The Multi-Map Orbit Hopping Mechanism

Xiaowen Zhang
Computer Science Dept.
College of Staten Island / CUNY
2800 Victory Blvd, Staten Island, NY 10314, USA

Abstract Inspired by the idea of frequency hopping mechanism for secure digital communications, in this paper we present a mixing method – the multi-map orbit hopping mechanism, which can be used as a building block in cipher designs. We discuss orbit hopping patterns, possible maps for the mechanism, and analysis to the mechanism.

Keywords: map, hopping, cipher design.

1 Introduction

According to Shannon [8], a good cryptographic cipher should have diffusion and confusion properties to thwart statistical analysis. Mixing is a way to obtain the above two properties. We see mixing phenomena in everyday life, too. A drop of red ink in a glass of water spreads through and turns the entire water into pink after a while. By a lottery machine numbers are mixed thoroughly before each drawing.

Motivated by the frequency hopping communications, we come up with the multi-map orbit hopping mechanism (Mmohom) to build ciphers. The Mmohom is used to mix numbers out of multi-orbits / multi-maps. The mechanism itself will be explained in the following sections. In the rest of this section we’ll review the frequency hopping communication.

In spread spectrum communication the bandwidth $W$ for the transmission of digital information is much bigger than the information rate $R$. The bandwidth expansion factor $(W/R)$ is much bigger than unity. There are two types of spread spectrum systems: direct sequence (DS) and frequency hopping (FH). A spread-spectrum transmission offers three major advantages over a fixed-frequency transmission [5, 10]:

- Suppressing narrowband interference due to jamming and self interference due to multipath propagation.
- Gaining message privacy since spread spectrum transmissions are difficult to intercept.
- Sharing a frequency band with many types of conventional transmissions with minimal interference, since spread spectrum signals are transmitted at lower power level of background noise.

![Figure 1: Frequency hopping pattern.](image)

2 The mechanism

2.1 Terminology

Here we explain some terms used in the Mmohom. The map is a synonym for function and it pairs each of its inputs with exactly one output [10]. Let A be
the input set (i.e. domain) and \( B \) be the output set (i.e. range). A map \( f \) is defined as \( y = f(x) \), it maps each element \( x \in A \) to exactly one element \( y \in B \). We can denote the map as \( f : A \rightarrow B \). For example, input and output \( x, y \in \mathbb{N}, y = f_1(x) = x^2 \) is a map that maps each natural number \( x \) to another natural number \( x^2 \), e.g., \( 1 \to 1, 2 \to 8, 3 \to 27, \ldots \).

The Orbit (a.k.a. trajectory) refers to the ordered set of intermediate states assumed by a dynamical map as a result of time evolution [10]. An orbit is generated by repeatedly evolving (iterating or calling) the map a certain number of times. An orbit is determined by a map, a seed (also called start point), and the number of times to evolve (you can get the end point). Here we give a definition of orbit in state machine \((\mathbb{N}, \mathcal{F})\) from [2], provided that all states are natural numbers.

**Orbit**: Given \( x_0 \in \mathbb{N} \) and a map \( f \), we define the \( n \)-th orbit of \( x_0 \) under \( f \) to be the sequence of points \( x_0, x_1 = f(x_0), x_2 = f^2(x_0), \ldots, x_n = f^n(x_0) \). The point \( x_0 \) is the seed of the orbit, \( n \) is the number of times to iterate. Here \( f^n \) means iterating \( f \) \( n \) times.

With the same map multiple orbits can be produced by using different seeds. We can obtain another orbit by adding an offset \( \epsilon \) to the initial seed \( x_0 \).

### 2.2 The Mmohom

The multi-map orbit hopping mechanism (see Figure 2) is for further mixing and muddling the points sampled from multiple orbits / multiple maps. It consists of a hopping pattern generator, an orbit control switch, and multiple orbits. Instead of hopping around different frequencies in FH system, here the mechanism hops among multiple orbits.

**Orbit hopping pattern**, like the frequency hopping pattern in FH system, is a pseudorandom sequence generated by the hopping pattern generator. If the total number of orbits is \( n \), then orbit hopping pattern contains \( n \) orbit indices from 0 to \( n - 1 \). For instance if we have 10 orbits \( \text{orbit-0} \sim \text{orbit-9} \) from a map \( f \), a possible hopping pattern could be a sequence \{4, 7, 9, 1, 3, 5, 0, 2, 8, 6\}.

The orbit hopping pattern is used by the orbit control switch to determine the order/way that the system uses multiple orbits. Use the above orbit hopping pattern for a 10-orbit mechanism we know that the switch will connect orbit-4 first, then orbit-7, orbit-9, ..., and orbit-6 in this order.

The time the switch stays connected with one specific orbit is determined by a factor called \#samples – the number of points to be sampled from the orbit, which is an analogous concept to the signaling interval in FH system. In the orbit hopping mechanism we specify one \#samples for each map, i.e., \#samples is the same for any orbit coming from the same map. For another map the \#samples may change.

The \( n \) orbits in Figure 2 are generated by multiple maps, say \( k \) maps: \( M_0, M_1, \ldots, M_{k-1} \). Each map contributes different \#orbits – number of orbits towards the \( n \) orbits. If map \( M_0 \) gives birth to \( n_0 \) orbits, map \( M_1 \) gives \( n_1 \) orbits, ..., map \( M_{k-1} \) gives \( n_{k-1} \) orbits, then \( \sum_{i=0}^{k-1} n_i = n \). For demonstration purpose orbit hopping only happens among orbits which come from one map. Of course we could make the mechanism hop among the entire \( n \) orbits which come from \( k \) maps. Similarly we could make a map hopping mechanism hop among multiple maps in the similar way as in orbit hopping mechanism. We leave these topics for future investigation.

### 3 The selected maps

All maps used in the Mmohom should have good cryptographic properties and be well-studied. Here we'll briefly introduce some maps that have been tested in the Mmohom-based ciphers.

#### 3.1 Chaotic maps

A chaotic map is a discrete map (difference equation) which generates chaotic sequence. A one-dimensional map has the format of \( x_{n+1} = f(x_n) \), and a two-dimensional map has the format of \((x_{n+1}, y_{n+1}) = f(x_n, y_n)\). Here are some chaotic
maps [1, 7, 10] which could be used to design ciphers through the Mmhom:

- Logistic: \( x_{n+1} = ax_n(1-x_n), x_n \in [0,1], a \in [3.85, 4] \);
- Chebyshev of degree \( k \): \( x_{n+1} = \cos(2^k \cos^{-1} x_n), -1 < x_n \leq 1, k = 1, 2, 3, \ldots \);
- Kolmogorov\(^1\): \( T_n(x, y) = (q_1(x - F_i), F_i + y) \mod [0, 1] \).

3.2 Primitive polynomials modulo 2

In a broad sense we treat primitive polynomials as maps when the Mmhom is applied to the LFSR-based stream ciphers. A tap polynomial of length \( n \): \( p(x) = x^n + \ldots + 1 \) over a finite field \( \mathbb{F}_2 \) is said to be primitive if it is irreducible and \( p(x) \) divides \( x^{2^n} - 1 \) but not \( x^d - 1 \) for any \( d \) that divides \( 2^n - 1 \) [9]. For example, \( p(x) = x^{32} + x^{10} + x^9 + x^9 + x^8 + x^7 + x^4 + x + 1 \) is a primitive polynomial modulo 2.

3.3 T-functions

Klimov and Shamir [3, 4] proposed the T-functions (Triangular-function) as a new class of invertible maps, with the property to be computable from the least significant bits (LSB) to the most significant bits (MSB) (i.e., information is propagated from right to left). A T-function is a mapping from \( n \)-bit to \( n \)-bit (a collection of memory words) with bitwise triangular structure such that bit \( i \) of the output depends only on bits \( 0 \)-th through \( i \)-th of the input. For instance, \( f : x \rightarrow x \oplus (x^2 \mod 2^n) \) is a T-function.

4 Analysis of the mechanism

4.1 Pseudo-cycle

If the state space is in non-negative integer, the orbit of \( x_0 \) is a sequence of numbers by iterating the map \( f(x) \). There is a cycle for any finite state machine. If \( n \) seeds \( x_{0,0}, x_{0,1}, x_{0,2}, \ldots, x_{0,n-1} \) are used, then there will be \( n \) sequences by iterating the map starting from those seeds.

How can we increase cycle by multiple orbits? Suppose a 2-bit finite state machine is used, then the total number of states is \( 2^2 = 4 \). This PRNG (pseudo-random number generator) can only generate 4 numbers \( \{0, 1, 2, 3\} \). For instance, starting from the initial state 0, then the generated sequence of orbit-0 is \( \{0, 1, 3, 2, \ldots, 0, 1, 3, 2\} \). If starting from another initial state, say 1, then PRNG generates a sequence of orbit-1 \( \{1, 2, 0, 3, \ldots, 1, 2, 0, 3\} \). They both have cycle length of 4. If the two orbits are combined into a long one by hopping, that is to pick one number at one orbit and then shift to the other orbit to pick another number, the resulted sequence is \( \{0, 1, 1, 2, 3, 0, 2, 3, \ldots, 0, 1, 1, 2, 3, 0, 2, 3\} \). It seems that the cycle is doubled (see Figure 3).

![Figure 3: Combining two orbits into one, the "cycle" is doubled.](image)

This does not contradict with the fact that the state size determines the cycle. The reason is that each state is reused twice in one "cycle." We call it a pseudo-cycle, and it does not mean that the state space is expanded.

**Pseudo-cycle**: a permutation of \( n \) elements in more than \( n \) positions. Some elements may take more than one position.

One pseudo-cycle consists of multiple smaller cycles. In Figure 3 the pseudo-cycle is \( \{0, 1, 1, 2, 3, 0, 2, 3\} \), it has smaller cycles \( \{0, 1, 1, 2, 3\}, \{1, 1\}, \ldots \). If we consider it in another way, making two adjacent states as one new "state", above sequence will become \( \{01, 12, 30, 23\} \). It has cycle of 4 of distinct elements.

**Theorem**: If \( n \) sequences generated by \( n \) orbits of \( x_{0,0}, x_{0,1}, x_{0,2}, \ldots, x_{0,n-1} \) have cycles \( t_0, t_1, \ldots, t_n-1 \), then the new sequence generated by orbit hopping mechanism has a pseudo-cycle \( t_k \geq n \times \min(t_0, t_1, \ldots, t_{n-1}) \). Here \( \min(t_0, t_1, \ldots, t_{n-1}) \) takes the minimum of \( \{t_0, t_1, \ldots, t_{n-1}\} \).

\(^1\)Given a unit square \( E \), a partition \( \pi = (p_1, \ldots, p_k) \) \((0 < p_i < 1, \sum_{i=1}^{k} p_i = 1)\) of a unit interval, let \( F_i \) defined by \( F_i = 0, F_i = F_{i-1} + p_{i-1}, F_0 = F_0 \).
4.2 Conjugate permutation

Let \( \pi \) and \( \tau \) be any two permutations of a set \( S_n \), i.e., \( \pi, \tau \in S_n \) (where \( S_n = \{1, 2, \ldots, n\} \)), and \( \tau = (i_1 i_2 \ldots i_m) \) is a cycle of length \( m \) (\( 1 \leq m \leq n \)), then

\[
\pi \tau \pi^{-1} = (\pi i_1 \pi i_2 \ldots \pi i_m),
\]

i.e., the conjugate of a cycle \( \tau \) of length \( m \) by an arbitrary permutation \( \pi \) is again a cycle of length \( m \) with entries obtained by applying \( \pi \) to all entries of the original cycle \( \tau \) [6]. \( \sigma \) and \( \pi \) are called conjugate permutations by \( \tau \) if \( \sigma = \pi \tau \pi^{-1} \) has the same cycle structure as \( \pi \).

Given \( \pi \) and \( \sigma \) in \( S_n \), find a conjugate permutation \( \tau \) in \( S_n \) such that \( \sigma = \tau \pi \tau^{-1} \). For example, \( \pi = (1, 2, 3)(4, 5, 6) \) and \( \sigma = (7, 8, 9)(1, 3, 5) \) are permutations in \( S_9 \). We can find \( \tau \in S_9 \) such that

\[
\tau(1, 2, 3)(4, 5, 6)\tau^{-1} = (7, 8, 9)(1, 3, 5).
\]

It can be written as \( \tau(1, 2, 3)\tau^{-1}\tau(4, 5, 6)\tau^{-1} = (7, 8, 9)(1, 3, 5) \). So we assume that \( \tau(1, 2, 3)^{-1} = (7, 8, 9) \) and \( \tau(4, 5, 6)^{-1} = (1, 3, 5) \), which means that \( \tau(1) = 7, \tau(2) = 8, \tau(3) = 9, \tau(4) = 1, \tau(5) = 3, \tau(6) = 5 \). So we get \( \tau = (6, 5, 3, 9)(4, 1, 7)(2, 8) \).

The Mmohom can be treated as a permutation \( \tau \) on the state space. By applying it to a regular permutation \( \pi \), we get another permutation \( \sigma \) which, we hope, has better diffusion property than \( \pi \).

5 Conclusion

The Mmohom is a conceptual extension of the frequency hopping mechanism in spread spectrum communication, which offers three major advantages over the fixed-frequency transmission: anti-jamming, message privacy, and signal hiding. Mmohom has a significant large pseudo-cycle for integer maps, better mixing, better confusion and diffusion, among other cryptographic desired properties. Our intention is to use this mechanism as a heuristic approach to build ciphers based on some well-studied maps.

References