Study on permutation polynomials over a ring of modulo 2^w and their applications to cryptography

Atsushi Iwasaki

February 2017

Contents

Chapter 1

Introduction

1.1 Cryptography

1.1.1 History

Cryptography has been used from the distant past. At least, ancient Egypt used cryptography approximately 4000 years ago [1]. One of the historically most famous cryptosystem is “Caesar cipher” which was developed by Gaius Iulius Caesar in the first century BC. That is one kind of simple substitution cipher and its algorithm is very simple. That shifts alphabet. In the early 20th century, it used code book and cipher text was made by hands, or it was implemented by machines which can be used only for encryption and decryption. “Enigma”, which was used by Germany, is the most famous such machine. (Now, almost all cryptography is implemented by digital computers and digital signal processors.) Except some cases, the encryption (or decryption) algorithm had to be secret. The machines were not inexpensive and making use of them needed much cost. Thus, cryptography was used for special cases such as military affairs and foreign diplomacy. Another reason why it was not used by ordinal people is that they did not need it.

In 1970s, two revolutions in cryptography happened. One is that “Digital Encryption Standard (DES)” which is one kind of “common key cryptography” was standardized and the specification was published [2]. By the publishing, the security of DES can be tested by many researchers all over the world. To withstand attacks by the researchers, DES had ensured the security until when the attack which can break DES was developed in 1990s [3]. Now, “Advanced Encryption Standard (AES)” is used as the successor of DES [4]. The specification of AES also has been published and any attack which can practically break AES is not known. The other revolution is that Diffie and Hellman published a new concept “public key cryptography” [5]. In common key cryptography, the encryption algorithm can be published like DES but the key used in the encryption must be secret. In public key cryptography, there are two kind of keys, secret-key and public-key, and the encryption algorithm and the public-key can be published. Both revolutions accelerated research and development of cryptography.

Now, by development of the Internet, ordinary people use the Internet, and send and receive much data which is important for ordinary people such as his credit card number. Then, now, almost all people unconsciously use cryptography. In addition, function of cryptography became not being restricted to keeping a secret. As other functions, it is now used for electronic signature and so on.

1.1.2 What future cryptography needs

In the near future, “Internet of Things (IoT)” will be realized. Kevin Ashton is the first person using the word “IoT” [6]. At that time, he used the word as merchandise management system using RFID. Now, the meaning of the word was expanded. IoT is a concept that every possible “Things” such as cars, home electric appliances, smartphones, buildings, infrastructure, wearable devices and so on will connect with the Internet. “Things” will load sensor, gather information about situation and events around them and communicate the information to each other. At the same time, they automatically optimize their own operation based on the information or are controlled by other machine or human. IoT is expected to make our life convenient and efficient, and generate new sense of values.

Information security is to keep following three properties:

- Confidentiality
- Integrity
- Availability

In the society which IoT is realized, information security will be needed more than ever before. Because “Things” around us gather information about our private, the amount of information whose confidentiality must be kept increases. Because “Things” work based on the information, if integrity and availability of the information are not kept, the society gets flustered. In particular, for infrastructure, defect of integrity and availability cause disaster.

In such situation, what property is needed for cryptography ? The author thinks that the following properties are the answer:

- Rapidity
- Lightness
- Safety

The above properties are needed even today and the author think that the demand becomes stronger. Rapidity is needed because the amount of information whose confidentiality must be kept increases. Lightness is needed because even small devices such as wearable devices must use cryptography. Safety is needless to say.

1.1.3 Present cryptography

Roughly speaking, present cryptography can be devided to two types, public-key cryptography including key-exchange and electronic signature and common-key cryptography. Both of them are needed for information security.

In generally speaking, public-key cryptography often uses a finite field or a ring of modulo N, where N is the product of very large prime numbers. (See Ref. [5, 7, 8, 9] as examples.) Operations on such sets bring with residure. Residue takes more time than multiplication and addition and so that is hindrance to making cryptography be faster.

Common-key cryptography often uses “S-box”, “linear feedback sift-register (LFSR)” or both of them (See Ref. [10, 11, 12, 13, 14, 15] as examples.). DES and AES also use S-box. S-box is a non-linear operation and it is usually implemented with a reference table and so that is hindrance to making cryptography be light. LFSR is a generator which generate a M-sequence. M-sequence has useful properties for cryptography such as long period. M-sequence is, however, predicable if a part of M-sequence is known because LFSR is based on a simple linear operation. In practical situation, some LFSRs or other parts is combined when common-key cipher is constructed. In some cases, period of such combined system cannot be exactly estimated.

1.2 Permutation polynomials

A polynomial $f(X)$ is called permutation polynomial over a finite ring R when $f(X)$ is a bijection over R. There are many studies about permutation polynomials, and permutation polynomials are applied for many field including cryptography. RSA cryptosystem, which is one of the most famous public-key cryptography, is one of such applications [7]. Almost all studies, R is a finite field. (See Ref. [16, 17] as the survey.) Operations over a finite field, however, take much time as previously mentioned. The author thinks that we need to discuss another approach in order to develop better cipher.

1.2.1 Permutation polynomials over a ring of modulo 2^w

Although almost all studies on permutation polynomials are over finite fields, we focus on a ring of modulo 2^w. The definition of a permutation polynomial over a ring of modulo 2^w is as follows:

Definition 1.2.1. A finite degree polynomial $f(X)$ with integer coefficients is called a permutation polynomial over a ring of modulo 2^w if

$$\forall w \geq 0,\ \{f(\bar{X}) \mod 2^w | \bar{X} \in \mathbb{Z}/2^w\mathbb{Z}\} = \mathbb{Z}/2^w\mathbb{Z}.$$

Study about permutation polynomials over the ring is very important because they are compatible with digital computers and digital signal processors. They can calculate values of permutation polynomials over the ring faster than over a finite field because 2 power residue operation is practically negligible. Then, they are in particular expected to be useful for cryptography and pseudo random number generator. The compatibility is also useful for light implementation of such applications either by software or hardware. We, therefore, believe that we can construct cipher which can be used in IoT by the polynomials.

Indeed, RC6 which is a common-key cipher uses permutation polynomial over a ring of modulo 2^w [18] and it was elected as a final candidate of AES. It means that usability and safety of RC6 is appreciated to a certain extent by National Institute of Standard and Technology (NIST) which is a agency of United States Federal Government. As another example, Vector Stream Cipher (VSC) is also a common-key cipher which uses permutation polynomial over a ring of modulo 2^w [19]. VSC recorded more than 25Gbps by hardware implementation. As far as the author knows, VSC was one of fastest common-key cipher at the time when VSC was developed in 2004 and any attack which can practically break VSC has not been known at the moment. These examples show the usability of permutation polynomials over a ring of modulo 2^w.

There is another reason why we focus on the polynomials. Because the value of permutation polynomials over the ring can be fastly calculated, we need not to use a reference table when the polynomials are used as a part of common-key cryptography. Thus, it can be said that using the permutation polynomials contributes to light implementation.

In addition, because a ring of modulo 2^w and permutation polynomials over the ring have recursive structure, we can expect that the periodicity of orbits which the polynomials draw over a ring is clearly. If the degrees of the polynomials are equal to or more than

2, the polynomials have non-linearity. Then, ensuring of periodicity and non-linearity are expected to stand side by side. Indeed, we discuss the periodicity of the polynomials and the possibility of using them for common-key cipher in Chapters 4 and 5, and they are affirmed.

Although permutation polynomials over a ring of modulo 2^w have such usability and the good exampls exist, they have not been studied very well. Except some applications (See Ref. [18, 20, 19] as examples) including linear congruential method, they have not been used in the field of cryptography and pseudo random number generator. We, therefore, research them in this thesis and pioneer a new field.

1.3 Outline of this thesis

This thesis is constructed as follows:

In Chapter 2, we discuss about a key-exchange protocol with permutation polynomials over a ring of modulo 2^w [20]. A key-exchange protocol with odd degree Chebyshev polynomials over the ring was proposed as one of application of permutation polynomials over the ring. We analyses the protocol and generalize the result for a key-exchange protocol with more general permutation polynomials.

In Chapter 3, we deal with Vector Stream Cipher (VSC), which is cipher constructed by permutation polynomials over a ring of modulo 2^w [19]. The security of VSC has been already investigated by several researchers, and some security problems against for the theoretical attacks were reported though any practical attack breaking VSC has not been reported so far. We propose some improvements in order to avoid such attacks and develop new ciphers " Vector Stream Cipher 2.0" and " Vector Stream Cipher 2.1" which have proven against distinguishing attack with linear masking.

In Chapter 4, we introduce a new class of permutation polynomials called "one-stroke polynomials over a ring of modulo 2^w". In general, a map used in cipher and pseudo random number generator should have long period for their security and randomness. One-stroke polynomial over a ring of modulo 2^w is a permutation polynomial which has the maximum period over the ring. We derive a necessary and sufficient condition for permutation polynomials to be one-stroke polynomials over a ring of modulo 2^w. Such condition for low degree polynomials was derived at more approximately half a century ago [21], but condition for an arbitrary degree has not been derived until we derive in this thesis. After that, we introduce some properties of one-stroke polynomials over a ring of modulo 2^w.

In Chapter 5, we propose some methods of combining one-stroke polynomials over a ring of modulo 2^w for pseudo random number generator and stream cipher. A system using only one one-stroke polynomial over the ring is too simple, and so it should not be used for pseudo random number generators and, in particular, for stream cipher. Another reason why the system should not be used for stream cipher is that we need longer period than 2^w in many cases, where w is the length of the variable used in the system. Then, we propose a method to make a more complex system which preserve the period of one-stroke polynomials over a ring of modulo 2^w and a method to make a system which has a longer period than 2^w.

Finally, we conclude this thesis.

Chapter 2

Cryptanalysis of key-exchange with permutation polynomials over a ring of modulo 2^w

One application of permutation polynomials over a ring of modulo 2^w is a key-exchange protocol with odd degree Chebyshev polynomials over the ring [20]. Odd degree Chebyshev polynomials are proven to be permutation polynomials over a ring of modulo 2^w [20] and they are commutative each other. Then, the protocol is constructed by replacing the discrete logarithm problem of Diffie-Hellman key-exchange protocol [5] with the degree decision problem of Chebyshev polynomials over the ring. Thus, the security is related to difficulty of the degree decision problem over the ring. If the problem can efficiently be solved, the protocol is not secure.

Although a key-exchange protocol with Chebyshev polynomials over the real-number interval $[-1, 1]$ was proposed earlier than that over a ring of modulo 2^w [22], the degree decision problem of Chebyshev polynomials over $[-1, 1]$ was solved and so the key-exchange protocol is regarded as being not secure [23]. However, we cannot directly adapt the method to solve the problem over a ring of modulo 2^w because residue operations do not appear in the problem over $[-1, 1]$. It was also shown that the degree decision problem over a ring of modulo 2^w can efficiently be solved when the given argument of Chebyshev polynomial is even [24], but the degree decision problem with odd argument has not been solved.

It is conjectured that the difficulty of the problem over a ring of modulo 2^w is related to the periodicity of Chebyshev polynomial over the ring. Here, odd degree Chebyshev polynomials have two kinds of periodicity. One is " orbital period", and the other is "degree period". Since odd degree Chebyshev polynomials are permutation polynomials over a ring of modulo 2^w, when an odd degree Chebyshev polynomial iterate affecting a factor of the ring, we can observe an orbit on the ring. The " orbital period " is the period of the orbit. The "degree period" is observed when changing the degree of Chebyshev polynomials with fixed arguments of polynomials. Although there are studies about both kinds of period [25, 26, 27], they have not been completely studied so far.

In this chapter, we clarify both kinds of the periodicity. After that, we show that the degree decision problem over a ring of modulo 2^w can efficiently be solved even if the given argument of Chebyshev polynomial is odd, and so the key-exchange protocol is not secure. It takes only $O(w^4)$ times to solve the problem. The fact does not mean, however, that Chebyshev polynomials are not useful for all the fields in cryptography.

We solved the problem in 2015 [28], and Kawano and Yoshioka independently solved the problem at almost the same time [29, 30]. The solving method proposed in this chapter is more general than them. Although the degree of the Chebyshev polynomial used in the key-exchange protocol is restricted to odd number, our method proposed in this chapter can solve the problem even if the solution is even.

In addition, we discuss the reason why the key-exchange protocol with Chebyshev polynomials is not secure and generalize the discussion. We show that a key exchange protocol with a set of permutation polynomials is not secure if the permutation polynomials in the set satisfy some conditions which odd degree Chebyshev polynomials also satisfy. If it takes $O(g(w))$ times to calculate the value of permutation polynomials for given argument and iteration number, it takes only $O(w \cdot g(w))$ times to break the key-change protocol with the polynomials.

2.1 Key-exchange

Let us consider the case that there are two persons, Alice and Bob, and Alice would like to communicate a secret message to Bob. Assume that they can use a public channel only. It means that their talk can be heard by anyone. In this case, they should use cryptography. Alice encrypt the secret message with a key and send the cipher text. Bob received the cipher text and decrypt it with the same key. (See Fig. 2.1.) Then, Bob get the secret message and any other person cannot know because they do not know the key. Cryptography like this protocol is called "common key cryptography".

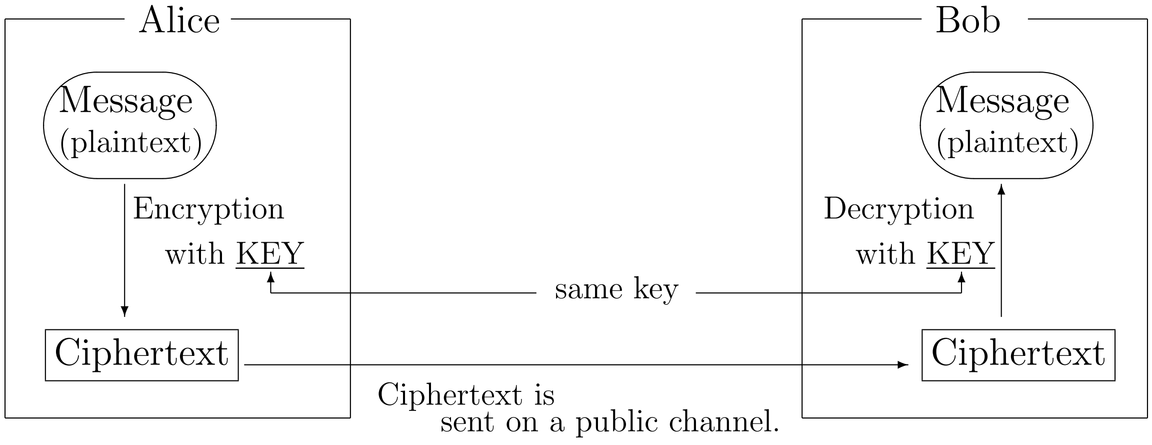


Figure 2.1: Common key cryptography

Although the above protocol is useful for general cases, there is a big issue. That is "how to share the key between Alice and Bob". Using a credible courier is one solution, but they cannot do so because there is the assumption that they can use a public channel only. The assumption is realistic in many cases.

"Key-exchange" is a concept which can solve the issue. By using it, Alice and Bob can secretly share the key on a public channel. The first practical key-exchange protocol was developed by Diffie and Hellman in 1976 [5]. It is believed that the security of the protocol is based on discrete logarithm problem (DLP).

Definition 2.1.1. Discrete logarithm problem is as follows: find x such that

$$y = g^x \mod p$$

for given prime number p, its primitive root$g$ and y.

Some other protocols have been also developed. One of the most famous such protocols uses elliptic curve [9].

2.1.1 Diffie-Hellman key-exchange

For the later section, let us introduce the key-exchange protocol developed by Diffie-Hellman. That is as follows:

1. Alice and Bob agree a sufficient large prime number p and its primitive root g on a public channel.
2. Alice determine a secret sufficient large number N_A and calculate $M_A := g^{N_A} \mod p$. Similarly, Bob determine a secret sufficient large number N_B and calculate $M_B := g^{N_B} \mod p$.
3. Alice send M_A to Bob on a public channel and Bob send M_B to Alice.
4. Alice calculate $M_{BA} := M_B^{N_A} \mod p$ and Bob calculate $M_{AB} := M_A^{N_B} \mod p$.

At the step 4,

$$\begin{aligned} M_{BA} =& M_B^{N_A} \mod p \\ =& (g^{N_B} \mod p)^{N_A} \mod p \\ =& (g^{N_A} \mod p)^{N_B} \mod p \\ =& M_A^{N_B} \mod p \\ =& M_{AB}. \end{aligned}$$

Thus, Alice and Bob share the number $K := M_{BA} (= M_{AB})$.

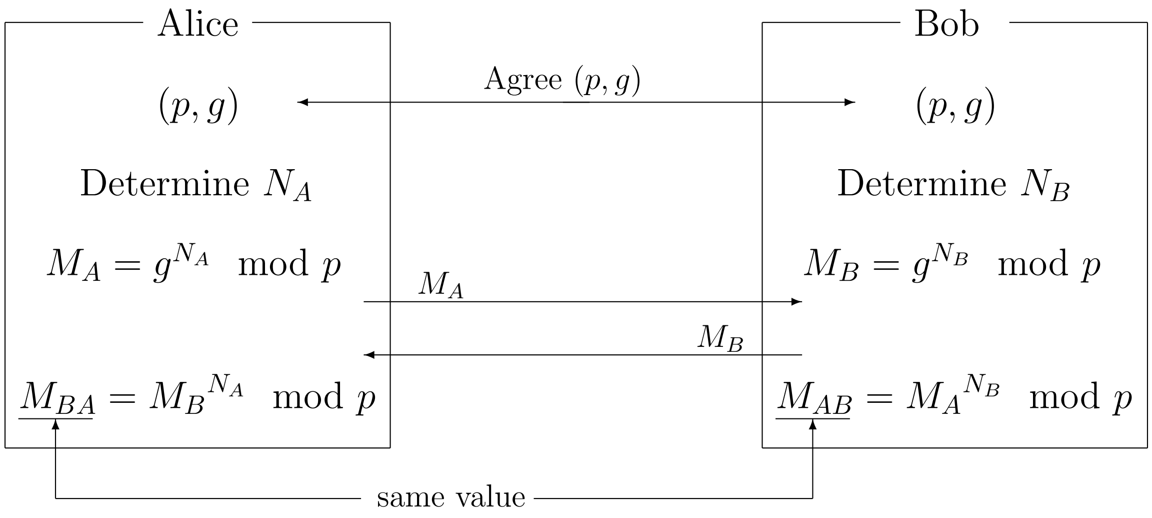


Figure 2.2: Diffie-Hellman key-exchange

It is believe that it is too difficult for any other person to calculate the value of K only with the knowledge of p, g, M_A and M_B. That is based on the following conjectures:

- To calculate the value of N_A or N_B is the most effective approach to calculate the value of K.
- DLP is computationally difficult.

Although these conjectures are not proven, many researchers believe them.

2.2 Key-exchange with Chebyshev polynomials over a ring of modulo 2^w

In general, a residue operation takes much time, and so Diffie-Hellman key-exchange protocol also takes much time. Faster protocol, therefore, should be developed and permutation polynomials over a ring of modulo 2^w are perhaps useful for the purpose. One of such protocol is a key-exchange with odd degree Chebyshev polynomials. In this section, we introduce the protocol.

2.2.1 Chebyshev polynomials

Before explaining the protocol, we introduce Chebyshev polynomials.

Definition 2.2.1. Assume that p is an integer. A Chebyshev polynomial of p degree $T_p(X)$ is defined as a polynomial satisfying

$$\forall\theta \in \mathbb{R},\ T_p(\cos\theta) = \cos p\theta.$$

For example,

$$\begin{aligned}
T_1(X) &= X,\\
T_2(X) &= 2X^2 - 1,\\
T_3(X) &= 4X^3 - 3X,\\
T_4(X) &= 8X^4 - 8X^2 + 1,\\
T_5(X) &= 16X^5 - 20X^3 + 5X.
\end{aligned}$$

By the definition, there are some simple properties. It is clear that arbitrary Chebyshev polynomials are commutative such that

$$\forall p, q \in \mathbb{Z},\quad T_p \circ T_q(X) = T_q \circ T_p(X) = T_{pq}(X).$$

It is also clear that the following relation is satisfied.

$$\forall p, q \in \mathbb{Z},\quad 2T_{pq}(X) = T_{p+q}(X) + T_{p-q}(X).$$

It is known that arbitrary odd degree Chebyshev polynomials are permutation polynomials over a ring of modulo 2^w, it means that they are bijection over the ring. The above simple properties are preserved even if the domain and range are restricted to the ring.

It takes only $O(w^3)$ times to calculate the value of Chebyshev polynomial over a ring of modulo 2^w for a given argument and degree.

2.2.2 Key-exchange with Chebyshev polynomials

Based on the properties of Chebyshev polynomials, the key-exchange protocol with odd degree Chebyshev polynomials was proposed. The protocol replaced DLP of Diffie-Hellman key-exchange protocol with the following degree decision problem (DDP) over a ring of modulo 2^w.

Definition 2.2.2. A degree decision problem over a ring of modulo 2^w is as follows: find $p \in \mathbb{Z}/2^w\mathbb{Z}$ satisfying

$$\bar{Y} \equiv T_p(\bar{X}) \mod 2^w,$$

with given $\bar{X},\ \bar{Y} \in \mathbb{Z}/2^w\mathbb{Z}$.

Then, the key-exchange protocol is as follows:

1. Alice and Bob agree a sufficient large number w and $\bar{X}$ on a public channel.
2. Alice determine a secret sufficient large number N_A and calculate $M_A := T_{N_A}(\bar{X}) \mod 2^w$. Similarly, Bob determine a secret sufficient large number N_B and calculate $M_B := T_{N_B}(\bar{X}) \mod 2^w$.
3. Alice send M_A to Bob on a public channel and Bob send M_B to Alice.
4. Alice calculate $M_{BA} := T_{N_A}(M_B) \mod 2^w$ and Bob calculate $M_{AB} := T_{N_A}(M_A) \mod 2^w$.

At the step 4,

$$\begin{aligned} M_{BA} =& T_{N_A}(M_B) \mod 2^w \\ =& T_{N_A}\left(T_{N_B}(\bar{X}) \mod 2^w\right) \mod 2^w \\ =& T_{N_B}\left(T_{N_A}(\bar{X}) \mod 2^w\right) \mod 2^w \\ =& T_{N_B}(M_A) \mod 2^w \\ =& M_{AB}. \end{aligned}$$

Thus, Alice and Bob can share the $K := M_{BA}(= M_{AB})$.

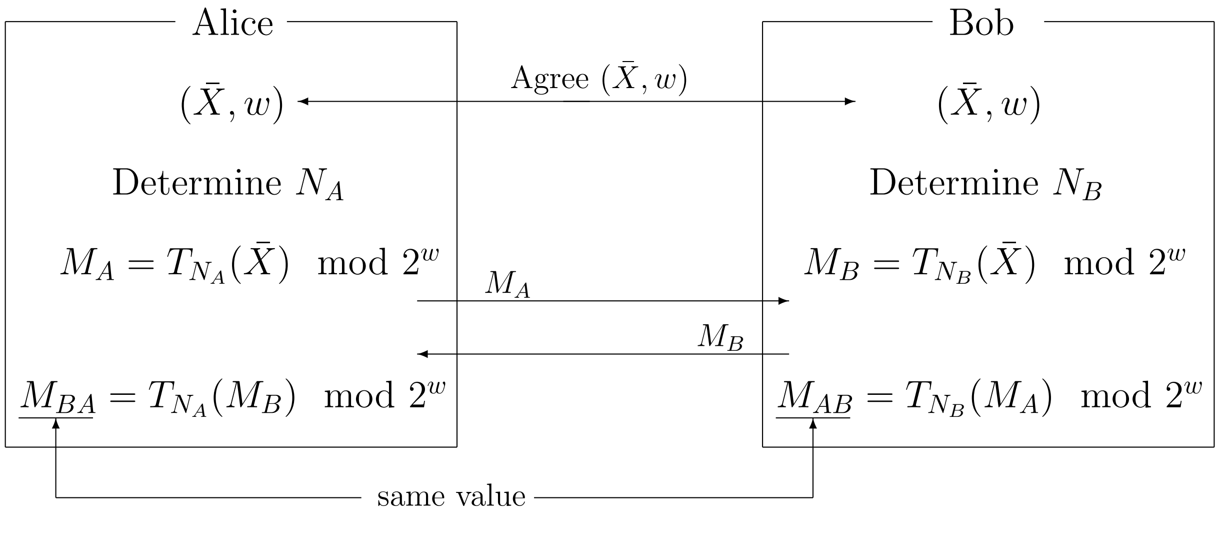

Alice
Bob
Agree $(\bar{X}, w)$
$(\bar{X}, w)$
$(\bar{X}, w)$
Determine N_A
Determine N_B
$M_A = T_{N_A}(\bar{X}) \mod 2^w$
$M_B = T_{N_B}(\bar{X}) \mod 2^w$
M_A
M_B
$M_{BA} = T_{N_A}(M_B) \mod 2^w$
$M_{AB} = T_{N_B}(M_A) \mod 2^w$
same value

Figure 2.3: Key-exchange with Chebyshev polynomials

Since the calculation the value of Chebyshev polynomial for a given argument on a ring of 2^w takes only $O(w^3)$ times, the protocol also takes only $O(w^3)$ times. We think that the security of this protocol depends on the difficulty of DDP by the analogy of Diffie-Hellman key-exchange. At least, if the DDP can be perfectly solved, the key-exchange is not secure.

2.3 Cryptanalysis of key-exchange with odd degree Chebyshev polynomials over a ring of modulo 2^w

In this section, we show a method to perfectly solve DDP. It means that, unfortunately, the key-exchange protocol is not secure. The difficulty of DDP strictly relates to periodicities, orbital period and degree period. Then, we firstly investigate the both of them.

2.3.1 Orbital period of odd degree Chebyshev polynomials over a ring of modulo 2^w

We prove a theorem about orbital period. Firstly, we prove some lemmas which are needed for proving the theorem.

Lemma 2.3.1. Assume that $X_1 = (2A_1 - 1) \cdot 2^{k_1} \pm 1$ and $X_2 = (2A_2 - 1) \cdot 2^{k_2}$, where A_1, A_2, k_1, $k_2 \in \mathbb{N}$ and $k_1 \geq 2$. For $r \geq 2$,

$$T_{2^r}(X_1) \equiv 1 \mod 2^{k_1+r+2}, \tag{2.1}$$

$$T_{2^r}(X_2) \equiv 1 \mod 2^{k_2+r+2}. \tag{2.2}$$

Proof. In the case $r = 2$,

$$\begin{aligned}
&T_{2^2}(X_1)\\
=&8X_1^4 - 8X_1^2 + 1\\
=&8\{(2A_1 - 1) \cdot 2^{k_1} \pm 1\}^4 - 8\{(2A_1 - 1) \cdot 2^{k_1} \pm 1\}^2 + 1\\
\equiv&1 \mod 2^{k_1+2+2},\\
&T_{2^2}(X_2)\\
=&8\{(2A_2 - 1) \cdot 2^{k_2}\}^4 - 8\{(2A_2 - 1) \cdot 2^{k_2}\}^2 + 1\\
\equiv&1 \mod 2^{k_2+2+2}.
\end{aligned}$$

Then, (2.1) and (2.2) are true.

Assume that (2.1) and (2.2) are true with $r = r_0$. We will consider the case $r = r_0 + 1$.

$$\begin{aligned}
&T_{2^{r_0+1}}(X_1)\\
=&2T_{2^{r_0}}(X_1)T_{2^{r_0}}(X_1) - T_0(X_1)\\
\equiv&2\{T_{2^{r_0}}(X_1)\}^2 - 1 \mod 2^{k_1+r_0+3}\\
\equiv&2\{T_{2^{r_0}}(X_1) \mod 2^{k_1+r_0+3}\}^2 - 1 \mod 2^{k_1+r_0+3}\\
\equiv&1 \mod 2^{k_1+r_0+3},\\
&T_{2^{r_0+1}}(X_2)\\
=&2T_{2^{r_0}}(X_2)T_{2^{r_0}}(X_2) - T_0(X_2)\\
\equiv&2\{T_{2^{r_0}}(X_2)\}^2 - 1 \mod 2^{k_2+r_0+3}\\
\equiv&2\{T_{2^{r_0}}(X_2) \mod 2^{k_2+r_0+3}\}^2 - 1 \mod 2^{k_2+r_0+3}\\
\equiv&1 \mod 2^{k_2+r_0+3}.
\end{aligned}$$

Then, (2.1) and (2.2) are true with $r = r_0 + 1$.

From the above, the lemma is true. □

Lemma 2.3.2. Assume that $X_1 = (2A_1 - 1) \cdot 2^{k_1} \pm 1$ and $X_2 = (2A_2 - 1) \cdot 2^{k_2}$, where A_1, A_2, k_1, $k_2 \in \mathbb{N}$ and $k_1 \geq 2$. For $r \geq 2$,

$$T_{2^r\pm1}(X_1) \equiv X_1 + 2^{k_1+r+1} \mod 2^{k_1+r+2}, \tag{2.3}$$

$$T_{2^r\pm1}(X_2) \equiv X_2 + 2^{k_2+r} \mod 2^{k_2+r+1}. \tag{2.4}$$

Proof. In the case $r = 2$,

$$\begin{aligned}
T_{2^2+1}(X) &= 16X^5 - 20X^3 + 5X,\\
T_{2^2-1}(X) &= 4X^3 - 3X.
\end{aligned}$$

Then, (2.3) and (2.4) are true.

Assume that (2.3) and (2.4) are true with $r = r_0$. We will consider the case $r = r_0 + 1$. By Lemma 2.3.1,

$$\begin{aligned}
&T_{2^{r_0+1}\pm1}(X_1)\\
=&2T_{2^{r_0}\pm1}(X_1)T_{2^{r_0}}(X_1) - X_1\\
\equiv&2\{T_{2^{r_0}\pm1}(X_1) \mod 2^{k_1+r_0+3}\}\{T_{2^{r_0}}(X_1) \mod 2^{k_1+r_0+3}\} - X_1 \mod 2^{k_1+r_0+3}\\
\equiv&X_1 + 2^{k_1+r_0+2} \mod 2^{k_1+r_0+3},\\
&T_{2^{r_0+1}\pm1}(X_2)\\
=&2T_{2^{r_0}\pm1}(X_2)T_{2^{r_0}}(X_2) - X_2\\
\equiv&2\{T_{2^{r_0}\pm1}(X_2) \mod 2^{k_2+r_0+2}\}\{T_{2^{r_0}}(X_2) \mod 2^{k_2+r_0+2}\} - X_2 \mod 2^{k_2+r_0+2}\\
\equiv&X_2 + 2^{k_2+r_0+1} \mod 2^{k_2+r_0+2}.
\end{aligned}$$

Then, (2.3) and (2.4) are true with $r = r_0 + 1$. From the above, the lemma is true. □

Lemma 2.3.3. Assume that $X_1 = (2A_1 - 1) \cdot 2^{k_1} \pm 1$ and $X_2 = (2A_2 - 1) \cdot 2^{k_2}$, where A_1, A_2, k_1, $k_2 \in \mathbb{N}$ and $k_1 \geq 2$. For $r \geq 2$,

$$T_{3\cdot2^r\pm1}(X_1) \equiv X_1 + 2^{k_1+r+1} \mod 2^{k_1+r+2}, \tag{2.5}$$
$$T_{3\cdot2^r\pm1}(X_2) \equiv X_2 + 2^{k_2+r} \mod 2^{k_2+r+1}. \tag{2.6}$$

Proof. The following calculations prove the lemma.

$$\begin{aligned}
&T_{3\cdot2^r\pm1}(X_1)\\
=&2T_{2^{r+1}\pm1}(X_1)T_{2^r}(X_1) - T_{2^r\pm1}(X_1)\\
\equiv&2\{T_{2^{r+1}\pm1}(X_1) \mod 2^{k_1+r+2}\}\{T_{2^r}(X_1) \mod 2^{k_1+r+2}\}\\
&\qquad - \{T_{2^r\pm1}(X_1) \mod 2^{k_1+r+2}\} \mod 2^{k_1+r+2}\\
\equiv&X_1 + 2^{k_1+r+1} \mod 2^{k_1+r+2},\\
&T_{3\cdot2^r\pm1}(X_2)\\
=&2T_{2^{r+1}\pm1}(X_2)T_{2^r}(X_2) - T_{2^r\pm1}(X_2)\\
\equiv&2\{T_{2^{r+1}\pm1}(X_2) \mod 2^{k_2+r+1}\}\{T_{2^r}(X_2) \mod 2^{k_2+r+1}\}\\
&\qquad - \{T_{2^r\pm1}(X_2) \mod 2^{k_2+r+1}\} \mod 2^{k_2+r+1}\\
\equiv&X_2 + 2^{k_2+r} \mod 2^{k_2+r+1}.
\end{aligned}$$

□

Lemma 2.3.4. Assume that $X_1 = (2A_1 - 1) \cdot 2^{k_1} \pm 1$ and $X_2 = (2A_2 - 1) \cdot 2^{k_2}$, where A_1, A_2, k_1, k_2, $B \in \mathbb{N}$ and $k_1 \geq 2$. For $r \geq 2$,

$$T_{(2B-1)\cdot2^r\pm1}(X_1) \equiv X_1 + 2^{k_1+r+1} \mod 2^{k_1+r+2}, \tag{2.7}$$
$$T_{(2B-1)\cdot2^r\pm1}(X_2) \equiv X_2 + 2^{k_2+r} \mod 2^{k_2+r+1}, \tag{2.8}$$

where B is a natural number.

Proof. It has been shown that (2.7) and (2.8) are true in the cases of $B = 1$ and 2. We consider the case $B \geq 3$. Assume that (2.7) and (2.8) are true at $B = B_0$ and at

$B = B_0 + 1$.

$$\begin{aligned}
&T_{(2B_0+3)\cdot 2^r \pm 1}(X_1)\\
\equiv&2T_{(2B_0+1)\cdot 2^r \pm 1}(X_1)T_{2^{r+1}}(X_1) - T_{(2B_0-1)\cdot 2^r \pm 1}(X_1) \mod 2^{k_1+r+2}\\
\equiv&2\{X_1 + 2^{k_1+r+2}\} \times 1 - \{X_1 + 2^{k_1+r+2}\}\\
\equiv&X_1 + 2^{k_1+r+2},\\
&T_{(2B_0+3)\cdot 2^r \pm 1}(X_2)\\
\equiv&2T_{(2B_0+1)\cdot 2^r \pm 1}(X_2)T_{2^{r+1}}(X_2) - T_{(2B_0-1)\cdot 2^r \pm 1}(X_2) \mod 2^{k_2+r+1}\\
\equiv&2\{X_2 + 2^{k_2+r+1}\} \times 1 - \{X_2 + 2^{k_2+r+1}\}\\
\equiv&X_2 + 2^{k_2+r+1}.
\end{aligned}$$

Then, (2.7) and (2.8) are true at $B = B_0 + 2$. From the above, the lemma is true. □

Lemma 2.3.5. Assume that p, m and X_0 are natural numbers satisfying

$$T_p(X_0) \equiv X_0 + 2^m \mod 2^{m+1}.$$

If $m \leq w$,

$$\begin{aligned}
&\{T_p^i(X_0) \mod 2^w | i = 0, 1, 2, \cdots, 2^{w-m} - 1\}\\
=&\{X_0 + k \cdot 2^m \mod 2^w | k = 0, 1, 2, \cdots, 2^{w-m} - 1\}.
\end{aligned}$$

Proof of Lemma 2.3.5 is shown in Ref. [26].

Theorem 2.3.6. The orbital periods are distributed according to Table 2.1.

Table 2.1: Orbital periods of odd degree Chebyshev polynomials. Here, A, B, r, k_1, k_2 $\in \mathbb{N}$, $2 \leq r \leq w-1$, $2 \leq k_1 \leq w-1$ and $k_2 \leq w-1$.

Initial Point	Degree	Orbital Period
arbitrary	1, $2^w - 1$	1
$(2A-1) \cdot 2^{k_1} \pm 1$	$(2B-1) \cdot 2^r \pm 1$	$\max(2^{w-k_1-r-1}, 1)$
$(2A-1) \cdot 2^{k_2}$	$(2B-1) \cdot 2^r \pm 1$	$\max(2^{w-k_2-r}, 1)$
0, 1, $2^w - 1$	arbitrary	1

Proof. By Lemmas 2.3.4 and 2.3.5, it is clear that the second and third lines of Table 2.1 are true.

Since $T_1(X) = X$ and $T_{2^w-1} \equiv (X) \mod 2^w$, the first line of Table 2.1 is true.

Assume that $X_1 = 1$. We can express $X_1 \equiv (2A_1 - 1) \cdot 2^w + 1 \mod 2^w$. Then, by Lemma 2.3.4,

$$\begin{aligned}
T_{(2B-1)\cdot 2^r \pm 1}(X_1) &\equiv X_1 + 2^{w+r+1} \mod 2^{w+r+2}\\
&\equiv X_1 \mod 2^w.
\end{aligned}$$

Similarly, we can get $T_{(2B-1)\cdot 2^r \pm 1}(0) \equiv 0 \mod 2^w$ and $T_{(2B-1)\cdot 2^r \pm 1}(2^w - 1) \equiv 2^w - 1 \mod 2^w$. Then, the fourth line of Table 2.1 is true. □

Example 2.3.1. Let us consider the orbital period with the initial point $X_0 = 5$ and the degree $p = 3$ over a ring of modulo 2^7. Since $5 = 2^2 + 1$ and $3 = 2^2 - 1$, the second line of Table 2.1 is applied. The orbital period is calculated as

$$\max(2^{7-2-2-1}, 1) = 4.$$

Indeed,

$$\begin{aligned} T_3(5) &\equiv 101 \mod 2^7, \\ T_3(101) &\equiv 69 \mod 2^7, \\ T_3(69) &\equiv 37 \mod 2^7, \\ T_3(37) &\equiv 5 \mod 2^7. \end{aligned}$$

Then, the orbital period is surely 4.

2.3.2 Degree period of Chebyshev polynomials

We prove a theorem about periodicity of degree. Firstly, we introduce the following some basic lemmas.

Lemma 2.3.7.

$$\forall p \in \mathbb{N},\ T_p\left(\frac{\alpha+\alpha^{-1}}{2}\right) = \frac{\alpha^p + \alpha^{-p}}{2}.$$

Lemma 2.3.8. Assume that s and t are natural numbers. Then,

$$a \equiv b \mod 2^s \Rightarrow a^{2^t} \equiv b^{2^t} \mod 2^{s+t}.$$

Proofs of the above two lemmas are shown in Ref. [25].

Lemma 2.3.9. Assume that $X_1 = (2A-1)\cdot 2^{k_1} \pm 1$ and $\alpha = X_1 + \sqrt{X_1^2 - 1}$, where A and k_1 are natural numbers and $2 \le k_1 \le w-4$. Then,

$$\frac{\alpha^{2^{w-k_1-1}} + \alpha^{-2^{w-k_1-1}}}{2} \equiv 1 \mod 2^w, \tag{2.9}$$

$$\frac{\alpha+\alpha^{-1}}{2} \cdot \frac{\alpha^{2^{w-k_1-1}} - \alpha^{-2^{w-k_1-1}}}{2} \equiv 0 \mod 2^w. \tag{2.10}$$

Proof. Since $k_1 \ge 2$,

$$\begin{aligned} \alpha^2 &= 2X_1^2 - 1 + 2X_1\sqrt{X_1^2-1} \\ &\equiv 1 + 2X_1\sqrt{X_1^2-1} \mod 2^{k_1+2}. \end{aligned}$$

Assume that $\exists t_0 \in \mathbb{N},\ \alpha^{2^{t_0}} \equiv 1 + 2^{t_0}X_1\sqrt{X^2-1}$. By Lemma 2.3.8,

$$\begin{aligned} \alpha^{2^{t_0+1}} &\equiv \{1 + 2^{t_0}X_1\sqrt{X_1^2-1}\}^2 \mod 2^{k_1+t_0+2} \\ &\equiv 1 + 2^{t_0+1}X_1\sqrt{X^2-1} \mod 2^{k_1+t_0+2}. \end{aligned}$$

Then, $\forall t \in \mathbb{N},\ \alpha^{2^t} \equiv 1 + 2^tX_1\sqrt{X^2-1}$, and so

$$\alpha^{2^{w-k_1-1}} \equiv 1 + 2^{w-k_1-1}X\sqrt{X^2-1} \mod 2^w.$$

By the same reason,

$$\alpha^{-2^{w-k_1-1}} \equiv 1 - 2^{w-k_1-1} X\sqrt{X^2-1} \mod 2^w.$$

Form the above,

$$\begin{aligned}
&\frac{\alpha^{2^{w-k_1-1}} + \alpha^{-2^{w-k_1-1}}}{2} \equiv 1 \mod 2^w,\\
&\frac{\alpha+\alpha^{-1}}{2} \cdot \frac{\alpha^{2^{w-k_1-1}} - \alpha^{-2^{w-k_1-1}}}{2}\\
\equiv&\sqrt{X_1^2-1} \cdot 2^{w-k_1-1} X_1\sqrt{X^2-1} \mod 2^w\\
\equiv&0 \mod 2^w.
\end{aligned}$$

□

Lemma 2.3.10. Assume that $X_2 = (2A-1)\cdot 2^{k_2}$ and $\alpha = X_2 + \sqrt{X_2^2-1}$, where A and k_2 are natural numbers satisfying $k_2 \le w-3$. Then,

$$\frac{\alpha^{2^{w-k_2}} + \alpha^{-2^{w-k_2}}}{2} \equiv 1 \mod 2^w, \tag{2.11}$$

$$\frac{\alpha^{2^{w-k_2}} - \alpha^{-2^{w-k_2}}}{2} \equiv 0 \mod 2^w. \tag{2.12}$$

Proof. Since $X_2 = (2A-1)\cdot 2^{k_2}$ and k_2 is natural number,

$$\alpha^4 \equiv 1 \mod 2^{k_2+2}.$$

Assume that $\exists t_0 \in \mathbb{N},\ \alpha^{2^{t_0}} \equiv 1 \mod 2^{k_2+t_0}$. Then,

$$\alpha^{2^{t_0+1}} \equiv 1^2 \mod 2^{k_2+t_0+1}.$$

Therefore, $\alpha^{2^{w-k_2}} \equiv 1 \mod 2^w$ and $\alpha^{-2^{w-k_2}} \equiv 1 \mod 2^w$. From the above,

$$\frac{\alpha^{2^{w-k_2}} + \alpha^{-2^{w-k_2}}}{2} \equiv 1 \mod 2^w, \tag{2.13}$$

$$\frac{\alpha^{2^{w-k_2}} - \alpha^{-2^{w-k_2}}}{2} \equiv 0 \mod 2^w. \tag{2.14}$$

□

Lemma 2.3.11. Assume that p is an odd number and $X_1 = (2A-1)\cdot 2^{k_1} \pm 1$, where A and k_1 are natural numbers satisfying $k_1 \le w-4$. Then,

$$T_{p+2^{w-k_1-1}}(X_1) \equiv T_p(X_1) \mod 2^w.$$

Proof. Assume that $\alpha = X + \sqrt{X^2-1}$. Then, by Lemmas 2.3.7 and 2.3.9,

$$\begin{aligned}
&T_{p+2^{w-k_1-1}}(X_1)\\
=&\frac{\alpha^{p+2^{w-k_1-1}} + \alpha^{-p-2^{w-k_1-1}}}{2}\\
=&\frac{\alpha^p+\alpha^{-p}}{2} \cdot \frac{\alpha^{2^{w-k_1-1}} + \alpha^{2^{w-k_1-1}}}{2} + \frac{\alpha^p-\alpha^{-p}}{2} \cdot \frac{\alpha^{2^{w-k_1-1}} - \alpha^{2^{w-k_1-1}}}{2}\\
\equiv&T_p(X_1) \mod 2^w.
\end{aligned}$$

□

Lemma 2.3.12. Assume that p is an odd number and $X_2 = (2A-1) \cdot 2^{k_2}$, where A and k_2 are natural numbers satisfying $k_2 \leq w-3$. Then,

$$T_{p+2^{w-k_2}}(X_2) \equiv T_p(X_2) \mod 2^w.$$

Proof. Assume that $\alpha = X + \sqrt{X^2-1}$. Then, by Lemmas 2.3.7 and 2.3.10,

$$\begin{aligned}
&T_{p+2^{w-k_2}}(X_2) \\
=&\frac{\alpha^{p+2^{w-k_2}} + \alpha^{-p-2^{w-k_2}}}{2} \\
=&\frac{\alpha^p + \alpha^{-p}}{2} \cdot \frac{\alpha^{2^{w-k_2}} + \alpha^{2^{w-k_2}}}{2} + \frac{\alpha^p - \alpha^{-p}}{2} \cdot \frac{\alpha^{2^{w-k_2}} - \alpha^{2^{w-k_2}}}{2} \\
\equiv&T_p(X_2) \mod 2^w.
\end{aligned}$$

□

Theorem 2.3.13. The degree periods of odd degree Chebyshev polynomials are distributed according to Table 2.2.

Table 2.2: Periods of degree. Here, A, k_1, $k_2 \in \mathbb{N}$, $2 \leq k_1 \leq w-4$ and $k_2 \leq w-3$.

X	Periodicity of Degree
$(2A-1) \cdot 2^{k_1} \pm 1$	$T_{p+2^{w-k_1-1}}(X) \equiv T_p(X) \mod 2^w$
	$T_{p+2^{w-k_1-2}}(X) \not\equiv T_p(X) \mod 2^w$
$(2A-1) \cdot 2^{k_2}$	$T_{p+2^{w-k_2}}(X) \equiv T_p(X) \mod 2^w$
	$T_{p+2^{w-k_2-1}}(X) \not\equiv T_p(X) \mod 2^w$
otherwise	$T_{p+2}(X) \equiv T_p(X) \mod 2^w$

Proof. It has already been shown that $T_{p+2^{w-k_1-1}}(X) \equiv T_p(X) \mod 2^w$ for $X = (2A-1) \cdot 2^{k_1} \pm 1$ and $T_{p+2^{w-k_2}}(X) \equiv T_p(X) \mod 2^w$ for $X = (2A-1) \cdot 2^{k_2}$. First, we consider the case $p \equiv \pm 1 \mod 2^w$. Assume that $X_1 = (2A-1) \cdot 2^{k_1} \pm 1$. By Theorem 2.3.6, the orbital period of $T_{\pm 1+2^{w-k_1-2}}$ with the initial value X_1 is 2. Then,

$$T_{\pm 1+2^{w-k_1-2}}(X_1) \not\equiv X_1 \mod 2^w.$$

On the other hand, $T_{\pm 1}(X_1) \equiv X_1 \mod 2^w$. Then,

$$T_{p+2^{w-k_1-2}}(X_1) \not\equiv T_p(X_1) \mod 2^w.$$

Next, assume that $p = (2B-1) \cdot 2^r \pm 1$, where B and r are natural numbers satisfying $2 \leq r \leq w-1$.

We consider the case $r \geq w-k_1-1$. By Theorem 2.3.6, the orbital period of T_p with the initial value X_1 is 1. On the other hand, since $\exists B' \in \mathbb{N}$, $p+2^{w-k_1-1} = (2B'-1) \cdot 2^{w-k_1-2} \pm 1$, the orbital period of $T_{p+2^{w-k_1-2}}$ with the initial value X_1 is 2. Then,

$$T_{p+2^{w-k_1-2}}(X_1) \not\equiv T_p(X_1) \mod 2^w.$$

We consider the case $r = w-k_1-2$. The orbital period of T_p with the initial value X_1 is 2. On the other hand, since $\exists B', r' \in \mathbb{N}$, $p + 2^{w-k_1-1} = (2B'-1) \cdot 2^{w-k_1-2+r'} \pm 1$, the orbital period of $T_{p+2^{w-k_1-2}}$ with the initial value X_1 is 1. Then,

$$T_{p+2^{w-k_1-2}}(X_1) \not\equiv T_p(X_1) \mod 2^w.$$

We consider the case $2 \leq r \leq w - k_1 - 3$. Since the orbital period of T_p with the initial value X_1 is 2^{w-k_1-r-1},

$$\forall i, j \in \mathbb{Z}/2^{w-k_1-r-1}\mathbb{Z},\ i \neq j \Rightarrow T_p^i(X_1) \not\equiv T_p^j(X_1) \mod 2^w.$$

By the definition of Chebyshev polynomials, $\forall i \in \mathbb{N},\ T_p^i(X_1) = T_{p^i}(X_1)$. If i is an odd number, $\exists B_i \in \mathbb{N},\ p^i = (2B_i - 1) \cdot 2^r \pm 1$. Then, if i and j are odd numbers and satisfy $i < j \leq 2^{w-k_1-r-2}$,

$$T_{(2B_i-1)\cdot 2^r \pm 1}(X_1) \not\equiv T_{(2B_j-1)\cdot 2^r \pm 1}(X_1) \mod 2^w.$$

$$\{2(B + 2^{w-k_1-r-2}) - 1\} \cdot 2^r \pm 1 = p + 2^{w+k_1-1}.$$

By Lemma 2.3.11,

$$T_{p+2^{w-k_1-1}}(X_1) \equiv T_p(X_1) \mod 2^w.$$

Then,

$$T_{p+2^{w-k_1-2}}(X_1) \not\equiv T_p(X_1) \mod 2^w.$$

From the above, for an arbitrary odd number p,

$$T_{p+2^{w-k_1-2}}(X_1) \not\equiv T_p(X_1) \mod 2^w.$$

By the same way, for an arbitrary odd number p,

$$T_{p+2^{w-k_1-2}}((2A-1) \cdot 2^{k_2}) \not\equiv T_p((2A-1) \cdot 2^{k_2}) \mod 2^w.$$

Assume that X_0 cannot be written as $(2A-1) \cdot 2^{k_1}$ nor $(2A-1) \cdot 2^{k_2}$. By Theorem 2.3.6, the orbital period of T_p with the initial value X_0 is 1 where p is an arbitrary odd number. Then,

$$T_{p+2}(X_0) \equiv T_p(X_0) \mod 2^w.$$

From the above, the theorem is true. □

Example 2.3.2. Let us consider the degree period with $X_0 = 5$ over a ring of modulo 2^6. Since $5 = 2^2 + 1$, Theorem 2.3.13 states that

$$T_p(X_0) \equiv T_{p+2^{6-2-1}}(X_0) \mod 2^6,$$
$$T_p(X_0) \not\equiv T_{p+2^{6-2-2}}(X_0) \mod 2^6$$

for an arbitrary odd number p. Indeed,

$$\begin{aligned}
T_3(5) &\equiv 37 \mod 2^6,\\
T_5(5) &\equiv 37 \mod 2^6,\\
T_7(5) &\equiv 5 \mod 2^6,\\
T_9(5) &\equiv 5 \mod 2^6,\\
T_{11}(5) &\equiv 37 \mod 2^6,\\
T_{13}(5) &\equiv 37 \mod 2^6,\\
T_{15}(5) &\equiv 5 \mod 2^6,\\
T_{17}(5) &\equiv 5 \mod 2^6.
\end{aligned}$$

Then, the statement is surely true.

2.3.3 Method of solving degree decision problem

In this section, we show a method to solve a degree decision problem by using Lemma 2.3.4, Theorem 2.3.6 and Theorem 2.3.13.

Remark. Degree decision problem over a ring of modulo 2^w is as follows: find $p \in \mathbb{Z}/2^w\mathbb{Z}$ satisfying

$$\bar{Y} \equiv T_p(\bar{X}) \mod 2^w$$

with given $\bar{X}$ and $\bar{Y}$.

Assume that $p = p(l) \cdot 2^l$ where $p(l)$ is an odd number and l is a non-negative natural number and $\bar{X}_l = T_{2^l}(X) \mod 2^w$. Then,

$$\begin{aligned} T_p(\bar{X}) &= T_{p(l)\cdot 2^l}(\bar{X}) \\ &= T_{p(l)}\left(T_{2^l}(\bar{X})\right) \\ &= T_{p(l)}(\bar{X}_l). \end{aligned}$$

Then, it is enough to solve the following reduced degree decision problem for $l = 0, 1, \cdots, w-1$.

Definition 2.3.1. Reduced degree decision problem is as follows: find an odd number $p(l)$ satisfying

$$\bar{Y} \equiv T_{p(l)}(\bar{X}_l) \mod 2^w, \tag{2.15}$$

where $\bar{X}_l \equiv T_{2^l}(\bar{X})$.

We show a method to solve the problem with a fixed l. There are three cases.

Case 1: $\bar{X}_l$ can be expressed as the following form

$$\bar{X}_l = (2A-1)\cdot 2^k \pm 1,$$

where A and k are natural numbers satisfying $2 \leq k \leq w-4$. In this case, the algorithm to solve the problem is as follows:

1. If $\bar{Y} \equiv \bar{X}_l \mod 2^w$, output $p(l) = 1$ and finish this algorithm.
2. Find a natural number $r \geq 2$ satisfying

   $$\bar{Y} \equiv \bar{X}_l + 2^{k+r+1} \mod 2^{k+r+2}.$$

   If r satisfying the condition does not exist, finish this algorithm since any odd number $p(l)$ does not satisfy (2.15).
3. Set $q \leftarrow 2^r + 1$ and $m \leftarrow k + r + 3$.
4. If $Y \not\equiv T_q(\bar{X}) \mod 2^m$, $q \leftarrow q + 2^{m-k-2}$.
5. If $m \geq w$, output $p(l) = q$ and finish this algorithm. Else, $m \leftarrow m+1$ and return step 4.

The operation of step 1 is obviously proper.

By the Lemma 2.3.4, if we cannot find r at the step 2, there are the three possible cases: $\bar{Y} \equiv T_1(\bar{X}_l) \mod 2^w$, $\bar{Y} \equiv T_{2^w-1}(\bar{X}_l) \mod 2^w$ and $\bar{Y} \not\equiv T_p(\bar{X}) \mod 2^w$ for an arbitrary odd number p. By the theorem 2.3.6, $T_1(\bar{X}_l) \equiv T_{2^w-1}(\bar{X}_l) \equiv \bar{X}_l \mod 2^w$. Since the possibility of $\bar{Y} \equiv \bar{X}_l \mod 2^w$ is removed at the step 1, $\bar{Y} \not\equiv T_p(\bar{X}) \mod 2^w$ for an arbitrary odd number p if we cannot find r. Therefore, the operation of the step 2 is proper.

By the theorem 2.3.13, if $\bar{Y} \equiv T_q(\bar{X}_l) \mod 2^{m-1}$ and $\bar{Y} \not\equiv T_q(\bar{X}_l) \mod 2^m$, $\bar{Y} \not\equiv T_{q+2^{m-k-2}}(\bar{X}_l) \mod 2^m$. Then, the operations of the steps 3-5 are proper.

From the above, this algorithm is proper.

Since it takes $O(w^3)$ times to calculate the value of $T_q(\bar{X}) \mod 2^w$, this algorithm requires $O(w^4)$ times if there exists an odd number $p(l)$ satisfying (2.15) and $O(w)$ times if there does not exist.

Case 2: $\bar{X}_l$ can be expressed as the following form

$$\bar{X}_l = (2A-1)\cdot 2^k,$$

where A and k are natural numbers satisfying $k \leq w-3$. In this case, the algorithm to solve the problem is as follows:

1. If $\bar{Y} \equiv \bar{X}_l \mod 2^w$, output $p(l) = 1$ and finish this algorithm.

2. Find a natural number $r \geq 2$ satisfying

$$\bar{Y} \equiv \bar{X}_l + 2^{k+r} \mod 2^{k+r+1}.$$

   If r satisfying the condition does not exist, finish this algorithm since any odd number $p(l)$ does not satisfy (2.15).

3. Set $q \leftarrow 2^r \pm 1$ and $s \leftarrow k+r+2$.

4. If $Y \not\equiv T_q(\bar{X}) \mod 2^m$, $q \leftarrow q + 2^{m-k-1}$.

5. If $m \geq w$, output $p(l) = q$ and finish this algorithm. Else, $m \leftarrow m+1$ and return 4).

By the same way as the case 1, it is shown that this algorithm is proper. This algorithm also requires $O(w^4)$ times if there exists an odd number $p(l)$ satisfying (2.15) and $O(w)$ times if there does not exist.

Case 3: otherwise. By the theorem 2.3.13, $T_{p(l)}(X) \mod 2^w$ is constant for any $p(l)$. Then, if $\bar{Y} \equiv \bar{X}_l$, an arbitrary odd number $p(l)$ satisfies (2.15). Else, any odd number $p(l)$ does not satisfy (2.15).

From the above, the original degree decision problem whose domain of searching degree is not restrict odd numbers can be efficiently solved. Since it takes only $O(w^3)$ times to calculate the value of $\bar{X}_l$ for each l, the method to solve the problem takes only $O(w^4)$ times if there exists a solution p and $O(w^3)$ times if there does not exist.

Example 2.3.3. Let us find p satisfying

$$865 = T_p(7) \mod 2^{11}.$$

In this example,

$$\begin{aligned} w =& 11, \\ \bar{Y} =& 865 = 27 \cdot 2^5 + 1. \end{aligned}$$

First, we consider the case $l = 0$.

$$\bar{X}_0 \equiv 7 = 2^3 - 1.$$

This corresponds to the case 1. At the step 2 of the algorithm, we cannot find r satisfying the condition. Then, any odd number $p(0)$ is not satisfy (2.15).

Next, we consider the case $l = 1$.

$$\bar{X}_1 = T_2(7) \mod 2^{11} = 97 = 3 \cdot 2^5 + 1.$$

This corresponds to the case 1. At the step 2 of the algorithm, $r = 2$ is chosen. At the step 3 we set $q \leftarrow 5 = 2^2 + 1$ and $m \leftarrow 10 = 5 + 2 + 3$.

$$T_5(97) \not\equiv 865 \mod 2^{10}.$$

Then, $q \leftarrow q + 2^3 = 13$ and $m \leftarrow m + 1$.

$$T_{13}(97) \equiv 865 \mod 2^{11}.$$

Since $m = 11 \geq w$, finish this algorithm.

From the above, $p = 13 \cdot 2^1 = 26$ is a solution of the problem.

2.4 Generalization

In this section, we discuss generalization of the former sections. We consider a broader class of set of permutation polynomials than the set of odd degree Chebyshev polynomials.

There are two direct and essential reasons why the degree decision problem of Chebyshev polynomials over a ring of modulo 2^w is efficiently solved. One reason is that the periodicity of degree is completely made clear. The other reason is that Chebyshev polynomials over the ring have some recursive properties. In general, such recursive properties are common of permutation polynomials over the ring. There are also two reasons why we could make the periodicity of degree clear. One reason is that the orbital period is made clear, and the other reason is that the relation between the orbital period and the degree period is known to be the established. In connection with the later reason, the arbitrary odd degree Chebyshev polynomials over the ring can be expressed as an iteration of third degree Chebyshev polynomials over the ring.

Based on the above, we consider the following set of permutation polynomials: assume that

$$\{P_1(X), P_2(X), \cdots, P_n(X)\}$$

is a set of permutation polynomials over a ring of modulo 2^w, that the value $P_i(\bar{X}) \mod 2^w$ can be efficiently calculated for given i and $\bar{X}$, that there exists a permutation polynomial $f(X)$ which satisfies that

$$\forall i,\ \exists j \text{ s.t. } \forall \bar{X},\ P_i(\bar{X}) \equiv f^j(\bar{X}) \mod 2^w,$$

and so the polynomials $P_i(X)$ are commutative each other and that $f(X)$ satisfies at least one of the following conditions.

- The values of $f^j(\bar{X}) \mod 2^w$ can be directly and efficiently calculated for given j and $\bar{X}$.
- We can efficiently find i which satisfies $P_i(X) \equiv f^j(X) \mod 2^w$ for given j.

In the case of odd degree Chebyshev polynomials, $f(X)$ is $T_3(X)$ and the both conditions are satisfied. In the case that the later condition is satisfied, we can calculated $F^j(\bar{X})$ mod 2^w by finding i and calculating $P_i(\bar{X}) \mod 2^w$ even if we cannot directly calculate the value.

In this situation, we can construct a key-exchange protocol with the set of polynomials. That replace Chebyshev polynomials at the key-exchange protocol with the polynomials in the set. The key-exchange protocol, however, is not secure. It means that the following iteration number decision problem (INDP) can efficiently be solved.

Definition 2.4.1. Iteration number decision problem is as follows: find an integer j satisfying

$$\bar{Y} \equiv f^j(\bar{X}) \mod 2^w,$$

where $\bar{Y}$ and $\bar{X}$ are given integers.

Theorem 2.4.1. INDP can efficiently be solved.

We show an algorithm to solve the problem. First, we introduce some lemmas.

Lemma 2.4.2. Assume that $\bar{Y}$, $\bar{X}$ and j are integers, m is a non-negative integer and they satisfy

$$\bar{Y} \equiv f^j(\bar{X}) \mod 2^m.$$

Then, there is a non-negative integer $l \leq m$ such that

$$\bar{Y} \equiv f^{j+2^l}(\bar{X}) \mod 2^m.$$

Proof. It is clear that the lemma is true in the case $m = 0$. We consider the case $m = 1$. Since $F(X)$ is a permutation polynomials over a ring of modulo 2^w,

$$f(0) \equiv 0 \mod 2 \text{ and } f(1) \equiv 1 \mod 2$$

or

$$f(0) \equiv 1 \mod 2 \text{ and } f(1) \equiv 0 \mod 2$$

is practical. Then,

$$f^{j+2}(\bar{X}) \equiv f^j(\bar{X}) \mod 2 \equiv \bar{Y} \mod 2.$$

Assume that m' is a non-negative smaller integer than m. We consider the case that there exists a non-negative number $l' \leq m'$ such that $\bar{Y} \equiv f^{j+2^{l'}}(\bar{X}) \mod 2^{m'}$. In this case,

$$\bar{Y} \equiv f^{j+2^{l'}}(\bar{X}) + c2^{m'} \mod 2^{m'+1},$$

where $c \in \{0, 1\}$.

$$\begin{aligned} f^{j+2^{l'+1}}(\bar{X}) &\equiv f^{2^{l'}}\left(F^{j+2^{l'}}(\bar{X})\right) \mod 2^{m'+1} \\ &\equiv f^{2^{l'}}\left(F^{j}(\bar{X}) + c2^{m'}\right) \mod 2^{m'+1} \\ &\equiv f^{2^{l'}}\left(F^{j}(\bar{X})\right) + c2^{m'} \mod 2^{m'+1} \\ &\equiv \bar{Y} \mod 2^{m'+1}. \end{aligned}$$

From the above, the lemma is true. □

Lemma 2.4.3. Assume that $\bar{Y}$, $\bar{X}$, j and j' are integers, m is a non-negative integer and they satisfy

$$\begin{aligned} \bar{Y} &\equiv f^{j}(\bar{X}) \mod 2^{m}, \\ \bar{Y} &\equiv f^{j'}(\bar{X}) \mod 2^{m+1}, \end{aligned}$$

and l is the minimum non-negative integer satisfying

$$\bar{Y} \equiv f^{j+2^{l}}(\bar{X}) \mod 2^{m}.$$

Then,

$$j' \equiv j \mod 2^{l}.$$

Proof. Assume that $j' = j + a2^{l} + b$ where a is an integer and b is a non-negative integer satisfying $b < 2^{l}$. Then,

$$\begin{aligned} f^{j'}(\bar{X}) =& f^{j+a2^{l}+b}(\bar{X}) \\ \equiv& f^{j+b}(\bar{X}) \mod 2^{m}. \end{aligned}$$

Since 2^{l} is the minimum natural number satisfying $f^{j}(\bar{X}) \equiv f^{j+2^{l}}(\bar{X}) \mod 2^{m}$, $b = 0$.
Then, the lemma is true. □

Lemma 2.4.4. Assume that $\bar{Y}$, $\bar{X}$ and j are integers, m is a non-negative integer and they satisfy

$$\bar{Y} \equiv f^{j}(\bar{X}) + 2^{m} \mod 2^{m+1},$$

and l is the minimum non-negative integer satisfying

$$\bar{Y} \equiv f^{j+2^{l}}(\bar{X}) \mod 2^{m}.$$

If there exists an integer j' such that $\bar{Y} \equiv f^{j'}(\bar{X}) \mod 2^{m+1}$,

$$\bar{Y} \equiv f^{j+2^{l}}(\bar{X}) \mod 2^{m+1}.$$

Proof. Assume that

$$\bar{Y} \equiv f^{j+2^{l}}(\bar{X}) + 2^{m} \mod 2^{m+1}.$$

For an arbitrary integer a,

$$\begin{aligned} f^{a2^l}(\bar{Y}) &\equiv f^{a2^l}\left(f^{j+2^l}(\bar{X}) + 2^m\right) \mod 2^{m+1} \\ &\equiv f^{a2^l}\left(f^{j+2^l}(\bar{X})\right) + 2^m \mod 2^{m+1} \\ &\equiv f^{(a+1)2^l}\left(f^j(\bar{X})\right) + 2^m \mod 2^{m+1} \\ &\equiv f^{(a+1)2^l}\left(f^j(\bar{X}) + 2^m\right) \mod 2^{m+1} \\ &\equiv f^{(a+1)2^l}\left(\bar{Y}\right) \mod 2^{m+1}. \end{aligned}$$

Then,

$$\begin{aligned} f^{j+a2^l}(\bar{X}) &\equiv f^{a2^l}(\bar{Y} + 2^m) \mod 2^{m+1} \\ &\equiv f^{a2^l}(\bar{Y}) + 2^m \mod 2^{m+1} \\ &\equiv f^{2^l}(\bar{Y}) + 2^m \mod 2^{m+1} \\ &\equiv f^{2^l}\left(f^j(\bar{X}) + 2^m\right) + 2^m \mod 2^{m+1} \\ &\equiv f^{2^l}\left(f^j(\bar{X})\right) \mod 2^{m+1} \\ &\equiv f^{j+2^l}(\bar{X}) \mod 2^{m+1} \\ &\equiv \bar{Y} + 2^m \mod 2^{m+1}. \end{aligned}$$

By Lemma 2.4.3, if there exists an integer j' such that $\bar{Y} \equiv f^{j'}(\bar{X}) \mod 2^{m+1}$, there exists an integer b such that

$$\bar{Y} \equiv f^{j+b2^l}(\bar{X}) \mod 2^w.$$

From the above, this lemma is true. □

By using the above lemmas, the problem can be solved. We propose the following algorithm to solve INDP:

1. Set $j \leftarrow 0$ and $l \leftarrow 0$.
2. If $\bar{Y} \equiv f^j(\bar{X}) \mod 2^w$, output j and finish this algorithm. Else, find m_j such that
$$\bar{Y} \equiv f^j(\bar{X}) + 2^{m_j} \mod 2^{m_j+1}.$$
3. If $\bar{Y} \equiv f^{j+2^l}(\bar{X}) \mod 2^{m_j+1}$, $j \leftarrow j + 2^l$ and return to 2).
4. If $l = w - 1$, finish this algorithm. (In this case, any integer j dose not satisfy $\bar{Y} \equiv f^j(\bar{X}) \mod 2^w$.) Else, $l \leftarrow l + 1$ and return to 3).

If it takes $O(g(w))$ times to calculate $f^j(\bar{X}) \mod 2^w$ for given $\bar{X}$ and j, this algorithm requires only $O(w \cdot g(w))$ times at the worst case. Then, Theorem 2.4.1 is proven.

2.5 Summary

We completely clarified the orbital periods and degree periods of odd degree Chebyshev polynomials over a ring of modulo 2^w. Both of them show a recursive property, and it is shown here that we can efficiently solve a degree decision problem by using the property. The proven fact here shows that the proposed key-exchange protocol with Chebyshev polynomials is not secure. It is also shown that a key-exchange protocol with more generalized permutation polynomials including Chebyshev polynomials is not secure. However, it does not mean that permutation polynomials over a ring of modulo 2^w including Chebyshev polynomials will not be applied to other fields such as common key cipher. The characteristics shown in here would provide useful clues for investigating further applications based on Chebyshev polynomials and other permutation polynomials over a ring of modulo 2^w.

Chapter 3

Improving security of Vector Stream Cipher

Vector Stream Cipher (VSC) is a stream cipher which was developed by Umeno, Kim and Hasegawa at Communication Research Laboratory (now called the National Institute of Information and Communications Technology) in 2004 [19]. Although, roughly speaking, stream cipher is usually constructed by operations over GF(2) including shift-register (see Ref. [12, 13, 14, 15] as examples), VSC is constructed by permutation polynomials over a ring of modulo 2^w. There are few applications of permutation polynomials over a ring of modulo 2^w, in particular, practical applications are very rare. VSC is one of such applications. Because permutation polynomials over a ring of modulo 2^w compatible with digital computers and digital signal processors, VSC is very fast. In particular, it has recorded 25Gbps by hardware implementation. The compatibility is also useful for light implementation either by software or hardware.

On the other hand, the security of VSC has been investigated by several researchers [31, 32, 33, 34], and some security problems against for the theoretical attacks were reported though any practical attack breaking VSC has not been reported so far. (In this thesis, "theoretical attack" means an attack, which currently needs too much computation and so cannot practically break cipher with today's computers. Thus, a cipher which is proven to be immune against theoretical attacks can be said to have a certain provable security, while the success of a theoretical attack leads to a potential risk to be attacked with future's computers. On the other hand, "practical attack" means an attack which can practically break cipher with today's computers.) In this chapter, we improve VSC in order to avoid them and design two new cipher systems, which we call "Vector Stream Cipher 2.0" and "Vector Stream Cipher 2.1", respectively.

Our purpose of this chapter is not only to develop new cipher but also to explore possibility of the permutation polynomials over a ring of modulo 2^w in the field of cryptography, in particular, stream cipher.

3.1 Pseudo random number generator and Stream cipher

A method which can generate a sequence of truly random numbers is not known, and I think that such method will not be developed, at least in near future. Sequences of random numbers, however, is needed in many field such that

- Monte Carlo simulation

- inspection of products
- system identification
- communication systems
- cryptography
- steganography

and so on. Therefore, methods generating a sequence of numbers which has similar property of truly random sequence approximately are used. Demanded precision of the approximation depends on applications. Such methods are classified into two types. One is "physical random number generator" and the other is "pseudo random number generator". Both of them are practically used.

3.1.1 Pseudo random number generator

Pseudo random number generator (PRNG) is an algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers. The algorithm is completely deterministic and depends only on seed, which is a parameter or an initial value of the algorithm. In comparison with physical random number generator, PRNG has the following properties:

- It can easily be implemented at low cost.
- Estimation of the precision is easy.

In addition, PRNG has a reproducibility. It means that we can get perfectly same sequence if we use the same seed. The property is useful for some applications including stream cipher.

$$\begin{tikzcd}
\text{Seed (= initial value, parameter)} \arrow[d, "\text{input}"] \\
\boxed{\text{PRNG}} \text{ (= algorithm)} \arrow[d, "\text{output}"] \\
100101011010\cdots \ : \text{pseudo random number}
\end{tikzcd}$$

Figure 3.1: Pseudo random number generator

3.1.2 Stream cipher

Common key cipher is classified into two cipher. One is "Block cipher" and the other is "Stream cipher". Stream cipher uses PRNG and the seed of the PRNG corresponds to the key. In general, stream cipher is faster than block cipher and it can be implemented more lightly than block cipher. Then, stream cipher gets attention since 2000 approximately. In general, the encryption is as follows:

1. Generate a sequence by using PRNG. (The "seed" = key)

2. Execute the exclusive-OR with the sequence and plaintext. (The result is the ciphertext.)

In practical situations, the step 1 and 2 are executed in parallel. In decryption, the roles of plaintext and ciphertext are exchanged.

$$\begin{tikzcd}
\boxed{\text{PRNG}} \arrow[d] & \text{Seed (= “Key”)} \arrow[l] \\
100101011010\cdots : \text{pseudo random number} & \\
\oplus & \\
111001100101\cdots : \text{plaintext} \arrow[d, Rightarrow] & \\
011100111111\cdots : \text{ciphertext} &
\end{tikzcd}$$

Figure 3.2: Stream cipher

There are many types of stream cipher, but almost stream cipher uses operations over GF(2) including sift-register. VSC does not use the operations, so it is a rare type. It is classified into “chaotic cipher”.

3.2 Vector Stream Cipher

In this section, we introduce Vector Stream Cipher 128 (VSC128) which is one kind of original VSC. VSC128 requires a 128-bit secret key and a 128-bit initial vector. The encryption algorithm is as follows:

1. Assume that A, B, C, D, X, Y, Z and W are 32-bit integer variables. Assign a secret key to A, B, C and D, and an initial vector to X, Y, Z and W.

2. Repeat the following operation 8 times. (We call the operation “round” of VSC128.)

   (a) Assume that a, b, c, d, x, y, z and w are 32-bit integer variables. Calculate the values of a, b, c, d, x, y, z and w as follows:

$$\begin{aligned}
a &= A - (A \mod 4) + 1 \mod 2^{32},\\
b &= B - (B \mod 4) + 1 \mod 2^{32},\\
c &= C - (C \mod 4) + 1 \mod 2^{32},\\
d &= D - (D \mod 4) + 1 \mod 2^{32},\\
x &= X - (X \mod 4) + 1 \mod 2^{32},\\
y &= Y - (Y \mod 4) + 1 \mod 2^{32},\\
z &= Z - (Z \mod 4) + 1 \mod 2^{32},\\
w &= W - (W \mod 4) + 1 \mod 2^{32}.
\end{aligned}$$

   (b) Assume that A', B', C', D', X', Y', Z' and W' are 32-bit integer variables.

Calculate the values of A', B', C', D', X', Y', Z' and W' as follows:

$$\begin{aligned}
A' &= A(2A+y) \mod 2^{32},\\
B' &= B(2B+x) \mod 2^{32},\\
C' &= C(2C+z) \mod 2^{32},\\
D' &= D(2D+w) \mod 2^{32},\\
X' &= X(2X+c) \mod 2^{32},\\
Y' &= Y(2Y+d) \mod 2^{32},\\
Z' &= Z(2Z+a) \mod 2^{32},\\
W' &= W(2W+b) \mod 2^{32}.
\end{aligned}$$

(c) Regard $(A', B', C', D', X', Y', Z', W')$ as a 256-bit sequence, and perform 5-bit left rotational shift. After that, copy the sequence to (A, B, C, D, X, Y, Z, W). Writing mathematically,

$$\begin{aligned}
A &= (A' << 5) \oplus (B' >> 27) \mod 2^{32},\\
B &= (B' << 5) \oplus (C' >> 27) \mod 2^{32},\\
C &= (C' << 5) \oplus (D' >> 27) \mod 2^{32},\\
D &= (D' << 5) \oplus (X' >> 27) \mod 2^{32},\\
X &= (X' << 5) \oplus (Y' >> 27) \mod 2^{32},\\
Y &= (Y' << 5) \oplus (Z' >> 27) \mod 2^{32},\\
Z &= (Z' << 5) \oplus (W' >> 27) \mod 2^{32},\\
W &= (W' << 5) \oplus (A' >> 27) \mod 2^{32}.
\end{aligned}$$

Here, "<<" means simple bit shift.

3. Assume that $D1$, $D2$, $D3$ and $D4$ are 32-bit plaintexts and $E1$, $E2$, $E3$ and $E4$ are the corresponding ciphertexts respectively. Then, calculate the values of $E1$, $E2$, $E3$ and $E4$ as follows.

$$\begin{aligned}
E1 &= D1 \oplus X,\\
E2 &= D2 \oplus Y,\\
E3 &= D3 \oplus Z,\\
E4 &= D4 \oplus W.
\end{aligned}$$

4. Repeat steps 2 and 3 until all the given plaintexts are encrypted.

3.3 Attacks for VSC128

In this section, we introduce some theoretical attacks for VSC128 or weakness points of VSC128.

3.3.1 Distinguishing attack with linear masking

Key-averaged linear probability (LP) of a function f_K is defined as

$$LP(\Gamma_{\bar{X}}, \Gamma_{\bar{Y}}) := \left(2\frac{\#\{(\bar{X}, K)\,|\bar{X}\cdot\Gamma_{\bar{X}} = f_K(\bar{X})\cdot\Gamma_{\bar{Y}}\}}{2^{2n}} - 1\right)^2,$$

where $\Gamma_{\bar{X}}$, $\Gamma_{\bar{Y}}$, $\bar{X}$ and K are integers which are expressed as n-bit integers. The K corresponds to a key. The $\Gamma_{\bar{X}}$ and $\Gamma_{\bar{Y}}$ are called "Linear-Mask". The operation "$\cdot$" means the inner product over GF(2).

Assume $g_{K_1,K_2} := f_{K_1} \circ f_{K_2}$. Key-averaged linear characteristic probability with considering multi pass ($LCPM$) of g_{K_1,K_2} is defined as

$$LCPM(\Gamma_{\bar{X}}, \Gamma_{\bar{Z}}) := \sum_{\Gamma_{\bar{Y}}} LP(\Gamma_{\bar{X}}, \Gamma_{\bar{Y}}) LP(\Gamma_{\bar{Y}}, \Gamma_{\bar{Z}}).$$

The fundamental function of VSC128 with a key K is described as

$$f_K(\bar{X}) = \bar{X}(2\bar{X} + K - (K \mod 4) + 1) \mod 2^{32},$$

where $\bar{X}$ and K are expressed as 32-bit integers. The value of LP of the fundamental function was studied [32]. The value is

$$LP(\Gamma_{\bar{X}}, \Gamma_{\bar{Y}})$$
$$= \begin{cases} 1 & \text{for } (\Gamma_{\bar{X}}, \Gamma_{\bar{Y}}) = (1,1), (2,3) \text{ or } (3,2) \\ 2^{-2\mathrm{maxbit}(\Gamma_{\bar{X}})+4} & \text{for } \mathrm{maxbit}(\Gamma_{\bar{X}}) = \mathrm{maxbit}(\Gamma_{\bar{Y}}) > 3 \text{ and } [\Gamma_{\bar{X}}]_{\mathrm{maxbit}(\Gamma_{\bar{X}})-1} = [\Gamma_{\bar{Y}}]_{\mathrm{maxbit}(\Gamma_{\bar{Y}})-1} \\ 0 & \text{otherwise} \end{cases},$$

where $[\alpha]_i$ means the i-th least bit of α and $\mathrm{maxbit}(\alpha) := \max\{i | [\alpha]_i = 1\}$. It was also studied that the maximum value of $LCPM$ with 8 rounds is 2^{-115} [33]. Because a secret key of VSC128 is 128-bit, a distinguishing attack is realized.

3.3.2 Chosen initial vector attack

It is reported that the output sequence (key-stream) of VSC128 have a statistical deviation if the initial vector is chosen among specific vectors, and so the distinguishing attack is realized if an attacker can chose an initial vector intentionally [34]. More concretely, the distinguishing attack is practical if there are 2^{32} initial vectors whose 96 bits from the least significant bit are 0.

3.3.3 Collision of key stream

The round of VSC128 is not bijection. For example, assume that A, B, C, D, X, Y, Z and W are odd numbers and $\tilde{A} := A \oplus 0x80000000$, $\cdots$, $\tilde{W} := W \oplus 0x80000000$. Then, the values of $(A, \cdots, W)$ and $(\tilde{A}, \cdots, \tilde{W})$ after the round are equal. The fact means that effective key length of VSC128 is smaller than 128-bit.

3.4 Improving security of VSC128

In this section, we propose improvement of VSC128 to avoid such attacks. We call the this improved VSC128 "Vector Stream Cipher 2.0 (VSC 2.0)".

3.4.1 Increasing the repetition number of rounds

Since the maximum $LCPM$ becomes smaller when repetition number of rounds increases, we try to calculate the maximum $LCPM$ with more rounds than 8.

Assume LP_{round} means $LCPM$ of the round, and $\Gamma W_i = 1 << (i-1)$ and $\Gamma Z_j = 1 << (j-1)$ are Linear-Masks which mask W and Z respectively on the round, where i and j are natural numbers which are smaller than 32. LP_{round} is calculated as follows:

$$LP_{round}(\Gamma W_1, \Gamma) = \begin{cases} 1 & \text{for } \Gamma = \Gamma W_6 \\ 0 & \text{otherwise} \end{cases}.$$

$$LP_{round}(\Gamma W_2, \Gamma) = \begin{cases} 1 & \text{for } \Gamma = \Gamma W_7 \oplus \Gamma W_6 \\ 0 & \text{otherwise} \end{cases}.$$

For $3 \le i \le 27$,

$$LP_{round}(\Gamma W_i \oplus \alpha_{i-2}, \Gamma) = \begin{cases} 2^{-2i+4} & \text{for} \Gamma = \Gamma W_{i+5} \oplus (\beta_{i-2} << 5) \\ 0 & \text{otherwise} \end{cases}$$

where α_{i-2} and β_{i-2} are arbitrary $(i-2)$bit integers. For $i = 28$ and 29,

$$LP_{round}(\Gamma W_i \oplus \alpha_{i-2}, \Gamma) = \begin{cases} 2^{-2i+4} & \text{for } \Gamma = \Gamma Z_{i-27} \\ 0 & \text{otherwise} \end{cases}.$$

For $i = 30, 31$ and 32,

$$LP_{round}(\Gamma W_i \oplus \alpha_{i-2}, \Gamma) = \begin{cases} 1 & \text{for } \Gamma = \Gamma Z_{i-27} \oplus \beta_{i-29} \\ 0 & \text{otherwise} \end{cases}.$$

$LP_{round}(\Gamma Z_i, \cdot)$ is the same as the above.

Assume $LCPM_k$ means $LCPM$ with k rounds.

$$\begin{aligned} LCPM_9(\Gamma W_3, \Gamma Z_{16}) = \sum_{\alpha_1,\alpha_4,\alpha_6,\alpha_{11},\alpha_{16},\alpha_{21}} & [LP_{round}(\Gamma W_3, \Gamma W_8 \oplus (\alpha_1 << 5)) \\ & LP_{round}(\Gamma W_8 \oplus (\alpha_1 << 5), \Gamma W_{13} \oplus (\alpha_6 << 5)) \\ & LP_{round}(\Gamma W_{13} \oplus (\alpha_6 << 5), \Gamma W_{18} \oplus (\alpha_{11} << 5)) \\ & LP_{round}(\Gamma W_{18} \oplus (\alpha_{11} << 5), \Gamma W_{23} \oplus (\alpha_{16} << 5)) \\ & LP_{round}(\Gamma W_{23} \oplus (\alpha_{16} << 5), \Gamma W_{28} \oplus (\alpha_{21} << 5)) \\ & LP_{round}(\Gamma W_{28} \oplus (\alpha_{21} << 5), \Gamma Z_1) \\ & LP_{round}(\Gamma Z_1, \Gamma Z_6) \\ & LP_{round}(\Gamma Z_6, \Gamma Z_{11} \oplus (\alpha_4 << 5)) \\ & LP_{round}(\Gamma Z_{11} \oplus (\alpha_4 << 5), \Gamma Z_{16})] \\ = 2^{-129}. & \end{aligned}$$

For more detail, see the Table 3.1.

We repeat similar calculation, and get that 2^{-129} is the largest value of $LCPM_9$. It means that the linear attack needs more work than searching all over the key space and so is not practical. Then, we change the repetition number of rounds at step 2 of VSC128's algorithm from 8 to 9 in the specification of VSC 2.0.

Table 3.1: The values of LP_{round}.

$\Gamma_{\bar{X}}$	$\Gamma_{\bar{Y}}$	$LP_{round}(\Gamma_{\bar{X}}, \Gamma_{\bar{Y}})$
ΓW_3	$\Gamma W_8 \oplus (\alpha_1 << 5)$	2^{-2}
$\Gamma W_8 \oplus (\alpha_1 << 5)$	$\Gamma W_{13} \oplus (\alpha_6 << 5)$	2^{-12}
$\Gamma W_{13} \oplus (\alpha_6 << 5)$	$\Gamma W_{18} \oplus (\alpha_{11} << 5)$	2^{-22}
$\Gamma W_{18} \oplus (\alpha_{11} << 5)$	$\Gamma W_{23} \oplus (\alpha_{16} << 5)$	2^{-32}
$\Gamma W_{23} \oplus (\alpha_{16} << 5)$	$\Gamma W_{28} \oplus (\alpha_{21} << 5)$	2^{-42}
$\Gamma W_{28} \oplus (\alpha_{21} << 5)$	ΓZ_1	2^{-52}
ΓZ_1	ΓZ_6	1
ΓZ_6	$\Gamma Z_{11} \oplus (\alpha_4 << 5)$	2^{-8}
$\Gamma Z_{11} \oplus (\alpha_4 << 5)$	ΓZ_{16}	2^{-18}

3.4.2 Preprocessing

If we replace an initial vector with its hash value when the initial vector is given, we can avoid chosen initial vector attack because an attacker cannot calculate an inverse of a hash value effectively. Because the attack needs 2^{32} initial vectors whose 96 bit from the least significant bit of are 0, an attacker who will perform the attack is needed to calculate the hash function $2^{96} \times 2^{32} = 2^{128}$ times.

Although there are many hash functions in the world, we design a new hash function with the round because we should keep the program small. The algorithm of the new hash function is as follows.

1. Set A=0xfedcba98, B=0x01234567, C=0x89abcdef and D=0x76543210.
2. Assign a given initial vector to X, Y, Z and W.
3. Perform the round 30 times.
4. Output X, Y, Z and W.

Because any person who does not know the given initial vector cannot know the values of A, B, C and D at step 4 of the above algorithm, it is difficult for him to calculate the value of the given initial vector even if he knows the output. Then, we expect that the algorithm can be regarded as a hash function. We, therefore, add the operation replacing a given initial vector with its hash value with the algorithm to VSC128.

If the values of A, B, C and D at step 1 are all "0", the preprocessing is weak for a preimage attack. The value of D must not be odd number and the reason is shown in the next section. Then, we choose A =0xfedcba98, B =0x01234567, C =0x89abcdef and D =0x76543210. The reputation number at step 3 is chosen to be safe enough. We confirm experimentally it at section 3.6.

3.4.3 Avoiding collision

Firstly, we prove the following theorem.

Theorem 3.4.1. Consider a map $g : (\mathbb{Z}/2^n\mathbb{Z})^m \to (\mathbb{Z}/2^n\mathbb{Z})^m$, which is described as

$$g(A_0, A_1, \cdots, A_{m-1}) = (A'_0, A'_1, \cdots, A'_{m-1}),$$
$$A'_i = A_i \left(2A_i + a\left(A_{(i+1 \mod m)}\right)\right) \mod 2^n \quad ({}^\forall i \in \mathbb{Z}/m\mathbb{Z}),$$

where $A_1, \cdots, A_m$ and $A'_1, \cdots, A'_m$ are elements of $\mathbb{Z}/2^n\mathbb{Z}$ and $a(A_i) = A_i - (A_i \mod 4) + 1$ $(^\forall i \in \mathbb{Z}/m\mathbb{Z})$. Assume O_n a subset of $\mathbb{Z}/2^n\mathbb{Z}$, which is constructed of odd numbers in $\mathbb{Z}/2^n\mathbb{Z}$. Then, if we restrict the domain of g to $(\mathbb{Z}/2^n\mathbb{Z})^m$ except $(O_n)^m$, g becomes a bijective map on $(\mathbb{Z}/2^n\mathbb{Z})^m \backslash (O_n)^m$.

Proof. Since $g(X) \in (\mathbb{Z}/2^n\mathbb{Z})^m \backslash (O_n)^m$ for all $X \in (\mathbb{Z}/2^n\mathbb{Z})^m \backslash (O_n)^m$ obviously, g can be regarded as a map from $(\mathbb{Z}/2^n\mathbb{Z})^m \backslash (O_n)^m$ to $(\mathbb{Z}/2^n\mathbb{Z})^m \backslash (O_n)^m$. Then, we prove injectivity.

Assume $(A_0, A_1, \cdots, A_{m-1})$ and $(\tilde{A_0}, \tilde{A_1}, \cdots, \tilde{A_{m-1}})$ elements of $(\mathbb{Z}/2^n\mathbb{Z})^m \backslash (O_n)^m$ satisfying
$(A_0, A_1, \cdots, A_{m-1}) \neq (\tilde{A_0}, \tilde{A_1}, \cdots, \tilde{A_{m-1}})$. There are integers $0 \le s_i \le n$ and $2 \le t_i \le n$ $(i \in \mathbb{Z}/m\mathbb{Z})$ such that

$$\begin{aligned}
&(s_0, s_1, \cdots, s_{m-1}, t_0, t_1, \cdots, t_{m-1}) \neq (n, n, \cdots, n),\\
&s_i \le t_i \quad (^\forall i \in \mathbb{Z}/m\mathbb{Z}),\\
&\tilde{A_i} = A_i + (2p_i - 1)2^{s_i} \mod 2^n \quad (^\forall i \in \mathbb{Z}/m\mathbb{Z}),\\
&a(\tilde{A_i}) = a(A_i) + (2q_i - 1)2^{t_i} \mod 2^n \quad (^\forall i \in \mathbb{Z}/m\mathbb{Z}),
\end{aligned}$$

where p_i and q_i are natural numbers.

Assume that

$$\begin{aligned}
&(A'_0, A'_1, \cdots, A'_{m-1}) = g(A_0, A_1, \cdots, A_{m-1}),\\
&(\tilde{A_0}', \tilde{A_1}', \cdots, \tilde{A_{m-1}}') = g(\tilde{A_0}, \tilde{A_1}, \cdots, \tilde{A_{m-1}}).
\end{aligned}$$

To simplify notation, we define new notations A_m, $\tilde{A_m}$, t_m and q_m as A_0, $\tilde{A_0}$, t_0 and q_0 respectively. Then,

$$\begin{aligned}
\tilde{A_i}' &= \tilde{A_i}\{2\tilde{A_i} + a(\tilde{A_{i+1}})\} \mod 2^n\\
&= \{A_i + (2p_i - 1)2^{s_i}\}\{2A_i + (2p_i - 1)2^{s_i+1} + a(A_{i+1}) + (2q_{i+1} - 1)2^{t_{i+1}}\} \mod 2^n\\
&= [A_i\{2A_i + a(A_{i+1})\} + A_1(2p_i - 1)2^{s_i+2} + A_i(2q_{i+1} - 1)2^{t_{i+1}} + (2p_i - 1)^2 2^{2s_i+1}\\
&\qquad + a(A_{i+1})(2q_{i+1} - 1)2^{s_i} + (2p_i - 1)(2q_{i+1} - 1)2^{s_i+t_{i+1}}] \mod 2^n\\
&= A'_i + (2r_i - 1)2^{s_i} + A_1(2q_{i+1} - 1)2^{t_{i+1}} \mod 2^n.
\end{aligned}$$

Here, r_i $(i = 0, 1, \cdots, m)$ are natural numbers. There are two cases.

Case 1: There are $i, j \in \mathbb{Z}/m\mathbb{Z}$ such that $s_i \neq t_j$. In this case, there is $k \in \mathbb{Z}/m\mathbb{Z}$ such that $s_k < t_{k+1} \le n$, because $s_l \le t_l$ $(^\forall l \in \mathbb{Z}/m\mathbb{Z})$. Then,

$$\begin{aligned}
&[(2r_k - 1) \cdot 2^{s_k} + A_k(2q_{k+1} - 1)2^{t_{k+1}} \mod 2^n]_i = 0 \quad (i = 1, 2, \cdots, s_k),\\
&[(2r_k - 1) \cdot 2^{s_k} + A_k(2q_{k+1} - 1)2^{t_{k+1}} \mod 2^n]_{s_k+1} = 1.
\end{aligned}$$

Then, the $(s_k + 1)$-th least bit of $\tilde{A_k}'$ is different from that of A'_k.

Case 2: $s_0 = s_1 = \cdots = s_{m-1} = t_0 = t_1 = \cdots = t_{m-1} < n$. In this case, there are $k \in \mathbb{Z}/m\mathbb{Z}$ and $B \in \mathbb{Z}/2^{n-1}\mathbb{Z}$ such that $A_k = 2B$. Then,

$$\begin{aligned}
&(2r_k - 1)2^{s_k} + A_k(2q_{k+1} - 1)2^{t_{k+1}} \mod 2^n\\
=&(2r_k - 1)2^{s_k} + 2B(2q_{k+1} - 1)2^{s_k} \mod 2^n\\
=&[2\{r_k + B(2q_{k+1} - 1)\} - 1]2^{s_k} \mod 2^n.
\end{aligned}$$

Since $s_k < n$, $\tilde{A_k}' \neq A'_k$.

Therefore, $(\tilde{A_0}', \tilde{A_1}', \cdots, \tilde{A_{m-1}}') \neq (A'_0, A'_1, \cdots, A'_{m-1})$. □

Using the theorem, the round becomes a bijection if we restrict the domain to the case that at least one of A, B, C, D, X, Y, Z or W is even. Then, we introduce a new rule, " Keep the value of D is even". To keep the new rule, we add to the algorithm two modifications as follows.

- Change the length of secret key from 128-bit to 127-bit, and assign secret key to A, B, C and significant 31-bit of D at step 1 of the VSC128's algorithm. Set the least bit of D '0' simultaneously.
- At step 2(b) of the algorithm, if D is even, D' is also even. Then, change the step 2(c) as follows, and we can keep D even after performing the round. The mechanism is shown in Fig. 3.3.

$$\begin{aligned}
A &= (A' << 5) \oplus (B' >> 27) \mod 2^{32},\\
B &= (B' << 5) \oplus (C' >> 27) \mod 2^{32},\\
C &= (C' << 5) \oplus (D' >> 27) \mod 2^{32},\\
D &= (D' << 5) \oplus ((X' >> 27) << 1) \mod 2^{32},\\
X &= (X' << 5) \oplus (Y' >> 27) \mod 2^{32},\\
Y &= (Y' << 5) \oplus (Z' >> 27) \mod 2^{32},\\
Z &= (Z' << 5) \oplus (W' >> 27) \mod 2^{32},\\
W &= (W' << 5) \oplus (A' >> 27) \mod 2^{32}.
\end{aligned}$$

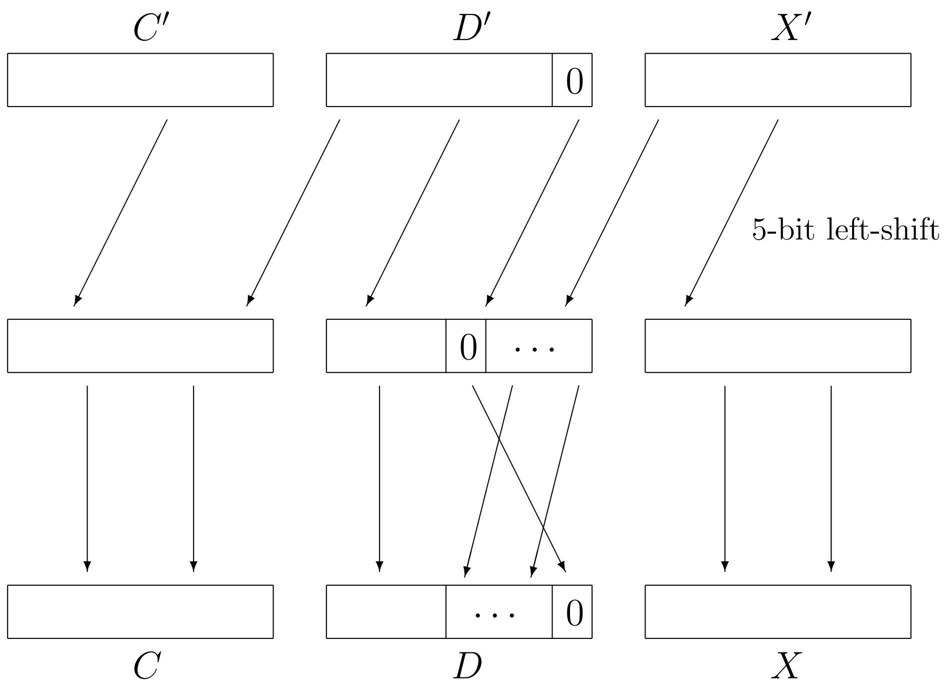


Figure 3.3: The new operation to keep D even

3.4.4 Algorithm of VSC 2.0

The algorithm of VSC 2.0 which is based on the above is as follows:

1. Assume that A, B, C, D, X, Y, Z and W are 32-bit integer variables. Set A=0xfedcba98, B=0x01234567, C=0x89abcdef and D=0x76543210 and assign an initial vector to X, Y, Z and W.

2. Repeat the following operation 30 times. (The operation is the "round" of VSC 2.0.)

   (a) Assume that a, b, c, d, x, y, z and w are 32-bit integer variables. Calculate the values of a, b, c, d, x, y, z and w as follows:

$$\begin{aligned}
a &= A - (A \mod 4) + 1 \mod 2^{32},\\
b &= B - (B \mod 4) + 1 \mod 2^{32},\\
c &= C - (C \mod 4) + 1 \mod 2^{32},\\
d &= D - (D \mod 4) + 1 \mod 2^{32},\\
x &= X - (X \mod 4) + 1 \mod 2^{32},\\
y &= Y - (Y \mod 4) + 1 \mod 2^{32},\\
z &= Z - (Z \mod 4) + 1 \mod 2^{32},\\
w &= W - (W \mod 4) + 1 \mod 2^{32}.
\end{aligned}$$

   (b) Assume that A', B', C', D', X', Y', Z' and W' are 32-bit integer variables. Calculate the values of A', B', C', D', X', Y', Z' and W' as follows:

$$\begin{aligned}
A' &= A(2A + y) \mod 2^{32},\\
B' &= B(2B + x) \mod 2^{32},\\
C' &= C(2C + z) \mod 2^{32},\\
D' &= D(2D + w) \mod 2^{32},\\
X' &= X(2X + c) \mod 2^{32},\\
Y' &= Y(2Y + d) \mod 2^{32},\\
Z' &= Z(2Z + a) \mod 2^{32},\\
W' &= W(2W + b) \mod 2^{32}.
\end{aligned}$$

   (c) Regard $(A', B', C', D', X', Y', Z', W')$ as a 256-bit sequence, and perform 5-bit left rotational shift. After that, copy the sequence to (A, B, C, D, X, Y, Z, W). After that, 1-bit left rotational shift for low-ranking 6-bit of D. Writing mathematically,

$$\begin{aligned}
A &= (A' << 5) \oplus (B' >> 27) \mod 2^{32},\\
B &= (B' << 5) \oplus (C' >> 27) \mod 2^{32},\\
C &= (C' << 5) \oplus (D' >> 27) \mod 2^{32},\\
D &= (D' << 5) \oplus ((X' >> 27) << 1)) \mod 2^{32},\\
X &= (X' << 5) \oplus (Y' >> 27) \mod 2^{32},\\
Y &= (Y' << 5) \oplus (Z' >> 27) \mod 2^{32},\\
Z &= (Z' << 5) \oplus (W' >> 27) \mod 2^{32},\\
W &= (W' << 5) \oplus (A' >> 27) \mod 2^{32}.
\end{aligned}$$

3. Assign a secret key to A, B, C and D except the least significant bit of D. Set the least significant bit of D to 0.

4. Perform the round 9 times.
5. Assume that $D1$, $D2$, $D3$ and $D4$ are 32-bit plaintexts and $E1$, $E2$, $E3$ and $E4$ are the corresponding ciphertexts respectively. Then, calculate the values of $E1$, $E2$, $E3$ and $E4$ as follows:

$$\begin{aligned} E1 &= D1 \oplus X, \\ E2 &= D2 \oplus Y, \\ E3 &= D3 \oplus Z, \\ E4 &= D4 \oplus W. \end{aligned}$$

6. Repeat steps 4 and 5 until all the given plaintexts are encrypted.

3.5 Further improvements of VSC 2.0

Although the security of VSC 2.0 is better than the original VSC, something to be considered for optimization of cipher design is left. In particular, we think that the following two things should be considered for further improvements.

- The key length of VSC 2.0 is 127bit. Thus, the key space is the half of the original VSC128.
- The step 2(c) of VSC 2.0 algorithm is slower than that of VSC128.

Both of them are caused by what makes the round a bijection. To improve them, we introduce the following theorem.

Theorem 3.5.1. Assume that $g : (\mathbb{Z}/2^n\mathbb{Z})^m \to (\mathbb{Z}/2^n\mathbb{Z})^m$ is described as

$$\begin{aligned} &g(A_0, A_1, \cdots, A_{m-1}) = (A'_0, A'_1, \cdots, A'_{m-1}), \\ &A'_i = A_i \left(2A_i + 4A_{(i+1 \mod m)} + 1\right) \mod 2^n \quad ({}^\forall i \in \mathbb{Z}/m\mathbb{Z}). \end{aligned}$$

Here, $A_1, \cdots, A_m$ and $A'_1, \cdots, A'_m$ are elements of $\mathbb{Z}/2^n\mathbb{Z}$. Then, g is a bijection.

Proof. It is enough to prove injectivity.

Assume that $(A_0, A_1, \cdots, A_{m-1})$ and $(\tilde{A}_0, \tilde{A}_1, \cdots, \tilde{A_{m-1}})$ are elements of $(\mathbb{Z}/2^n\mathbb{Z})^m$ satisfying

$$(A_0, A_1, \cdots, A_{m-1}) \neq (\tilde{A}_0, \tilde{A}_1, \cdots, \tilde{A_{m-1}}).$$

There are non-negative numbers s_i $(i \in \mathbb{Z}/m\mathbb{Z})$ satisfying

$$\begin{aligned} &s_i \leq n \quad ({}^\forall i \in \mathbb{Z}/m\mathbb{Z}), \\ &(s_0, s_1, \cdots, s_{m-1}) \neq (n, n, \cdots, n), \\ &\tilde{A}_i = A_i + (2p_i - 1)2^{s_i} \mod 2^n \quad ({}^\forall i \in \mathbb{Z}/m\mathbb{Z}), \end{aligned}$$

where p_i are natural numbers. Assume that

$$\begin{aligned} (A'_0, A'_1, \cdots, A'_{m-1}) &= g(A_0, A_1, \cdots, A_{,-1}), \\ (\tilde{A}_0{}', \tilde{A}_1{}', \cdots, \tilde{A}'_{m-1}) &= g(\tilde{A}_0, \tilde{A}_1, \cdots, \tilde{A}_{m-1}). \end{aligned}$$

Then, there exists $k \in \mathbb{Z}/m\mathbb{Z}$ such that $s_k \leq s_i$ ($^\forall i \in \mathbb{Z}/m\mathbb{Z}$). Obviously, $s_k < n$. To simplify notation, we define new symbols A_m, $\tilde{A}_m$, s_m and p_m as A_0, $\tilde{A}_0$, s_0 and p_0, respectively. Then,

$$\begin{aligned}
&\tilde{A}_k{}' \\
&= \tilde{A}_k\{2\tilde{A}_k + 4\tilde{A}_{k+1} + 1\} \mod 2^n \\
&= \{A_k + (2p_k - 1)2^{s_k}\}\{2A_k + (2p_k - 1)\cdot 2^{s_k+1} + 4A_{k+1} + (2p_{k+1} - 1)\cdot 2^{s_{k+1}+2} + 1\} \mod 2^n \\
&= A'_k + (2p_k - 1)\cdot 2^{s_k} + r\cdot 2^{s_k+1} + r'\cdot 2^{s_{k+1}+2} \mod 2^n.
\end{aligned}$$

Here, r and r' are natural numbers. Since $s_k \leq s_{k+1}$ and $s_k < n$, $\tilde{A}_k{}' \neq A'_k$. From the above, g is a bijection. □

3.5.1 Algorithm of VSC 2.1

By using Theorem 3.5.1, we can further improve VSC 2.0. We propose a new cipher "Vector Stream Cipher 2.1 (VSC 2.1)". The algorithm proposed here as VSC 2.1 is as follows:

1. Assume that A, B, C, D, X, Y, Z and W are 32-bit integer variables. Set A=0xfedcba98, B=0x01234567, C=0x89abcdef and D=0x76543210 and assign an initial vector to X, Y, Z and W.

2. Repeat the following operation 30 times. (The operation is the "round" of VSC 2.1.)

   (a) Assume that a, b, c, d, x, y, z and w are 32-bit integer variables. Calculate the values of a, b, c, d, x, y, z and w as follows:

$$\begin{aligned}
a &= 4A + 1 \mod 2^{32}, \\
b &= 4B + 1 \mod 2^{32}, \\
c &= 4C + 1 \mod 2^{32}, \\
d &= 4D + 1 \mod 2^{32}, \\
x &= 4X + 1 \mod 2^{32}, \\
y &= 4Y + 1 \mod 2^{32}, \\
z &= 4Z + 1 \mod 2^{32}, \\
w &= 4W + 1 \mod 2^{32}.
\end{aligned}$$

   (b) Assume that A', B', C', D', X', Y', Z' and W' are 32-bit integer variables. Calculate the values of A', B', C', D', X', Y', Z' and W' as follows:

$$\begin{aligned}
A' &= A(2A + y) \mod 2^{32}, \\
B' &= B(2B + x) \mod 2^{32}, \\
C' &= C(2C + z) \mod 2^{32}, \\
D' &= D(2D + w) \mod 2^{32}, \\
X' &= X(2X + c) \mod 2^{32}, \\
Y' &= Y(2Y + d) \mod 2^{32}, \\
Z' &= Z(2Z + a) \mod 2^{32}, \\
W' &= W(2W + b) \mod 2^{32}.
\end{aligned}$$

(c) Regard $(A', B', C', D', X', Y', Z', W')$ as a 256-bit sequence, and perform 5-bit left rotational shift. After that, copy the sequence to (A, B, C, D, X, Y, Z, W). Writing mathematically,

$$\begin{aligned}
A &= (A' << 5) \oplus (B' >> 27) \mod 2^{32},\\
B &= (B' << 5) \oplus (C' >> 27) \mod 2^{32},\\
C &= (C' << 5) \oplus (D' >> 27) \mod 2^{32},\\
D &= (D' << 5) \oplus (X' >> 27)) \mod 2^{32},\\
X &= (X' << 5) \oplus (Y' >> 27) \mod 2^{32},\\
Y &= (Y' << 5) \oplus (Z' >> 27) \mod 2^{32},\\
Z &= (Z' << 5) \oplus (W' >> 27) \mod 2^{32},\\
W &= (W' << 5) \oplus (A' >> 27) \mod 2^{32}.
\end{aligned}$$

3. Assign a secret key to A, B, C and D.

4. Perform the round 9 times.

5. Assume that $D1$, $D2$, $D3$ and $D4$ are 32-bit plaintexts and $E1$, $E2$, $E3$ and $E4$ are the corresponding ciphertexts respectively. Then, calculate the values of $E1$, $E2$, $E3$ and $E4$ as follows:

$$\begin{aligned}
E1 &= D1 \oplus X,\\
E2 &= D2 \oplus Y,\\
E3 &= D3 \oplus Z,\\
E4 &= D4 \oplus W.
\end{aligned}$$

6. Repeat steps 4 and 5 until all the given plaintexts are encrypted.

By Theorem 3.5.1, the round of VSC 2.1 is bijection. The maximum $LCPM_9$ of VSC 2.1 is the same as that of VSC 2.0. The key length of VSC 2.1 is 128bit, it is longer than that of VSC 2.0. VSC 2.1 is expected to be faster than VSC 2.0 because step 2(c) is more simple than that of VSC 2.0.

3.6 Experiments

In this section, we perform some experiments for VSC 2.0 and VSC 2.1.

3.6.1 Speed

We measure the speeds of performing VSC128, VSC 2.0, VSC 2.1 and AES-128. The environment in which we measure is shown in Table 3.2. As results, we got Table 3.3. VSC2.1 is slightly slower than the original VSC128, but faster than VSC 2.0.

3.6.2 Property of the preprocessing

We investigate properties of the preprocessing of VSC 2.0 and VSC 2.1. Step 2 of VSC 2.0 algorithm and that of VSC 2.1 algorithm are preprocessing, respectively. The detail of the experiment is as follows:

Table 3.2: Environment on which we measure speed

CPU	1.3 GHz Intel Core i5
Memory	4 GB 1600 MHz DDR3
OS	OS X 10.9.5 (13F34)
Compiler	gcc 4.2.1
Optimization option	-Ofast

Table 3.3: Encryption speeds

Algorithm	Speed(Mbps)
VSC128	1202.254889
VSC 2.0	1039.222464
VSC 2.1	1113.193866
AES-128 ECB	366.901621

1. Select an input randomly. (We call the input I_1.)
2. Select a bit of I_1 and reverse the bit. (We call the value I_2.)
3. Apply the preprocessing to I_1 and I_2. (We call the outputs I'_1 and I'_2 respectively.)
4. Measure the Hamming distance between I'_1 and I'_2.
5. Repeat step 1-4 1000000 times. Calculate the average of the Hamming distance which are measure at step 3.

As a result, we got Table 3.4. Since the output length of the both preprocessing are 128bit, the result shows that the preprocessing have a good property.

Table 3.4: Hamming distance

Algorithm	Average Hamming distance
VSC 2.0	64.000107
VSC 2.1	63.995965

3.6.3 Randomness of key stream

We performed randomness test described by NIST SP800-22 [35] for key streams generated by VSC128, VSC 2.0 and VSC 2.1. The test was performed for 11 sets. Each set is constructed of 1000 sequences. (Exceptionally, the sets 10 and 11 are constructed of 255 sequences respectively. A sequence of the set 10 is generated with an initial condition (key and initial vector) whose one bit is "1" and the others are "0". VSC 2.0 requires the least bit of D is "0". Then, the set 10 is constructed of only 255 sequences. A sequence of the set 11 is generated with an initial condition whose two bits are "0" and the others are "1". One of the two is the least bit of D. Then, set 11 is also constructed of only

255 sequences.) Each sequence is constructed of 1000000bits, which are generated by VSC128, VSC 2.0 or VSC 2.1 with a secret key and an initial vector. The secret key and the initial vector are chosen randomly, but random pattern is dependent on a set. Table 3.5 shows the result. The randomness test is constructed of 188 test items. Even if sequences are exactly random, there are cases that the sequence does not pass all test items. Therefore, the result show that there are no problem about randomness of the key stream of VSC128, VSC 2.0 and VSC 2.1.

Table 3.5: Results of randomness tests

	Numbers of test items which the set passed		
Set No.	VSC128	VSC 2.0	VSC 2.1
1	188	188	187
2	188	187	188
3	188	186	188
4	187	188	188
5	187	188	188
6	188	187	188
7	188	187	188
8	188	188	187
9	188	188	188
10	188	188	188
11	188	188	188

In addition, we performed statistical test proposed in Appendix A for VSC128, VSC 2.0 and VSC 2.1. We performed the test for 100 sets per one generator. Each set consists 1000 sequences and each sequence consists 1000000 bits. As the results, all sets passed the test. These results also show that there are no problem about randomness of the key stream of VSC128, VSC 2.0 and VSC 2.1.

3.7 Summary

The original VSC was developed based on the chaos theory. Some theoretical attacks for VSC have been reported so far. Here, we proposed VSC 2.0 which are based on VSC, and it dramatically improves the security of VSC. As a main result, the securities of VSC 2.0 for the linear attack are provable. VSC 2.0 are very fast because of the simplicity of the algorithm, although it is little bit slower than the original VSC. VSC 2.0 can be also used as a pseudo random numbers generator. We think that chaos encryption algorithm based on the chaos theory is of important class of encryption algorithm. VSC 2.0 are rare chaotic encryption algorithm with *provable security*. We should proceed further in investigating proven security properties such as the distribution of periods of key stream for the chaotic encryption algorithm VSC 2.0. We also proposed further improving of VSC as a certain optimization of cipher design. Our proposed VSC 2.1 is faster than VSC 2.0. We think that VSC 2.1 is more secure than VSC 2.0 because of the following two reasons. First is that the key length of VSC 2.1 is longer than that of VSC 2.0. Second is that any theoretical attacks which were reported as workable attacks for the original VSC128 are not workable for VSC 2.1. In particular, VSC 2.1 has the provable security for the distinguishing attack with linear masking. Thus, VSC 2.1 is a precious example of secure chaotic cipher which is using permutation polynomials over a ring of modulo 2^w, and suggests that the permutation polynomials and chaos theory are useful for cryptography.

Chapter 4

One-stroke polynomials over a ring of modulo 2^w

There are two important studies about permutation polynomials over a ring of modulo 2^w. One is about periods of the polynomials. For cryptography and PRNG, such periods are expected to be longer. Then, a necessary and sufficient condition to maximize the periods of the permutation polynomials should be explored. When the period of the permutation polynomial over the ring is maximized, there exists only one orbit passed by the polynomial over the ring and the orbit passes all the elements of the ring. Since a map which draws such only one orbit is called "one-stroke map" [36], we call such permutation polynomials over the ring "one-stroke polynomials over a ring of modulo 2^w" in this thesis. The necessary and sufficient condition that specifies one-stroke polynomials with the assumption that the degree of the permutation polynomials are restricted to 1 or 2 is known [37, 21]. One-stroke polynomials whose degrees are 1 or 2 are used in a linear congruential method and a quadratic congruential method, which are pseudo random number generators. A sufficient condition without any assumption has also been known [37], but a necessary and sufficient condition without the assumption has not been known as far as the authors know.

The other is more fundamental. In order to study about permutation polynomials over a ring of modulo 2^w, we should know which polynomials are permutation polynomials over the ring. The necessary and sufficient condition that specifies permutation polynomials have been already studied [38].

Based on the above, we study about the one-stroke polynomials over a ring of modulo 2^w whose degrees are arbitrary.

4.1 Permutation polynomials over a ring of modulo 2^w

In this section, we introduce some properties of permutation polynomials over a ring of modulo 2^w. The necessary and sufficient condition that specifies permutation polynomials over the ring is given by the following theorem [38]:

Theorem 4.1.1. [Rivest, 2001] A polynomial $f(X) = \sum_{i=0}^{N} a_i X^i$, where the coefficients are integers, is a permutation polynomial over a ring of modulo 2^w if and only if

$$a_1 \equiv 1 \mod 2, \tag{4.1}$$

$$(a_2 + a_4 + a_6 + \cdots) \equiv 0 \mod 2, \tag{4.2}$$

$$(a_3 + a_5 + a_7 + \cdots) \equiv 0 \mod 2. \tag{4.3}$$

The following lemma is used in order to prove Theorem 4.1.1. We also use the lemma in the next section.

Lemma 4.1.2. Let $f(X)$ is a polynomial with integer coefficients. Then, $f(X)$ is a permutation polynomial over a ring of modulo 2^w if and only if

$$\forall \bar{X},\ \forall w \geq 1,\ f(\bar{X} + 2^{w-1}) \equiv f(\bar{X}) + 2^{w-1} \mod 2^w.$$

The following lemma is also used in the next section.

Lemma 4.1.3. Let $f(X)$ is a permutation polynomial over a ring of modulo 2^w. Then, $f^j(X)$ is also a permutation polynomial over the ring for arbitrary integer j, where $f^j(X) := f \circ f^{j-1}(X)$ and $f^1(X) := f(X)$.

4.2 One-stroke polynomial over a ring of modulo 2^w

In this section, we derive a necessary and sufficient condition that coefficients of one-stroke polynomials over a ring of modulo 2^w satisfy. Firstly, we exactly define one-stroke polynomials over a ring of modulo 2^w.

Definition 4.2.1. Let $f(X)$ is a permutation polynomial over a ring of modulo 2^w. If $f(X)$ satisfy

$$\forall w \geq 1,\ \forall \bar{X},\ \{f^i(\bar{X}) \mod 2^w | i \in \mathbb{Z}/2^w\mathbb{Z}\} = \mathbb{Z}/2^w\mathbb{Z},$$

$f(X)$ is called a one-stroke polynomial over a ring of modulo 2^w.

Example 4.2.1. We consider polynomials $F(X) = 4X^3 + X + 1$ and $G(X) = 6X^3 + 2X^2 + X + 1$. Both of them are permutation polynomials over a ring of modulo 2^w. Fig. 4.1 and 4.2 show orbits on a ring of modulo 2^w passed by $F(X)$ and $G(X)$, respectively. In Fig. 4.1, each orbit passes all elements of the ring where the orbit is passed on. It means that $F(X)$ is a one-stroke polynomial over a ring of modulo 2^w. On the other hand, $G(X)$ is not a one-stroke polynomial over a ring of modulo 2^w because there is not an orbit which passes all elements of $\mathbb{Z}/2^3\mathbb{Z}$.

$$\begin{tikzcd}
0 \arrow[r] & 1 \arrow[d] \\
3 \arrow[u] & 2 \arrow[l]
\end{tikzcd}$$

(a)

$$\begin{tikzcd}
0 \arrow[r] & 1 \arrow[r] & 6 \arrow[r] & 7 \arrow[d] \\
3 \arrow[u] & 2 \arrow[l] & 5 \arrow[l] & 4 \arrow[l]
\end{tikzcd}$$

(b)

$$\begin{tikzcd}
0 \arrow[r] & 1 \arrow[r] & 6 \arrow[r] & 7 \arrow[r] & 4 \arrow[r] & 5 \arrow[r] & 10 \arrow[d] \\
3 \arrow[u] & & & & & & 11 \arrow[d] \\
2 \arrow[u] & 13 \arrow[l] & 12 \arrow[l] & 15 \arrow[l] & 14 \arrow[l] & 9 \arrow[l] & 8 \arrow[l]
\end{tikzcd}$$

(c)

Figure 4.1: Orbits passed by $F(X)$. (a) Orbit on $\mathbb{Z}/2^2\mathbb{Z}$. (b) Orbit on $\mathbb{Z}/2^3\mathbb{Z}$. (c) Orbit on $\mathbb{Z}/2^4\mathbb{Z}$.

Next, we introduce some lemmas. By the definition, the following two lemmas are obviously true.

$$\begin{tikzcd}0 \arrow[r] & 1 \arrow[d] \\ 3 \arrow[u] & 2 \arrow[l]\end{tikzcd}$$

(a)

$$\begin{tikzcd}0 \arrow[r] & 1 \arrow[d] & 4 \arrow[r] & 5 \arrow[d] \\ 3 \arrow[u] & 2 \arrow[l] & 7 \arrow[u] & 6 \arrow[l]\end{tikzcd}$$

(b)

Figure 4.2: Orbits passed by $G(X)$. (a) Orbit on $\mathbb{Z}/2^2\mathbb{Z}$. (b) Orbit on $\mathbb{Z}/2^3\mathbb{Z}$.

Lemma 4.2.1. Let $f(X)$ is a permutation polynomial over a ring of modulo 2^w. Then, $f(X)$ is a one-stroke polynomial over the ring if and only if

$$f^i(0) \equiv 0 \mod 2^w \Leftrightarrow i \equiv 0 \mod 2^w.$$

Lemma 4.2.2. Let $f(X)$ is a permutation polynomial over a ring of modulo 2^w. Then, $f(X)$ is a one-stroke polynomial over the ring if and only if

$$f^{2^w}(0) \equiv 0 \mod 2^w,$$
$$f^{2^{w-1}}(0) \not\equiv 0 \mod 2^w.$$

Lemma 4.2.3. Let $f(X)$ is a permutation polynomial over a ring of modulo 2^w. Then, $f(X)$ is a one-stroke polynomial over the ring if and only if

$$\forall w \geq 1,\ f^{2^{w-1}}(0) \equiv 2^{w-1} \mod 2^w. \tag{4.4}$$

Proof. Assume that $f(X)$ is a one-stroke polynomial over the ring. By the definition,

$$\forall w \geq 1,\ \exists i \leq 2^w,\ \text{s.t. } f^i(0) \equiv 2^{w-1} \mod 2^w.$$

Then, by Lemma 4.1.2 and 4.1.3,

$$f^{2i}(0) \equiv f^i(2^{w-1}) \mod 2^w \equiv 0 \mod 2^w.$$

By Lemma 4.2.1, $2i = 2^w$. Then, $i = 2^{w-1}$.

Conversely, assume that (4.4) is true. Then, by Lemma 4.1.2 and 4.1.3,

$$f^{2^w}(0) \equiv f^{2^{w-1}}(2^{w-1}) \mod 2^w \equiv 0 \mod 2^w.$$

By Lemma 4.2.2, $f(X)$ is a one-stroke polynomial over the ring. □

Corollary 4.2.4. Let $f(X)$ is a permutation polynomial over a ring of modulo 2^w. Then, $f(X)$ is a one-stroke polynomial over the ring if and only if

$$\forall \bar{X},\ \forall w \geq 1,\ f^{2^{w-1}}(\bar{X}) \equiv \bar{X} + 2^{w-1} \mod 2^w.$$

Lemma 4.2.5. Assume that $f(X)$ is a permutation polynomial over a ring of modulo 2^w and $f(X)$ satisfy $f^2(0) \equiv 2 \mod 4$ and $f^4(0) \equiv 4 \mod 8$. Then,

$$\forall w \geq 2,\ f^{2^{w-1}}(0) \equiv 2^{w-1} \mod 2^w.$$

Proof. Assume that $f^2(X) = \sum b_i X^i$ and $f^4(X) = \sum c_i X^i$, where all b_i and c_i are integers. By the assumption of the lemma, $b_0 \equiv 2 \mod 4$ and $c_0 \equiv 4 \mod 8$. Since $f(X)$ is a permutation polynomial over the ring, by Lemma 4.1.3, $f^2(X)$ is also permutation polynomial over the ring. Then, by the Theorem 4.1.1, $b_1 \equiv 1 \mod 2$. Since $f^4(X) = f^2 \circ f^2(X)$,

$$\begin{aligned} c_1 =& b_1^2 + 2b_2b_1b_0 + 3b_3b_1b_0^2 + 4b_4b_1b_0^3 + \cdots \\ \equiv& b_1^2 \mod 4 \ \ (\because b_0 \equiv 2 \mod 4) \\ \equiv& 1 \mod 4 \ \ (\because b_1 \equiv 1 \mod 2). \end{aligned}$$

Assume that there exists an integer $\bar{w} \geq 3$ such that $f^{2^{\bar{w}-1}}(0) \equiv 2^{\bar{w}-1} \mod 2^{\bar{w}}$ and the first degree's coefficient of the $f^{2^{\bar{w}-1}}(X)$ is 1 under modulo 4. We express $f^{2^{\bar{w}-1}}(X)$ and $f^{2^{\bar{w}}}(X)$ as $f^{2^{\bar{w}-1}}(X) = \sum d_i X^i$ and $f^{2^{\bar{w}}}(X) = \sum e_i X^i$, where all d_i and e_i are integers. By the assumption, $d_1 \equiv 1 \mod 4$ and $d_0 \equiv 2^{\bar{w}-1} \mod 2^{\bar{w}}$.

$$\begin{aligned} e_1 =& d_1^2 + 2d_2d_1d_0 + 3d_3d_1d_0^2 + 4d_4d_1d_0^3 + \cdots \\ \equiv& d_1^2 \mod 4 \ \ (\because d_0 \equiv 2^{\bar{w}-1} \mod 2^{\bar{w}}) \\ \equiv& 1 \mod 4 \ \ (\because d_1 \equiv 1 \mod 2), \\ e_0 =& d_0 + d_1d_0 + d_2d_0^2 + d_3d_0^3 + \cdots \\ \equiv& d_0 + d_0d_1 \mod 2^{\bar{w}+1} \ \ (\because d_0 \equiv 2^{\bar{w}-1} \mod 2^{\bar{w}}) \\ \equiv& 2^{\bar{w}} \mod 2^{\bar{w}+1} \ \ (\because d_1 \equiv 1 \mod 4). \end{aligned}$$

Then, $f^{2^{\bar{w}}}(0) \equiv 2^{\bar{w}} \mod 2^{\bar{w}+1}$ and the first degree's coefficient of $f^{2^{\bar{w}}}(X)$ is 1 under modulo 4.

From the above, the lemma is true. □

Finally, we introduce a necessary and sufficient condition that specifies one-stroke polynomials over a ring of modulo 2^w.

Theorem 4.2.6. Let $f(X) = \sum_{i=0}^{N} a_i X^i$ is a polynomial, where all a_i are integers. Then, $f(X)$ is a one-stroke polynomial over a ring of modulo 2^w if and only if

$$\begin{aligned} a_0 \equiv& 1 \mod 2, \\ a_1 \equiv& 1 \mod 2, \\ (a_2 + a_4 + a_6 + \cdots) \equiv& 0 \mod 2, \\ (a_3 + a_5 + a_7 + \cdots) \equiv& 2a_2 \mod 4, \\ (a_1 + a_2 + a_3 + \cdots) \equiv& 1 \mod 4. \end{aligned}$$

Proof. If $f(X)$ is a one-stroke polynomial over the ring, $f(X)$ is a permutation polynomial over the ring. Then, by Theorem 4.1.1, Lemmas 4.2.3 and 4.2.5, $f(X)$ is a one-stroke polynomial over the ring if and only if (4.1), (4.2), (4.3) and

$$f(0) \equiv 1 \mod 2, \ f^2(0) \equiv 2 \mod 4, \ f^4(0) \equiv 4 \mod 8.$$

Since $f(0) = a_0$,

$$f(0) \equiv 1 \mod 2 \Leftrightarrow a_0 \equiv 1 \mod 2.$$

Since $f^2(0) = a_0 + a_1a_0 + a_2a_0^2 + \cdots + a_Na_0^N$, if $a_0 \equiv 1 \mod 2$, (4.1) and (4.3),

$$\begin{aligned} f^2(0) \equiv& a_0(1 + a_1 + a_3 + a_5 + \cdots) + (a_2 + a_4 + a_6 + \cdots) \mod 4 \\ \equiv& 1 + a_1 + a_2 + a_3 + \cdots + a_N \mod 4. \end{aligned}$$

Then,

$$\begin{aligned}
&f^2(0) \equiv 2 \mod 4,\ a_0 \equiv 1 \mod 2,\ (4.1) \text{ and } (4.3)\\
\Leftrightarrow\ &(a_1 + a_2 + a_3 + \cdots) \equiv 1 \mod 4,\ a_0 \equiv 1 \mod 2,\ (4.1) \text{ and } (4.3).
\end{aligned}$$

We express $f^2(X)$ as $f^2(X) = \sum b_i X^i$, where all b_i are integers. If $f(X)$ is a permutation polynomial over the ring, $f^2(X)$ is also a permutation polynomial over the ring by Lemma 4.1.3, and so $b_1 \equiv 1 \mod 2$ by Theorem 4.1.1. If $b_0 \equiv 2 \mod 4$ and $b_1 \equiv 1 \mod 2$,

$$\begin{aligned}
f^4(0) =& b_0 + b_1 b_0 + b_2 b_0^2 + b_3 b_0^3 + \cdots\\
\equiv& 2(1 + b_1 + 2b_2) \mod 8.
\end{aligned}$$

If $a_0 \equiv 1 \mod 2$, (4.1), (4.2) and (4.3),

$$\begin{aligned}
b_2 =& a_2 a_1 + \sum_{i=2}^{N} a_i \left\{ \frac{i(i-1)}{2} a_1^2 a_0^{i-2} + i a_2 a_0^{i-1} \right\}\\
\equiv& a_2 + \sum_{i=2}^{N} a_i \left\{ \frac{i(i-1)}{2} + i a_2 \right\} \mod 2\\
\equiv& a_2 + \sum_{i=2}^{N} a_i \left\{ \frac{i(i-1)}{2} \right\} \mod 2\ \ (\because (4.3))\\
\equiv& a_2 + (a_3 + a_7 + a_{11} + \cdots) + (a_2 + a_6 + a_{10} + \cdots) \mod 2\\
\equiv& (a_3 + a_7 + a_{11} + \cdots) + (a_6 + a_{10} + a_{14} \cdots) \mod 2,\\
b_1 =& a_1^2 + 2a_2 a_1 a_0 + 3a_3 a_1 a_0^2 + \cdots + N a_N a_1 a_0^N\\
\equiv& 1 + a_1(3a_3 + 5a_5 + 7a_7 + \cdots) + a_1 a_0 (2a_2 + 4a_4 + 6a_6 \cdots) \mod 4\\
\equiv& 1 + a_1(3a_3 + a_5 + 3a_7 + \cdots) + a_1 a_0 (2a_2 + 2a_6 + 2a_{10} \cdots) \mod 4\\
\equiv& 1 + 2a_1(a_3 + a_7 + a_{11} \cdots) + a_1(a_3 + a_5 + a_7 + \cdots) + 2(a_2 + a_6 + a_{10} \cdots) \mod 4\\
\equiv& 1 + 2a_2 + 2(a_3 + a_7 + a_{11} + \cdots) + (a_3 + a_5 + a_7 + \cdots) + 2(a_6 + a_{10} + a_{14} \cdots) \mod 4.
\end{aligned}$$

Then,

$$f^4(0) \equiv 4 + 2\{2a_2 + (a_3 + a_5 + a_7 + \cdots)\} \mod 8.$$

Therefore,

$$\begin{aligned}
&f^4(0) \equiv 4 \mod 8, b_0 \equiv 2 \mod 4, (4.1), (4.2) \text{ and } (4.3)\\
\Leftrightarrow& (a_3 + a_5 + a_7 + \cdots) \equiv 2a_2 \mod 4,\ b_0 \equiv 2 \mod 4, (4.1), (4.2) \text{ and } (4.3).
\end{aligned}$$

From the above, the theorem is true. □

Example 4.2.2. Let $f(X) = 4X^3 + X + 1$ and $g(X) = 2X^3 + X + 1$. The both of them are permutation polynomials over a ring of modulo 2^w. Theorem 4.2.6 states that $f(X)$

is a one-stroke polynomial over the ring and $g(X)$ is not. Indeed,

$$\begin{aligned}
f(0) &\equiv 1 \mod 8,\\
f(1) &\equiv 6 \mod 8,\\
f(6) &\equiv 7 \mod 8,\\
f(7) &\equiv 4 \mod 8,\\
f(4) &\equiv 5 \mod 8,\\
f(5) &\equiv 2 \mod 8,\\
f(2) &\equiv 3 \mod 8,\\
f(3) &\equiv 0 \mod 8.
\end{aligned}$$

Then, the period of the orbit which $f(X)$ draws over $\mathbb{Z}/2^w\mathbb{Z}$ is maximum. On the other hand,

$$\begin{aligned}
g(0) &\equiv 1 \mod 8,\\
g(1) &\equiv 4 \mod 8,\\
g(4) &\equiv 5 \mod 8,\\
g(5) &\equiv 0 \mod 8.
\end{aligned}$$

Then, $g(X)$ is not a one-stroke polynomial over the ring. The results are consistent with Theorem 4.2.6.

4.3 Some properties of one-stroke polynomials over a ring of modulo 2^w

In this section, we introduce some properties of one-stroke polynomials over a ring of modulo 2^w.

4.3.1 Commutativity

We show a theorem about commutativity of one-stroke polynomials.

Theorem 4.3.1. Assume that $f(X)$ is a one-stroke polynomial over a ring of modulo 2^w and $g(X)$ is a permutation polynomial over the ring. If there exist $\bar{w} \geq 1$ such that

$$\forall \bar{X} \in \mathbb{Z}/2^{\bar{w}}\mathbb{Z},\ f \circ g(\bar{X}) \equiv g \circ f(\bar{X}) \mod 2^{\bar{w}},$$

then, there is exists $j \leq 2^{\bar{w}}$ such that

$$\forall X,\ g(X) \equiv f^j(X) \mod 2^{\bar{w}}.$$

Proof. Assume that $\bar{X}$ is a fixed value in $\mathbb{Z}/2^{\bar{w}}\mathbb{Z}$. Since $f(X)$ is a one-stroke polynomial,

$$\exists j \leq 2^{\bar{w}} \text{ s.t. } g(\bar{X}) \equiv f^j(\bar{X}) \mod 2^{\bar{w}}.$$

If $f^i \circ g(\bar{X}) \equiv g \circ f^i(\bar{X}) \mod 2^{\bar{w}}$,

$$\begin{aligned}
f^{i+1} \circ g(\bar{X}) =& f \circ \left(f^i \circ g(\bar{X})\right)\\
=& f \circ \left(g \circ f^i(\bar{X})\right)\\
=& f \circ g\left(f^i(\bar{X})\right)\\
=& g \circ f\left(f^i(\bar{X})\right)\\
\equiv& g \circ f^{i+1}(\bar{X}) \mod 2^{\bar{w}}.
\end{aligned}$$

Then,

$$\begin{aligned} g\left(f^i(\bar{X})\right) \equiv& f^i \circ g(\bar{X}) \mod 2^{\bar{w}} \\ \equiv& f^j\left(f^i(\bar{X})\right) \mod 2^{\bar{w}}. \end{aligned}$$

Then, since i is arbitrary and $f(X)$ is a one-stroke polynomial,

$$\forall X,\ g(X) \equiv f^j(X) \mod 2^{\bar{w}}.$$

□

The theorem is about one-stroke polynomials over a ring of modulo 2^w, but it can be easily more generalized. It is, however, not the theme of this thesis.

4.3.2 Calculation methods

Under the assumption that the degree of one-stroke polynomial $f(X)$ is lower than w, we show that following values can be calculated with polynomial order times of w.

(A) $\bar{X}$ satisfying $\bar{Y} \equiv f(\bar{X}) \mod 2^w$ for given $\bar{Y}$.

(B) j satisfying $\bar{Y} \equiv f^j(\bar{X}) \mod 2^w$ for given $\bar{X}$ and $\bar{Y}$.

(C) $\bar{Y}$ satisfying $\bar{Y} \equiv f^j(\bar{X}) \mod 2^w$ for given $\bar{X}$ and j.

In chapter 2, similar problem for permutation polynomials over the ring is discussed. Here, we use not only properties of permutation polynomials over the ring but also those of one-stroke polynomials over the ring.

Method to calculate (A).

The following algorithm can calculate (A).

```
1 Set $X' \leftarrow 0$ and $m \leftarrow 1$.
2 If $\bar{Y} \not\equiv f(X') \mod 2^m$, $X' \leftarrow 2^{m-1}$.
3 If $m = w$, output $X'$ and finish this algorithm. Else, $m \leftarrow m + 1$ and return to (ii).
```

In the step 2, if $\bar{Y} \equiv f(X') + 2^{m-1} \mod 2^m$, $\bar{Y} \equiv f(X' + 2^{m-1}) \mod 2^m$ by Lemma 4.1.2. Therefore, this algorithm can calculate (A).

Since the degree of $f(X)$ is lower than w, it requires $O(w)$ multiplications and $O(w)$ additions on $\mathbb{Z}/2^w\mathbb{Z}$ to calculate the value of $f(X) \mod 2^w$ for given X. Thus, the calculation requires $O(w^3)$ times. Since the calculation is used $O(w)$ times in the above algorithm, the above algorithm requires $O(w^4)$ times.

Method to calculate (B).

In order to calculate (B), we introduce polynomials $h_{2^i}(X)$ $(i = 0, 1, 2, \cdots, w-1)$ described as

$$h_{2^i}(X) := \left(f^{2^i}(X) \mod 2^w\right) \mod X^{\lceil \frac{w}{i} \rceil}.$$

The polynomials $h_{2^i}(X)$ have the following properties. If $\bar{X} \equiv 0 \mod 2^i$,

$$h_{2^i}(\bar{X}) \equiv f^{2^i}(\bar{X}) \mod 2^w.$$

If $\bar{X} \equiv 0 \mod 2^{i+1}$,

$$h_{2^i}(\bar{X}) \equiv 2^i \mod 2^{i+1},$$

and if $\bar{X} \equiv 2^i \mod 2^{i+1}$,

$$h_{2^i}(\bar{X}) \equiv 0 \mod 2^{i+1}.$$

If we know $h_{2^i}(X)$, we can calculate $h_{2^{i+1}}(X)$ as $h_{2^{i+1}}(X) = h_{2^i} \circ h_{2^i}(X) \mod X^{\lceil \frac{w}{i+1} \rceil}$. Because the degrees of $h_{2^i}(X)$ and $h_{2^{i+1}}(X)$ are lower than $\lceil \frac{w}{i} \rceil$, the calculation requires $O(\lceil \frac{w}{i} \rceil^3)$ multiplications and $O(\lceil \frac{w}{i} \rceil^3)$ additions. Then, the calculation requires $O(w^2 \lceil \frac{w}{i} \rceil^3)$ times.

By the estimation, it takes $O(w^5)$ times to calculate the list $\{h_{2^0}(X),\ h_{2^1}(X), h_{2^2}(X),\ \cdots,\ h_{2^{w-1}}(X)\}$.

We show a method to calculate (B) by using $h_{2^i}(X)$. If we find j' and j'' such that

$$0 \equiv f^{j'}(\bar{Y}) \mod 2^w \text{ and } 0 \equiv f^{j''}(\bar{X}) \mod 2^w,$$

we can calculate as $j \equiv j'' - j' \mod 2^w$. We, therefore, assume that $\bar{Y}$ equals to 0 without loss of generality. Assume that $j = \sum_{i=0}^{w-1} \epsilon(i) 2^i$ where $\epsilon(i) \in \{0, 1\}$. Then, $f^j(\bar{X}) \equiv f^{\epsilon(w-1)2^{w-1}} \circ f^{\epsilon(w-2)2^{w-2}} \circ \cdots \circ f^{\epsilon(0)2^0}(\bar{X}) \mod 2^w$. By Lemma 4.2.1, if $f^j(X) \equiv 0 \mod 2^w$, then $f^{\epsilon(m)2^m} \circ f^{\epsilon(m-1)2^{m-1}} \circ \cdots \circ f^{\epsilon(0)2^0}(\bar{X}) \equiv 0 \mod 2^{m+1}$ for arbitrary m. Thus, by the properties of $h_{2^i}(X)$,

$$f^j(\bar{X}) \equiv h_{2^{w-1}}^{\epsilon(w-1)} \circ h_{2^{w-2}}^{\epsilon(w-2)} \circ \cdots h_{2^0}^{\epsilon(0)} \mod 2^w.$$

From the above, the following algorithm outputs j satisfying $f^j(\bar{X}) \equiv 0 \mod 2^w$.

1 Set $i \leftarrow 0$, $j \leftarrow 0$ and $X' = \bar{X}$.

2 If $X' \equiv 2^i \mod 2^{i+1}$, $X' \leftarrow h_{2^i}(X') \mod 2^w$ and $j \leftarrow j + 2^i$.

3 If $i = w - 1$, output j and finish this algorithm. Else, $i \leftarrow i + 1$ and return to step 2.

It takes $O(w^2 \lceil \frac{w}{i} \rceil)$ times to calculate the value of $h_{2^i}(\bar{X})$ for given $\bar{X}$. Then, this algorithm requires $O(w^3 \log w)$ times, but calculating (B) requires $O(w^5)$ because we must calculate the list $\{h_{2^0}(X),\ h_{2^1}(X), h_{2^2}(X),\ \cdots,\ h_{2^{w-1}}(X)\}$.

Method to calculate (C).

By using the above algorithm, we can find j' such that $f^{j'}(\bar{X}) \equiv 0 \mod 2^w$, and so it is enough to show an algorithm to calculate $f^k(0) \mod 2^w$ for given k. Assume that $k = \sum_{i=0}^{w-1} \epsilon(i) 2^i$ where $\epsilon(i) \in \{0, 1\}$. Then, $f^k(0) \equiv f^{\epsilon(0)2^0} \circ f^{\epsilon(1)2^1} \circ \cdots \circ f^{\epsilon(w-1)2^{w-1}}(0) \mod 2^w$. By Lemma 4.2.1, $f^{\epsilon(m)2^m} \circ f^{\epsilon(m+1)2^{m+1}} \circ \cdots \circ f^{\epsilon(w-1)2^{w-1}}(0) \equiv 0 \mod 2^{m+1}$ for arbitrary m. Thus, by the properties of $h_{2^i}(X)$,

$$f^k(0) \equiv h_{2^0}^{\epsilon(0)} \circ h_{2^1}^{\epsilon(1)} \circ \cdots h_{2^{w-1}}^{\epsilon(w-1)} \mod 2^w.$$

Then, the following algorithm outputs $f^k(0) \mod 2^w$.

```
1 Set $i \leftarrow w-1$, $X' = 0$.
2 If $(i+1)$-th least significant bit of $k$ is 1, $X' \leftarrow h_{2^i}(X') \mod 2^w$.
3 If $i = 0$, output $X'$ and finish this algorithm. Else, $i \leftarrow i-1$ and return to step 2.
```

This algorithm also requires $O(w^3 \log w)$ times, but calculating (C) requires $O(w^5)$ by the same reason why the method to calculate (B) requires $O(w^5)$ times.

4.4 Summary

We derived the necessary and sufficient condition to specify one-stroke polynomials over a ring of modulo 2^w. The condition enables us to construct many long sequences with maximum periods such that the distribution of points of the sequences are uniform over the ring. In addition, one-stroke polynomials have some interesting properties. One-stroke polynomials will be applied for many fields, including cryptography and PRNG.

Chapter 5

Methods of combining one-stroke polynomials over a ring of modulo 2^w for pseudo random number generator and stream cipher

In the former chapter, we introduced the new class of permutation polynomials "one-stroke polynomials over a ring of modulo 2^w". They should not, however, be applied for stream cipher and PRNG as they are. A system using only one one-stroke polynomial over the ring is too simple, and so there are many cases that a sequence made by one-stroke polynomial does not have good randomness and that inner state of the PRNG can be calculated [39, 40, 41]. We, therefore, discuss how to apply the polynomials to stream cipher and PRNG, in this chapter. Some one-stroke polynomials over the ring should be used. Then, we propose a method of combining one-stroke polynomials over the ring. Orbits made by the polynomials have long periods, and the property is useful for the applications. We should propose a method which can utilize the property. In the method, the period is preserved.

In addition, we propose a combining method made a period longer than that of the original one-stroke polynomials. The reason why we propose the method is that we need a longer period than 2^w in many cases, where w is the length of the variable used in Stream cipher and PRNG. By the method, we can construct a PRNG which has *arbitrary* period.

The rapidity of computation is also useful for the applications. Then, the methods proposed in this chapter can be performed by parallel computing.

5.1 Reproduct property

Lemma 4.2.3 states that an orbit made by a one-stroke polynomial over a ring of modulo 2^w has a recursive property. The property essentially relates to the periodicity. Then, we introduce some lemmas relating to the property for later sections.

Lemma 5.1.1. Assume that $f(X)$ is a one-stroke polynomial over a ring of modulo 2^w. Then,

$$\forall \bar{X},\ f(\bar{X}+1) \equiv f(\bar{X}) + 1 \mod 4.$$

Proof. Assume that $f(X) = \sum_{i=0}^{n} a_i X^i$. If $\bar{X} = 4k$ where k is an integer,

$$\begin{aligned} f(\bar{X}) =& \sum_{i=0}^{n} a_i (4k)^i \\ \equiv& a_0 \mod 4. \end{aligned}$$

If $\bar{X} = 4k + 1$ where k is an integer,

$$\begin{aligned} f(\bar{X}) =& \sum_{i=0}^{n} a_i (4k+1)^i \\ \equiv& \sum_{i=0}^{n} a_i \mod 4 \\ \equiv& a_0 + 1 \mod 4 \ (\because \text{Theorem 4.2.6}). \end{aligned}$$

If $\bar{X} = 4k + 2$ where k is an integer,

$$\begin{aligned} f(\bar{X}) =& \sum_{i=0}^{n} a_i (4k+2)^i \\ \equiv& a_0 + 2a_1 \mod 4 \\ \equiv& a_0 + 2 \mod 4 \ (\because \text{Theorem 4.2.6}). \end{aligned}$$

If $\bar{X} = 4k + 3$ where k is an integer,

$$\begin{aligned} f(\bar{X}) =& \sum_{i=0}^{n} a_i (4k+3)^i \\ \equiv& \sum_{i\text{: odd}} 3a_i + \sum_{i\text{: even}} a_i \mod 4 \\ \equiv& a_0 + \sum_{i=1}^{n} a_i + 2a_1 + 2(a_3 + a_5 + a_7 + \cdots) \mod 4 \\ \equiv& a_0 + 3 \mod 4 \ (\because \text{Theorem 4.2.6}). \end{aligned}$$

Then, the lemma is true. □

Lemma 5.1.2. Assume that $f(X)$ is a one-stroke polynomial over a ring of modulo 2^w. Then, there exists $C \in \{0, 1\}$ depending on $f(X)$ such that

$$\forall \bar{X},\ f(\bar{X} + 2) \equiv f(\bar{X}) + 2 + 4C \mod 8.$$

Proof. Assume that $f(X) = \sum_{i=0}^{n} a_i X^i$. Then,

$$f(\bar{X} + 2) \equiv \sum_{i=0}^{n} a_i \bar{X}^i + 2 \sum_{i=1}^{n} i a_i \bar{X}^{i-1} + 4 \sum_{i=2}^{n} \frac{i(i-1)}{2} a_i \bar{X}^{i-2} \mod 8.$$

Then,

$$\begin{aligned} f(\bar{X}_{odd} + 2) \equiv& f(\bar{X}_{odd}) + 2 \sum_{i=1}^{n} i a_i + 4 \sum_{i=2}^{n} \frac{i(i-1)}{2} a_i \mod 8, \\ f(\bar{X}_{even} + 2) \equiv& f(\bar{X}_{even}) + 2a_1 + 4a_2 \mod 8, \end{aligned}$$

where $\bar{X}_{odd}$ and $\bar{X}_{even}$ are arbitrary odd number and even number, respectively. By Theorem 4.2.6,

$$\begin{aligned}
&2\sum_{i=0}^{n} ia_i\\
\equiv&2a_1 + 2(a_3 + a_5 + a_7 + \cdots) + 4(a_3 + a_7 + a_{11} + \cdots)\\
&\qquad\qquad\qquad\qquad\qquad + 2^{m+1}(a_2 + a_6 + a_{10} + \cdots) \mod 8\\
\equiv&2a_1 + 2^{m+1}a_2 + 4(a_3 + a_7 + a_{11} + \cdots) + 2^{m+1}(a_2 + a_6 + a_{10} + \cdots) \mod 8,\\
&4\sum_{i=2}^{n} \frac{i(i-1)}{2}a_i\\
\equiv&4(a_2 + a_3 + a_6 + a_7 + a_{10} + a_{11} + \cdots) \mod 8.
\end{aligned}$$

Then, $2\sum_{i=1}^{n} ia_i + 4\sum_{i=2}^{n} \frac{i(i-1)}{2}a_i \equiv 2^m a_1 + 4a_2 \mod 8$.

Since a_1 is odd, $2a_1 + 4a_2 \equiv 2 \mod 4$.

From the above, the lemma is true. □

Lemma 5.1.3. Assume that $f(X)$ is a one-stroke polynomial over a ring of modulo 2^w. Then, there exists $C',\ C'' \in \{0,1\}$ depending on $f(X)$ such that

$$\begin{aligned}
\forall \bar{X}_{odd}\text{: odd number, } f(\bar{X}_{odd} + 2^m) \equiv& f(\bar{X}_{odd}) + 2^m + C'2^{m+1} \mod 2^{m+2},\\
\forall \bar{X}_{even}\text{: even number, } f(\bar{X}_{even} + 2^m) \equiv& f(\bar{X}_{even}) + 2^m + C''2^{m+1} \mod 2^{m+2},
\end{aligned}$$

for arbitrary integer $m \geq 2$.

Proof. Assume that $f(X) = \sum_{i=0}^{n} a_i X^i$. Then,

$$\begin{aligned}
f(\bar{X}_{odd} + 2^m) \equiv& f(\bar{X}_{odd}) + 2^m \sum_{i=1}^{n} ia_i \mod 2^{m+2},\\
f(\bar{X}_{even} + 2^m) \equiv& f(\bar{X}_{even}) + 2^m a_1 \mod 2^{m+2}.
\end{aligned}$$

By Theorem 4.2.6,

$$\begin{aligned}
&2^m\sum_{i=0}^{n} ia_i\\
\equiv&2^m a_1 + 2^m(a_3 + a_5 + a_7 + \cdots) + 2^{m+1}(a_3 + a_7 + a_{11} + \cdots)\\
&\qquad\qquad\qquad\qquad\qquad + 2^{m+1}(a_2 + a_6 + a_{10} + \cdots) \mod 2^{m+2}\\
\equiv&2^m a_1 + 2^{m+1}a_2 + 2^{m+1}(a_3 + a_7 + a_{11} + \cdots) + 2^{m+1}(a_2 + a_6 + a_{10} + \cdots) \mod 2^{m+2}\\
\equiv&2^m \mod 2^{m+1},\\
&2^m a_1\\
\equiv&2^m \mod 2^{m+1}.
\end{aligned}$$

Then, the lemma is true. □

5.2 Method of combining one-stroke polynomials over a ring of modulo 2^w with preserving the periodicity

In this section, we propose a method of combining some one-stroke polynomials over a ring of modulo 2^w. The purpose is to make a PRNG which can generate a sequence with good randomness. The reason why we use one-stroke polynomials is that an orbit made by a one-stroke polynomial is long and that such a long periodicity is in general needed to PRNG. Then, we discuss a combining method which can preserve the long period.

5.2.1 Proposed Method

Firstly, we introduce a concept which generalize the reproduct property of an orbit made by one-stroke polynomials over a ring of modulo 2^w.

Definition 5.2.1. If a sequence $\{x_n \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ satisfy

$$\forall n, \forall m < w,\ x_{n+2^{m+a}} \equiv x_n + 2^m \mod 2^{m+1},$$
$$\forall n,\ x_{n+2^{w+a}} \equiv x_n \mod 2^w,$$

then, we say that the sequence has (a, w)-reproduct property.

Of course, an orbit made by a one-stroke polynomial over $\mathbb{Z}/2^{\bar{w}}\mathbb{Z}$ is a sequence which has $(0, \bar{w})$-reproduct property. We should utilize (a, w)-reproduct property for PRNG. We expect that the following proposition will be useful.

Lemma 5.2.1. Assume that $\{x_n \in \mathbb{Z}/2^{w-1}\mathbb{Z}\}_{n\in\mathbb{Z}}$ is a sequence which has $(0, w)$-reproduct property, and that $\{y_n \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ is a sequence satisfying

$$\forall n,\ y_{n+1} = f(y_n) + 4Ax_n \mod 2^w,$$

where $f(X)$ is a one-stroke polynomial over a ring of modulo 2^w and A is an integer. Then, $\{y_n \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ has $(0, w)$-reproduct property.

Proof. By Corollary 4.2.4, for all n,

$$\begin{aligned} y_{n+1} &\equiv f(y_n) + 4Ax_n \mod 2 \\ &\equiv f(y_n) \mod 2 \\ &\equiv y_n + 1 \mod 2. \end{aligned}$$

Assume that $y_{n+1} \equiv y_n + 1 + 2d_n \mod 4$, where $d_n \in \{0, 1\}$. Then, by Lemma 4.1.2 and 5.1.1,

$$\begin{aligned} y_{n+2} &\equiv f(y_{n+1}) + 4Ax_n \mod 4 \\ &\equiv f(y_n + 1 + 2d_n) \mod 4 \\ &\equiv f(y_n) + 1 + 2d_n \mod 4 \\ &\equiv y_{n+1} + 1 + 2d_n \mod 4 \\ &\equiv y_n + 2 \mod 4. \end{aligned}$$

Assume that $y_{n+2} \equiv y_n + 2 + 4d_n \mod 8$, where $d_n \in \{0, 1\}$. Then, by Lemma 4.1.2 and 5.1.2,

$$\begin{aligned} y_{n+3} &\equiv f(y_{n+2}) + 4Ax_n \mod 8 \\ &\equiv f(y_n + 2 + 4d_n) \mod 8 \\ &\equiv f(y_n) + 2 + 4(d_n + C) + 4Ax_n \mod 8 \\ &\equiv y_{n+1} + 2 + 4(d_n + C) \mod 8, \\ y_{n+4} &\equiv y_{n+2} + 2 + 4(d_n + C + C) \mod 8 \\ &\equiv y_{n+2} + 2 + 4d_n \mod 8, \\ &\equiv y_n + 4 \mod 8, \end{aligned}$$

where $C \in \{0, 1\}$.

Assume that there is a natural number $m \geq 2$ satisfying

$$\forall n,\ y_{n+2^m} \equiv y_n + 2^m + 2^{m+1} d_n \mod 2^{m+2},$$

where $d_n \in \{0, 1\}$. Then, by Lemma 4.1.2 and 5.1.3,

$$\begin{aligned} y_{n+2^m+1} =& f(y_{n+2^m}) + 4Ax_{n+2^m} \mod 2^w \\ \equiv& f(y_n + 2^m + 2^{m+1} d_n) + 4A(x_n + 2^m) \mod 2^{m+2} \\ \equiv& f(y_n) + 2^m + 2^{m+1}(d_n + C_n) + 4Ax_n \mod 2^{m+2} \\ \equiv& y_{n+1} + 2^m + 2^{m+1}(d_n + C_n) \mod 2^{m+2}, \end{aligned}$$

where $C \in \{0, 1\}$ and it depends on which y_n is odd or even. Repeating similar calculations,

$$\begin{aligned} y_{n+2^{m+1}} =& y_{n+2^m+2^m} \\ \equiv& y_{n+2^m} + 2^m + 2^{m+1}(d_n + \sum_{i=0}^{2^m-1} C_{n+i}) \mod 2^{m+2} \\ \equiv& y_n + 2^m + 2^{m+1} + 2^{m+1} \sum_{i=0}^{2^m-1} C_{n+i} \mod 2^{m+2}. \end{aligned}$$

Since $y_{n+2} \equiv y_n \mod 2$ for all n,

$$\sum_{i=0}^{2^m-1} C_{n+i} \equiv 0 \mod 2.$$

Then,

$$y_{n+2^{m+1}} \equiv y_n + 2^m + 2^{m+1} \mod 2^{m+2}.$$

From the above the lemma is true. □

Based on Lemma 5.2.1, we consider the following combined system:

$$\begin{aligned} x_{n+1}^{(1)} =& f_1(x_n^{(1)}) + \sum_{k=0}^{K} C_{1,k} x_n^{(k)} \mod 2^w, \\ x_{n+1}^{(2)} =& f_2(x_n^{(2)}) + \sum_{k=0}^{K} C_{2,k} x_n^{(k)} \mod 2^w, \\ &\vdots \\ x_{n+1}^{(K)} =& f_K(x_n^{(K)}) + \sum_{k=0}^{K} C_{K,k} x_n^{(k)} \mod 2^w, \end{aligned}$$

where $f_i(X)$ $(i = 1, 2, \cdots, K)$ are one-stroke polynomials over a ring of modulo 2^w and $C_{i,k}$ $((i = 1, 2, \cdots, K), (k = 1, 2, \cdots, K))$ are integers satisfying $C_{i,k} \equiv 0 \mod 4$. The system is more complex than a simple system without combination. Then, we can expect that the sequences $\{x_n^{(i)} \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ $(i = 1, 2, \cdots, K)$ have good randomnesses.

A point which we should pay attention to is the periods of the sequences $\{x_n^{(i)} \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ $(i = 1, 2, \cdots, K)$. By Lemma 5.2.1, we can prove that the sequences $\{x_n^{(i)} \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ $(i = 1, 2, \cdots, K)$ satisfy $(0, w)$-reproduct condition and that periods of the sequences are 2^w. The period without combination is preserved.

Theorem 5.2.2. The sequences $\{x_n^{(i)} \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ $(i = 1, 2, \cdots, K)$ satisfy $(0, w)$-reproduct condition.

Proof. It is clear that

$$\forall i,\ \forall n,\ x_{n+1}^{(i)} \equiv x_n^{(i)} + 1 \mod 2.$$

Assume that there exists an non-negative number $m \le w-1$ such that $\{x_n^{(i)} \mod 2^m\}_{n\in\mathbb{Z}}$ $(i = 1, 2, \cdots, K)$ have $(0, m)$-reproduct property. Then, by Lemma 5.2.1, $\{x_n^{(i)} \mod 2^{m+1}\}_{n\in\mathbb{Z}}$ $(i = 1, 2, \cdots, K)$ have $(0, m+1)$-reproduct property. From the above, the theorem is true. □

If we choose the coefficients $C_{i,k}$ as the form $2^{e(i,k)}$ where $e(i, k)$ are non-negative integer, the combination does not increase the number of multiplications because $C_{i,k} \times x_n^{(i)}$ can be calculated by bit-sift. Moreover, In particular hardware implementation, the proposed method can be performed by parallel computation, in particular on hardware. These mean that the proposed method can be fast.

In addition, the additions on the proposed method can be replaced with exclusive OR. The replaced method keeps the periodicity and rapidity.

5.2.2 Generalization of the proposed method

The proposed method in the former section can be more generalized. Assume that $f_i(X) = \sum_{j=0}^{N} a_{i,j} X^j$ $(i = 1, 2, \cdots, K)$ are one-stroke polynomials over a ring of modulo 2^w, and that sequences $\{x_n^{(i)} \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ $(i = 1, 2, \cdots, K)$ satisfy

$$\begin{aligned}
x_{n+1}^{(1)} &= \sum_{j=0}^{N} \left\{ a_{1,j} + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{1,j,k,l}(x_n^{(k)})^l \right\} (x_n^{(1)})^j \mod 2^w \\
x_{n+1}^{(2)} &= \sum_{j=0}^{N} \left\{ a_{2,j} + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{2,j,k,l}(x_n^{(k)})^l \right\} (x_n^{(2)})^j \mod 2^w \\
&\vdots \\
x_{n+1}^{(K)} &= \sum_{j=0}^{N} \left\{ a_{K,j} + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{K,j,k,l}(x_n^{(k)})^l \right\} (x_n^{(K)})^j \mod 2^w
\end{aligned}$$

where $C_{i,j,kl}$ satisfy the following conditions:

- For arbitrary (i, j, k, l) except $j = 0$, $C_{i,j,k,l} \equiv 0 \mod 4$.
- For arbitrary (i, k, l), $C_{i,0,k,l} \equiv 0 \mod 2$.
- For arbitrary i, $\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l} \equiv 0 \mod 4$.
- For arbitrary i, $\sum_{k=1}^{K}\sum_{l:\text{odd number larger than }1} C_{i,0,k,l} \equiv 0 \mod 4$.

This system is further more complex than the system in the former section, and so the sequences made by this system are expected to have better randomness.

Theorem 5.2.3. The sequences $\{x_n^{(i)} \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ $(i = 1, 2, \cdots, K)$ described the above satisfy $(0, w)$-reproduct condition.

Proof. Since $f_i(X)$ $(i = 1, 2, \cdots, K)$ are one-stroke polynomial over a ring of modulo 2^w, by Corollary 4.2.4,

$$\forall i,\ \forall n,\ x_{n+1}^{(i)} \equiv x_n^{(i)} + 1 \mod 2.$$

Assume that, for all n and i,

$$x_{n+1}^{(i)} \equiv x_n^{(i)} + 1 + 2d_n^{(i)} \mod 4,$$

where $d_n \in \{0, 1\}$. Then, by Lemma 4.1.2 and 5.1.1,

$$\begin{aligned}
x_{n+1}^{(i)} \equiv& f_i(x_n^{(i)}) + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}(x_n^{(k)})^l \mod 4\\
\equiv& f_i(x_n^{(i)}) + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}x_n^{(k)} \mod 4,\\
x_{n+2}^{(i)} \equiv& f_i(x_{n+1}^{(i)}) + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}(x_{n+1}^{(k)})^l \mod 4\\
\equiv& f_i(x_n^{(i)} + 1 + 2d_n^{(i)}) + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}(x_n^{(k)} + 1) \mod 4\\
\equiv& f_i(x_n^{(i)}) + 1 + 2d_n^{(i)} + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}(x_n^{(k)} + 1) \mod 4\\
\equiv& x_{n+1}^{(i)} + 1 + 2d_n^{(i)} + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l} \mod 4\\
\equiv& x_n^{(i)} + 2 \mod 4.
\end{aligned}$$

Assume that, for all n and i,

$$x_{n+1}^{(i)} \equiv x_n^{(i)} + 2 + 4d_n^{(i)} \mod 8,$$

where $d_n \in \{0,1\}$. Then, by Lemma 4.1.2 and 5.1.2,

$$\begin{aligned}
&x_{n+3}^{(i)}\\
\equiv&\sum_{j=0}^{N}\left\{a_{i,j}+\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,j,k,l}(x_{n+2}^{(k)})^l\right\}(x_{n+2}^{(i)})^j \mod 8\\
\equiv&\sum_{j=0}^{N}\left\{a_{i,j}+\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,j,k,l}(x_{n}^{(k)}+2+4d_n^{(i)})^l\right\}(x_{n+2}^{(i)})^j \mod 8\\
\equiv&\sum_{j=0}^{N}\left\{a_{i,j}+\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,j,k,l}(x_{n}^{(k)})^l\right\}(x_{n+2}^{(i)})^j+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}l(x_n^{(k)})^{l-1} \mod 8\\
\equiv&\sum_{j=0}^{N}\left\{a_{i,j}+\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,j,k,l}(x_{n}^{(k)})^l\right\}(x_{n}^{(i)}+2+4d_n^{(i)})^j\\
&\qquad\qquad+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}l(x_n^{(k)})^{l-1} \mod 8\\
\equiv&f_i(x_n^{(i)}+2+4d_n^{(i)})+\sum_{j=0}^{N}\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,j,k,l}(x_n^{(k)})^l(x_n^{(i)})^j\\
&\qquad+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}(x_n^{(k)})^l+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}l(x_n^{(k)})^{l-1} \mod 8\\
\equiv&f_i(x_n^{(i)})+2+4(d_n^{(i)}+D_i)+\sum_{j=0}^{N}\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,j,k,l}(x_n^{(k)})^l(x_n^{(i)})^j\\
&\qquad+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}(x_n^{(k)})^l+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}l(x_n^{(k)})^{l-1} \mod 8\\
\equiv&x_{n+1}^{(i)}+2+4(d_n^{(i)}+D_i)\\
&\qquad+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}(x_n^{(k)})^l+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}l(x_n^{(k)})^{l-1} \mod 8,
\end{aligned}$$

where $D_i \in \{0,1\}$. In the same way,

$$\begin{aligned}
&x_{n+4}^{(i)}\\
\equiv&x_{n+2}^{(i)}+2+4(d_n^{(i)}+D_i+D_i)\\
&\qquad+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}(x_n^{(k)})^l+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}l(x_n^{(k)})^{l-1}\\
&\qquad+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}(x_{n+1}^{(k)})^l+2\sum_{k=1}^{K}\sum_{l=1}^{L}C_{i,0,k,l}l(x_{n+1}^{(k)})^{l-1} \mod 8\\
\equiv&x_{n+2}^{(i)}+2+4d_n^{(i)} \mod 8\\
\equiv&x_n^{(i)}+4 \mod 8.
\end{aligned}$$

Assume that there exists a natural number $2 \le m \le w-1$ such that $\{x_n^{(i)} \mod 2^m\}_{n\in\mathbb{Z}}$ $(i = 1,2,\cdots,K)$ have $(0,m)$-reproduct property. Then, for all n and i, there exist $d_n^{(i)} \in \{0,1\}$

such that

$$x_{n+2^m}^{(i)} \equiv x_n^{(i)} + 2^m + 2^{m+1}d_n^{(i)} \mod 2^{m+2}.$$

Then, by Lemma 4.1.2 and 5.1.3,

$$\begin{aligned}
&x_{n+2^m+1}^{(i)} \\
&\equiv \sum_{j=0}^{N}\left\{a_{i,j} + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,j,k,l}(x_{n+2^m}^{(k)})^l\right\}(x_{n+2^m}^{(i)})^j \mod 2^{m+2} \\
&\equiv \sum_{j=0}^{N}\left\{a_{i,j} + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,j,k,l}(x_n^{(k)} + 2^m + 2^{m+1}d_n^{(k)})^l\right\}(x_{n+2^m}^{(i)})^j \mod 2^{m+2} \\
&\equiv \sum_{j=0}^{N}\left\{a_{i,j} + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,j,k,l}(x_n^{(k)})^l\right\}(x_{n+2^m}^{(i)})^j + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}\,l(x_n^{(k)})^{l-1}2^m \mod 2^{m+2} \\
&\equiv \sum_{j=0}^{N}\left\{a_{i,j} + \sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,j,k,l}(x_n^{(k)})^l\right\}(x_n^{(i)} + 2^m + 2^{m+1}d_n^{(i)})^j \\
&\qquad\qquad + 2^m\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}\,l(x_n^{(k)})^{l-1} \mod 2^{m+2} \\
&\equiv f_i(x_n^{(i)} + 2^m + 2^{m+1}d_n^{(m+1,i)}) + \sum_{j=0}^{N}\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,j,k,l}(x_n^{(k)})^l(x_n^{(i)})^j \\
&\qquad\qquad + 2^m\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}(x_n^{(k)})^l + 2^m\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}\,l(x_n^{(k)})^{l-1} \mod 2^{m+2} \\
&\equiv f_i(x_n^{(i)}) + 2^m + 2^{m+1}(d_n^{(i)} + D_{i,n}) + \sum_{j=0}^{N}\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,j,k,l}(x_n^{(k)})^l(x_n^{(i)})^j \\
&\qquad\qquad + 2^m\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}(x_n^{(k)})^l + 2^m\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}\,l(x_n^{(k)})^{l-1} \mod 2^{m+2} \\
&\equiv x_{n+1}^{(i)} + 2^m + 2^{m+1}(d_n^{(i)} + D_{i,n}) \\
&\qquad\qquad + 2^m\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}(x_n^{(k)})^l + 2^m\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}\,l(x_n^{(k)})^{l-1} \mod 2^{m+2},
\end{aligned}$$

where $D_{i,n} \in \{0,1\}$ and it depends on which $x_n^{(i)}$ is odd or even. In the same way,

$$\begin{aligned}
&x_{n+2^{m+1}}^{(i)} \\
&\equiv x_{n+2^m}^{(i)} + 2^m + 2^{m+1}\left(d_n^{(i)} + \sum_{s=0}^{2^m-1} D_{i,n+s}\right) \\
&\qquad + 2^m\sum_{s=0}^{2^m-1}\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}(x_{n+s}^{(k)})^l + 2^m\sum_{s=0}^{2^m-1}\sum_{k=1}^{K}\sum_{l=1}^{L} C_{i,0,k,l}\,l(x_{n+s}^{(k)})^{l-1} \mod 2^{m+2} \\
&\equiv x_{n+2^m}^{(i)} + 2^m + 2^{m+1}d_n^{(i)} \mod 2^{m+2} \\
&\equiv x_n + 2^{m+1} \mod 2^{m+2}.
\end{aligned}$$

From the above, the theorem is true. □

5.2.3 Experiment

As an example of the method proposed in the former section, we consider the following system:

$$
\begin{aligned}
x_{n+1}^{(1)} &= 2(x_n^{(1)})^2 + (4x_n^{(3)} + 3)x_n^{(1)} + (4x_n^{(2)} + 1) \mod 2^{64},\\
x_{n+1}^{(2)} &= 2(x_n^{(2)})^2 + (4x_n^{(4)} + 3)x_n^{(2)} + (4x_n^{(3)} + 1) \mod 2^{64},\\
x_{n+1}^{(3)} &= 2(x_n^{(3)})^2 + (4x_n^{(5)} + 3)x_n^{(3)} + (4x_n^{(4)} + 1) \mod 2^{64},\\
x_{n+1}^{(4)} &= 2(x_n^{(4)})^2 + (4x_n^{(6)} + 3)x_n^{(4)} + (4x_n^{(5)} + 1) \mod 2^{64},\\
x_{n+1}^{(5)} &= 2(x_n^{(5)})^2 + (4x_n^{(7)} + 3)x_n^{(5)} + (4x_n^{(6)} + 1) \mod 2^{64},\\
x_{n+1}^{(6)} &= 2(x_n^{(6)})^2 + (4x_n^{(8)} + 3)x_n^{(6)} + (4x_n^{(7)} + 1) \mod 2^{64},\\
x_{n+1}^{(7)} &= 2(x_n^{(7)})^2 + (4x_n^{(1)} + 3)x_n^{(7)} + (4x_n^{(8)} + 1) \mod 2^{64},\\
x_{n+1}^{(8)} &= 2(x_n^{(8)})^2 + (4x_n^{(2)} + 3)x_n^{(8)} + (4x_n^{(1)} + 1) \mod 2^{64}.
\end{aligned}
$$

Assume that this system outputs a following 32-bit number for each n,

$$
\begin{aligned}
&((x_n^{(1)} \oplus x_n^{(2)} \oplus x_n^{(3)} \oplus x_n^{(4)}) >> 32)\\
&\quad \oplus ((x_n^{(5)} \oplus x_n^{(6)} \oplus x_n^{(7)} \oplus x_n^{(8)}) >> 48) \oplus (((x_n^{(5)} \oplus x_n^{(6)} \oplus x_n^{(7)} \oplus x_n^{(8)}) >> 16) \& 0\text{xffff0000}.
\end{aligned}
$$

Then, we can regard the system as PRNG, and it can be also used as stream cipher.

Randomness test

We performed randomness test described by NIST SP800-22 for sequences generated by the system. The detail is as follows:

- We performed the test for 100 sets.
- Each set consists of 1000 sequences.
- Each sequence consists of 1000000 bit.
- Initial state of the system was randomly decided for each set.

As the result, we got Table 5.1.

Table 5.1: Results of randomness test

Numbers of test items which a set passed	Numbers of set
188	71
187	22
186	6
185	1

In addition, we performed statistical test proposed in Appendix A for 100 sets. As the results, all sets passed the test.

These results show that the test did not find a problem about randomness. Main purpose to propose the new methods is to make PRNGs whose sequences have a good randomness. Thus, we can say that the experiment was successful.

Encryption Speed

We measured the encryption speed of the system when we use it as stream cipher without key-expansion. The environment in which we measured is shown in Table 5.2. In this experiment, we did not use parallel parallel computing.

As a result, the system recorded 4936.824780Mbps. If calculated in cycle/Byte, it is 2.11cycle/Byte. Fastest stream cipher is K-Cipher2, as far as I know, and the maximum speed of K-Cipher2 have been reported as 2.88cycle/Byte [14]. Investigation of the security of our method is not enough, and so we cannot state that our method is the fastest stream cipher. The result, however, shows possibility of one-stroke polynomials over a ring of modulo 2^w in the field of stream cipher.

Table 5.2: Environment on which we measure speed

CPU	1.3 GHz Intel Core i5
Memory	4 GB 1600 MHz DDR3
OS	OS X 10.9.5（13F34)
Compiler	gcc 4.2.1
Optimization option	-Ofast

5.3 Method of combining one-stroke polynomials over a ring of modulo 2^w with extending the period

Although a period of an orbit made by one-stroke polynomials over a ring of modulo 2^w is long, it is limited to 2^w. For example, if we use 64-bit processor, the period equals to 2^{64} or is shorter. If we need a sequence with longer period, multiple-precision calculation is a solution. Multiple-precision calculation, however, causes speed reduction. One reason is that multiplication of m-bit numbers needs $O(m^2)$ times. Another reason is that multiple-precision calculation needs more memory assesses.

Then, in this section, we propose another method of combining some one-stroke polynomials over a ring of modulo 2^w. The purpose is to get a longer period without speed reduction.

Firstly, we prove the following Theorem 5.3.2.

Lemma 5.3.1. Assume that a sequence $\{x_n \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ has a (a, w)-reproduct property. Then,for arbitrary n,

$$\sum_{i=0}^{2^{w+a-1}-1} [x_{n+i}]_w \oplus [x_{n+i-1}]_w \equiv 1 \mod 2,$$

where $[x]_m$ means the m-th least significant bit of a variable x.

Proof. For arbitrary n and m $(m > n)$,

$$[x_{m-1}]_w \equiv [x_{n-1}]_w + \sum_{i=0}^{m-n-1} [x_{n+i}]_w \oplus [x_{n+i-1}]_w \mod 2.$$

Since $\{x_n\}_{n\in\mathbb{Z}}$ has a (a,w)-reproduct property,

$$[x_{n+2^{w+a-1}-1}]_w = [x_{n-1}]_w \oplus 1.$$

Assume that $m = n + 2^{w+a-1}$. Then,

$$\sum_{i=0}^{2^{w+a-1}-1} [x_{n+i}]_w \oplus [x_{n+i-1}]_w \equiv 1 \mod 2.$$

□

Theorem 5.3.2. Assume that a sequence $\{x_n \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ has a (a,w)-reproduct property, $f(X)$ is a one-stroke polynomial over a ring of modulo 2^w,a sequence $\{y_n \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ satisfy

$$y_{n+1} = f(y_n) + [x_n]_w \oplus [x_{n-1}]_w \mod 2^{w'}.$$

Then, $\{y_n \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ has $(w+a-1, w')$-reproduct property.

Proof. By Corollary 4.2.4, for arbitrary n,

$$y_{n+1} = y_n + 1 + [x_n]_w \oplus [x_{n-1}]_w \mod 2.$$

Then, by Lemma 5.3.1,

$$\begin{aligned} y_{n+2^{w+a-1}} \equiv & y_n + 2^{w+a-1} + \sum_{i=0}^{2^{w+a-1}} [x_{n+i}]_w \oplus [x_{n+i-1}]_w \mod 2 \\ \equiv & y_n + 1 \mod 2. \end{aligned}$$

Assume that $\{y_n \mod 2\}_{n\in\mathbb{Z}}$ has a $(w+a-1,1)$-reproduct property. Then, for all n, there exist $d_n \in \{0,1\}$ such that

$$y_{n+2^{w+a+m-1}} \equiv y_n + 1 + 2d_n \mod 4.$$

Then, by Lemma 4.1.2 and 5.1.1,

$$\begin{aligned} y_{n+2^{w+a-1}+1} \equiv & f(y_{n+2^{w+a-1}}) + [x_{n+2^{w+a-1}}]_w \oplus [x_{n+2^{w+a-1}-1}]_w \mod 4 \\ \equiv & f(y_n + 1 + 2d_n) + [x_{n+2^{w+a-1}}]_w \oplus [x_{n+2^{w+a-1}-1}]_w \mod 4 \\ \equiv & f(y_n) + 1 + 2d_n + [x_n]_w \oplus [x_{n-1}]_w \mod 4 \\ \equiv & y_{n+1} + 1 + 2d_n \mod 4. \end{aligned}$$

Then,

$$\begin{aligned} y_{n+2^{w+a}} = & y_{n+2^{w+a-1}+2^{w+a-1}} \\ \equiv & y_{n+2^{w+a-1}} + 1 + 2d_n \mod 4 \\ \equiv & y_n + 2 \mod 4. \end{aligned}$$

Assume that, for all n, there exist $d_n \in \{0,1\}$ such that

$$y_{n+2^{w+a}} \equiv y_n + 2 + 4d_n \mod 8.$$

Then, by Lemma 4.1.2 and 5.1.2,

$$\begin{aligned} y_{n+2^{w+a}+1} \equiv& f(y_{n+2^{w+a}}) + [x_{n+2^{w+a}}]_w \oplus [x_{n+2^{w+a}-1}]_w \mod 8 \\ \equiv& f(y_n + 2 + 4d_n) + [x_{n+2^{w+a}}]_w \oplus [x_{n+2^{w+a}-1}]_w \mod 8 \\ \equiv& f(y_n) + 2 + 4(d_n^{(m+1)} + C) + [x_n]_w \oplus [x_{n-1}]_w \mod 8 \\ \equiv& y_{n+1} + 2 + 4(d_n + C) \mod 8, \end{aligned}$$

where $C \in \{0, 1\}$. In the same way,

$$\begin{aligned} y_{n+2^{w+a+1}} \equiv& y_{n+2^{w+a}} + 2 + 4(d_n + 2^{w+a}C) \mod 8 \\ \equiv& y_n + 4 \mod 8. \end{aligned}$$

Assume that there exists a natural number $2 \le m \le w-1$ such that $\{y_n \mod 2^{m+1}\}_{n\in\mathbb{Z}}$ has a $(w + a - 1, m + 1)$-reproduct property. Then, for all n, there exist $d_n^{\in}\{0, 1\}$ such that

$$y_{n+2^{w+a+m-1}} \equiv y_n + 2^m + 2^{m+1}d_n \mod 2^{m+2}.$$

Then, by Lemma 4.1.2 and 5.1.3,

$$\begin{aligned} y_{n+2^{w+a+m-1}+1} \equiv& f(y_{n+2^{w+a+m-1}}) + [x_{n+2^{w+a+m-1}}]_w \oplus [x_{n+2^{w+a+m-1}-1}]_w \mod 2^{m+2} \\ \equiv& f(y_n + 2^m + 2^{m+1}d_n) + [x_{n+2^{w+a+m-1}}]_w \oplus [x_{n+2^{w+a+m-1}-1}]_w \mod 2^{m+2} \\ \equiv& f(y_n) + 2^m + 2^{m+1}(d_n + C_n) + [x_n]_w \oplus [x_{n-1}]_w \mod 2^{m+2} \\ \equiv& y_{n+1} + 2^m + 2^{m+1}(d_n + C_n) \mod 2^{m+2}, \end{aligned}$$

where $C_n \in \{0, 1\}$ and it depends on which y_n is odd or even.. In the same way,

$$\begin{aligned} y_{n+2^{w+a+m}} =\equiv& y_{n+2^{w+a+m-1}} + 2^m + 2^{m+1}\left(d_n + \sum_{s=0}^{2^{w+a+m-1}} C_{n+s}\right) \mod 2^{m+2} \\ \equiv& y_n + 2^{m+1} \mod 2^{m+2}. \end{aligned}$$

From the above, the theorem is true. □

Based on Theorem 5.3.2, we consider the following system:

$$\begin{aligned} x_{n+1}^{(1)} =& f_1(x_n^{(1)}) \mod 2^{w_1}, \\ x_{n+1}^{(2)} =& f_2(x_n^{(2)}) + [x_n^{(1)}]_{w_1} \oplus [x_{n-1}^{(1)}]_{w_1} \mod 2^{w_2}, \\ x_{n+1}^{(3)} =& f_2(x_n^{(3)}) + [x_n^{(2)}]_{w_2} \oplus [x_{n-1}^{(2)}]_{w_2} \mod 2^{w_3}, \\ &\vdots \\ x_{n+1}^{(K)} =& f_K(x_n^{(K)}) + [x_n^{(K-1)}]_{w_{K-1}} \oplus [x_{n-1}^{(K-1)}]_{w_{K-1}} \mod 2^{w_K}, \end{aligned}$$

where $f_i(X)$ $(i = 1, 2, \cdots, K)$ are one-stroke polynomials over a ring of modulo 2^w. Then, the sequence $\{x_n^{(K)} \in \mathbb{Z}/2^w\mathbb{Z}\}_{n\in\mathbb{Z}}$ has a $(w_1 + w_2 + \cdots + w_{K-1} - (K-1), w_K)$-reproduct property, and the period is $2^{w_1+w_2+\cdots+w_K-(K-1)}$. If it takes $O(g_i(w_i))$ times to calculate the value of $f(\bar{X}) \mod 2^{w_i}$ for a given $\bar{X}$, it takes only $O(g_1(w_1) + g_2(w_2) + \cdots + g_K(w_K))$ times to push the system forward 1 step. If we use parallel computing, it takes only $O(\max\{g_1(w_1), g_2(w_2), \cdots, g_K(w_K)\})$.

5.4 Summary

In this chapter, we proposed two combining methods. One is a method of making a PRNG which can generate a sequence with good randomness. The other is a method of making a PRNG which can generate a sequence with a longer period. Both methods take a little time only. As applications, we can use both methods at the same time. Then, I believe that one-stroke polynomials will be practically used in cryptography in the near future.

Chapter 6

Conclusion

We discussed permutation polynomials over a ring of modulo 2^w and their applications to cryptography. In this chapter, we summarize the possibility of the polynomials and discuss the future works.

6.1 Key-exchange and public-key cryptography

In Chapter 2, we analyzed key-exchange protocol using odd degree Chebyshev polynomials over a ring of modulo 2^w. We proved that the key-exchange protocol is *not* secure, which means that there is an algorithm which can break the protocol with polynomial order times. We generalized the result and proved that an iteration number decision problem can be solved with polynomial order times for more general permutation polynomials over the ring which satisfy some conditions. In Chapter 4, we introduced one-stroke polynomials over a ring of modulo 2^w and showed an algorithm which can solve an iteration number decision problem with $O(w^5)$ times for *arbitrary* one-stroke polynomial over a ring of modulo 2^w. These results suggest that it is unfortunately difficult to construct a key-exchange protocol using permutation polynomials over a ring of modulo 2^w. This means that it is also difficult to construct a public-key cryptography using the polynomials. I think that we need a breakthrough if we construct a new key-exchange protocol and a public-key cryptography using permutation polynomials over a ring of modulo 2^w, and it is a future work.

6.2 Stream cipher and pseudo random number generator

On the other hand, we got an opposite result in the fields of stream cipher and pseudo random number generators. In Chapter 3, we improved Vector Stream Cipher consisted of permutation polynomials over a ring of modulo 2^{32}, and proposed Vector Stream Cipher 2.0 and 2.1. Both of them are fast, light and more secure than the original VSC.

In Chapter 4, we introduce a new class of permutation polynomials called "one-stroke polynomials over a ring of modulo 2^w". They have the orbit of the maximum periods. Then, they are expected to be useful for stream cipher and pseudo random number generators because long periodicity is important for such applications. Indeed, we show the possibility of one-stroke polynomials in Chapter 5. In Chapter 5, we discuss how to use one-stroke polynomials for such applications. We proposed a method of combining many one-stroke polynomials and making a system more complex. We showed that system using such method. We also proposed a method of combining one-stroke polynomials and making a system which has a longer period. Using the methods, non-linearity and ensuring of the periodicity can stand by stand. Compared with linear feedback shift register, that is an advantage of one-stroke polynomials.

These results suggest usefulness of permutation polynomials over a ring of modulo 2^w for stream cipher and pseudo random number generators.

6.3 Future work and possibility of permutation polynomials over a ring of modulo 2^w

There are some what we have to do before using permutation polynomials over a ring of modulo 2^w for cryptography, in particular, stream cipher. Futher estimation of security is needed. In practical, we are needed to repeat designing cipher with permutation polynomials, estimating the security and redesigning cipher.

There are three points of view:

- As standard indicators, "linear characteristic probability" and "linear complexity" are often used for estimating security of stream cipher. We have to design cipher which has sufficiently large these values. In addition, in order to calculate the values of the indicators for stream cipher using permutation polynomials, we have to develop a new methodology.
- We should show that stream cipher using permutation polynomials have registance to known attacks, such as Time-Memory Trade off Attack, Coorelation Attack and Guess-and-Determine Attacks.
- We have to investigate weak points indigenous to permutation polynomials and make the countermeasure.

The author conjectures that the first point will be solved soon because permutation polynomials have non-linearity. Although the second and third points are more difficult than the first point, the author conjectures that they can also be solved. We showed that we can construct a very first system using permutation polynomials over a ring of modulo 2^w in Chapter 5. Then, if we need more security, we can increase number of used permutation polynomials because there is a margin.

From the above, practical cipher using permutation polynomials over a ring of modulo 2^w will be developed. Since it will be a very first and light cipher, it may be used for IoT if it is verified that the security is superior.

Acknowledgement

The author expresses his gratitude to Professor Ken Umeno for his guidance and for helpful comments on this study. The author is also grateful to Professor Toshinobu Kaneko, Associate Professor Akihiro Yamaguchi and Dr. Hidetoshi Okutomi for their helpful comments.

The author would like to thank Dr. Shin-itiro Goto and Mr. Hirofumi Tsuda for discussion and advice. The author would like to thank Assosiate professor Daisaburo Yoshioka, Associate professor Fumiyoshi Kuwashima, Associate professor Akito Igarashi, Assistant professor Akihiro Sato, Dr. Shun Ogawa and Mr. Ken-ichi Okubo for discussion.

Appendix A

Randomness test based on variance of power spectrum –an alternative approach different from DFT test–

NIST SP 800-22 [35] is one of the most famous randomness test suite in the world. Randomness test is useful for many fields and that is, in particular, indispensable for estimation of security of cryptography. The first version of NIST SP 800-22 was published in 2000. Now, revision 1a which is the resent version of NIST SP 800-22 have been published.

A.1 DFT test and history of its improvements

Discrete Fourier Transform Test (DFTT) is a test included in NIST SP800-22. DFTT is a hypothesis test and the null hypothesis is "Given sequences are independent and truly random sequences". The algorithm of DFTT is as follows:

1 Receive M sequences $X_1, X_2, \cdots, X_M$ as the input. Here, each X_i is a n-bit sequence, and each bit is 0 or 1.

2 Convert each bit x to $2x - 1$. (Then, each bit become 1 or -1.)

3 For each X_i, calculate p-value p_i.

4 Perform a hypothesis test for $\{p_1, p_2, \cdots, p_M\}$ under the null hypothesis "$p_1, p_2, \cdots, p_M$ are independent and the distribution is uniform on the interval $[0, 1]$ " (Second-level-test).

The null hypothesis of DFTT is reject if and only if the null hypothesis of step 4 is reject. It means that if the null hypothesis of DFTT is true, $p_1, p_2, \cdots, p_M$ must be independent and the distribution is uniform on the interval $[0, 1]$. The independency of $p_1, p_2, \cdots, p_M$ is clearly true if $X_1, X_2, \cdots, X_M$ are independent. Then, the most important point of DFTT is calculation of p-value at step 3. If X_i is a truly random sequence, the corresponding p-value p_i must be follow uniform distribution on $[0, 1]$.

The algorithm of calculating p-value in the first version is as follows:

1 For given n-bit sequence X, perform discrete Fourier Transform and get the Fourier spectrum series $|S_0(X)|, |S_1(X)|, \cdots, |S_{\frac{n}{2}-1}(X)|$.

2 Count N_1 which is the number of $|S_i(X)| < \sqrt{3n}$.

3 Calculate d as follows:

$$d = \frac{N_1 - 0.95\frac{n}{2}}{\sqrt{(0.95)(0.05)\frac{n}{2}}}.$$

4 Calculate p-value p as follows:

$$p = \operatorname{erfc}\left(\frac{|d|}{\sqrt{2}}\right).$$

This algorithm is based on the following conjecture: for sufficient large n, if X is truly random sequence, $\frac{2}{n}|S_1(X)|^2, \cdots, \frac{2}{n}|S_{\frac{n}{2}-1}(X)|^2$ independently follow χ_2^2-distribution and N_1 follows $\mathcal{B}(\frac{n}{2}, 0.95)$ where $\mathcal{B}$ is the binomial distribution. If the conjecture is true, d and p approximately follow the normal distribution and uniform distribution on $[0, 1]$ respectively. The conjecture is, however, probably not true because even sequences generated by Mersenne twister which can be regarded as theoretically good generator do not pass the DFTT. There are following two problems:

- For an arbitrary fixed $j < n$ except $j = 0$, it is true that $\frac{2}{n}|S_j(X)|^2$ follows χ^2-distribution if X is truly random sequence and n is sufficient large. However, $|S_0(X)|, |S_1(X)|, \cdots, |S_{\frac{n}{2}-1}(X)|$ are not probably independent because they share the same argument X.
- The threshold $\sqrt{3n}$ is an approximate value, but the precision is not good.

In 2003, Kim et al. pointed out the above problems [43]. Hamano also pointed out them [44]. Kim et al. proposed the following improvement:

- Change the calculation of d as follows:

$$d = \frac{N_1 - 0.95\frac{n}{2}}{\sqrt{(0.95)(0.05)\frac{n}{4}}}.$$

- Set the threshold as $\sqrt{-n\log(0.05)}$.

The first improving point is based on fitting using pseudo random sequences. The second point has a theoretical basis if the first point is proper. DFTT in the resent version of NIST SP 800-22 adopted the improvement.

Pareschi et al. did more precise fitting using pseudo random sequences generated by BBS [45] and propose further improvement [46]. That is to change the calculation of d as follows:

$$d = \frac{N_1 - 0.95\frac{n}{2}}{\sqrt{(0.95)(0.05)\frac{n}{3.8}}}.$$

Although DFTT has been improved, we think that the following problems are left:

- The purpose of randomness test is to estimate randomness of given sequences. Then, fitting using pseudo random sequences is inconsistent even if it is theoretically ensured that PRNG generating the sequences is superior.
- Even if the PRNG is perfectly superior, fitting is merely numerical.
- There is no basis that the calculation of d is written as the form

$$d = \frac{N_1 - 0.95\frac{n}{2}}{\sqrt{(0.95)(0.05)\frac{n}{a}}},$$

  where a is a parameter.

Okada et al. proposed an approach to solve the problems [47]. We, however, think that the approach made another problem. That is to break independency of p-values.

A.2 Variance of power spectrum

The reason why the problems are left is that it is too difficult to derive the reference distribution of N_1. The purpose of DFTT is to investigate whether the fluctuation of Fourier spectrum is suitable or not. Then, we should introduce another indicator which reflects the fluctuation and whose reference distribution can analytically be derived.

Firstly, we consider the variance of Fourier spectrum as the indicator. It is, however, difficult to derive the reference distribution of the variance because the definition of Fourier spectrum is

$$|S_j(X)| := \sqrt{\left(\sum_{k=0}^{n-1} x_k \cos\left(\frac{2\pi kj}{n}\right)\right)^2 + \left(\sum_{k=0}^{n-1} x_k \sin\left(\frac{2\pi kj}{n}\right)\right)^2},$$

where x_k is k-th bit of X and the square root makes it difficult.

Then, we consider the variance of power spectrum as the indicator. The definition is as follows:

$$V_n(X) := \frac{1}{n}\sum_{j=0}^{n-1}\left\{\frac{1}{n}|S_j(X)|^2 - \frac{1}{n}\sum_{r=0}^{n-1}\left(\frac{1}{n}|S_r(X)|^2\right)\right\}^2.$$

Of course, power spectrum is not Fourier spectrum, but the variance of power spectrum probably reflects fluctuation of Fourier spectrum.

A.2.1 Distribution of $V_n(X)$

In order to derive the distribution of $V_n(X)$, let us calculate the average and variance of $V_n(X)$. Firstly, we deform $V_n(X)$.

$$\begin{aligned}
V_n(X) :=& \frac{1}{n}\sum_{j=0}^{n-1}\left\{\frac{1}{n}|S_j(X)|^2 - \frac{1}{n}\sum_{r=0}^{n-1}\left(\frac{1}{n}|S_r(X)|^2\right)\right\}^2 \\
=& \frac{1}{n^3}\sum_{j=0}^{n-1}|S_j(X)|^4 - 1 \quad (\because \text{Parseval's theorem}) \\
=& \frac{1}{n^3}\sum_{j=0}^{n-1}\left\{\left(\sum_{k=0}^{n-1} x_k\cos\left(\frac{2\pi kj}{n}\right)\right)^2 + \left(\sum_{k=0}^{n-1} x_k\sin\left(\frac{2\pi kj}{n}\right)\right)^2\right\}^2 - 1 \\
=& \frac{1}{n^3}\sum_{j=0}^{n-1}\sum_{a=0}^{n-1}\sum_{b=0}^{n-1}\sum_{c=0}^{n-1}\sum_{d=0}^{n-1} x_a x_b x_c x_d \Big\{ \\
& \cos\left(\frac{2\pi aj}{n}\right)\cos\left(\frac{2\pi bj}{n}\right)\cos\left(\frac{2\pi cj}{n}\right)\cos\left(\frac{2\pi dj}{n}\right) \\
& + \sin\left(\frac{2\pi aj}{n}\right)\sin\left(\frac{2\pi bj}{n}\right)\sin\left(\frac{2\pi cj}{n}\right)\sin\left(\frac{2\pi dj}{n}\right) \\
& + \cos\left(\frac{2\pi aj}{n}\right)\cos\left(\frac{2\pi bj}{n}\right)\sin\left(\frac{2\pi cj}{n}\right)\sin\left(\frac{2\pi dj}{n}\right)\Big\} - 1 \\
=& \frac{1}{4n^3}\sum_{j=0}^{n-1}\sum_{a=0}^{n-1}\sum_{b=0}^{n-1}\sum_{c=0}^{n-1}\sum_{d=0}^{n-1} x_a x_b x_c x_d \Big\{ \\
& \cos\left(\frac{2\pi(a+b+c-d)j}{n}\right) + \cos\left(\frac{2\pi(a+b-c+d)j}{n}\right) \\
& - \cos\left(\frac{2\pi(a-b+c+d)j}{n}\right) - \cos\left(\frac{2\pi(a-b-c-d)j}{n}\right) \\
& + 2\cos\left(\frac{2\pi(a-b+c-d)j}{n}\right) + 2\cos\left(\frac{2\pi(a-b-c+d)j}{n}\right)\Big\} - 1.
\end{aligned}$$

We introduce a new notation:

$$\delta_x := \begin{cases} 1 & (x = 0, \pm n) \\ 0 & (\text{otherwise}) \end{cases} .$$

Then,

$$\begin{aligned}
V_n =& \frac{1}{4n^2}\sum_{a=0}^{n-1}\sum_{b=0}^{n-1}\sum_{c=0}^{n-1}\sum_{d=0}^{n-1} x_a x_b x_c x_d \Big\{ \\
& \delta_{a+b+c-d} + \delta_{a+b-c+d} - \delta_{a-b+c+d} \\
& - \delta_{a-b-c-d} + 2\delta_{a-b+c-d} + 2\delta_{a-b-c+d}\Big\} - 1 \\
=& \frac{1}{n^2}\sum_{a=0}^{n-1}\sum_{b=0}^{n-1}\sum_{c=0}^{n-1}\sum_{d=0}^{n-1} x_a x_b x_c x_d \delta_{a-b+c-d} - 1. \qquad (A.1)
\end{aligned}$$

Let us calculate the average of $V_n(X)$. Since each x_i takes 1 or -1 with the same probability, terms in (A.1) except terms hold $x_a x_b x_c x_d = 1$ vanish when we take the

Table A.1: Combinatorial number satisfying $(a,b,c,d) \in B_1$, $(a',b',c',d') \in R(a,b,c,d)$

$a-b+c-d$	$a'-b'+c'-d'$	Combinatorial number
0	0	$\frac{8}{3}n^3 + O(n^2)$
0	n	$O(n^2)$
0	$-n$	$O(n^2)$
n	0	$O(n^2)$
n	n	$\frac{4}{3}n^3 + O(n^2)$
n	$-n$	$\frac{4}{3}n^3 + O(n^2)$
-n	0	$O(n^2)$
-n	n	$\frac{4}{3}n^3 + O(n^2)$
-n	$-n$	$\frac{4}{3}n^3 + O(n^2)$

average. Then,

$$\begin{aligned}\mathbb{E}[V_n] =& \frac{1}{n^2}\Big\{ \sum_{a=b=c=d} \delta_{a-b+c-d} + \sum_{a=b\neq c=d} \delta_{a-b+c-d} \sum_{a=c\neq b=d} \delta_{a-b+c-d} + \sum_{a=d\neq b=c} \delta_{a-b+c-d} \Big\} - 1 \\ =& \begin{cases} 1 & (n\text{: even}) \\ 1-\frac{1}{2n} & (n\text{: odd}) \end{cases} \longrightarrow 1 \ \ (n \to \infty).\end{aligned}$$

Next, let us calculate the variance of $V_n(X)$. We introduce new notations:

$$\begin{aligned} B_1 &:= \big\{(a,b,c,d) \in \{0,1,\cdots,n-1\}^4 | a,b,c,d \text{ take different values each other}\big\}, \\ B_2 &:= \big\{(e,f,g) \in \{0,1,\cdots,n-1\}^3 | e,f,g \text{ take different values each other}\big\}. \end{aligned}$$

Then,

$$\begin{aligned} &\mathbb{E}\left[(V_n(X) - \mathbb{E}[V_n(X)])^2\right] \\ =&\mathbb{E}\Big[\big\{ \sum_{(a,b,c,d)\in B_1} x_a x_b x_c x_d \delta_{a-b+c-d} + \sum_{(e,f,g)\in B_1} x_f x_g \delta_{2e-f-g} \big\}^2\Big] \\ =&\frac{1}{n^4}\Big\{\mathbb{E}[(\sum_{(a,b,c,d)\in B_1} x_a x_b x_c x_d \delta_{a-b+c-d})^2] \\ &+ \mathbb{E}[(\sum_{(a,b,c,d)\in B_1} x_a x_b x_c x_d \delta_{a-b+c-d})(\sum_{(e,f,g)\in B_2} x_f x_g \delta_{2e-f-g})] + \mathbb{E}[(\sum_{(e,f,g)\in B_2} x_f x_g \delta_{2e-f-g})^2]\Big\}. \end{aligned}$$

Let us consider the first term.

$$\begin{aligned} &\mathbb{E}[(\sum_{(a,b,c,d)\in B_1} x_a x_b x_c x_d \delta_{a-b+c-d})^2] \\ =&\mathbb{E}[\sum_{(a,b,c,d)\in B_1} \sum_{(a',b',c',d')\in B_1} x_a x_b x_c x_d x_{a'} x_{b'} x_{c'} x_{d'} \delta_{a-b+c-d} \delta_{a'-b'+c'-d'}] \\ =&\mathbb{E}[\sum_{(a,b,c,d)\in B_1} \sum_{(a',b',c',d')\in R(a,b,c,d)} \delta_{a-b+c-d} \delta_{a'-b'+c'-d'}], \end{aligned}$$

where $R(a,b,c,d) = \{\text{permutation of } a,b,c,d\}$. By Table. A.1,

$$\mathbb{E}[(\sum_{(a,b,c,d)\in B_1} x_a x_b x_c x_d \delta_{a-b+c-d})^2] = 8n^3 + O(n^2).$$

By the same way, the second term equals to 0 and the third term is $O(n^2)$. Then,

$$\mathbb{E}\left[(V_n(X) - \mathbb{E}[V_n(X)])^2\right] = \frac{8}{n} + O\left(\frac{1}{n^2}\right) \qquad \longrightarrow 0 \ (n \to \infty).$$

A.2.2 Scaling

The variance of $V_n(X)$ vanishes as $n \to \infty$, and so we cannot, unfortunately, use $V_n(X)$ as a new indicator. Then, we consider the following scaled variance of power spectrum as the indicator:

$$\begin{aligned}\tilde{V}_n(X) :=& \sqrt{\frac{n}{8}}\left\{V_n(X) - \mathbb{E}[V_n(X)]\right\} \\ =& \frac{1}{(2n)^{\frac{3}{2}}}\Big\{\sum_{(a,b,c,d)\in B_1} x_a x_b x_c x_d \delta_{a-b+c-d} + \sum_{(e,f,g)\in B_2} x_f x_g \delta_{2e-f-g}\Big\}.\end{aligned}$$

In order to derive the distribution of $\tilde{V}_n(X)$, let us calculate the moment.

$$\begin{aligned}\mathbb{E}[(\tilde{V}_n)^m] =& \frac{1}{(2n)^{\frac{3}{2}m}}\sum_{l=0}^{m} {}_mC_l \mathbb{E}\Big[\Big\{\sum_{(a,b,c,d)\in B_1} x_a x_b x_c x_d \delta_{a-b+c-d}\Big\}^{m-l} \\ & \times \Big\{\sum_{(e,f,g)\in B_2} x_f x_g \delta_{2e-f-g}\Big\}^{l}\Big] \\ =& \frac{1}{(2n)^{\frac{3}{2}m}}\mathbb{E}\Big[\Big\{\sum_{(a,b,c,d)\in B_1} x_a x_b x_c x_d \delta_{a-b+c-d}\Big\}^{m}\Big] + (\text{Lower order terms}),\end{aligned}$$

$$\begin{aligned}\mathbb{E}\Big[\Big\{\sum_{(a,b,c,d)\in B_1} x_a x_b x_c x_d \delta_{a-b+c-d}\Big\}^{m}\Big] =& \sum_{(a(1),b(1),c(1),d(1))\in B_1} \cdots \sum_{(a(m),b(m),c(m),d(m))\in B_1} \\ & \mathbb{E}\left[\{x_{a(1)}x_{b(1)}x_{c(1)}x_{d(1)}\}\cdots\{x_{a(m)}x_{b(m)}x_{c(m)}x_{d(m)}\}\right] \\ & \times \delta_{a(1)-b(1)+c(1)-d(1)}\cdots\delta_{a(m)-b(m)+c(m)-d(m)}.\end{aligned}$$

Then, we should count number of $\{a(1), b, (1), c(1), d(1)\}, \cdots,$ $\{a(m), b(m), c(m), d(m)\}$ satisfying

$$\begin{aligned}&\mathbb{E}\left[\{x_{a(1)}x_{b(1)}x_{c(1)}x_{d(1)}\}\cdots\{x_{a(m)}x_{b(m)}x_{c(m)}x_{d(m)}\}\right] = 1, \\ &\delta_{a(1)-b(1)+c(1)-d(1)}\cdots\delta_{a(m)-b(m)+c(m)-d(m)} = 1.\end{aligned}$$

In order to count the number, we consider the following model:

- Basically, each variable can freely take n values. (In other words, each variable has one degree of freedom.)
- Each variable has a hand.
- Variables which have a same argument are included in a same set. (See Fig. A.1.)
- Each hand must be connected with another hand by final time.
- Each hand must not be connected with two or more other hands.
- Hands of variables included in a same set must not be connected each other.

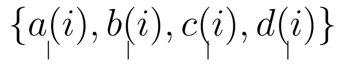


Figure A.1: Model of set of variables

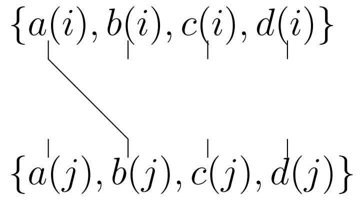


Figure A.2: An example connecting a set with another set by one-connection

- There are the following constraint conditions:
  - Each set has one constraint condition. If values of three variables in a set are fixed, the other variable in the set is automatically determined.
  - If a hand of variable A is connected with another hand of variable B, the value of A must equal to the value of B.

As example, variables in Fig. A.1 have 3(=4-1) degree of freedom. We consider the case that add a set $\{a(j), b(j), c(j), d(j)\}$ to Fig. A.1 and connect $a(i)$'s hand with $b(j)$'s hand. (See Fig. A.2.) In this case, the value of $b(j)$ must equal to $a(i)$. Then, degree of freedom increases and the amount of the change is +2(=4-1-1). Hands which are not connected with other hands (we call "open hands") also increase and the amount of the change is +2.

We consider a general case. Table A.2 shows the relation between number of anew connecting hands, amount of change of degree of freedom and open hands. In initial states, there is a set and variables included in the set have 3 degree of freedom. Fig. A.1 is an example of initial state. Since we must not leave hands which are not connected with other hands, connecting a set with another set like Fig. A.3 maximizes degree of freedom per one variable.

Table A.2: Relation between number of connected hands and change of degree of freedom and open hands

Number of connected hands	Degree of freedom	Open hands
1	+2	+2
2	+1	±0
3	±0	-2
4	±0	-4

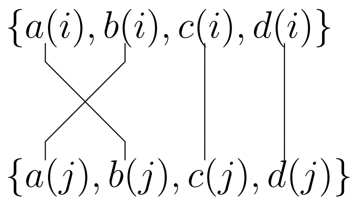


Figure A.3: An example of optimum connection

From the above, when m is even,

$$\begin{aligned}\mathbb{E}[(\tilde{V}_n)^m] =& \frac{1}{(2n)^{\frac{3}{2}m}}\mathbb{E}\Big[\Big\{\sum_{(a,b,c,d)\in B_1} x_a x_b x_c x_d \delta_{a-b+c-d}\Big\}^m\Big] + (\text{Lower order terms})\\ =& \frac{1}{(2n)^{\frac{3}{2}m}}\mathbb{E}\Big[\Big\{\sum_{(a,b,c,d)\in B_1} x_a x_b x_c x_d \delta_{a-b+c-d}\Big\}^2\Big]^{\frac{m}{2}}\\ &\quad \times (\text{The combinatorial total number to make } m/2 \text{ pairs of sets by } m \text{ sets})\\ &\qquad + (\text{Lower order terms})\\ =&(m-1)!! + (\text{Lower order terms}) \quad \longrightarrow (m-1)!! \ \ (n\to\infty).\end{aligned}$$

Here, $x!! := x \times (x-2) \times (x-4) \times \cdots \times 3 \times 1$. When m is odd, since we cannot make $m/2$ pairs of sets by m sets,

$$\mathbb{E}[(\tilde{V}_n)^m] = 0 + (\text{Lower order terms}) \qquad \longrightarrow 0 \ \ (n\to\infty).$$

Summarizing the above,

$$\lim_{n\to\infty} \mathbb{E}[(\tilde{V}_n)^m] = \begin{cases}(m-1)!! & (m\text{: even})\\ 0 & (m\text{: odd})\end{cases}.$$

Then, the moments of $\tilde{V}_n(X)$ converge to those of the standard normal distribution. It means that distribution of $\tilde{V}_n(X)$ "converges" to the standard normal distribution.

A.2.3 Numerical simulation

We made cumulative distributions of $\tilde{V}_n(X)$ by M n-bit sequences generated by Mersenne twister and compared them and the standard normal distribution. Fig. A.4 shows the results. Roughly speaking, the cumulative distribution of $\tilde{V}_n(X)$ correspond to the standard normal distribution when $n \geq 10^4$.

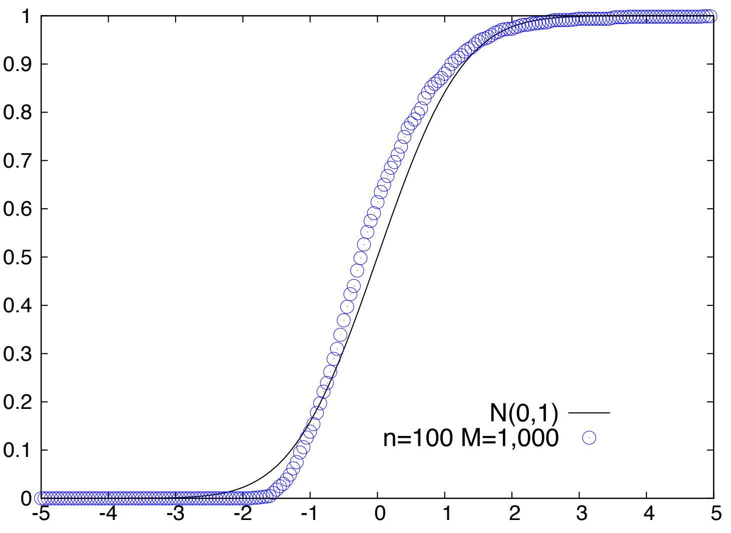
A.3 Proposal method

Based on the former section, we propose a new method using $\tilde{V}_n(X)$. The proposal method replaces p-value calculation of DFTT with the follows:

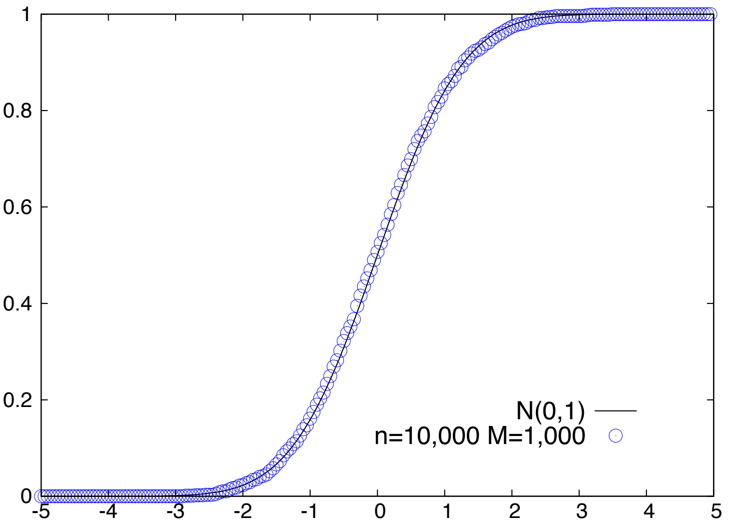
1 For given n-bit sequence X, perform discrete Fourier Transform and get the Fourier spectrum series $|S_0(X)|, |S_1(X)|, \cdots, |S_{\frac{n}{2}-1}(X)|$.

2 Calculate $\tilde{V}_n(X)$ as follows:

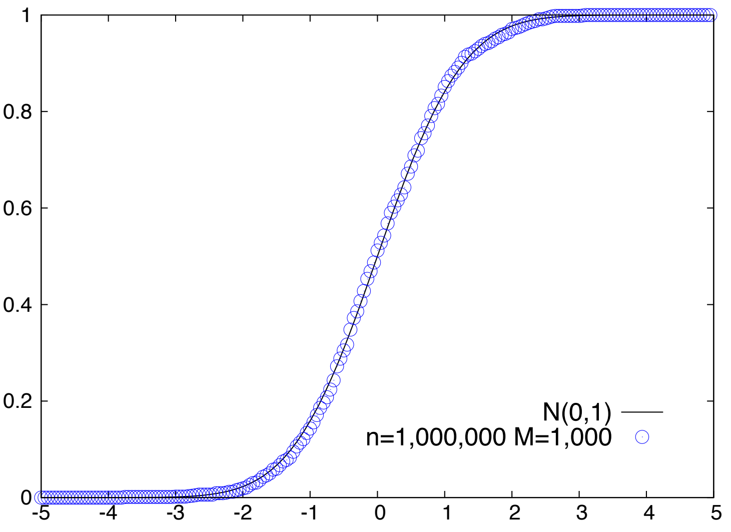
$$\tilde{V}_n(X) \leftarrow \frac{1}{\sqrt{2n^5}}\sum_{j=0}^{\frac{n}{2}-1}|S_j(X)|^4 - \sqrt{\frac{n}{2}}.$$



(a)



(b)



(c)

Figure A.4: Comparison cumulative distributions of $\tilde{V}_n(X)$ with the standard normal distribution

3 Calculate p-value p as follows:

$$p = \operatorname{erfc}\left(\frac{|\tilde{V}_n(X)|}{\sqrt{2}}\right).$$

At the step 2, we use symmetry of Fourier spectrum series and deformed $\tilde{V}_n(X)$ in order to reduce calculation.

The proposal method needs approximately n multiplications more than calculation of p-value of DFTT.

Discrete Fourier Transform, however, needs $O(n \log n)$ and so the proposal method needs $O(n \log n)$ times. That is same as DFTT.

A.4 Evaluation of detection power of test

In order to evaluate detection power of test, we performed some experiments for the proposal method, the present DFTT (proposed by Kim et al.) and Pareschi et al.'s method.

Before explaining the experiments, we explain the second-level-test. When we consider the case that M n-bit sequences are tested, we get M p-values $p_1, p_2, \cdots, p_M$. The second-level-test test is hypothesis test and the null hypothesis is that "$p_1, p_2, \cdots, p_M$ must be independent and the distribution is uniform on the interval $[0, 1]$". For the M p-values, we perform the following two tests:

- **Success rate**
  Count r which is the number of $p_i > 0.01$. Under the null hypothesis, r follows $\mathcal{B}(0.99, M)$. Then, if $|r - 0.99M| < 3 \times$ (the standard deviation), we make the null hypothesis pass. Else, we reject the null hypothesis.

- **Uniformity**
  Under the null hypothesis, we perform χ^2-test for $\{p_1, p_2, \cdots, p_M\}$ and get new one p-value $\bar{p}$. If $\bar{p} > 0.0001$, we make the null hypothesis pass. Else, we reject the null hypothesis.

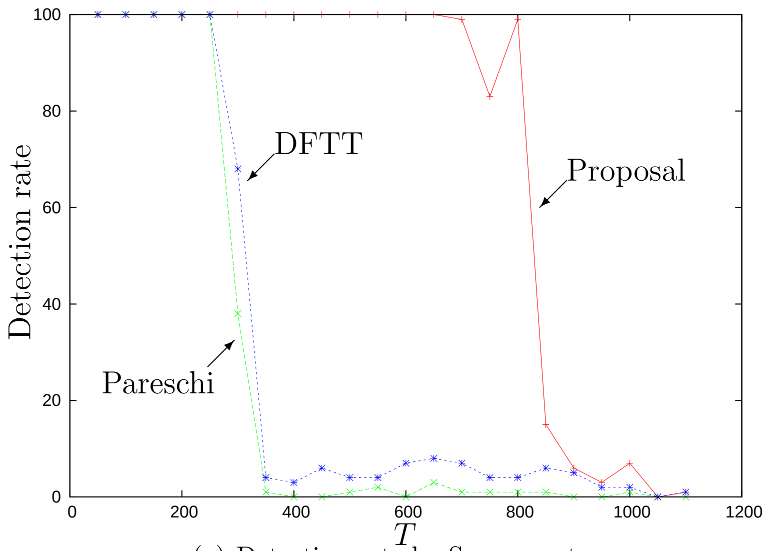
A.4.1 Experiment 1

We explain the first experiment. We generated sequences with Mersenne twister. After that, for each sequence, we did the following replacement:

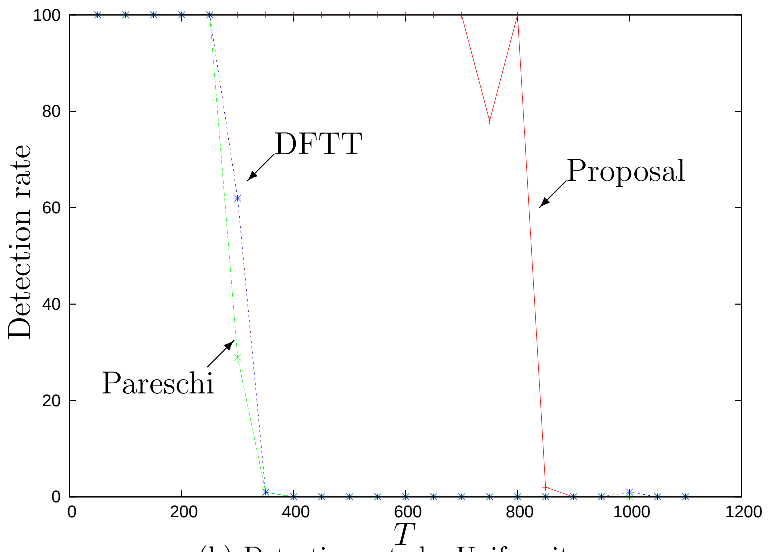
$$\begin{aligned} x_i =& 1 \quad (i \equiv T \mod 2T), \\ x_i =& -1 \quad (i \equiv 0 \mod 2T), \end{aligned}$$

where T is a parameter. Clearly, the sequences do not have good randomness. For each fixed T, we performed test 100 times. One test was performed for 1000 sequences and each sequence is 1000000-bit. For fair evaluation, sequences used by evaluation of the proposal method, the present DFTT and Pareschi et al.'s method were same.

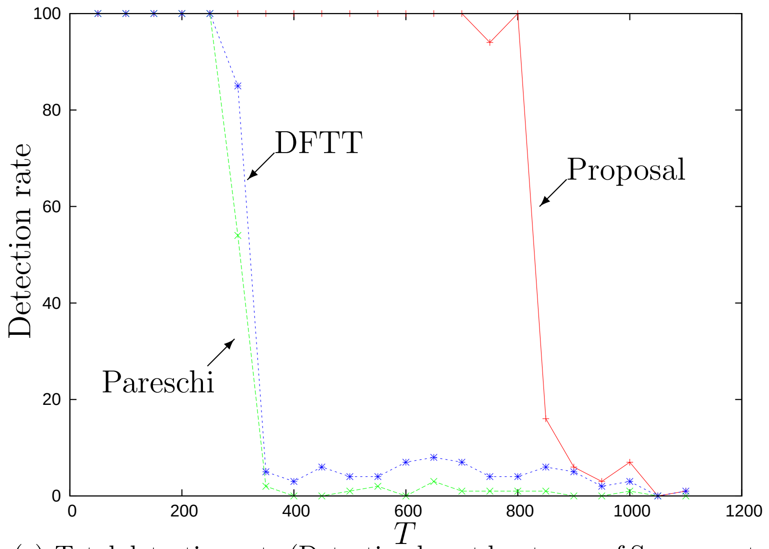
As the result, we got Fig. A.5. The proposal method could detect non-negligible deviations for large T as compared with the present DFTT and Pareschi et al.'s method. It shows that detection power of the proposal method is better than those of the others.



(a) Detection rate by Success rate



(b) Detection rate by Uniformity



(c) Total detection rate (Detection by, at least, one of Success rate or Uniformity)

Figure A.5: Result of Experiment 1

Table A.3: Number of detection by the proposal method

	Success rate	Uniformity	Total
VSC128	0	0	0
VSC 2.0	0	0	0
VSC 2.1	0	0	0
VSC-I	0	0	0
MT	0	0	0
AES-128 CTR	0	0	0
LCG	82	82	82
QCG-I	95	98	99
QCG-II	100	100	100
CCG	100	100	100
XORG	21	1	21

Table A.4: Number of detection by the present DFTT

	Success rate	Uniformity	Total
VSC128	5	0	5
VSC 2.0	1	0	1
VSC 2.1	3	0	3
VSC-I	2	0	2
MT	2	0	2
AES-128 CTR	3	0	3
LCG	82	82	82
QCG-I	2	0	2
QCG-II	100	100	100
CCG	100	100	100
XORG	8	0	8

A.4.2 Experiment 2

We performed the proposal method, the present DFTT and Pareschi et al.'s method for sequences generated by some existing pseudo random number generators (including stream ciphers and block cipher). We used VSC128 [19], VSC 2.0 [48], VSC 2.1 [49], VSC-I which is described as an example in Ref. [50], Mersenne twister, AES, LCG [35], QCG-I [35], QCG-II [35], CCG [35] and XORG [35] as the generators. For one generator, 100 test were performed. One test was performed for 1000 sequences and each sequence is 1000000-bit. For fair evaluation, sequences used by evaluation of the proposal method, the present DFTT, Pareschi et al.'s method were same.

As the results, we got Table A.3, A.4 and A.5. Although there are cases that the present DFTT detects non-negligible deviations compared with our method and Pareschi et al.'s method, we think that they are Type-1 error because Pareschi et al.'s method is numerically optimized method of the present DFTT. The results of QCG-I and XORG shows that detection power of the proposal method is better than those of the others.

Table A.5: Number of detection by Pareschi et al.'s method

	Success rate	Uniformity	Total
VSC128	0	0	0
VSC 2.0	0	0	0
VSC 2.1	0	0	0
VSC-I	0	0	0
MT	0	0	0
AES-128 CTR	0	0	0
LCG	82	82	82
QCG-I	0	0	0
QCG-II	99	100	100
CCG	100	100	100
XORG	2	0	2

A.5 Summary

We reviewed the problems of DFTT and proposed a new method. The proposal method uses the fact that the moments of scaled variance of power spectrum converge to those of the standard normal distribution. The proposal method needs $O(n \log n)$ times, that is same as the present DFTT. Some experiments showed that the proposal method has better detection power than the present DFTT.

As a future work, we have to investigate detail of convergence of the distribution. As another future work, we have to investigate independency between the proposal method and the other tests included in NIST SP800-22 because randomness of given sequences should be judged by all results of tests included in NIST SP800-22.

Bibliography

[1] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC press, 1996.

[2] National Bureau of Standards, “Data Encryption Standard”, FIPS-Pub.46. (1977).

[3] M. Matsui, “Linear cryptanalysis method for DES cipher”, *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer Berlin Heidelberg, 1993.

[4] National Institute of Standards and Technology, “Specification for the ADVANCED ENCRYPTION STANDARD (AES)”, FIPS-Pub.197. (2001).

[5] W. Diffie and M. Hellman, “New directions in cryptography”, *IEEE Trans. on Information Theory*, 22.6 (1976): 644-654.

[6] K. Ashton, “That ‘internet of things’ thing”, *RFiD Journal*, 22.7 (2009): 97-114.

[7] R. L. Rivest, A. Shamir and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, *Communications of the ACM*, 21.2 (1978): 120-126.

[8] National Institute of Standards and Technology, “Digital Signature Standard (DSS)”, FIPS-Pub.186-4. (2013).

[9] V. S. Miller,“Use of Elliptic Curves in Cryptography”,*Conference on the Theory and Application of Cryptographic Techniques*. Springer Berlin Heidelberg, 1985.

[10] M. Matsui, “New block encryption algorithm MISTY”, *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1997.

[11] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima and T. Tokita, “Camellia: A 128-bit block cipher suitable for multiple platforms—design andanalysis”, *International Workshop on Selected Areas in Cryptography*. Springer Berlin Heidelberg, 2000.

[12] D. Watanabe, S. Furuya, H. Yoshida, K. Takaragi and B. Preneel, “A new keystream generator MUGI”, *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 2002.

[13] Hitachi, Ltd., “Stream Cipher Enocoro Specification Ver. 2.0”, http://www.hitachi.co.jp/rd/yrl/crypto/enocoro (2010).

[14] KDDI Corporation, “Stream Cipher KCipher-2”, http://www.cryptrec.go.jp/english/cryptrec_13_spec_cypherlist_files/PDF/21_00espec.pdf (2010).

[15] P. Ekdahl and T. Johansson, "A new version of the stream cipher SNOW", *International Workshop on Selected Areas in Cryptography*. Springer Berlin Heidelberg, 2002.

[16] R. Lidl and G. L. Mullen, "When does a polynomial over a finite field permute the elements of the field?", The *American Mathematical Monthly*, 95.3 (1988): 243-246.

[17] R. Lidl and G. L. Mullen, "When does a polynomial over a finite field permute the elements of the field?, II", *The American Mathematical Monthly*, 100.1 (1993): 71-74.

[18] R. L. Rivest, M. J. B. Robshaw, R. Sidney and Y. L. Yin, "The RC6 Block Cipher", https://people.csail.mit.edu/rivest/pubs/RRSY98.pdf.

[19] K. Umeno, S. Kim and A. Hasegawa, "128bit VSC Specification," http://www.chaosware.com/vsc128.pdf (2004) (In Japanese).

[20] K. Umeno, "Key exchange by Chebyshev polynomials modulo 2^w", *Proc. of INACISC* (2005): 95-97.

[21] R. Coveyou, "Random Number Generation Is Too Important to Be Left to Chance", *Studies in Applied Mathematics*, III (1970), 70-111.

[22] L. Kocarev and Z. Tasev, "Public-key encryption based on Chebyshev maps", *Proc. of the 2003 International Symposium on Circuits and Systems, IEEE*, Vol. 3, 2003.

[23] P. Bergamo, P. D'Arco, A. S. Santis and L. Kocarev, "Security of public-key cryptosystems based on Chebyshev polynomials", *IEEE Trans. on Circuits and Systems I: Regular Papers*, 52.7 (2005): 1382-1393.

[24] M. Ishii, and A. Yoshimoto, "Applications for cryptography of the structure of the group of reduced residue classes of residue ring of $\mathbb{Z}/2^w\mathbb{Z}$", *Trans. of The Japan Society for Industrial and Applied Mathematics*, 19.1 (2009): 57-71 (In Japanese).

[25] M. Ishii, "Periodicity of Chebyshev polynomials over the residue ring of $\mathbb{Z}/2^r\mathbb{Z}$ and an electronic signature", *Trans. of The Japan Society for Industrial and Applied Mathematics*, 18.2 (2008): 257-265 (In Japanese).

[26] D. Yoshioka, "On some properties of Chebyshev polynomial sequences modulo 2^k", *Nonlinear Theory and Its Applications*, IEICE 6.3 (2015): 443-452.

[27] A. Iwasaki and K. Umeno, "Periodical property of Chebyshev polynomials on the residue class rings of modulo 2^w", *IEICE Technical Report, CAS2014-67, NLP2014-61* (2014): 81-86 (In Japanese).

[28] A. Iwasaki and K. Umeno, "Period of orbit and degree of Chebyshev polynomial on a ring of modulo 2^w ", *IEICE Technical Report, NLP2015-61* (2015): 129-134 (In Japanese).

[29] K. Kawano and D. Yoshioka, "A solution on the degree determination problem of Chebyshev polynomials over the residue ring $\mathbb{Z}/2^k\mathbb{Z}$ ", *IEICE Technical Report, NLP2015-77* (2015), pp. 53-56 (In Japanese).

[30] D. Yoshioka and K. Kawano, "Periodic Properties of Chebyshev Polynomial Sequences Over the Residue Ring $\mathbb{Z}/2^k\mathbb{Z}$", *IEEE Trans. on Circuits and Systems II: Express Briefs*, 63.8 (2016): 778-782..

[31] H. Tanaka, K. Nemoto, T. Miki and M. Sato, “Security Evaluation of 128bit VSC,” *Technical Report of IEICE, IT3003-104, ISEC2003-144, WBS2003-222* (2004): 179-184 (In Japanese).

[32] M. Nakamura, T. Nosaka and T. Kaneko, “A Study on the Linear Property of Basic Function in VSC128,” *Proc. ESS Conf. IEICE* (2004): 126.

[33] T. Nosaka, M. Nakamura and T. Kaneko, “A Study on Linear Cryptanalysis of VSC128,” *Proc. ESS Conf. IEICE* (2004): 127.

[34] Y. Tunoo, T. Saito, K. Myao, T. Suzaki and T. Kawabata, “Distinguishing Attack With Chosen Initial Vector Against VSC128,” *Proc. SASC Workshop* (2004): 212-219.

[35] NIST, “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,” Special Publication 800-22 Revision 1a, http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf (2010).

[36] K. Umeno, Complex systems and communications, in: *Information Systems as Complex Systems*, Waseda University Advanced Institute for Complex Systems ed., Kyouritsu-syuppansya (2007): 181-250 (In Japanese).

[37] D. E. Knuth, *The Art of Computer Programming. Vol. 2*, Addison-Wesley, Upper Saddle River, 1981.

[38] R. L. Rivest, “Permutation polynomials modulo 2^w”, *Finite Fields and Their Applications*, 7.2 (2001): 287-292.

[39] A. M. Frieze, R. Kannan and J. C. Lagarias, “Linear congruential generators do not produce random sequences”, *Proc. of 25th Annual Symposium on. IEEE Foundations of Computer Science*, 1984.

[40] D. E. Knuth, “Deciphering a linear congruential encryption”, *IEEE Trans. on Information Theory*, 31.1 (1985): 49-52.

[41] J. Stern, “Secret linear congruential generators are not cryptographically secure”, *Proc. of 28th Annual Symposium on. IEEE Foundations of Computer Science*, 1987.

[42] S. Kiyomoto, T. Tanaka, K. Sakurai, “Development of super high speed stream cipher, K-Cipher2”, http://www.fbi-award.jp/sentan/jusyou/2012/6.pdf (2012) (In Japanese).

[43] S. Kim, K. Umeno and A. Hasegawa, “On the NIST Statistical Test Suite for Randomness”, *Technical report of IEICE, ISEC2003-87* (2003).

[44] K. Hamano, “The Distribution of the Spectrum for the Discrete Fourier Transform Test Included in SP800-22”, *IEICE Trans. Fundamentals*, vol. E88-A, No. 1 (2005)

[45] L. Blum, M. Blum, and M. Shub “A Simple Unpredictable Pseudo-Random Number Generator”, *SIAM Journal on Computing*, 15.2 (1986): 364-383.

[46] F. Pareschi, R. Rovatti and G. Setti, “On Statistical Test Includeed in the NIST SP800-22 Test Suite and Based on the Binomial Distribution”, *IEEE trans. Information Forensics and Security*, 7.2 (2012): 491-505.

[47] H. Okada and K. Umeno, “Randomness Evaluation with the Discrete Fourier Transform Test Based on Exact Analysis of the Reference Distribution”, arXiv:1701.01960 to apper in *IEEE Trans. Information Forensics and Security* (2017).

[48] A. Iwasaki and K. Umeno, “Improving security of Vector Stream Cipher”, *NOLTA*, IEICE, 7.1 (2016): 30-37.

[49] A. Iwasaki and K. Umeno, “Further improving security of Vector Stream Cipher ”, *IEICE Technical Report, ISEC2016-28, SITE2016-22, ICSS2016-28, EMM2016-36* (2016) (In Japanese).

[50] A. Iwasaki and K. Umeno, “Methods of combining one-stroke polynomials over a ring of modulo 2^w for pseudorandom number generator and stream cipher”, Proc. SCIS2017 (2017) (In Japanese).

List of author ’ s papers related to this thesis

1 A. Iwasaki and K. Umeno,“Improving Security of Vector Stream Cipher”, *Nonlinear Theory and Its Applications*, IEICE 7.1 (2016): 30-37.

2 A. Iwasaki and K. Umeno,“One-stroke polynomials over a ring of modulo 2^w”, *JSIAM Letters*, 9 (2017): 5-8.

3 A. Iwasaki, K. Umeno, “Further improving security of Vector Stream Cipher ”, arXiv:1607.08311 (2016) (submitted to NOLTA).

4 A. Iwasaki and K. Umeno,“Three Theorems on odd degree Chebyshev polynomials and more generalized permutation polynomials over a ring of module 2^w, arXiv:1602.08238 (2016) (submitted to JJIAM).

5 A. Iwasaki and K. Umeno, “Periodical property of Chebyshev polynomials on the residue class rings of modulo 2^w”, *IEICE Technical Report, CAS2014-67, NLP2014-61* (2014): 81-86 (In Japanese).

6 A. Iwasaki and K. Umeno, “Period of orbit and degree of Chebyshev polynomial on a ring of modulo 2^w ”, *IEICE Technical Report, NLP2015-61* (2015): 129-134 (In Japanese).

7 A. Iwasaki and K. Umeno, “Further improving security of Vector Stream Cipher ”, *IEICE Technical Report, ISEC2016-28, SITE2016-22, ICSS2016-28, EMM2016-36* (2016): 115-120 (In Japanese).

8 A. Iwasaki and K. Umeno, “Methods of combining one-stroke polynomials over a ring of modulo 2^w for pseudorandom number generator and stream cipher”, *Proc. SCIS2017* (2017) (In Japanese).

9 A. Iwasaki and K. Umeno, “Randomness test based on variance of power spectrum –an alternative approach different from DFT test–”, *Proc. SCIS2017* (2017) (In Japanese).

Figures

Source of figures

Fig. 2.1 : Original.
Fig. 2.2 : Original.
Fig. 2.3 : Original.
Fig. 3.1 : Original.
Fig. 3.2 : Original.
Fig. 3.3 : Original.
Fig. 4.1: No.2 of the former page.
Fig. 4.2: No.2 of the former page.
Fig. A.1: No.9 of the former page.
Fig. A.2: No.9 of the former page.
Fig. A.3: No.9 of the former page.
Fig. A.4: No.9 of the former page.
Fig. A.5: No.9 of the former page.