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

Muhammad Shaiful Azrin Md Alimon[1], L.M. Kamarudin[1,2], Azizi Harun[1], Ammar Zakaria[1] and Shaufikah Shukri[1]

[1]*Center of Excellence in Advanced Sensor Technology, University Malaysia Perlis, Malaysia*
[2]*School 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 comprehand, which can be used to provide confidentiality from the third party. Vigenere cipher is a polyalphabetic Ceaser cipher, which means it shifting the character in plaintext to become character of mod of its key character. Even thou 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 mod 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 indéchiffrable* (French for 'the indecipherable cipher'), because for 300 years it

Alimon, M., Kamarudin, L., Harun, A., Zakaria, A. and Shukri, S.
Shalala Cipher, a New Implementation of Vigenere Cipher for Wireless Sensor Node Security.
DOI: 10.5220/0006358403630369
In *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017)*, pages 363-369
ISBN: 978-989-758-245-5

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 uncrack able 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

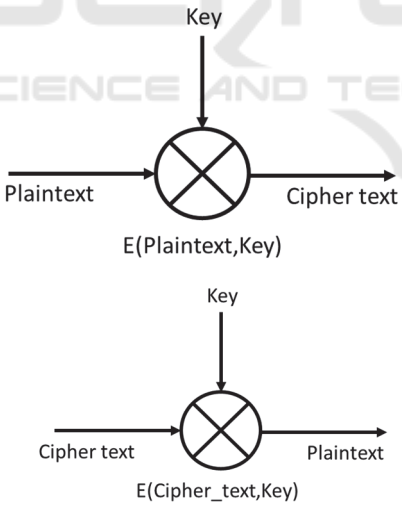

Figure 1: Original Vigenere Cipher Algorithm, Left for Encryption and Right for Decryption.

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.

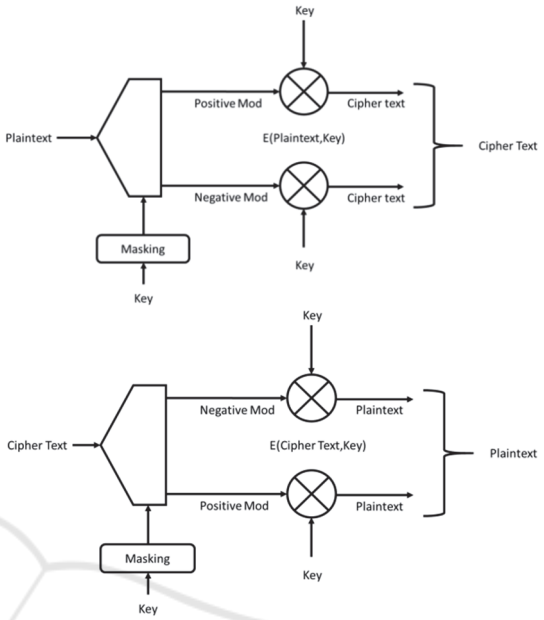

Diagram 2: Proposed Method Algorithm to Enhance Vigenere Cipher Resilience.

Diagram 2 shows the algorithm of a Shalala Cryptography method which is consist of two block diagram, each for encryption and decryption. Top block is used for encryption, it works by:

	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 **Then** Masking with 1st bit of Masking Character
3	**If** result of Masking Block is '1'
4	**Do**
5	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**
10	Forward the Nth position of the plaintext using Negative Mod
11	Encrypt using Negative Mod
12	Forward result as Nth position encrypted

13 **Increment**
14 increasing the Masking bit and N[th] character key's bit followed with next character
15 **If** the entire bit of Nth position of the key were used
16 **Do**
17 **Take** the next position of the key and take the 1[st] bit **Then** Masking with 1[st] bit of Masking Character
18 **While [Ciphertext.length() < Plaintext.length()]**
19 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 1[st] bit **Then** Masking with 1[st] bit of Masking Character
3 **If** result of Masking Block is '1'
4 **Do**
5 Forward the Nth position of the cipher text using Negative Mod
6 Decrypt using Negative Mod
7 Forward result as Nth position decrypted
8 **Else If** result of Masking Block is '0'
9 **Do**
10 Forward the Nth position of the plaintext using Positive Mod
11 Decrypt using Positive Mod
12 Forward result as Nth position decrypted
13 **Increment**
14 increasing the Masking bit and N[th] character key's bit followed with next character
15 **If** the entire bit of Nth position of the key were used
16 **Do**
17 **Take** the next position of the key and take the 1[st] bit **Then** Masking with 1[st] bit of Masking Character
18 **While [Plaintext.length() < Ciphertext.length()]**
19 Repeat at number 3

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 number 32 to 126. Taken E as encryption and K as key, the algorithm can be written as follow (S. Garg, 2016).

$$Ci = EK(Pi) = (Pi + Ki) \bmod 96 \quad \text{(equ.1)}$$

And decryption D using the key K

$$Pi = DK(Ci) = (Ci - Ki) \bmod 96 \quad \text{(equ.2)}$$

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.

Kbit[n] = K0….Km in 8 bit arrays of its byte representational as in ASCII Table and n is the position of bit

Mask = 8 bit array, predetermined by the user

M() = Masking function that masking Kbit[n] with Mask[n] using X-OR logic

The encryption algebraic is like follow:

$$C_i = EK(P_i) = \begin{cases} (P_i + K_i) & if\ M(Kbit[n], mask) = 1 \\ (P_i - K_i) & if\ M(Kbit[n], mask) = 0 \end{cases} \quad \text{(equ.3)}$$

And decryption D

$$P_i = DK(P_i) = \begin{cases} (C_i - K_i) & if\ M(Kbit[n], mask) = 1 \\ (C_i + K_i) & if\ M(Kbit[n], mask) = 0 \end{cases} \quad \text{(equ.4)}$$

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

Table 1: Encryption using Shalala Cryptography Method.

Plaintext	2	0	1	6	B	a	t	m	a	n	v	s	O	n	i	#	@
ASCII Plaintext	50	48	49	54	66	48	116	109	97	110	118	115	79	110	105	35	64
Keyword	B	a	t	B	a	t	B	a	t	B	a	t	B	a	t	B	a
ASCII Keyword	66	97	116	66	97	116	66	97	116	66	97	116	66	97	116	66	97
Kbit	B = dec66 or B = hex42								a = dec97 or a = hex97								hex74
	0	1	0	0	0	0	1	1	0	1	0	1	0	1	1	1	0
Masking	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1
X-OR	0	0	1	0	1	1	1	1	0	0	1	1	1	0	1	1	0
Result	T	N	<	s	$	1	7	O	l	l	5	~	q	P	^	E	“

Table 2: Decryption using Shalala Cryptography Method.

Cipher Text	T	N	<	s	$	1	7	O	l	l	5	~	q	P	^	E	“
ASCII CT	84	78	60	115	36	49	55	79	108	49	53	126	113	80	94	69	34
Keyword	B	a	t	B	a	t	B	a	t	B	a	t	B	a	t	B	a
ASCII Keyword	66	97	116	66	97	116	66	97	116	66	97	116	66	97	116	66	97
Kbit	B = dec66 or B = hex42								a = dec97 or a = hex97								74
	0	1	0	0	0	0	1	1	0	1	0	1	0	1	1	1	0
Masking	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1
X-OR	0	0	1	0	1	1	1	1	0	0	1	1	1	0	1	1	0
Result	2	0	1	6	B	a	t	m	a	n	v	s	O	n	i	#	@

Table 3: Encryption using Original Vigenere Cipher.

Plaintext	2	0	1	6	B	a	t	m	a	n	v	s	O	n	i	#	@
ASCII Plaintext	50	48	49	54	66	48	116	109	97	110	118	115	79	110	105	35	64
Keyword	B	a	t	B	a	t	B	a	t	B	a	t	B	a	t	B	a
ASCII Keyword	66	97	116	66	97	116	66	97	116	66	97	116	66	97	116	66	97
Result	T	q	&	X	$	V	7	O	V	1	X	h	q	P	^	E	“

4 ANALYSIS AGAINST THE ADVISARY

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 4: Decryption using Original Vigenere Cipher.

Plaintext	T	q	&	X	$	V	7	O	V	1	X	h	q	P	^	E	“
ASCII Plaintext	84	113	38	88	36	86	55	79	86	49	88	104	113	80	94	69	34
Keyword	B	a	t	B	a	t	B	a	t	B	a	t	B	a	t	B	a
ASCII Keyword	66	97	116	66	97	116	66	97	116	66	97	116	66	97	116	66	97
Result	2	0	1	6	B	a	t	m	a	n	v	s	O	n	i	#	@

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:

“Batman Batman”

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.

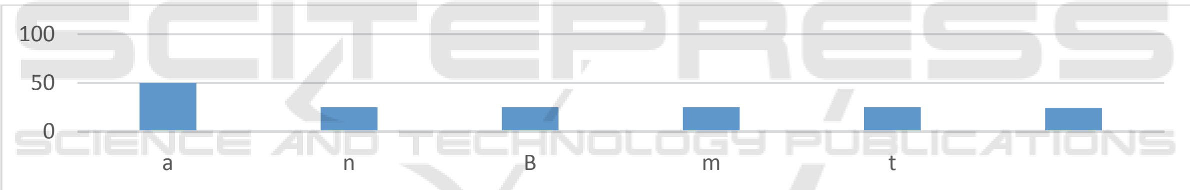

Diagram 3: Plaintext Character Distribution (6 character).

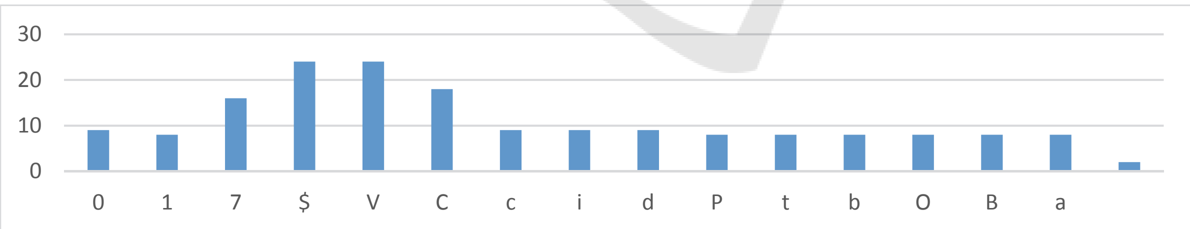

Diagram 4: Original Vigenere Cipher Character Distribution (16 character).

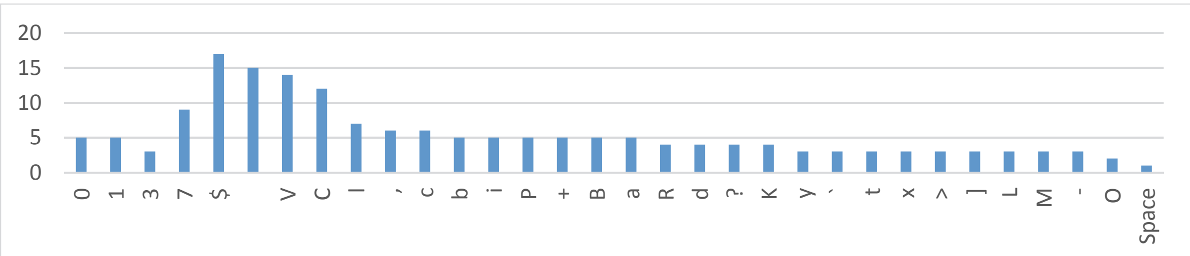

Diagram 5: Proposed Method Character Distribution (32 character).

4.2 Charles Babbage Analyses

Vigenere cipher is a combination of a series of Ceaser 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 Ceaser 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.

```
dCi0CcB$V7OV1a7$Vb$PtdCi0CcB$V7OV1a
7$Vb$PtdCi0CcB$V7OV1a7$Vb$PtdCi0CcB
$V7OV1a7$Vb$PtdCi0CcB$V7OV1a7$Vb$Pt
dCi0CcB$V7OV1a7$Vb$PtdCi0CcB$V7OV1a
7$Vb$PtdCi0CcB$V7OV1a7$Vb$PtdCi0Cc
```

Diagram 6

```
d  KCyB$l7,l1a7$Vb$-
+dCi0Cy]$lROVL>M$3b$-t
i0CcB$VR,V1a7$3x$-+dC K
c]$VROlLa7$Vb$P+
Ci0Cc]`VR,V1>M?Vx$P+d
0CcB$V7,l1a7$Vx?P+ CiK
yB`V7,VL>7$Vb$Pt
i0CcB`l7,l1aM?3b?Pt C KCc
```

Diagram 7

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 Vegenere 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 cypher 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.

Study	Proposed Method
ROM Consumption (kB)	4,850
RAM Consumption (kB)	883
Processing Time (ms)	8

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 out 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

Stallings, William. *Network Security Essentials*. 1st ed. Pearson Education. Print.

S. Garg, et al, "Extended Vigenere Cipher with Stream Cipher". *International Journal of Engineering Science and Computing*, (2016) 5176-5180

Beutelspacher, Albrecht. *Cryptology*. 1st ed. Washington: The Mathematical association of America, 1994. Print.

Analysis, Frequency et al. "Frequency Analysis Tool - Letter Counter - Online Software Tool ★". *Dcode.fr*. N.p., 2017. Web. 15 Feb. 2017.

"Decrypting Text". *Richkni.co.uk*. N.p., 2017. Web. 15 Mar. 2017.