Shalala Cipher, a New Implementation of Vigenere Cipher for Wireless Sensor Node Security

Muhammad Shaiful Azrin Md Alimon¹, L.M. Kamarudin¹,², Azizi Harun¹, Ammar Zakaria¹ and Shaufikah Shukri¹

¹Center of Excellence in Advanced Sensor Technology, University Malaysia Perlis, Malaysia
²School of Computer & Communication Engineering, University Malaysia Perlis, Malaysia

Keywords: 8-bit Cipher, Cryptography, WSN Security, Vigenere Cipher, Shalala.

Abstract: Cryptography is a science that deals on the method of converting plaintext into cipher text, usually with the help of encryption keys and encryption algorithm. Current standard implementation of cryptography is proved to require high resource in memory which is not suitable to be implemented in low memory embedded system, thus a lightweight cryptography are required. Vigenere cipher is one of the encryption algorithms that was easy to implement and comprehend, which can be used to provide confidentiality from the third party. Vigenere cipher is a polyalphabetic Caesar cipher, which means it shifting the character in plaintext to become character of mode of its key character. Even though Vigenere cipher are a classical cipher that doing its cipher using character compared to modern cipher that doing its cipher using bits and byte, it still can be implemented in modern computer by representing its character based on ASCII Table. Furthermore, because of this, its character can be extended to contain all alphabet and special character in the ASCII Table, thus allowing it to encrypt every character that can be represented using ASCII Table. Originally the resilience of Vigenere cipher is based on two factors: (1) the length of its key, and (2) the randomness of character in its key. Even though increasing the length and the used random character provide a great resilience, it takes away the fun of using dictionary word to encrypt the plaintext. To overcome that, this paper, proposing a method of implementing a Pseudorandom Path that change the flow of mode use by the Vigenere Cipher to encrypt and decrypt either positive mod or negative mod. Usually Vigenere Cipher use positive mode to encrypt, while negative mod to decrypt, however this paper propose a method to combine both mod during encryption or decryption with the supervision from the Pseudorandom Path. This method of implementation and its result were discussed in this paper and named as Shalala Cryptography. The results were compared between using original resilience and a Shalala Cryptography method. The resource requirement to implement this algorithm using C++ language is also shown, which shows a lightweight cryptography scheme in term of RAM consumption and fast processed time, which are suitable to be used in WSN or IoT environment.

1 INTRODUCTION

Cryptography, a Greek word for secret writing, it’s an art of applied mathematics and science used to hide plaintext to become cipher text. Cryptography provides confidentiality between two parties, provided that both parties use the same method and know the key, while securing the message during massage delivery from getting known by the third party. Thus making cryptography is one of the essential ingredients in secure network communication (William Stalling, 2004).

There are two types of encryption key:-
1. Symmetric encryption – both parties use the same key for encryption and decryption, and
2. Asymmetric encryption – where both parties used private key and public key for encryption and decryption respectively.

In this paper, Vigenere cipher is used as the cryptographic algorithm, which is a symmetric encryption that used the same key for both parties. It is considered as le chiffre indechiffrable (French for 'the indecipherable cipher'), because for 300 years it

---

Alimon, M., Kamarudin, L., Harun, A., Zakaria, A. and Shukri, S.
DOI: 10.5220/0006358403630369
Copyright © 2017 by SCITEPRESS – Science and Technology Publications, Ltd. All rights reserved
cannot be cracked, not until it is cracked by Friedrich Kasiski during 1863 using a frequency analysis. Even though by increasing the length of its key as long as the plaintext and using a random character as its key can create an uncrackable cipher, it takes away the fun of using Vigenere cipher, which was to use dictionary words as its keyword. Nowadays, this cracking process is much easier to implement, by writing a small call that automatically brute forcing the cipher text using all the words in dictionary and this kind of attack were known as “Dictionary Attack”.

To enhance this Vigenere cipher, while maintaining the fun, this paper proposes Shalala Cryptography, a method that combines both positive mod and negative mod of character shifting with a supervision of Pseudorandom Path as shown in section 2, and the result of its implementation can be seen in section 3, followed with section 4 that shows the resilience test, comparing the original Vigenere Cipher with the Shalala Cipher. Last, is section 5, shows the resource requirement and the example of its application.

2 PROPOSED METHOD

It is good to note that Vigenere Cipher used in this paper is using an extended Vigenere Cipher that contain every character that can be represented by ASCII Table, thus creating a mod96 instead of mod26 as in Original Vigenere Cipher. Diagram 1 shows the implementation diagram of the original Vigenere Cipher Algorithm using mod96 during encryption and decryption while Diagram 2 shows the algorithm of a Shalala Cryptography method using the same mod.

Diagram 2: Proposed Method Algorithm to Enhance Vigenere Cipher Resilience.

Algorithm 1: Encryption Algorithm

1. **Take** Character at Nth position of the plaintext and key
2. **Take** Character at 1th position of the key and take the 1st bit
3. **Then** Masking with 1st bit of Masking Character
4. **If** result of Masking Block is ‘1’
5. **Do** Forward the Nth position of the plaintext using Positive Mod
6. **Encrypt** using Positive Mod
7. **Forward** result as Nth position encrypted
8. **Else If** result of Masking Block is ‘0’
9. **Do** Forward the Nth position of the plaintext using Negative Mod
10. **Encrypt** using Negative Mod
11. **Forward** result as Nth position encrypted

Diagram 1: Original Vigenere Cipher Algorithm, Left for Encryption and Right for Decryption.
Increment increasing the Masking bit and Nth character key’s bit followed with next character
If the entire bit of Nth position of the key were used
Do
Take the next position of the key and take the 1st bit
Then Masking with 1st bit of Masking Character

While |Ciphertext.length()| < Plaintext.length()
Repeat at number 3

Meanwhile the bottom block is used for decryption, it works by:

Algorithm 2: Decryption Algorithm
1 Take Character at Nth position of the cipher text and key
2 Take Character at 1th position of the key and take the 1st bit Then Masking with 1st bit of Masking Character
3 If result of Masking Block is ‘1’
   Do
      Forward the Nth position of the cipher text using Negative Mod
      Decrypt using Negative Mod
      Forward result as Nth position decrypted
   Else If result of Masking Block is ‘0’
      Do
      Forward the Nth position of the plaintext using Positive Mod
      Decrypt using Positive Mod
      Forward result as Nth position decrypted
      Increment
      increasing the Masking bit and Nth character key’s bit followed with next character
   If the entire bit of Nth position of the key were used
      Do
         Take the next position of the key and take the 1st bit
         Then Masking with 1st bit of Masking Character
   While |Plaintext.length()| < Ciphertext.length()

This implementation can also be represented in algebraic form shown in equ.1 and equ.2 for original Vigenere using mod96 and equ.3 and equ.4 for the Shalala Cryptography method using the same mod. By representing each character used in mod96 with ASCII Table, those characters can be represented as number from 32 to 126. Taken E as encryption and K as key, the algorithm can be written as follow (S. Garg, 2016).

\[
Ci = EK(Pi) = (Pi + Ki) \mod 96
\]  
(equ.1)

And decryption D using the key K

\[
Pi = DK(Ci) = (Ci - Ki) \mod 96
\]  
(equ.2)

Where

P = P0...Pn is the message,
C = C0...Cn is the ciphertext and
K= K0....Km is the used key.

Representing the proposed method in algebraic form require an if-else conditional statement. Take:

\[
P = P0...Pn is the message,
C = C0...Cn is the ciphertext and
K= K0....Km is the used key.
\]

\[
K_{bit}[n] = K0....Km in 8 bit arrays of its byte representation as in ASCII Table and n is the position of bit
\]

\[
Mask = 8 bit array, predetermined by the user
\]

M() = Masking function that masking K_{bit}[n] with Mask[n] using X-OR logic

The encryption algebraic is like follow:

\[
C_i = EK(P_i) = \\
\{(P_i + K_i) \text{ if } M(K_{bit}[n], mask) = 1 \} \\
\{(P_i - K_i) \text{ if } M(K_{bit}[n], mask) = 0 \} \text{ (equ.3)}
\]

And decryption D

\[
P_i = DK(P_i) = \\
\{(C_i - K_i) \text{ if } M(K_{bit}[n], mask) = 1 \} \\
\{(C_i + K_i) \text{ if } M(K_{bit}[n], mask) = 0 \} \text{ (equ.4)}
\]
3 EXAMPLE AND RESULT OF IMPLEMENTATION

Let’s take an example to understand the Shalala Cryptography algorithm in a better way where its result can be analyzed. This are the parameter used for this example for encryption:

Plaintext: 2016BatmanvsOni#@
Keyword: Bat
Masking key: 10010011

4 ANALYSIS AGAINST THE ADVISORY

It is good to note that the analysis method used in this paper is to compare the cipher text generated between the original Vigenere Cipher and the proposed method against Frequency Analysis and Dictionary Attack.

Table 1: Encryption using Shalala Cryptography Method.

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>ASCII</th>
<th>Keyw</th>
<th>ASCII</th>
<th>Kbit</th>
<th>X-OR</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016Batm</td>
<td>50 48</td>
<td>66 66</td>
<td>119 119 119</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
</tr>
<tr>
<td>anvsO</td>
<td>66 66 97 97 116 116 66 66 97 97 116 66 97 116 66 97 97 66 97</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>2 0 1 6 Batm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ni #</td>
<td>66 66 97 97 116 116 66 66 97 97 116 66 97 116 66 97 97 66 97</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>2 0 1 6 Batm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>@</td>
<td>66 66 97 97 116 116 66 66 97 97 116 66 97 116 66 97 97 66 97</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>2 0 1 6 Batm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Decryption using Shalala Cryptography Method.

<table>
<thead>
<tr>
<th>Cipher Text</th>
<th>ASCII</th>
<th>Keyword</th>
<th>ASCII</th>
<th>Kbit</th>
<th>X-OR</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>TN &lt; s $</td>
<td>84 78 60 115 36 49 55 79</td>
<td>B a t</td>
<td>66 97 116 66 97 97 116 66 97 97 116 66 97 97 66 97</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
</tr>
<tr>
<td>1 7 O l</td>
<td>119 119 119</td>
<td>B a t</td>
<td>66 97 116 66 97 97 116 66 97 97 116 66 97 97 66 97</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
</tr>
<tr>
<td>5 ~ q P ^</td>
<td>74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74</td>
<td>B a t</td>
<td>66 97 116 66 97 97 116 66 97 97 116 66 97 97 66 97</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
</tr>
<tr>
<td>E “ “</td>
<td>74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74</td>
<td>B a t</td>
<td>66 97 116 66 97 97 116 66 97 97 116 66 97 97 66 97</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
</tr>
</tbody>
</table>

Table 3: Encryption using Original Vigenere Cipher.

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>ASCII</th>
<th>Keyword</th>
<th>ASCII</th>
<th>Kbit</th>
<th>X-OR</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016Batm</td>
<td>50 48</td>
<td>B a t</td>
<td>66 97 116 66 66 97 97 116 66 97 97 116 66 97 97 66 97</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
</tr>
<tr>
<td>anvsO</td>
<td>66 97 116 66 66 97 97 116 66 97 97 116 66 97 97 66 97</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ni #</td>
<td>66 97 116 66 66 97 97 116 66 97 97 116 66 97 97 66 97</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>@</td>
<td>66 97 116 66 66 97 97 116 66 97 97 116 66 97 97 66 97</td>
<td>0 1 0 1 1 0 1 1 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td>0 0 1 0 1 1 1 1 0 0 1 1 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4: Decryption using Original Vigenere Cipher.

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>ASCII</th>
<th>Keyword</th>
<th>ASCII</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>T q &amp; X $ V 7 O V 1 X h q P ^ E &quot;</td>
<td>84 113 38 88 36 86 55 79 86 49 88 104 113 80 94 69 34</td>
<td>B a t B a t B a t B a t B a t B a t B a t B a</td>
<td>66 97 116 66 97 116 66 97 116 66 97 116 66 97</td>
<td>2 0 1 6 B a t m a n v s O n i # @</td>
</tr>
</tbody>
</table>

4.1 Frequency Analysis

It is a part of descriptive statistic that measure the frequency of event to occur, in term cryptanalysis however is the study of the frequency of letters or groups of letters to be happen in a cipher text. Based on the example shown in section 3. The process is repeated using much longer sentence as described as follow:


There are total of 25 repeated “Batman”, hence it is expected to reveal some pattern if using original Vigenere cipher. The result of the process between original and Shalala Cryptography can be compared using the distribution diagram produced using both result and were shown in Diagram 4 and 5. Diagram 3 shows the original plaintext character distribution.

Good cryptography is a cryptography that can hide the plaintext character to become more distributed across the character. Diagram 5 have shown that the Shalala Cryptography provide a more distributed character compare to the original.

Diagram 3: Plaintext Character Distribution (6 character).

Diagram 4: Original Vigenere Cipher Character Distribution (16 character).

4.2 Charles Babbage Analyses

Vigenere cipher is a combination of a series of Caesar cipher, if the key used were “Bat” it means that the cipher text is divided into 3 subtexts and each subtexts is cipher using separate Caesar shift, which were in this case it use modB, moda, and modt. According to Charles Babbage, Vigenere cipher can be cracked by:

- Identify the repeated sequence
- Determine the spacing, thus knowing the length of its key
- Either implement frequency analysis or start guessing systematically

Diagram 6 shows the result of cipher text of original Vigenere cipher implementation, the highlighted region shows the repeated sequence and strikethrough shows the spacing.

```
Diagram 6
```

Meanwhile diagram 7 shows the result of cipher text of proposed method, the highlighted region shows the repeated sequence and strikethrough shows the spacing. Places where repeating sequence happened in original Vigenere were not happened in Shalala method, thus proving that the proposed method to be frequency-analysis-proof.

4.3 Dictionary Attack

It is a method to systematically guessing the keyword to beat the decryption algorithm to decrypt the cipher text, such keywords can be every words in dictionary. Based on the Diagram 6, it shows that the length of its keyword were 3 characters because of 3 set of repeating sequence, and according to here[5] there were only 1015 of English words that can be found in dictionary.

The process to implement dictionary attack is to guess a single word from the expected plaintext, let say it is “Batman”. Then a small code can be written that feed those dictionary word and cipher text into the decryption algorithm, upon completed, a quick search on “Batman” is conducted at the resultant plaintext. If the word “Batman” is found the process end, if not it is repeated until all the dictionary word were used. Again if it still fail, change “Batman” to another word, and repeat the process.

Because of Charles Babbage analysis on the Shalala method doesn’t show any sequence, and it is concluded to be frequency-analysis-proof, thus it also concluded that the proposed method have successfully enhance the Vigenere Cipher against Dictionary Attack.

5 RESOURCE REQUIREMENT AND EXAMPLE OF ITS APPLICATION

The Shalala Cryptography has been tested using Arduino Board. It is very lightweight and proved to be high cryptography resilience. Table 1 shows the resource requirement of the algorithm written using C++, and because of this it is very suitable to be implemented in Internet of Things application to provide resilience during communication.

Table 5: Hardware Requirement.

<table>
<thead>
<tr>
<th>Study</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM Consumption (kB)</td>
<td>4,850</td>
</tr>
<tr>
<td>RAM Consumption (kB)</td>
<td>883</td>
</tr>
<tr>
<td>Processing Time (ms)</td>
<td>8</td>
</tr>
</tbody>
</table>

6 CONCLUSION

Vigenere Cipher is a lightweight cryptography implementation that was easy to comprehend and implement. Depending on its implementation it can be uncrack able cipher by making the keyword as long as plaintext and in random sequence, however by doing so, it takes the fun of using dictionary word to encrypt the plaintext. In this paper, a new implementation using Vigenere as its core cryptography with the idea of maintaining the fun of
using the dictionary word has been proposed, the idea is to encrypt or decrypt the text using a combination of positive mod and negative mod of Vigenere character supervised by the Pseudorandom Path. Pseudorandom Path give output ‘0’ or ‘1’ based on the byte represented by the keyword. To further increase the resilience, a Masking Block is implemented on top of the Pseudorandom Path, that X-OR the bit from Pseudorandom Path with the bit from the Masking Block. Example of this implementation can be seen in section 3, and the implementation technique with its diagram is shown in section 2. Depending on the number that were produce by Pseudorandom Path after Masking, the plaintext will follow a different encryption path, 0 is for positive mod and 1 is for negative mod, which were vice versa during decryption process. As a conclusion compared with the original implementation of Vigenere Cipher, this proposed method were frequency proofed and cannot be dictionary attacked even thou it used a dictionary word as it encryption key. Furthermore, because of this algorithm is lightweight, it is very suitable to be implement in motes used in the Internet of Things environment. The good things about it is by changing the masking key, it will change the entire cipher text output, thus if it implement on 8bit devices, it will create a unique 255 difference output, for each sentence.

REFERENCES