Implementation of Steganography in Simai Notation for Maimai Rhythm Game Charts

M. Athaullah Daffa Kusuma M (13522044)[1,2,3]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13522044@std.stei.itb.ac.id, [2]athaullah@atha.pro, [3]daffakusumamulia@gmail.com

Abstract—Simai Notation is a text-based notation used to represent charts in the rhythm game Maimai. This notation stores information about note positions with specific values that appear at specific times. This study proposes a steganography method to embed hidden messages into Simai files by leveraging the modifiable characteristics of note positions without damaging the chart's functionality. The proposed method converts plaintext messages into an octal number system, where each ASCII character is represented as three octal digits (0–7), which are then mapped to note positions (1–8). The embedding process is performed at random positions using a pseudo-random number generator (PRNG), with a user-provided seed serving as a secret key.

Keywords—Pseudo-random, Rhythm Game, Steganography, Simai notation

I. Introduction

Maimai is an arcade-based rhythm game developed by Sega that has gained popularity not only in its home country of Japan but also in various other countries, including Indonesia. This game is renowned for its international appeal, with versions available in multiple languages, including Japanese, English (International Version), and Simplified Chinese (Chinese Version).

The gameplay in Maimai revolves around rhythmically pressing eight buttons surrounding the screen or interacting directly with the screen itself. Players synchronize their actions with the rhythm of the chosen song, responding to symbols in the form of rings or slides that emanate from the center of the screen. Maimai accommodates both single-player and multiplayer modes, supporting up to four players simultaneously.

Each song in Maimai has a chart that defines the note patterns that players must perform. This chart determines when and at which position players must provide input, as well as the type of notes to be responded to. Within the Maimai ecosystem, particularly in simulator versions and the chart-making community, there exists a notation format known as Simai Notation. Simai is a text-based markup language used to represent Maimai charts in a structured manner, encompassing tempo information (BPM), beat divisions, note positions (1-8), and various note types such as tap, hold, slide, and touch. The Simai format has become one of the standards in the Maimai chart-making community, enabling creators to create, share, and modify charts with ease.

Steganography is defined as the science and art of hiding secret messages in such a way that no one suspects the existence of the message [1]. The primary goal of steganography is to ensure that the message remains undetected. Unlike cryptography, which scrambles messages to make them unreadable while their existence remains known, steganography conceals the very existence of the message itself. This technique has been applied to various digital media such as images, audio, video, and text. However, the implementation of steganography on non-conventional media such as music notation formats remains relatively unexplored, opening research opportunities to develop new methods suited to the unique characteristics of such media.

The characteristics of Simai Notation present an interesting opportunity for steganographic implementation. The note positions in Maimai are represented by values 1-8, corresponding to the eight input areas surrounding the screen. This value range has a direct correspondence with the octal number system (0-7), where each octal digit can be mapped to a note position by adding one. Since each ASCII character can be represented in three octal digits, text messages can be embedded into chart files by strategically modifying note positions without fundamentally altering the chart structure.

This research proposes the implementation of steganographic techniques on Simai Notation by utilizing this octal-to-note position mapping. The proposed method employs a random embedding approach using a pseudo-random number generator (PRNG) with a seed as the secret key, providing an additional security layer as the embedding positions cannot be predicted without knowing the seed used. Additionally, the system is designed with an automatic header to store message length information, enabling a seamless extraction process without requiring additional information from the sender other than the seed.

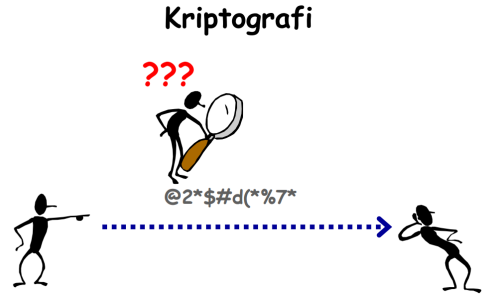
II. Basic Theory

A. Steganography and How It Differs from Cryptography

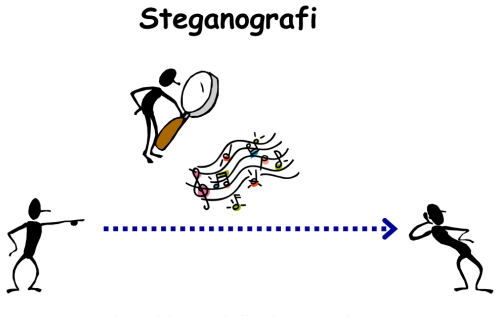
The term steganography originates from the Greek language, derived from two words: "steganos" (στεγανός) meaning hidden or covered, and "graphien" (γραφία) meaning writing. Thus, steganography literally translates to "covered writing" [1]. Steganography is defined as the science and art of hiding secret messages in such a way that no one suspects the existence of the message [1]. The primary objective of steganography is to ensure that the hidden message remains undetected. Although both cryptography and steganography aim to secure information, they employ fundamentally different approaches.

Cryptography focuses on hiding the meaning or content of a message. The objective is to make the message unreadable by third parties (adversaries). A cryptographic system transforms plaintext into ciphertext, which appears as scrambled or meaningless data. While the existence of the encrypted message is apparent, its content remains protected.

Steganography focuses on hiding the existence of a message. The objective is to avoid suspicion from third parties (adversaries). A steganographic system embeds the secret message within an innocuous cover medium, making the very presence of communication invisible to observers. The basic difference from Cryptography and Steganography shown on the image (1) and (2) below.



(1)



(2)

B. The Evolution of Steganography: Historical and Digital Methods

The age of steganography is as old as the age of cryptography, and the history of both has progressed in parallel [1]. Ancient civilizations employed various steganographic techniques, from tattooing messages on shaved heads to using invisible inks. In the modern digital era, steganography has evolved to accommodate digital media and communication channels.

Digital steganography refers to the concealment of digital messages within other digital documents [1]. The digital document used as the medium for hiding messages is called the carrier file or cover object. Common carrier files include digital images, audio files, video files, and text documents. The key requirement for a carrier file is that it must have sufficient redundancy to accommodate the hidden data without noticeable degradation.

The general model of digital steganography involves the following components:

- Cover Object: The original carrier file before embedding
- Secret Message: The information to be hidden
- Stego Key: A secret key used in the embedding/extraction process
- Stego Object: The carrier file after the secret message has been embedded

C. Random Embedding Method

One variant of steganographic embedding techniques is the random embedding method [1]. To make message concealment more secure, message bits are not stored in sequential positions within the carrier file but are instead selected randomly.

A Pseudo-Random Number Generator (PRNG) is used to generate random numbers that determine the embedding positions [1]. The seed for the random number generator serves as the key (stego-key). This approach provides an additional security layer because:

1. The embedding positions cannot be predicted without knowing the seed
2. Even if an adversary suspects the presence of hidden data, they cannot extract it without the correct seed
3. The random distribution of embedded data makes statistical detection more difficult

The extraction process requires the same seed to regenerate the identical sequence of random positions, allowing the hidden message to be retrieved in the correct order.

D. Octal Number System

The octal number system is a base-8 numeral system using eight digits: 0, 1, 2, 3, 4, 5, 6, and 7. Each octal digit represents exactly three binary digits, making it a more compact representation than binary. In the context of this research, the octal system was selected due to its direct correspondence with Maimai's note positions (1-8). By mapping octal digits (0-7) to note positions (1-8) through a simple addition of one, each octal digit can be represented as a single note position. Standard ASCII characters (0-127) can be represented in a maximum of three octal digits, where the highest octal value is 177 (127 in decimal).

III. Simai Notation

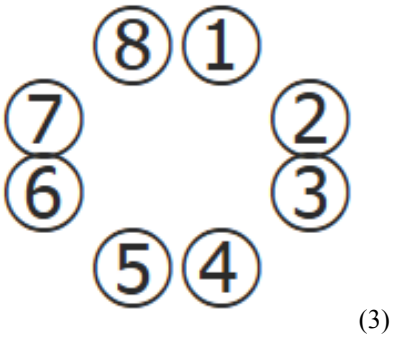
A. Definition and History

Simai Notation was created by Celeca in 2013 for denoting the various types of notes that appear in Maimai using plain text [2]. By using combinations of half-width alphanumeric characters according to defined rules, any Maimai chart can be accurately denoted using only plain text. This notation was first used in Simai, a chart simulator for Maimai, which was first released on February 5, 2013, by Celeca [2].

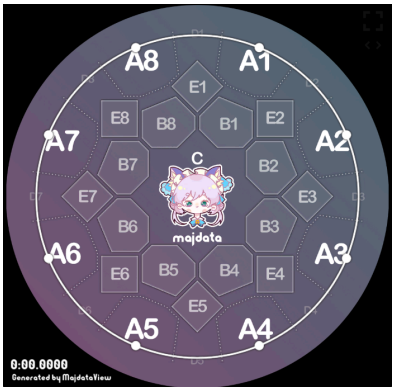
Since its introduction, various simulators inspired by Maimai have been created by volunteers, and Simai Notation has often been adopted as their chart notation format [2]. Although Celeca closed Simai to the public on February 5, 2023, there remains a significant user base that continues to use Simai Notation, which is why official documentation has been maintained [2].

B. Note Position Numbering System

In Maimai, there are eight buttons and many sensor areas surrounding the screen. These positions are numbered from 1 to 8 in clockwise order as illustrated in Figure (1), and touch sensor positions as illustrated in Figure (2).



(3)



(4)

C. BPM and Length Divider Definition

A chart in Simai Notation is composed of a series of commas and note notations [2]. Every comma occupies a certain length of time. For example, when a comma has a length of 1 second and there are 10 commas in the chart, the chart will have a total length of 10 seconds.

The notation “1,1,1,1,1,1,1,1,1,1,” represents a chart with 10 TAP notes at position 1, with a density of 1 note per second. The end of a chart is denoted by the letter "E" [2]. Without this ending notation, the music may continue playing indefinitely. Line breaks, spaces, and tabs can be inserted within the notation string for better readability; these characters are ignored during parsing.

The per-comma length can be adjusted by defining the BPM (Beats Per Minute) and/or the length divider. The BPM value is enclosed in round brackets such as “(4)”. The dividing value is enclosed in curly brackets. For example, “{4}” specifies quarter notes, “{8}” specifies eighth notes, and “{1}” specifies whole notes.

D. Note Types

1. Tap
   Tap is the most basic note type in Maimai. Tap requires the player to tap the note at the defined position. The notation consists of a button number followed by a comma. For example Tap at position 1 is “1,” and Tap at position 5 is “5,”.

2. Hold
   Hold requires the player to hold the note for the defined times at the defined position. In Simai Notation, Hold notes require specification of both the button number and the hold duration. The duration is written in the format [divider:multiplier]. For example Hold at position 5 with half-note duration is “5h[2:1],” and Hold at position 4 with 5.678 seconds duration is “4h[#5.678],”

3. Slide
   Slide notes require the player to tap at the starting point and slide at the track shape within defined duration. Slide notes require multiple parameters: starting point, track shape, ending point, and tracing length. The basic format is “[start][shape][end][duration]”. For example “1-4[8:3]” represents a straight slide from position 1 to position 4 with a tracing length of three eighth notes. All the available slide shapes are included in (5).

Table 1 Available Slide Shapes

Notation	Slide Shape
-	Straight line
>, <, ^	Arc shapes (right, left, auto shortest path)
v	V-shape through center
p, q	Curved shapes around center

s, z	Thunderbolt shapes
pp, qq	Grand curved shapes
V	Grand V-shape with turning point
w	Fan/Wi-fi shape (three endpoints)

(5)

4. Touch
   Touch notes require the player to touch some sensors. Touch notes were introduced in the Maimai DX series. The screen contains 34 sensor areas divided into five groups:
   - Group A : Adjacent to buttons (A1-A8)
   - Group B : Between Group A and center (B1-B8)
   - Group C : Center Area (C or C1/C2)
   - Group D : Between Group A Areas (D1-D8)
   - Group E : Adjacent to group B (E1-E8)

   Touch Notation uses the group letter followed by the position number. For example Touch at sensor B1 is “B1,”. The details of the position shown at Figure (4).

5. Touch Hold
   Touch Hold combines Touch and Hold notation. Touch Hold requires the player to Touch and Hold at some sensors. For example Touch Hold at center with three quarter-note duration is “Ch[4:3]”.

6. Each (Simultaneous Notes)
   When two or more notes occur at exactly the same time, they are treated as Each and are separated by “/”. For example Tap at position 1 and Hold at position 8 is “1/8h[2:1],”.

7. EX Notes
   EX Notes is the same with other non-touch notes, the only difference is that it has an expanded Critical Perfect timing window. For example Ex-Tap at position 1 is “1x,” and Ex-Hold at position 3 is “3hx[X:Y],”.

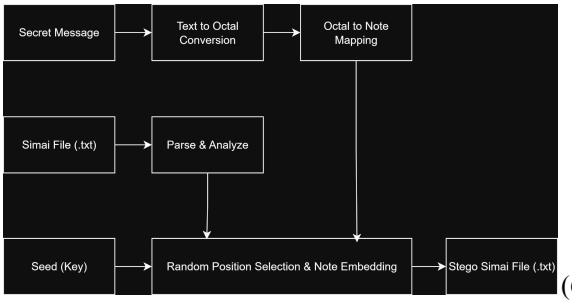
8. Firework Effect
   The firework effect creates a rainbow-colored radial visual when a Touch is activated. Firework effect can be applied to all touch and touch hold notes. Add "f" after the sensor number, for example “B7f,” is Touch at B7 with firework effect.

9. Break
   Break can be applied to all tap, hold, and slide notes. Break is the same with other non-touch notes, the only difference is that Break has a strict timing window with a bigger score than the others. Break can be added with EX Notes too, for example “3bx,” is a Break EX Note at position 3.

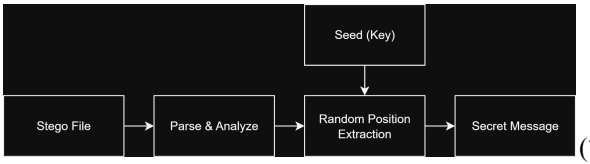
IV. Steganography in Simai Notation

A. System Overview

The proposed steganographic system consists of two main processes: embedding and extraction. Figure (6) illustrates the Process of Embedding, and Figure (7) illustrates the process of Extracting.



(6)



(7)

B. Text to Octal

The first step in the embedding process is converting the secret message into a sequence of octal digits. Since Maimai note positions range from 1 to 8, there exists a direct correspondence with the octal number system (0-7) through a simple offset of +1. Each ASCII character is converted to its octal representation using three digits. The standard ASCII character set (0-127) can be represented with octal values ranging from 000 to 177 in octal.

For example, the message “Hi!” will be converted to 110151041 (H is 72, in octal is 110, i is 105, in octal is 151, and ! is 33, in octal is 041). And will be embedded in the Simai as 221262152.

C. Data Format with Length Header

To enable automatic extraction without requiring the receiver to know the message length beforehand, the system prepends a 3-digit octal header containing the message length. With a 3-digit octal header, the maximum message length that can be embedded is 777 in octal, which is 511 in decimal. This is enough already, since simai notes count for the longest and hardest chart in Maimai history is around 2000 (including unmodifiable note type that will be discussed later).

D. Modifiable Note Types Analysis

Not all note types in Simai Notation can be safely modified for steganographic purposes. The analysis identifies which notes can be modified without disrupting chart functionality.

1. Modifiable Notes
   The following note types have position values (1-8) that can be modified:
   - Tap Notes
     Tap notes consist of a position number optionally followed by modifiers like break and EX. The position can be freely changed. For example if we want to embed "8" in "4,", we can change it into "8,".
   - Hold Notes
     Hold notes have a position number followed by 'h' and duration. Only the position is modified, the duration remains unchanged.
   - Touch Notes
     Touch notes have a group prefix (A, B, D, E) followed by a position number. Only the position is modified, the group prefix is preserved.
   - Simultaneous Notes
     Simultaneous notes can be modified, even if the modifying results have the same position as simultaneous notes. For example if i have "2/2,", it will not trigger the invalid input.

2. Non-Modifiable Notes
   The following note types are excluded from modification:
   - Slide Notes
     Slide notes have complex structures with shape characters and precise positional relationships. Modifying Slide positions would alter the visual path and potentially create invalid combinations. For example "1-4[8:3],” if i want to change it into "4-4[8:3],", it will create an invalid input (straight line cannot go from and to itself).
   - Center Touch
     Center TOUCH notes use only the letter 'C' without a position number, making them unsuitable for embedding.

D. Embedding and Extraction Algorithm

The embedding process utilizes a pseudo-random number generator (PRNG) with a seed to determine embedding positions, providing security through unpredictability. Before embedding, the system analyzes the chart to identify all modifiable positions (counting all modifiable notes). After the analysis, using the seed, positions are randomly selected for embedding. The Extraction process reverses the embedding process, using the same seed, getting the same position for the embedded notes, and that we can get the secret message.

V. Result and Analysis

A. Program Results

Here are the program results.

```
Masukkan pesan yang akan disisipkan: aku cinta kriptografi
Masukkan seed (angka atau teks): kripto

[INFO] Jumlah note yang bisa dimodifikasi: 607
[INFO] Panjang pesan: 21 karakter
[INFO] Total digit oktal yang disisipkan: 66
Masukkan path output file: umiyuri-stego.txt

[SUCCESS] Pesan berhasil disisipkan!
  - Jumlah digit yang disisipkan: 66
  - File disimpan ke: umiyuri-stego.txt
```

(8)

Figure (8) user wants to embed a secret message "aku cinta kriptografi" to "Umiyuri Kaiteitan" chart with "kripto" as the seed. So if the user wants to extract the secret message, the user needs to input the "kripto" again as the seed (or key).

```
Masukkan seed (angka atau teks): kripto

[SUCCESS] Pesan berhasil diekstrak!
  Pesan: aku cinta kriptografi
```

(9)

Figure (9) user successfully extract the secret message. But if the user inputs the wrong seed, the extracted message will be wrong as in Figure (10).

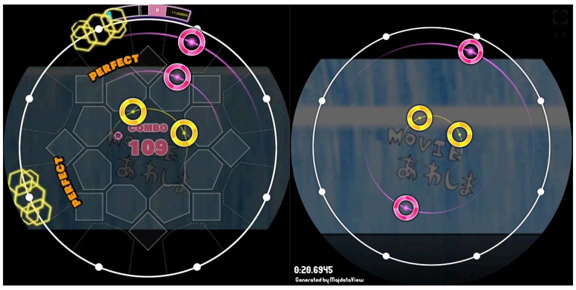
```
Masukkan seed (angka atau teks): wrongseed

[SUCCESS] Pesan berhasil diekstrak!
  Pesan: ªńĘÝ|6cőêȞaŪû_³ŸÂĉIp²Śĕ½ǨãÁħŜõŔīƩǖ–ĿJGńWżìŎĂôÿĚè9ǽǒÎ½äfţ
```

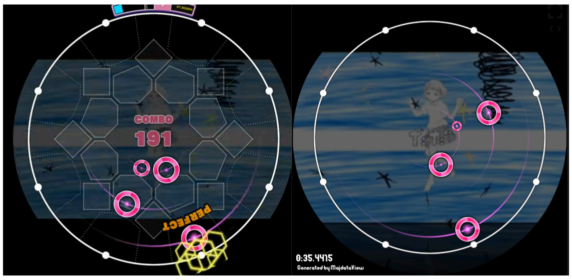
(10)

B. Visual Comparison Analysis

Here are some figures to compare the embedded charts and the original charts.



(11)



(12)

Figure (11) and Figure (12) are compared side by side in some parts of the charts of "Umiyuri Kaiteitan". As we can see, in the figure (11), the original part "1,1,2/8," is

changed to “1,5,2/8,” in the stego part. Figure (12) the original part “4,5,4,5,” is changed to “4,2,4,2,”. Although the chart is certainly changed and notifiable (also became harder to play for some cases), the stego chart is still playable.

VI. Conclusion

This research has successfully created a steganography method to hide secret messages inside Simai Notation files, which are used to make charts for the Maimai rhythm game. The method works by converting text into octal numbers (0-7) and then mapping them to note positions (1-8) in the chart. A random number generator with a seed is used to choose where to hide the data, and the seed works as a secret key for both hiding and extracting the message. The system also adds a length header automatically, so the receiver does not need to know how long the message is beforehand.

The testing results show that the method works well. All messages were hidden and extracted with 100% accuracy when using the correct seed. The charts with hidden messages still work normally in the game. All notes appear at the right time, the scoring system works correctly, and the chart structure stays the same. The method can also hide quite long messages, with Master-level charts able to store around 300-400 characters.

However, there is a major problem with this method. People can notice that the chart has been changed. The note positions become random and create weird patterns that look different from normal charts. When playing the chart, users will feel that the hand movements are awkward and unnatural. For example, if the original chart has a simple pattern like pressing buttons 1-2-1-2-1-2, the stego chart might change it to something like 2-8-4-5-6-1, which requires jumping across the screen. This makes the chart feel strange and much harder to play.

In conclusion, hiding messages in Simai Notation is possible and works correctly, but it is not truly hidden because people can notice the changes. The charts become weird in their patterns and harder to play. Therefore, this method is best used when the chart is not meant to be played seriously, such as for keeping records, proving who made the chart, or sending secret messages when it is okay if people notice something is different. Future research could try to make the embedding smarter by keeping the original patterns as much as possible, so the charts do not become so strange and difficult.

VII. Acknowledgment

I would like to thank God Almighty who has blessed me so that the work on this paper can run smoothly from start to finish. Then, I would like to thank my family, including my parents, who have supported me in studying at ITB. I also thank all the lecturers and assistants, especially Mr. Dr. Ir. Rinaldi Munir, M.T. for his guidance in studying the IF4020 Cyptography Subject. Lastly, I would like to thank my friends who have introduced me to the world of maimai, given me paper ideas and helped me work on this paper.

References

[1] Munir, Rinaldi. 2025. “Steganografi (Bagian 1)”, https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/08-Steganografi-Bagian1-2025.pdf. [Accessed: Dec. 21, 2025]

[2] L, Admin. 2023. “simai [simulator of maimai]”, https://w.atwiki.jp/simai/pages/1003.html. [Accessed: Dec. 21, 2025]

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Desember 2025

M. Athaullah Daffa Kusuma M (13522044)