جامعة أم القرى
UMM AL-QURA UNIVERSITY

REPORT

Implementing a Basic Cryptosystem

Applied Cryptography

Created By:

Rahaf Saud Alhazmi	**444005613**
Jood Alharbi	**444001600**
Ghada Alhajjaji	**444001531**
Raghad kadem	**443009968**

Supervised By:

Dr. SHOROUQ Alansari

2023 - 2024

1. Introduction:

A simple fast and efficient cryptographic system, it is considered a stream substitution using text based key

Cryptosystem:

Key

Key generating :

- The key starts with any word of your choice
- Using our key generation method/algorithm to fill key to fit length of plain text
- shift the letters of the original key (like Caesar cipher) and add to original key Cat + dbu New key catdbu
- repeat the shifting and adding process till the key reach's length of plain text **but** every time we repeat the process the shift +1
- (the first occurrence (the original key) would shift by 0, the second occurrence shift by 1, the third occurrence shift by 2, and so on.)

Key characteristics:

- Keyspace : the number of possible combinations for each position in the key 26^n
- 26 letters in the English language,
- n representing the length of plain text

ex: there are 26^10 different key combinations for a plain text consisting of 10 letters

- Modular Arithmetic: shifting wraps around the alphabet cipher letter= n mod 26
- Key is a combination of letters and is always the same length of plain text

Char Encryption Method:

- the corresponding letters in the key and plaintext are **added** together computing a sum
- "sum" modules 26
- The results represents the resulting cipher character letter placement in the alphabet

Char Decryption Method:

decryption is the inverse of encryption.

- the key generation process to decrypt is the same.
- the corresponding letters in the key and plaintext are **subtracted** computing a sum.
- "sum" modules 26
- The results represents the resulting cipher character letter placement in the alphabet

2. Explanation:

- EX: for the encryption
- Plaintext (cyber) (c=3,y=25,b=2,e=5,r=18)
- Key (catdb) (c=3,a=1,t=20,d=4,b=2)
- Sum (3+3,25+1,2+20,5+4,18+2)
- 6 mod 26 =6 , 26 mod 26 =0+26, 22 mod 26 =22, 9 mod 26 =9, 20 mod 26 =20

6=F,26=Z,22=V,9=I,20=T

- Cipher text (FZVIT)

3. Features:

- No letter frequency : A single letter can have multiple different encryptions
- Dynamic Key Length Adjustment: If the key is shorter than the plaintext or ciphertext, the program dynamically extends the key to match the length of the input.
- Modulo Operation: The program uses modulo operations to handle character positions and ensure that the result remains within the valid range of alphabet characters.
- Simple and small algorithm
- Fast and efficient
- Low error

4. vulnerabilities and issues:

- Limited Key Space: The key space is limited to letters. In a secure encryption scheme, a larger key space would preferable for resistance against brute-force attacks adding numbers would be better

- Low diffusion

- Vulnerable against insertion attack

5. potential threats:

- Brute-Force Attacks: The limited key space and lack of key strengthening mechanisms make the code susceptible to brute-force attacks. An attacker might attempt to guess the key by systematically trying all possible combinations.
- Insertion attack : retransmission of the plan text with a chosen byte inserted by attacker using the same key stream
- Social engineering : if attacker known the key it easy to generating key
- Side channel attack : exploit weaknesses in their implementation

(Pseudo code for encrypt)

```
function encryption():
    input plaintext
    input key
    cipher = ""
    x = length(plaintext) - length(key)
    if length(key) < length(plaintext):
        for i in range(x):
            char1 = key.charAt(i % 26)
            num1 = char1 - 'a' + 1
            encryptedChar = (num1 % 26 + 'a')
            key = key + encryptedChar
maxLength = max(length(plaintext), length(key))
    p = 0
        for i in range(maxLength):
        char1 = plaintext.charAt(i)
        if char1 == ' ':
            append " " to cipher
            continue
        char2 = key.charAt(p)
        num1 = char1 - 'a' + 1
        num2 = char2 - 'a' + 1
          sum = (num1 + num2) % 26
        if sum == 0:
            sum = sum + 26
        encryptedChar = (sum - 1 + 'a')
        append encryptedChar to cipher
        p = p + 1
    output cipher
```

(Pseudo code for decrypt)

```
function decryption():
    input ciphertext
    input key
    plaintext = ""
    x = length(ciphertext) - length(key)
    if length(key) < length(ciphertext):
        for i in range(x):
            char1 = key.charAt(i % 26)
            num1 = char1 - 'a' + 1
            encryptedChar = (num1 % 26 + 'a')
            key = key + encryptedChar
    p = 0
    for i in range(length(ciphertext)):
        resultChar = ciphertext.charAt(i)
        if resultChar == ' ':
            append " " to plaintext
            continue
        char2 = key.charAt(p)
        num1 = resultChar - 'a' + 1
        num2 = char2 - 'a' + 1
        diff = (num1 - num2) % 26
        if diff <= 0:
            diff += 26
        encryptedChar = (diff - 1 + 'a')
        append encryptedChar to plaintext
        p = p + 1
    output plaintext
```

Summary :

An cryptosystem based on stream cipher, simple and effective, with no frequency of letters, and based on fast and simple mathematical operations it has key space 26^n.

Reference :

Paar book

Work distribution:

Task	Rahaf	Ghada	Jood	Raghad
Encryption Algorithm	●	●	●	●
Decryption Algorithm	●	●	●	
Document	●	●	●	
Presentation		●	●	●