www.pro-technix.com/information/crypto/pages/vernam_base.html

[7] Vigenère Tableau. Retrieved November 2008 from the Wikipedia Website http://en.wikipedia.org/wiki/Vigenère_cipher

[8] Pfleeger, C. P. & Pfleeger, S. L. (2003). *Security in Computing*. Upper Saddle River, New Jersey: Prentice Hall.

[9] Kubota, T. (2008). Introduction to I18N. Retrieved November 2008 from the Debian Web site http://www.debian.org/doc/manuals/intro-i18n/ch-intro.en.html

[10] Stinson, D. R. (2002). *Cryptography: theory and practice*. Boca Raton: Chapman &Hall/CRC.

