Implementation Of The Rivest Cipher 4 Method Web-Based Employee Data

Dwi Kuswanto 1, Ach. Khozaimi 2, Deden Nur Eka Abdi 3
1,2,3 Universitas Trunojoyo Madura, Bangkalan, Indonesia

Corresponding email: dwi.kuswanto@trunojoyo.ac.id

Abstract. Data security has always been an exciting topic to discuss in the era of globalization and industrial revolution 5.0. Along with the development of Cryptographic techniques, which continue to develop. Security of data storage techniques is essential in a data security system. The method used in this research is the Rivest Chiper 4 Algorithm. This research takes a case study of employee data storage security in a company that uses local storage as a medium for storing employee data. A web-based employee data collection system with the Rivest Chiper 4 algorithm implemented on the server side. The research results show that in testing the Avalanche Effect Rivest Chiper 4 algorithm with three key character length variations, an average value of 50.87% was obtained. The test results show that the average Avalanche Effect value is more than 50%, indicating that small changes to the plaintext can impact the ciphertext. With the help of Cryptool 1, using Brute Force time testing results with variations in key length, the password cracking time was 33 years with a 6-character key length. The longer the key is used, the longer the completion process will take to crack the ciphertext. Meanwhile, plaintext length is linearly correlated with the length of Brute Force testing time but is insignificant. Hardware performance also affects the estimated time of Brute Force.

Keywords. Rivest Chiper 4; kriptografi; Avalanche Effect; Brute Force.

INTRODUCTION

Data security has always been an exciting topic to discuss in the era of globalization and Industrial Revolution 5.0. Along with the development of cryptographic techniques, which continue to develop. The security of data storage techniques is essential in a data security system. The method used in this research is the Rivest Chiper 4 algorithm. This research takes the form of a case study of employee data storage security in a company that uses local storage as a medium for storing employee data. A web-based employee data collection system with the Rivest Chiper 4 algorithm implemented on the server side.

This research implemented the RC4 encryption algorithm in the company website system, which runs on the server side. In addition, the quality of encryption was analysed using the Avalanche Effect method and a brute force attack to measure the estimated time needed to crack all possible encryption keys.

This research aims to implement the RC4 encryption algorithm method in a web-based employee data collection system and test the quality of encryption using the Avalanche Effect method and the estimated time required for the RC4 algorithm with a brute force attack experiment. Implementing the RC4 algorithm on the company's website system is hoped to protect employee data and minimize the risk of leaking sensitive information. Implementing
the RC4 algorithm is also expected to increase efficiency and security in the company's employee data collection process.

**METHOD**

In order to assess the web application system under development, there are two different kinds of requirements: functional and non-functional. *Functional requirements* are requirements that contain the processes carried out by the system. Non-functional requirements are requirements that focus on the operating characteristics of the system.

- **Functional Requirements**
  - The user inputs data as a string that will be encrypted and decrypted.
  - The user presses the input button, which will immediately encrypt the input form.
  - The server can carry out the encryption process on data that the user has entered.
  - The server can carry out the decryption process from the database to the interface without destroying the data.

- **Non-functional Requirements**
  - An analysis of non-functional requirements is needed to support system development with the minimum specifications required to work and run well.

**System planning**

System design explains how the author designs a system. Figure 1 explains the system flowchart as follows:

1. Admin starts accessing the company website.
2. Admin logs in
3. Admin carries out the process of adding employee data
4. Admin fills in the employee data form
5. If the data is valid, the program completes the submission process. If the data is invalid, then repeat the form-filling process.
6. Data is entered into the database
7. Admin can review data that has been encrypted and decrypted
8. Done

![Flowchart Sistem](image-url)
• System Architecture

The following is the website application system architecture by implementing the RC4 cryptographic algorithm. Figure 2 displays the architectural layout.

![System Architecture Diagram](image)

**Figure 2: System Architecture**

In Figure 2, when a user accesses a website application and processes CRUD data, the system will read the data as plaintext and carry out a cryptographic algorithm process using the RC4 method. The ciphertext will be kept in the website database after the encryption process. The decryption process is the process of calling up data from the database by the user. The user calls data from the database, and the system will carry out the decryption process from ciphertext to plaintext so that the user can easily read the data. This study handles the encryption and decryption procedure on the server side.

After sending data via the HTTP protocol, the server will capture the data; then, the server carries out an encryption process before being entered into the database.

• Field Encryption

The following database table shows the field columns used to encrypt important employee data.

![Field Database Table](image)

**Figure 3: Field Database**

In Figure 3 it can be seen that the fields in the encryption process are:

- a. Alamat
- b. Nik_ta
- c. No_kk
- d. No_ktp
- e. No_hp
- f. No_keluarga
- g. Email
### Desain Application

The design of this application is made as simple and efficient as possible to make it easier for users to operate it. The following is the form and login design for users to fill in their data:

![Image of Form Data Pegawai](image1.png)

**Gambar 4: Form Data Pegawai**

### Testing Scenarios

After designing the system, it continues with the testing process for the RC4 encryption algorithm. The following is a research and testing scenario:

1. The testing the quality of the RC4 encryption method by calculating the Avalanche Effect Value.

![Image of Flowchart Testing Avalanche Effect](image2.png)

**Gambar 6: Flowchart Testing Avalanche Effect**
After designing the system, it continues with the testing process for the RC4 encryption algorithm. The following is a research testing scenario:

Based on Figure 6, the Flowchart flow for testing the Avalanche effect can be explained as follows:

a. Employees enter plaintext employee data  
b. Enter the key for encryption process  
c. The system carries out the encryption process  
d. The encryption process produces ciphertext  
e. Test the Avalanche Effect by entering plaintext and ciphertext first  
f. Comparing bit changes in ciphertext and plaintext  
g. Calculating bit changes with the Avalanche effect formula  
h. Get the Avalanche Effect percentage

\[
\text{Avalanche Effect} = \frac{\sum \text{Perubahan bit}}{\sum \text{seluruh bit ciphertext}} \times 100\%
\]

2. Testing RC4 encryption using Brute Force Attack

Based on Figure 7, the Flowchart flow for testing Brute Force encryption for RC4 can be explained as follows:

a. Employees enter plaintext employee data  
b. Enter the key for the encryption process  
c. Carry out the encryption process  
d. Get the ciphertext  
e. Using key constraints  
f. Enter the Constraint key  
g. Carry out the RC4 encryption brute force process  
h. Reads brute force results of RC4 encryption  
i. Sort the results with the smallest entropy value  
j. Choose brute force results from the smallest entropy value.

**RESULT AND DISCUSSION**

The user interface display of the employee website includes the login page, employee data input and employee details.
The employee detail display is the result of RC4 cryptographic decryption.

### Implementation of RC4 on the Website

There are several stages to implementing RC4 on a website. The first stage is creating the UI and Database. The author uses Native UI with Bootstrap and uses a MySQL database. RC4 decryption has the same algorithm as encryption because the nature of the RC4 encryption algorithm is symmetric. RC4 decryption uses the encryption function and ciphertext, which will be XORed with a keystream with the same key.

First, capture the array from the POST form. After capturing the value, several variables will be decrypted. The $ SQL function will insert variables in the employee table with columns that correspond to the variables in displaying data by carrying out the decryption process. The function of HEX2BIN is to convert hexadecimal numbers to binary numbers.
Website Database

Data that has undergone an encryption process on the server side will be stored in the website database. Data that has undergone an encryption process will become a hexadecimal number. The ciphertext is then stored in the appropriate field.

Figure 11: Hasil enkripsi pada database

Avalanche Effect Test Results

Test results were obtained from 3 variations of key characters (5, 9 and 16, respectively) with the same plaintext and key. Plaintext is obtained from the names of employees at the company. The test scheme compares the number of plaintext bit changes with the total amount of plaintext multiplied by 100%.

Table 1. Avalanche effect test results with 5 key characters

<table>
<thead>
<tr>
<th>Name</th>
<th>Key</th>
<th>Ciphertext</th>
<th>Plaintext</th>
<th>Bit change</th>
<th>Rate %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Ahmad Faiz</td>
<td>12345</td>
<td>0101010101</td>
<td>101010101</td>
<td>101010101</td>
<td>101010101</td>
</tr>
<tr>
<td>2 Muhammad Ramadon</td>
<td>12345</td>
<td>0101010101</td>
<td>101010101</td>
<td>101010101</td>
<td>101010101</td>
</tr>
<tr>
<td>3 Almuhendi Sinarutik</td>
<td>12345</td>
<td>0101010101</td>
<td>101010101</td>
<td>101010101</td>
<td>101010101</td>
</tr>
<tr>
<td>4 Bagus Kurniady Wibowo</td>
<td>12345</td>
<td>0101010101</td>
<td>101010101</td>
<td>101010101</td>
<td>101010101</td>
</tr>
</tbody>
</table>
Table 2. Avalanche effect test results with 9 key characters

<table>
<thead>
<tr>
<th>Name</th>
<th>Key Characters</th>
<th>Ciphertext</th>
<th>Hexa Plaintext</th>
<th>Hexa Cipher</th>
<th>Bit yang berubah</th>
<th>Total Hit</th>
<th>Avalanche effect (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADIYANTO</td>
<td>T, T, N, O, E, Y, N, E, Y</td>
<td>00000101 01000010 01000010 00000010 01000010 01000010 00000010 01010010 01000010</td>
<td>01010010 01000010 01000010 00000010 01000010 01000010 00000010 01010010 01000010</td>
<td>94</td>
<td>176</td>
<td>33.4%</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Avalanche effect test results with 16 key characters

<table>
<thead>
<tr>
<th>Name</th>
<th>Key Characters</th>
<th>Ciphertext</th>
<th>Hexa Plaintext</th>
<th>Hexa Cipher</th>
<th>Bit yang berubah</th>
<th>Total Hit</th>
<th>Avalanche effect (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOHAMMAD KARIM</td>
<td>N, N, Y, N, T, N, O, E, Y, N, E, Y</td>
<td>00000101 01000010 01000010 00000010 01000010 01000010 00000010 01010010 01000010</td>
<td>01010010 01000010 01000010 00000010 01000010 01000010 00000010 01010010 01000010</td>
<td>16</td>
<td>136</td>
<td>55.1%</td>
<td></td>
</tr>
</tbody>
</table>

The table above makes it possible to compute the average percentage of the Avalanche Effect:

\[
\text{Rata} - \text{rata Avalanche Effect} = \frac{47.15 + 53.62 + 51.85}{3} \times 100% = \frac{152.62}{3} \times 100% = 50.87% \\

The average results of the Avalanche Effect test showed that it was 50.87%, indicating that the average Avalanche Effect value was more than 50%. This indicates that small changes to the plaintext can impact the ciphertext.
■ Brute Force Results

In Brute force testing, the RC4 encryption algorithm uses CrypTool 1 software for testing in brute force attacks. The table above shows that in the Brute Force process in the RC4 encryption algorithm, the key length greatly influences the length of the attack process. With the plaintext "Deden Nur Eka Abdi" and with key lengths "3, 4, 5, and 6", the resulting Brute Force processing time data is 56 seconds each for key length 3, 4 hours for key length 4, 40 days for key length 5, and 33 years for key length 6. The length of time in the Brute Force process results from the estimated system completing the experiment for all possibilities. Password cracking time refers to the permutation of the key length. The longer the key is used, the longer the completion process will take to crack the ciphertext using the Brute Force method.

Table 4: Results of Brute Force time testing compared to a critical length.

<table>
<thead>
<tr>
<th>No</th>
<th>Plaintext (Nama)</th>
<th>Kunci</th>
<th>Panjang Kunci</th>
<th>Waktu Bruteforce</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Deden Nur Eka</td>
<td>123</td>
<td>3</td>
<td>56 detik</td>
</tr>
<tr>
<td>2</td>
<td>Abdi</td>
<td>1927</td>
<td>4</td>
<td>4 jam</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Setro</td>
<td>5</td>
<td>40 hari</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>madiun</td>
<td>6</td>
<td>33 tahun</td>
</tr>
</tbody>
</table>

Table 5: Brute Force testing with 3 key bits

<table>
<thead>
<tr>
<th>No</th>
<th>Jumlah Karakter</th>
<th>Waktu Bruteforce</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>59 detik</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>1 menit 4 detik</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>1 menit 5 detik</td>
</tr>
<tr>
<td>4</td>
<td>400</td>
<td>1 menit 9 detik</td>
</tr>
<tr>
<td>5</td>
<td>500</td>
<td>1 menit 10 detik</td>
</tr>
</tbody>
</table>

Table 6: Brute Force testing with 4 key bits

<table>
<thead>
<tr>
<th>No</th>
<th>Jumlah Karakter</th>
<th>Waktu Bruteforce</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>4 jam 16 menit</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>4 jam 18 menit</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>4 jam 20 menit</td>
</tr>
<tr>
<td>4</td>
<td>400</td>
<td>4 jam 21 menit</td>
</tr>
<tr>
<td>5</td>
<td>500</td>
<td>4 jam 25 menit</td>
</tr>
</tbody>
</table>

Table 7: Brute Force testing with 5 key bits

<table>
<thead>
<tr>
<th>No</th>
<th>Jumlah Karakter</th>
<th>Waktu Bruteforce</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>46 hari 14 jam</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>47 hari 5 jam</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>47 hari 14 jam</td>
</tr>
<tr>
<td>4</td>
<td>400</td>
<td>47 hari 21 jam</td>
</tr>
<tr>
<td>5</td>
<td>500</td>
<td>47 hari 22 jam</td>
</tr>
</tbody>
</table>
Plaintext length can affect Brute Force testing duration, but not very much, according to the Brute Force testing chart with plaintext character length. Hardware performance also affects the estimated time for the Brute Force to complete the attack process.

CONCLUSION
From the results of several research trials that have been carried out, it can be concluded.
1. The Avalanche Effect method in testing carried out with crucial character lengths of 5, 9 and 16, respectively, obtained an average Avalanche Effect of 50.87%. This shows that the average Avalanche Effect value is more than 50%, indicating that small changes to the plaintext can impact the ciphertext.
2. With the help of Cryptool 1 for Brute Force time testing results with variations in key length, it was obtained that the password cracking time was 33 years with a 6-character key length. The longer the key is used, the longer the completion process will take to crack the ciphertext. Meanwhile, plaintext length is linearly correlated with the length of Brute Force testing time but is not significant. Hardware performance also affects the estimated time of Brute Force.

REFERENCES