Chaos-Based Device for Symmetric Stream Type Cipher

Veronica Bodenstein, Danny Bray

Wednesday April 12, 2023

1 Overview

In this project we will construct a chaotic system, namely a driven double pendulum, for the purpose of encryption via a symmetric stream-type cipher. Research related to chaotic maps encryption is relatively new, with novel ways to incorporate chaos for a greater security of keys. Using a physical system to create the cipher key allows for a completely secure way to create a ciphered text.

2 Project Description

General:

A driven double pendulum is by nature a chaotic system, meaning it is highly dependent on, and sensitive, to initial conditions. Utilizing this fact, we can extract sequences of numbers producing a randomized key. Fig. 1 shows the animated motion of two double pendulums started at near similar initial conditions. Each time the pendulum starts, it's path will be different unless the starting conditions are exactly identical. In a physical system however, this is nearly impossible, making it an interesting case for encryption. Using hall affect sensors we will detect when the pendulum passes certain positions and use this to encode a cipher.

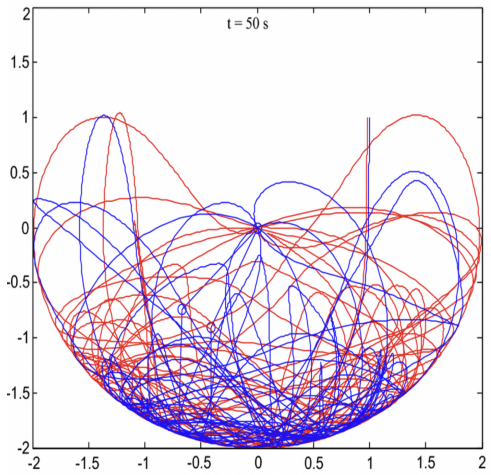


Figure 1: Diagram of Bifurcation for a Double Pendulum
from Wireless Personal Communications Article [1]

The pendulum we construct will be driven by a motor starting from a particular set of four to five initial conditions. These include the position of both pendulum arms, the input voltage, and frequency of the motor.

Encryption Process:

The basis for encryption will be an ASCII numerical key, most commonly binary; we also consider the hexadecimal system, either of which can be used. Conversion between text and an ASCII equivalent is done very efficiently, making it highly practical for cryptography.

We will create a stream cipher either with the key being the length of the text or a partial length with a loop. Each key will only be used once, for each time a message is put through our cipher machine, the pendulum will be released and a new key generated. This makes for very high security.

Currently we have two options for producing keys and depending on the speed of data acquisition via Hall affect sensor, we will determine the type we will use. The first is done via an addition operator, and is a slow variation.

It goes as follows (very slow variation):

Text: Hello World:
01001000011001010110110001101100011011110010000001010111011011110111001001101100011001 00

Key Generated by our chaotic device:
14010493474508746584930001843648439345475023038565830203494575647489020474565849302037 45

Ciphertext
150114934855097566949400029447484444658602403856584 12. . . .

We note that this doesn't need to be translated into binary, it will remain a string of numbers. This obviously is inefficient when the messages become very long. So the second method we describe will most likely be used. In this variation, we will use the known XOR operator used in stream ciphers. It goes as follows:

Text: Hello World:
01001000011001010110110001101100011011110010000001010111011011110111001001101100011001 00

Key Generated by our chaotic device: 2460094408:
00110010001101000011011000110000001100000011100100110100001101000011000000111000001110 00

Ciphertext
01111010010100010101101001011100010111110001100101100011010110110100001001010100010111 00

Translates:
zQZ_c[BT

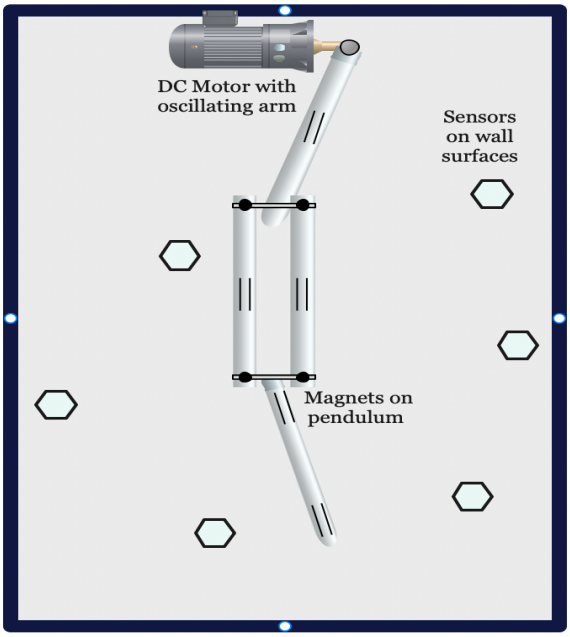
In this example the text is short so the chaotic system wouldn't need to run for very long, the whole process would be done quite fast.

The Chaos Machine!

How it works: The driven double pendulum will be constructed with the materials listed in the next section. Fig. 2a shows a rough sketch of the device. The DC motor is connected to an oscillating arm which drives the double pendulum.

The process begins when a message is given to the program to be encrypted and initial conditions specified (though, in reality, the latter in unnecessary because each time they will be inherently different anyways). Once the device is triggered, the fixed positions of the arms are released and the motor begins oscillating. The hall affect sensors on the surface walls will collected data by running a stream where each time the magnets on the pendulum pass near the sensor the stream output will read one where otherwise it will read zero (the sensors will likely begin collecting data after a brief period of time so that the system will already have diverged for non-identical initial conditions. This allows for more variability in the first portion of the key).

The magnets used will be made of neodymium since they have a very strong magnetic field for their size and price. A few different sizes will be bought in order to make sure that the hall sensors will be activated neither the entire time nor none of the time. In order to make each sensor active only when the magnet is a certain distance away, we can dampen the strength by using testing different type magnets, or by placing the sensor or magnets behind some type of surface.



(a) A few details of the pendulum system with motor driving an oscillating arm attached to a double pendulum

(b) A raw CAD design of our device, to be finalized in week 3

Figure 2: Preliminary sketches of our device are shown.

After a given amount of time that the pendulum is in motion, the program stops the collection of data when the length of the collected sequence sum equals the length of the message, or when the length of the binary sequence is equivalent to the binary message (we create the sum by a numpy array, by adding the separate arrays into one large one). Then, the program will generate a key, and use it to encrypt the message as described previously. That is the basic structure of the process by which we obtain the cipher.

The reason this type of chaotic process works for encryption is due to sensitivity to initial conditions. Only when the conditions are exactly the same, will we get the same motion.

A few notes about the sensors:

The number of sensors we have in the system depends on which number system we use, binary, decimal, or hexadecimal. We will change the number of sensors to allow for each number to occupy a single digit in the numerical system. In specific, we will have a total of 8-14 sensors all collecting a stream of data, we then add all streams producing a sequence of seemingly random numbers which we then convert to binary, acting as the key.

The sensors will be placed in varying spots along the wall where the pendulum will swing. They will be spaced out across the wall in order to be able to be activated from many different places of the pendulum arms. The exact placement will be determined as the pendulums are assembled.

Hall effect sensors allow for an increase in voltage from an increase in magnetic field density, so we will set a minimum amount for all the sensors.

Challenges:

There are three main challenges this project faces; encryption speed, randomness of the collected data, and exchange of keys for the symmetric cipher. The latter being a lesser concern, as many symmetric ciphers are currently in use with secure key exchange. Nonetheless the following is something to think about regarding that problem.

An unimportant, yet interesting side note: Living in a completely perfect world, such as a virtual matrix reality, the chaos machine key is actually only the set of initial conditions! Picture two people with this chaos machine who can set their initial conditions to match identically. They don't have to send each other a length list of numbers, rather they can create that list themselves. Now, it becomes simple to exchange keys because the

initial conditions don't mean anything for the people who don't possess the machine itself. It basically becomes a public key, for if, say, person C, stumbles upon the initial conditions exchanged between person A and person B, he cannot actually compose the key because he doesn't have the machine. Now, if the key is the same length as the message, it is proven that this message will never be cracked via other known methods.

However, back in the real world it is not quite so, therefore a secure way to exchange these keys is important to think about. Time permitting, we may devise a method to distribute these arrays in a new way in order to share the key.

To check for randomness, we will collect the sequences of data and run randomness tests. We will also release the pendulum from near identical initial conditions and compare results between the two. We will address these challenges in the intermediary weeks were we refine the way our device takes data.

Depending on the work progress and finances we may also purchase a banner sensor in week 4. Banner sensors send and receive their own beam of light, allowing for very easy sensing simply using a black string along the pendulum. In addition, they take data very fast and return with a simple number, making them ideal for this project. The issue is their steep cost, which is why Hall effect sensors are the first choice.

Conclusion:

From this project we hope to create a reliable means of generating undecipherable keys, and create very secure cipher texts. Obviously, the speed of encryption is important so ideally we will make this process as fast as possible. The practicality of having a physical chaos machine is unfortunately quite low for the real world and mass use, so this will be a encryption machine for recreational purposes and playing around with chaos.

3 Milestones

Week 1 (April 3-7): Research and design various possible version of the device, Begin CAD.

Week 2 (April 10-14): Design a sound and final version of the pendulum device, have a rough draft of CAD, order all parts, begin GitHub Repository–to be worked on each week, as we complete our project.

Week 3 (April 17-21): Write the stream-type cipher code in python. Run code for sequences of test number sets and texts. Finalize CAD design of project.

Week 4 (April 24-28): Once all parts arrive, begin construction of the device. Make sure all components–ie. the motor, sensors, magnets– are functioning properly. Use python for motor and sensor control (time delays of sensor, etc). Figure out ideal placement for sensor and magnet detection–with the least interference.

Week 5 (May 1-5): Have a prototype of the device complete. Test the prototype for functionality, speed, and randomness (described in section two). Find the earliest point of divergence of the pendulum and set this as the starting point for data acquisition via the sensors. Also determine the length of time for data collection.

Week 6 (May 8-12): Rebuild all non-functioning parts and continue testing the prototype. Debug. Secure the wiring, get a clicker to start the pendulum, and use the solenoids for setting the pendulum arms. Final week to order more parts if they break, or if other issues arise. Continued testing of our device with the written cipher.

Week 7 (May 15-19): Continue modifying the prototype and running the code. Continue attempts to encrypt and decrypt. If we have fallen behind, use some time to fix errors, otherwise spend extra time to design a strategy, or protocol, for real applications of our device (secure key exchange, for example).

Week 8 (May 22-26): Finalize the design of the pendulum device. Make the system presentable, portable, efficient.

Week 9 (May 29-June 2): Practice using the chaos device. Prepare to present the project to the class.

Week 10 (June 5-9): Present. Use the chaotic system to encrypt and decrypt a message for the class.

4 Components and Budget

Part	Cost and (%15 added Cost)	Supplier
Gear DC Motor	\$50 (\$57.5) for 2 motors	Micro DC Motors
Metal Rods	\$20.7 OR \$18.73 (\$23.805)for 2	McMaster-Carr
Flanged Ball Bearings	\$32.1 for 5 (\$36.91)	McMaster-Carr
Bolts ($\frac{1}{4}$-20)	\$8 to \$10 for 25-100 (\$11.5)	McMaster-Carr
Wood	\$12.98 (\$14.927)	Home-Depot
Paint	\$9.98 (\$11.5)	Amazon
Screws	\$9.83 for 25 (\$11)	McMaster-Carr
Hall Effect Sensor	\$6.99 for 20 (\$8.04)	Amazon
Banner Sensor	\$57.97 (\$66.6)	Mouser Electronics
Magnets(.1 thick) (neodymium)	\$.68 (per piece) (\$7.82)	McMaster-Carr
H -bridge	\$9.99 (\$11.5)	Amazon
Washers	\$2.45 for 100 (\$2.817)	McMaster-Carr
Nuts($\frac{1}{4}$ -20)	\$4.58 for 100 (\$5.305)	McMaster-Carr
Solenoids (4)	\$10.99 (\$12.65) Each	Amazon
Zip ties	\$0.00	Personal Collection
Button	\$7.68 (\$8.832)	McMaster-Carr

Total with fifteen percent added –from table above– becomes \$290.71.

5 References

[1] Bifurcation diagram https://www.researchgate.net/figure/Bifurcation-diagram-for-double-pendulum-Color-figure-online$_fig2_3$38519778

[2] https://www.mdpi.com/1099-4300/22/11/1253

[3] https://ieeexplore.ieee.org/document/7856658