Journal of Engineering and Applied Sciences 12 (5): 1186-1191, 2017
ISSN: 1816-949X


# An Enhanced Encryption Algorithm for Database Protection Based on Dynamic Key and Reverse String

[1]Waleed Khalid Ahmed, [1, 2]Muamer N. Mohammed and [1]Norrozila Sulaiman
[1]Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, 26300 Kuantan, Pahang, Malaysia
[2]IBM Center of Excellence, University Malaysia Pahang, 26300 Kuantan, Pahang, Malaysia

**Abstract:** With the rapid growth of the embedded system market, especially in the area of ubiquitous e-services, an increasing number of embedded systems are required to deal with sensitive and private information. In this study, a new algorithm to secure and protect databases is proposed by using a Dynamic Key (DK) and reverse string for each character in the string (or field in the file of a database) and each string (or record in the file of a database). In this paper, a new encryption algorithm for database protection based on Dynamic Key and Reverse String (DKRS). The proposed algorithm (DKRS) specifically protecting the sensitive data because the (DKRS) is using dynamic key for each character in the string. The dynamic key is generated using special equation that depends on length of string, order of the character in the string, ASCII code of the character and string sequence. The (DKRS) algorithm guarantees that no duplicate encryption for each character and each record will be different. Also, (DKRS) algorithm ensures a different encryption for the duplicate data. The (DKRS) algorithm is using the reverse string before the encryption and many steps after that to add more complexity.

**Key words:** Dynamic key, revers string, cryptography, encryption, plaintext, ciphertext, caesar cipher

## INTRODUCTION

The enormous amount of information resources available and ease of communication have made the Internet the most valuable tool in various settings of human life. A complete solution to data security must possess confidentiality, integrity and authenticity (Arai *et al*., 2014). Confidentiality refers to the protection of data against unauthorized disclosure. Integrity refers to the prevention of unauthorized and improper data, data modification and detection of privilege abuse in role-based access control-administered databases. Availability refers to the prevention of and recovery from hardware and software errors and from malicious data access denials that make the database system unavailable (Arai *et al*., 2014). Traditional program verification focuses on program safety which specifies what operations are allowed on data during a single-program execution. Meanwhile, program security specifies what information can be inferred about an object across all possible program executions. Run-time mechanisms for enforcing security policies must track all possible execution states and could therefore be impossible (Schneider, 2000). Static analyses, particularly information flow-type systems have become an indispensable and complementary technique to prevent information leaks. Language-based security with information flow-type systems or security-type languages has become a promising approach to specify and statically enforce security policies. Digital media for information are dealt with via secure and non-secure channels to display messages sent across networks of hackers or third parties. Encryption of messages in this modern age of technology is necessary to ensure that data sent via communication channels are protected and difficult to decipher (Kester, 2012, 2013). Large amounts of data pass through the internet which is considered the most efficient and available means of communication. Therefore, the weaknesses of the Internet need to be addressed. Many researchers have proposed efficient algorithms to encrypt information from plaintext to ciphers (Khalaf *et al*., 2013; Kester, 2013). In information security, encryption is the process of converting clear information into unreadable words for unauthorized persons through algorithms. These algorithms usually use a key. The result of the process is called cipher text.

**Corresponding Author:** Waleed Khalid Ahmed, Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, 26300 Kuantan, Pahang, Malaysia

**Literature review:** Mousa *et al.* (2013) presented an encryption application that can work with data access tables. This application is based on the caesar encryption method (symmetric key) which was developed to generate one key to each record. The proposed algorithm computes the length of the key from the first word of the record. The cipher text is stored in a separate line in a text file that separates each field by a semicolon and so on until the end of the table. Naji (2014) proposed an algorithm called reverse encryption algorithm. The aim of the algorithm is to achieve the required balance between security and excellent performance. Several famous encryption methods (DES, 3DES, RC2, AES and blowfish) were evaluated. Experimental results showed that the proposed encryption algorithm might be superior to other encryption algorithms in terms of performance and database security. The algorithm design is based on an asymmetric key and a Dynamic Key (DK). Mousa *et al.* (2013) proposed a simple algorithm with a novel security approach in database encryption; it is also based on asymmetric key and DK. The encryption scheme is based entirely on table number, column value and row value, that is, every data field possesses a unique key based on its position and place instead of a common key value for all fields (Mousa *et al.*, 2013).

## MATERIALS AND METHODS

**Encipher model:** An encryption scheme has the following five components:

- Plaintext: this is a clear message that represents the input to the encryption algorithm
- Encryption algorithm: It is the method used for various substitutions and transformations of plaintext
- Secret key: the key is a value independent of the plaintext and encryption algorithm and it represents the input to the encryption algorithm. The algorithm produces different outputs depending on the key used at the time, length and date
- Ciphertext: this is a random or unclear message that represents the output of the encryption algorithm
- Decryption algorithm: this is the reverse of the encryption algorithm. It converts cipher text into plaintext or the original message is extracted from a random message

**Cryptography:** Cryptography is the science of modifying secret information to keep it hidden from attackers. Any cryptographic algorithm uses mathematics as the basis to encrypt and decrypt data. In the past, algorithms relied on the secrecy of the encryption algorithm itself but their security needs were inadequate for real-world needs. For this reason, the security of all modern algorithms relies on encryption and decryption keys; an encrypted message is decrypted if and only if the decryption key is mathematically related to the encryption key. The main goals of cryptography are authentication, privacy, integrity, non-repudiation and access control. Encryption which is basically a process, algorithm or method to create hidden data or secret information is part of cryptography. Figure 1 shows the encryption process in general. Decryption is the process to transform or convert encoded data into readable data; in this process, cipher text is regarded as the input and natural text as the output. Figure 2 shows the decryption process in general.

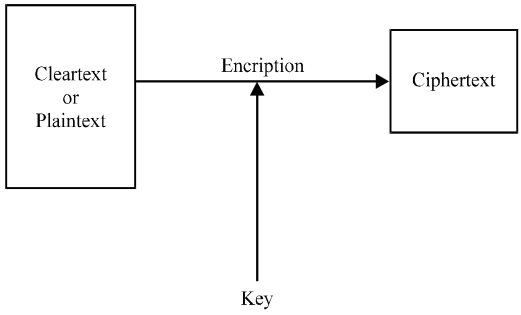


Fig. 1: Schematic of the encryption process

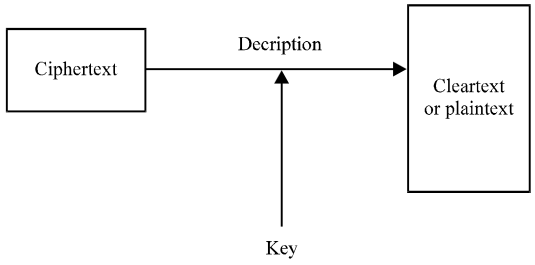


Fig. 2: Schematic of the decryption process

The encryption algorithm involves a set of steps and mathematical functions that consequently lead to encryption and decryption. The main objective of this algorithm is to make the encryption process as difficult as possible so that hackers would experience difficulty discovering the explicit text even if they had obtained the key. If a good cryptographic algorithm is utilized, then the best technique is to methodically test every possible combination of the key. Encipherment systems are classified into two categories, namely, symmetric-key (secret-key) cryptography and asymmetric-key (public-key) cryptography.

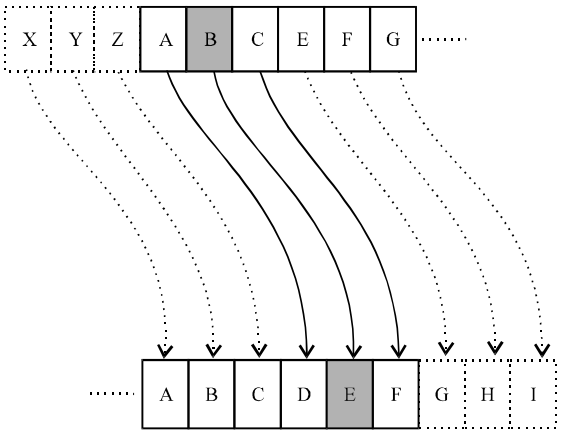


Fig. 3: Schematic of the caesar cipher mechanism

Asymmetric-key cryptography involves many mathematical computations for encryption and decryption; thus, it is slower than symmetric-key cryptography and is recommended for enciphering small messages. For the encryption of large messages, symmetric-key cryptography is preferable. The features of encryption systems depend on the following three independent dimensions (Kester, 2012).

**Type of operations used to transform plaintext to cipher text:** Two general principles represent the core of encryption algorithms. These two are substitution, in which each character or element in a plaintext is mapped into different elements and transposition, in which each character or element in a plaintext is reordered or rearranged.

**Number of keys used:** If the same key is utilized by the sender and receiver or if the same key is used for encryption and decryption algorithms, then this key is called a symmetric or secret key. By contrast, if different keys are adopted by the sender and receiver or if different keys are used for encryption and decryption algorithms, then this key is called an asymmetric or public key.

**Manner by which plaintext is processed:** The procedure in which plaintext is processed by inputting either one block or one string of elements at a time and producing an output block or string for each input is called the block cipher process. When plaintext is processed by inputting elements continuously one by one and outputting one element at a time, the process is called stream cipher.

**Classical encryption methods:** All classical encryption methods utilize two main techniques, namely, substitution and transposition (Stallings, 2005).

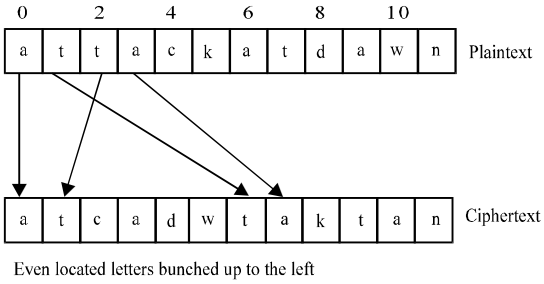


Fig. 4: Schematic of transposition cipher

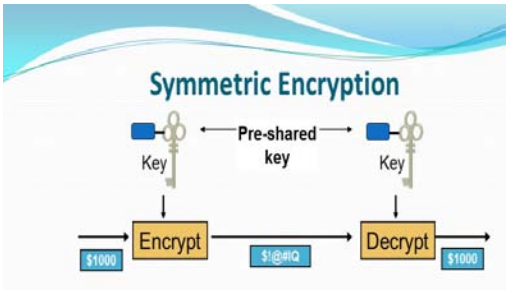


Fig. 5: Schematic of symmetric-key cipher

**Substitution:** The best example of this technique is the Caesar cipher in which each symbol in a plaintext is replaced with another symbol to generate a cipher text (Sharma *et al*., 2012). Figure 3 shows the substitution by the Caesar cipher algorithm.

**Transposition techniques:** Unlike substitution techniques, transposition techniques depend on some type of permutation on the plaintext letters only that are not changed. Symbols in the plaintext exchange their values with one another to form a cipher text (Stallings, 2005). Therefore, this technique is called transposition cipher. Figure 4 shows the transposition cipher algorithm.

**Symmetric-key and asymmetric-key encipherment:** Cryptography can be divided into two main types according to the type of the security key. These two types are symmetric and asymmetric encryption techniques.

**Symmetric-key encipherment:** The general idea behind symmetric-key encipherment is that the same key is used for both encryption and decryption. On the sender side, a plaintext is enciphered with a key (along with a set of functions and rules) and sent as a cipher text to the receiver (Khalaf *et al*., 2013; Ramaraj, 2012). Figure 5 shows the symmetric-key encryption.

**Asymmetric-key encipherment:** In asymmetric-key encipherment, a key pair of public and private keys is

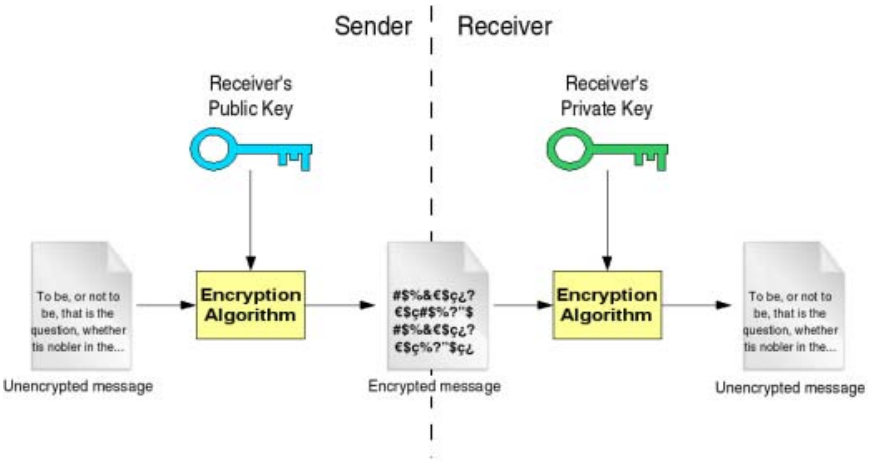


Fig. 6: Schematic of asymmetric-key cipher

used. Mathematical calculations are conducted to bind the two keys in a certain relation. Any of the keys can be used for either encryption or decryption and the other is used for the reverse operation. For example, the public key is used to encrypt a message that can be decrypted only by the corresponding private key and vice versa (Khalaf *et al*., 2013). Figure 6 shows the general architecture of asymmetric-key encryption.

**Asymmetric-key encipherment:** In asymmetric-key encipherment, a key pair of public and private keys is used. Mathematical calculations are conducted to bind the two keys in a certain relation. Any of the keys can be used for either encryption or decryption and the other is used for the reverse operation. For example, the public key is used to encrypt a message that can be decrypted only by the corresponding private key and vice versa (Khalaf *et al*., 2013). Figure 6 shows the general architecture of asymmetric-key encryption.

## RESULTS AND DISCUSSION

**Decryption:** The encryption is incorrect if we cannot decode and restore the original text. The algorithm may be good but in one direction only. Therefore, decryption is considered a good test to validate the proposed algorithm (Fig. 7). Logically, decryption must start from the last step for encryption. We then go back step by step until we obtain the plaintext. The steps of decryption are similar to the encryption steps, except for two important points. In one of the steps we need to recalculate the key value which is generated by step 6 of encryption. To acquire the key value, we apply the following equation:

$$O_{gn}(\text{Number}) = \text{Current Number} + (W_{div} \times 255)$$

Where:

$O_{gn}$(number) = The (Wsum) or the key before restricting it under the 255 limit

Current number = The number in the last series before this step

$W_{div}$ = It is already saved when we performed encryption; thus, it is necessary to determine the original value before we conduct the "mod" operation by reread step 6 for encryption. Step 7 of encryption is also considered

In another step, we need to recalculate the original ASCII code of the plaintext by applying the following equation:

$$W_{asc} = W_{Sum} - W_{Ln} - W_{or} - W_{seq}$$

Where:

$W_{asc}$ = Original ASCII of plaintext

$W_{sum}$ = DK

$W_{ln}$ = String length

$W_{or}$ = Character order in the string

$W_{seq}$ = String sequence

Many algorithms use number 26 as the basis of the alphabet under the assumption that the number of English characters is 26. Thus, many algorithms use number 92 as the basis of the alphabet under the assumption that the English alphabet contains uppercase, lowercase and numbers. Given that the purpose of encryption is to hide the plaintext as much as possible and confuse attackers as much as possible when the coding space increases, the probabilities and complexity also increase. Therefore, the

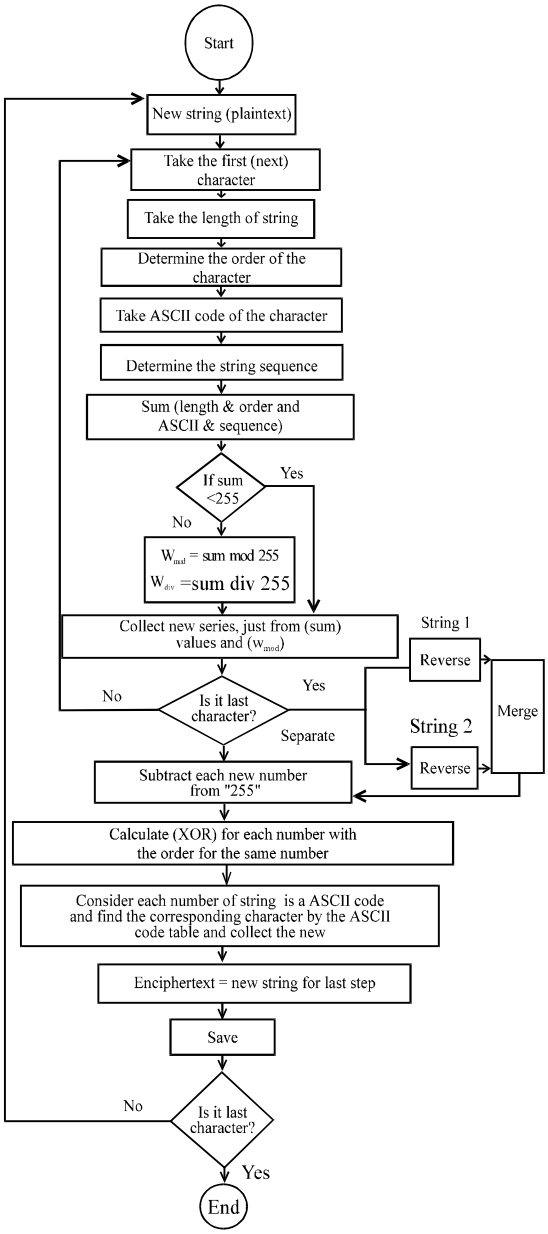


Fig. 7: Steps of the (DKRS) encryption algorithm

proposed algorithm uses number 255 as a basis because it provides all codes that can be printed by the keyboard.

**Encryption:** The proposed algorithm encrypts data through substitution techniques and uses DK and a reverse string. DK is generated by a special equation; it depends on string length, order of the characters in the string, the American Standard Code for Information Interchange (ASCII) code of the character and string sequence. The proposed DK guarantees the following:

- Each string or line has a different encryption
- Each character in the string has a different encryption
- No duplicate exists in the cipher text for the same character in the same plaintext string
- No duplicate exists in the cipher text for the same character in different plaintext strings
- No duplicate exists in the cipher text when a duplicate plaintext occurs

The proposed algorithm is divided into four main phases.

**Phase 1:** The use of key is improved by generating DKs.
**Phase 2:** The new string is divided into two strings which are reversed in two different directions and then remerged later.
**Phase 3:** Each number is subtracted from 255 to generate a new string. Each character is then calculated (XOR) with the order of the same character in the new string.
**Phase 4:** The cipher text of the previous phase is extracted by using substitution techniques.

For example, we apply the proposed algorithm to the text "Universiti Malaysia Pahang" step by step as follows:

**Step 1:** The text to be encrypted is obtained.
**Step 2:** The length of the first string ($W_{Ln}$) is calculated.
Here, the length is 26 with spaces.
**Step 3:** The order of the character in the string ($W_{or}$) is determined to judge if it is changing. The first character in the above example is U, meaning the order becomes 1.
**Step 4:** The ASCII code for the letter ($W_{asc}$) is obtained;
here, the ASCII code for U is 85.
**Step 5:** The string sequence ($W_{seq}$) is considered; here,
each character in the "Universiti Malaysia Pahang" string takes 1 as a sequence.
**Step 6:** The total results ($W_{Sum}$) for steps 2-5 are obtained. The ($W_{Sum}$) value represents a DK.
For the first character in our example (U):
$W_{sum}(U) = W_{Ln}+W_{or}+W_{asc}+W_{seq}$
$W_{sum}(U) = 26+1+85+1$
$W_{sum}(U) = 113$ and so on.
**Step 7:** We restrict the output of the previous step under the 255 limit based on the following comparison. If $W_{Sum}<255$, then we consider ($W_{Sum}$) the value of a new number of new strings. Otherwise, $W_{sum}\geq 255$. The (mod) and (div) for $W_{Sum}$ with 255 are obtained. Thus, the result of the (mod) operation represents the new number of new strings and the result of the (div) operation represents the multiples of 255 when we have a (mod) operation because we need this value to recalculate the plaintext in decryption.
**Steps 3-7:** are repeated to calculate the remaining characters in the current string. For the last example "Universiti Malaysia Pahang", we acquire a new string as follows:
113 139 135 149 133 147 149 140 152 142 70 116 137 149 139 164 159 150 143 79 128 146 154 148 162 156.
The number 26 exists which represents the string length with spaces.
**Step 8:** We divide the new string into two strings, reverse the two strings in two different directions and remerge them later. We divide the strings that include two or more characters because a text with only one character presents no meaning.
Here, we have two cases.
If the length of the string ($W_{Ln}$) is "even," the string is separated as follows:
**String1:** [1] to [($W_{Ln}$/2)],
**String2:** [($W_{Ln}$/2)+1] to [$W_{Ln}$].
If the length of the string ($W_{Ln}$) is "odd," the string is separated as follows:
**String1:** [1] to [integer ($W_{Ln}$/2)+1],
**String2:** [integer ($W_{Ln}$/2)+2] to [$W_{Ln}$].

**Note:** The lengths of the new two strings are not the same when the length of the string (WLn) is "odd."

We acquire two strings as follows:

String1: 113 139 135 149 133 147 149 140 152 142 70 116 137,
String2: 149 139 164 159 150 143 79 128 146 154 148 162 156.

We reverse each of the strings in two different directions. Accordingly, the strings are as follows:

String1: 137 116 70 142 152 140 149 147 133 149 135 139 113.
String2: 156 162 148 154 146 128 79 143 150 159 164 139 149,

We then remerge the new strings as follows:

137 116 70 142 152 140 149 147 133 149 135 139 113 156 162 148 154 146 128 79 143 150 159 164 139 149.

**Step 9:** We subtract each new number from 255. Thus, we obtain the following results.

118 139 185 113 103 115 106 108 122 106 120
116 142 99 93 107 101 109 127 176 112 105
96 91 116 106

**Step 10:** We perform (XOR) for each number with the order for the same number as follows:

New value = [Number (XOR) i].

I: the current order of the number. We then achieve the series

119 137 186 117 98 117 109 100 115 96 115
120 131 109 82 123 116 127 108 164 101 127
119 67 109 112.

**Step 11:** We convert each number for step 10 outputs into a character by the ASCII table under the assumption that the numbers represent the ASCII code and the new character denotes a ciphertext.

$C_f$ = Char (number)

$C_f$: the new character or (ciphertext)

$C_{har}$: the function to generate a character by ASCII number: the outputs for step 9

For our example,

$C_f$ (118) = Char (119) = w

$C_f$ (139) = Char (137) = ‰

$C_f$ (185) = Char (186) = ° and so on

According to the proposed algorithm when the plaintext is

**Universiti Malaysia Pahang**,

the cipher text is

w‰ ubumd's sxfmR tIew Cmp

**Note:** Special and unclear characters exist because the proposed algorithm is based on 255.

**Step 12:** Step 1 is repeated for the next string. The string sequence ($W_{seq}$) increases by 1.

## CONCLUSION

Data security is important, considering that the Internet has become one of the requirements of daily life. Encryption is one of the important solutions to protect data. In this study, a novel encryption algorithm for data protection using DK and reverse string was proposed. The algorithm was validated and the results were discussed. This algorithm can be applied to any form of text or any string whether in a database, network (wired, wireless) cloud and secondary storage. DK was generated by using a special equation that depends on string length, order of the character in the string, ASCII code of the character and string sequence. The text encrypted with the proposed method is complex and can be used for sensitive data because it cannot be decrypted by traditional crypto-analysis tools.

## REFERENCES


Arai, K., S. Kapoor and R. Bhatia, 2014. Intelligent System in Science and Information. Springer, Berlin, Germany, ISBN:978-3-319-14653-9, Pages: 414.

Kester, Q.A., 2012. A cryptosystem based on Vigenere cipher with varying key. Intl. J. Adv. Res. Comput. Eng. Technol., 1: 108-113.

Kester, Q.A., 2013. A hybrid cryptosystem based on Vigenere cipher and columnar transposition cipher. Intl. J. Adv. Technol. Eng. Res., 3: 141-147.

Khalaf, E.T., N. Sulaiman and M.N. Mohammad, 2013. Anti-forensic steganography method based on randomization. Global J. Technol., 4: 447-453.

Mousa, A., O.S. Faragallah, E.S. Rabaie and E.M. Nigm, 2013. Security analysis of reverse encryption algorithm for databases. Intl. J. Comput. Appl., 66: 19-27.

Naji, M.A., 2014. Implementation of encryption data table by using multi-keys. Intl. J. Soft Comput. Eng., 4: 14-17.

Ramaraj, E., 2012. A novel encryption approach in database security. Intl. J. Comput. Organ. Trends, 2: 16-20.

Schneider, F.B., 2000. Enforceable security policies. ACM Trans. Inform. Syst. Secur., 3: 30-50.

Sharma, R., R. Sharma and H. Singh, 2012. Classical encryption techniques. Intl. J. Comput. Technol., 3: 84-90.

Stallings, W., 2005. Cryptography and Network Security Principles and Practices. 4th Edn., Prentice Hall, USA.