Proceedings of the 10th International Conference of Control Systems, and Robotics (CDSR'23)
Ottawa, Canada – June 01 - 03, 2023
Paper No. 198
DOI: 10.11159/cdsr23.198

Music-Based Covert Channel in the Lord of the Rings Online

Jess Beckwith[1], Craig Gebo[1], Daryl Johnson[1]
[1]Rochester Institute of Technology
1 Lomb Memorial Drive, Rochester, USA
jwb5623@rit.edu; cjg2901@rit.edu; Daryl.Johnson@rit.edu

Abstract - This paper discusses a new music-based covert channel implemented within *The Lord of the Rings Online*, an online MMORPG. Many MMORPG video games have been used to implement covert channels to communicate between a sender and receiver while an observer is denied knowledge of the communication. Covert channels can be used as a method of secret communication between malicious actors or to provide a voice for those who are silenced by censorship. This covert channel is unique in that it utilizes the in-game music system to send messages by encoding ASCII characters into binary and then into music notes which can be played in the game. The implementation of this covert channel involves a sender who generates an ABC file containing the notes that represent the encoded message. The ABC file can be used to play the music within the game. The receiver uses audio recording software and music transcription software to extract the decoded message. A proof of concept implementation of this covert channel proved that it is effective for covertly communicating messages within *The Lord of the Rings Online*.

Keywords**:** covert channel, music, MMORPG, steganography

1. Introduction

A covert channel is a type of communication channel that denies an observer the knowledge that a conversation or communication is even taking place. An example would be Image Steganography, where two parties send and receive images between each other that contain hidden messages. If there was an observer they would be able to see the images but they would not know about the covert communication that is taking place.

Covert channels have been implemented within many different video games, many of which are MMORPGs (Massively Multiplayer Online Role Playing Games). Joseph Dane and Noel Kim proposed a covert channel in Runescape, which utilized the in-game trading system using the in-game items and their corresponding quantities to encode data. This encoding of messages was implemented using a REST API and bots that automatically send and receive the items which represent the message using the in-game trading system [1]. Ross Radford and Daryl Johnson proposed another covert channel within a different MMORPG, EVE Online. This covert channel involved encoding ASCII characters in the quantities of resource items that could be jettisoned from ships, the receiver could then check the jettisoned resources to decode the message [2]. Both of these covert channels use a similar method of encoding messages using in-game items and the ability to transfer items between players to send and receive the message.

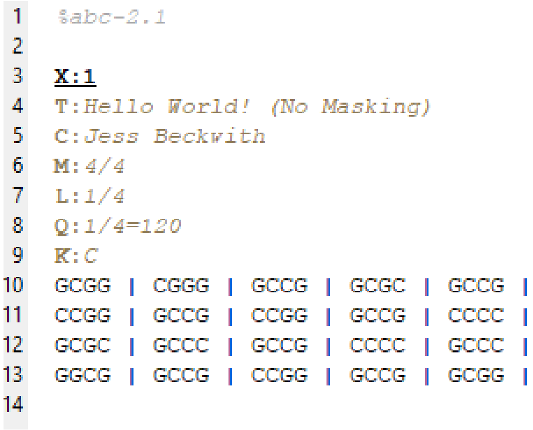
The covert channel proposed in this paper is implemented within another MMORPG, *The Lord of the Rings Online,* which will be referred to as “LOTRO” for the remainder of the paper. LOTRO is a game that allows you to play an original character in the world of Middle-earth, where you get to follow along with and even participate in the events of J.R.R. Tolkien’s The Lord of the Rings. While the vast majority of the game can be played as a single-player game, there are other players in the game that allow a person to group up into ”fellowships” with, or simply interact with through text chat and emotes as they pass by. The map is huge, full of quests and enemies, but contains several major intersections where player traffic is most commonly found. The most popular of these is the town of Bree, directly east of the Shire and central in the early stages of the game.

The covert channel within LOTRO utilizes the in-game music system, which allows players to play music at any time in any location with one of many different instruments available in the game. The music can be played manually by using your computer keyboard to play music, like a piano. However, there is only a range of 16 notes and this requires skill to perform, in addition to being harder to perfectly replicate across multiple performances. There is another option for playing music automatically using ABC notation which is a shorthand form of musical notation for computers. LOTRO offers the capability for ABC files to be written locally and stored in the game files to be played in-game. Using ABC files allows for the same music to be played every time, requires very little technical knowledge, and is very well-documented with free and open-source tools to support development.

2. Method

2.1. LOTRO Music System

ABC Notation is a computer-friendly representation of music. It has been in development since the late-90s and is still supported to the present. The concept is simple: each ABC file begins with the same headers that identify the title, the composer, the tempo, the key, and so on. Everything below the header identifies each note or rest in the song while being able to add details like length, octave, accidentals, and more for each note or rest. An example of ABC notation can be seen in Figure 1, followed by its music sheet translation in Figure 2.



```
%abc-2.1

X:1
T:Hello World! (No Masking)
C:Jess Beckwith
M:4/4
L:1/4
Q:1/4=120
K:C
GCGG | CGGG | GCCG | GCGC | GCCG |
CCGG | GCCG | CCGG | GCCG | CCCC |
GCGC | GCCC | GCCG | CCCC | GCCC |
GGCG | GCCG | CCGG | GCCG | GCGG |

```

Fig. 1: Hello World! (No Masking) ABC File

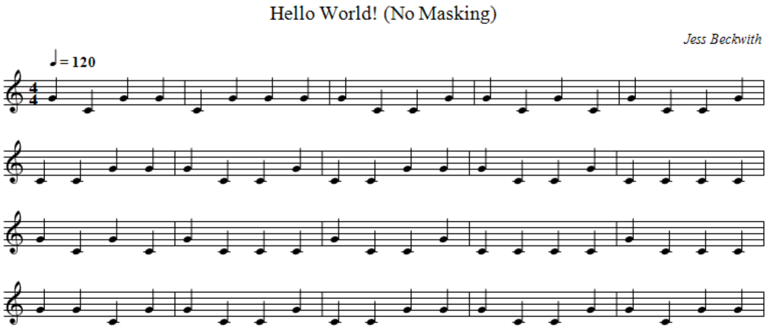


Fig. 2: Hello World! (No Masking) Sheet Music

2.2. Encoding Scheme

Every covert channel requires some sort of encoding scheme which allows the secret message to be encoded into the medium that is used to transmit the message. For this covert channel, the message needs to be encoded into music notes which can be played within LOTRO. The encoding scheme which was chosen involves converting ASCII characters to binary and representing the binary 0 with the music note G and the binary 1 with the music note C. This encoding scheme is called the two-note method, because two different musical notes are used to represent the two binary values. In the proof of concept discussed later, the message “HelloWorld” is used, Table 1 shows the encoding table for all of the characters contained within the message.

Table 1: Hello World Encoding Table

ASCII	Binary	Music Notes
H	01001000	GCGGCGGG
e	01100101	GCCGGCGC
l	01101100	GCCGCCGG
o	01101100	GCCGCCCC
W	01010111	GCGCGCCC
r	01110010	GCCCGGCG
d	01100100	GCCGGCGG

2.3. Masking

The best way to mask a message in the music is to simply write music around it. Two proof-of-concept versions of ”Hello World” were written. One was was developed by Jess Beckwith (Figure 3), one of this paper’s authors, while the other was written by Geneseo Community College student Nick Mazzola (Figure 4). The specifics of how the maskings were developed fall more under music composition and artistic expression than any scientific process, but to replicate this effect, two rules must be followed: 1) No encoded notes may be added, removed, or changed in the order they are played; and 2) Multiple notes (chords, other instruments or vocals, etc.) cannot be played at the same time. Violating either of these rules risks failing to send the complete message.

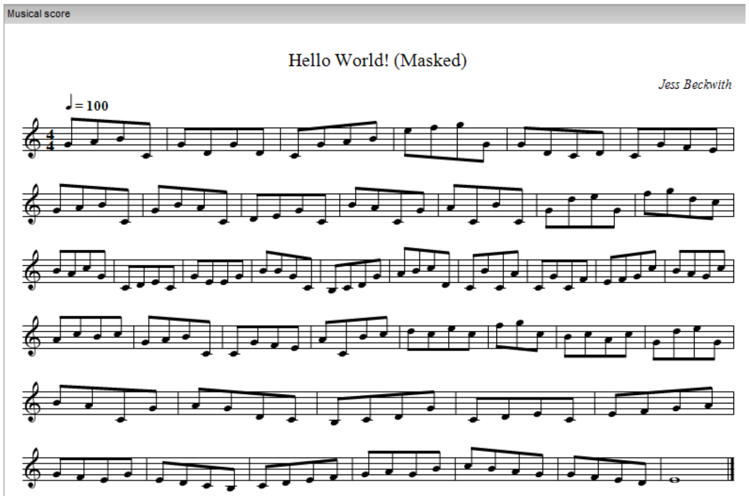


Fig. 3: Hello World! (Masked) Sheet Music by Jess Beckwith

There are a few suggestions that were found useful in developing effective maskings. The first would be to follow the concept of jazz improvisation scales, where a series of pre-selected notes are alternated in an arbitrary fashion. Using EasyABC (detailed in section 3), a sender could easily experiment with these and replay them to see if they “sound good”, which falls more under art than science. The second would be to use a standard note length when it comes to the encoded notes, to aid the receiver in having an accurate transcription. In Figure 4, this is demonstrated by making all but the final note eighth notes. Lastly, when there are two identical encoded notes that are directly next to each other, separate them with masking notes. In Figure 4, this is shown by writing arbitrary notes between every instance of two or more Gs or Cs that are adjacent to one another.

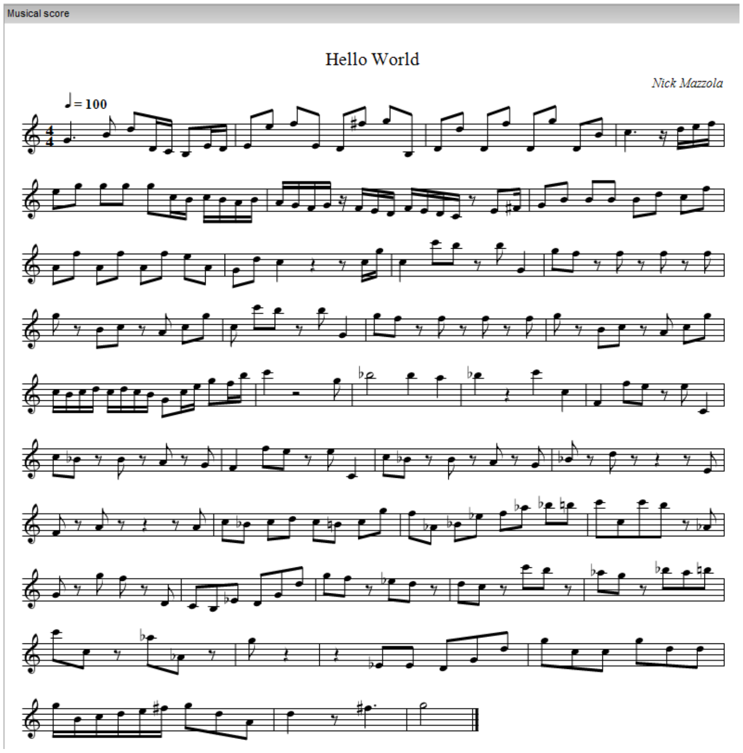


Fig. 4: Hello World! (Masked) Sheet Music by Nick Mazzola

2.4. Music Transcription

The biggest challenge of this covert channel was determining the method for converting the music back to the individual music notes to be decoded by the receiver. The first method that was explored was frequency analysis. However, this was not chosen and will be further discussed in the future work section. The chosen method was music transcription software because there were many options available and they were very easy to use.

The first music transcription program tested was Piano2Notes. This is a web-based program that allows users to upload an mp3 file and receive a PDF, MIDI, and MusicXML transcription of the music. One problem with this program is that it has restrictions on the free version, only allowing the first 30 seconds of the mp3 file to be transcribed. Also, we found that Piano2Notes did not have the most accurate transcription and we suspect this might be because it is intended for piano music. As a result of these problems, we chose not to use this program for music transcription.

The second music transcription program tested was Notation Software. This program provides the ability to convert MIDI files to sheet music. The problem with this is that another program would be needed to convert the mp3 file to an MIDI file to be transcribed using Notation Software. While this was definitely a possibility that could be considered, we decided to move forward with a program that could do the entire music transcription process by itself for simplicity and ease of use.

The third and final music transcription program that we tested was AnthemScore. This free trial of this program was chosen for the proof of concept. AnthemScore provided all of the capabilities that we needed and will be further discussed later [4].

2.5. LOTRO Instruments

Instruments are a core piece of the music system within LOTRO. In order to play music in-game, the player must have an instrument equipped. The instrument that is equipped affects how the music sounds in-game, with some sounding much better than others. This is very important for our covert channel because we want to use an instrument that clearly articulates the notes that are played. In order to determine which instrument was best we tested every instrument that was available for players up to level 10 and easily purchasable from the Bard NPC. There were a few instruments available at higher levels, but we restricted our testing to level 10 so that the covert channel would only require about 2-3 hours of gameplay to set up.

Table 2: LOTRO Instrument Tier List

Best	Clarinet, Lute
Average	Harp, Horn
Worst	Cowbell, Bassoon, Fiddle, Flute

The instruments were all tested using the same standard process. The same ABC file containing the hidden message "HelloWorld" was played using each of the instruments. The audio was recorded using Audacity and the mp3 file was transcribed using AnthemScore. The accuracy of each instruments transcription was used to create the tier list displayed in Table 2. As a result of the testing, the clarinet was chosen as the best instrument, as it produced a completely accurate music transcription.

3. Implementation

3.1. EasyABC

EasyABC is a free and open-source software developed on GitHub for the purpose of helping users write ABC files. This was the primary tool used for this purpose, as trying to write it from scratch or using only the documentation has proved difficult [6].

3.2. Audacity

For the implementation of this covert channel, we needed to choose a program to use for recording the in-game audio. Audacity was chosen because it is free to use and allows for the recording of all computer audio very easily. In order to accomplish this, all you need to do is set Audacity to record audio from your computer's audio output device and press record while the music is played in-game. It is also important to note that all audio settings within LOTRO must be set to 0 except for the setting for playing music. Once the audio has been recorded within Audacity, it should be exported as an MP3 file to be decoded [5].

3.3. AnthemScore

As we briefly mentioned previously, we selected AnthemScore as the program to use for music transcription. This program was chosen because it provides a free trial and provided very accurate results during our testing, specifically with the G and C notes. Most importantly, AnthemScore has the capability to take in an mp3 file and transcribe the audio into sheet music. This sheet music can be exported as an XML file which allows for the hidden message to be extracted and decoded using an automated script [4].

3.4. Encoding/Decoding Scripts

The encoding script is written in Python and can be used to automatically generate an ABC file containing an encoded message. The only user interaction required is when the script is initially run, it prompts the user to input the message that they want to be encoded in the ABC file. The script automatically converts the message to binary, then converts the binary to music notes based on the encoding scheme mentioned previously. Next, the script creates a basic ABC file and puts the music notes which represent the encoded message into the file. The ABC file that is produced by this script does not include any masking, but can be used to send a message very quickly if masking is not needed or could be used as a template to have the masking applied.

The decoding script is also written in Python and can be used to automatically decode an XML file exported from AnthemScore. Like the encoding script, the only user interaction required is when the program is first run, where it will prompt the user for the XML file to decode. The script then reads the XML file, parses the notes from it, converts those notes into binary, and converts the binary into ASCII. It concludes by printing the reassembled message to standard output. The completion time for the message "HelloWorld" was less than 1 second.

3.5. Proof of Concept

The sender and the receiver both agree to transmit a secret message over LOTRO. To prepare, the sender uses the encoding script to automatically generate an ABC file with the secret message. If the sender so wishes, masking can be added to better conceal the transmission when played in-game and can spend as much time as is considered reasonable on it. The ABC file is then placed in the local directory for the game to read, as directed by the LOTRO wiki page. Likewise, the receiver prepares by installing Audacity and AnthemScore on their computer and acquires the decoding script. Both sender and receiver then create new characters on LOTRO, proceed through the required tutorial section, and meet up in-game in the town of Bree. This can take up to an hour depending on familiarity with the game's mechanics and tutorial section.

Before transmission, the sender must find a Bard non-player character (NPC) vendor and purchase a clarinet for two silver pieces, which should be easily affordable after the tutorial. Once the sender and receiver meet up, they must stand within 25 meters of each other, which can be coordinated by using the minimap. The sender equips the clarinet, the receiver starts recording with Audacity and turns off all game audio, except for player music, and then the sender plays the message with two commands: "/music" and "/play message". Both remain within 25 meters of each other for the duration of the recording.

Once the recording is done, the receiver uses Audacity to export the message as an MP3 file, uploads the MP3 to AnthemScore, and then exports the sheet music as an XML file to the same directory as their decoding script. Finally, the receiver runs the decoding script on the XML file and reads the secret message. The total time required for this process varies based on the length of the message and the speed of the sender and receiver in-game, but the message length can theoretically be as large as the sender's hard drive can handle.

3.6. Bandwidth

Bandwidth is an important characteristic of covert channels. For this covert channel, the bandwidth is limited by the maximum tempo, in beats per minute, that can be successfully decoded. At this time we have not determined a maximum tempo which does not allow for successful decoding of the music. In an unmasked ABC file, each 8 notes is equivalent to one ASCII character. Given this information and the tempo of the music, the following equation can be used to calculate the bandwidth in characters per minute of the covert channel:

$$Bandwidth = Tempo / 8 \tag{1}$$

The bandwidth of a masked ABC file will be much lower than that of an unmasked ABC file. The main reason for this being that other notes are inserted into the music, so more than 8 notes are needed per character. Another potential limitation could be the maximum allowed length of an ABC file within LOTRO. This would not directly impact the bandwidth, but would limit the length of a single message. At this time, no maximum size restrictions have been identified from documentation or encountered during testing.

4. Future Work

4.1. Automation

There are two main opportunities for further automation of this covert channel. The first is to automate the masking process and the second is to fully automate the process of decoding the message using frequency analysis or pitch detection.

We have identified two possible methods for automating the masking process, the first of which is completely random masking and the second of which is machine-learning-based auto-generated music. The completely random masking could simply involve a modification to the encoding script which inserts random music notes in between the notes which represent the encoded message. This method would be easy to implement but would not result in real-sounding music. The machine-learning-based auto-generated music would involve a machine learning program that learns how to write good music, that would be able to add notes in such a way that it sounds like real music, similar to the manual masking method. These two automated masking methods would be very helpful in speeding up the ability to send messages using this covert channel.

There are various methods available for automating the process of frequency analysis or pitch detection. In the proof of concept demonstrated in this paper, we use a music transcription tool that performs the frequency analysis and maps the frequencies to music notes. The downside to this method is that the output from the transcription tool needs to be exported and manually put into our script to have the message decoded. If the frequency analysis was implemented using a python library such as Librosa then it could be possible to input an mp3 file directly into one script and get the decoded message. There are many other libraries available within different programming languages that could be used to implement frequency analysis. This approach requires much more time spent learning how frequency analysis works and how they convert the frequencies to music notes, however, if this is done then it could greatly speed up the covert channel.

4.2. Improved Music Transcription

As the maskings in Figure 3 and Figure 4 were developed, some notable issues were realized in the testing process. The implemented receiver program would sometimes have difficulty discerning the correct note lengths, and may interpret one long note as two short notes, or vice versa. This was due to a combination of two factors: note length, and the use of accidentals. For example, in Figure 3, there is an F sharp in the second to last measure, followed immediately by a G note, which is only a half-step difference. When interpreted by the receiver, this was confused as being two G notes, which caused a failure in receiving the message. To improve the effectiveness of this covert channel the receiver program into being able to discern subtle differences like accidentals and note length. As the work currently stands, Figure 3 cannot be accurately transcribed by a listener for decoding, but Figure 4 can. However, Figure 4 is more likely to put off some listeners as being bland or nonsensical, while Figure 3 would more effectively conceal any underlying intention for performance.

4.3. Alternate Mediums

LOTRO is a good medium to transmit the message because it is free, public, and unsuspecting of this covert channel. However, in any medium that music can be played, this covert channel is theoretically feasible. A sender could upload a video to YouTube, an MP3 to SoundCloud, or even perform it in real life. There are other factors that have to be accounted for in each of these mediums to minimize interference with the transmission, such as file compression or ambient noise, but the proof of concept is still a viable foundation.

4.4. Other Encoding Schemes

There were two other potential encoding schemes that we considered but did not implement in our proof of concept. These encoding schemes could be considered for future work to expand this covert channel.

The first encoding scheme is the one-note method, which involves choosing one music note and using the length of the note to represent the binary 0 and 1. For example, an eighth note would represent binary 0 and a half note would represent binary 1. The one problem with this method is that music transcription software might have trouble detecting the length of notes. During some testing of the music transcription software, it was identified that sometimes the note lengths were doubled, but the tempo was also doubled. This method could be viable if this problem was addressed.

The second encoding scheme is the table method, which involves each music note corresponding to a hex value. An example of the encoding table using all of the standard notes available within LOTRO can be seen in Table 3. The main benefit of this method is that ASCII characters can be encoded in two music notes instead of the eight notes which are required with our encoding scheme. The downside is that this encoding method is much more complicated because more music notes need to be accounted for which could result in a higher likelihood of error from the music transcription software detecting the difference between notes. If the music transcription software is effective enough then this method would be ideal for sending messages more quickly or for sending larger messages.

Table 3: Table Method Encoding Scheme

Music Note	Hex Value
c’	0
b	1
a	2
g	3
f	4
e	5
d	6
c	7
B	8
A	9
G	A
F	B
E	C
D	D
C	E
B’	F

4. Conclusion

A covert channel using music has been developed and tested in the MMORPG *The Lord of the Rings Online*. Using a combination of ABC notation, a musical expression for encoding, and a little set-up, a secret message can be transmitted through music. This covert channel is effectively free, but it is time consuming to actually prepare and requires some knowledge of music on the part of the sender. As a proof-of-concept, it sets a foundation for further research into music-based covert channels, whether through alternate mediums, further optimization, or any other factors not yet considered in this paper.

References

[1] J. Dane and N. Kim, “Runescape Based Covert Channel,” unpublished.

[2] R. Radford and D. Johnson, “Composite Covert Channels through EVE Online,” in *Proceedings of the International Conference on Security and Management (SAM),* Jul. 2013. [Online]. Available: https://worldcomp-proceedings.com/proc/p2013/SAM2108.pdf

[3] “Music - Lotro-Wiki.com,” lotro-wiki.com. [Online]. Available: https://lotro-wiki.com/index.php/music

[4] “AnthemScore | Music Transcription Software,” Lunaverus.com. [Online]. Available: https://www.lunaverus.com/

[5] “Audacity ® — Free, open source, cross-platform audio software for multi-track recording and editing.,” Audacityteam.org, May 13, 2019. [Online]. Available: https://www.audacityteam.org/

[6] N. Lindberg, “EasyABC,”, Github.com. [Online]. Available: https://github.com/jwdj/EasyABC