Cryptanalysis of Book-Caesar-RSA Cipher

Samanvitha Basole and Shubham Pachpute

CS 265 Cryptography and Computer Security

San Jose State University

March 20, 2018

1 Problem Statement

From the Mystery Twister website, we attempt to solve the BCR code challenge. In this challenge, we help Alice and Bob find the treasure. Given the book text:
$'SIERRA - ZERO - JULIET - SIX - YANKEE - ONE - ROMEO - PAPPAEIGHT$
$- KILO - FIVE - UNIFORM - XRAY - XXX - BRAVO - VICTORTWO - FOUR - T$
$ANGO - MIKE - OSCAR - HOTEL - DELTA - QUEBECKFOXTROT - ALPHA$
$- YYY - LIMA - INDIA - THREE - WHISKEYNOVEMBER - ECHO$
$- CHARLIE - GOLF - ZULU'$,
our task is to find the two words marked as XXX and YYY then use Book, Caesar, RSA ciphers to get the final output.

2 Approach

In this paper, we solve the BCR cipher consisting of a three-stage cascade: Altered book, Caesar, and RSA. Each stage uses the output of the previous one as input as shown in Figure 1. We discuss the high-level algorithm design of Book, Caesar, and RSA ciphers, then we present the algorithm pseudocode and part of the implementation. Next, we explain our decryption process and analysis achieved through computational experiments. Finally, we provide screenshots of our solution and submission.

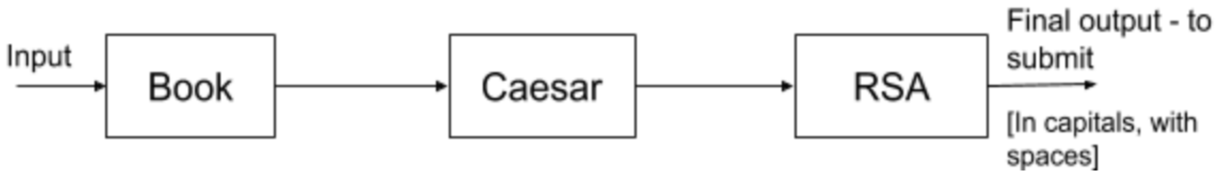

Figure 1: General Architecture

3 Algorithms Discussion

3.1 Book Cipher

Book cipher uses letters of subsequent words in a text to encode messages. Essentially, these words form the key. [example/description needed]

3.2 Caesar Cipher

Caesar cipher is a simple substitution cipher in which the alphabet is shifted a certain number of characters based on the key. It is a symmetric key algorithm, that is, the same key is used for encryption and decryption [1].
We show the following example to demonstrate encryption and decryption using Caesar cipher: Given a Caesar cipher with a left shift of 3, we first convert letters to numbers:

Letters:	A	B	C	D	E	F	...	X	Y	Z
Numbers:	0	1	2	3	4	5	...	23	24	25

To encrypt the plaintext 'ATTACK AT DAWN', we

- Transfom each letter in the plaintext into a number using the above scheme
- Apply the rule: (number + key) mod 26 to each transformed number
- Tranform each number back to its corresponding letter using the same scheme above

The result is the following ciphertext XQQXZH XQ AXTK.
Decryption is done in a similar way except the following rule is applied in the second step: (number - key) mod 26.

3.3 RSA

RSA is an algorithm that provides digital signature as well as encryption. RSA is based on factoring, that is, it is easy to factor two numbers but it is computationally NP-C to find two exact factors for a given number. This is a public key cryptosystem, and thus, the encryption key and the decryption key are different. [3].
The RSA algorithm is as follows [2]:

- Select p, q two large prime numbers
- Calculate $N = pq$
- Select integer e relatively prime to $(p-1)(q-1)$
- Calculate d as the multiplicative inverse of e modulo $(p-1)(q-1)$
- Public key (n, e)
- Private key d
- Consider plaintext M
- Convert plaintext to ciphertext by $C = M^e mod\ N$
- Convert ciphertext to plaintext by $M = C^d mod\ N$

4 Algorithm Design

4.1 Book

A part of the book cipher algorithm is listed below:

```
xxx1 = "7"
yyy1 = "9"
for j in range(0, len(decodedText1)):
    # print "IN FOR 1\n"
    if decodedText1[i] == "XXX":
        decodedText1[i] = xxx1
    if decodedText1[i] == "YYY":
        decodedText1[i] = yyy1
```

4.2 Caesar

The hex input consists of letters and numbers while the key consists of numbers as offset. The below code demonstrates the main logic of the algorithm:

```
output_str = ""
for eachChar in hex_string:
    try:
      # if eachChar is a number
        digit = int(eachChar)
        output_str += str((digit - int(YY)) % 10)
    except:
      # if eachChar is a letter
        output_str += letters[(letters.index(eachChar) + int(XX)) % 26]
```

4.3 RSA

5 Decryption and Analysis

...

6 Solution Submission

...

References

[1] Bruce Busta. Encryption in theory and practice. *The CPA Journal*, 72(11):42, 2002.

[2] S. A. Jaju and S. S. Chowhan. A modified rsa algorithm to enhance security for digital signature. In *2015 International Conference and Workshop on Computing and Communication (IEMCON)*, pages 1–5, Oct 2015.

[3] Karthik, Chinnasamy, and Deepalakshmi. Hybrid cryptographic technique using otp:rsa. In *2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, pages 1–4, March 2017.