[Rose-Hulman Undergraduate Mathematics Journal](https://scholar.rose-hulman.edu/rhumj)

Volume 21 | Issue 2 Article 2

A Card Trick Based on Error-Correcting Codes

Luis A. Perez
California Lutheran University, luisperez@callutheran.edu

Recommended Citation

Perez, Luis A. (2020) "A Card Trick Based on Error-Correcting Codes," *Rose-Hulman Undergraduate Mathematics Journal*: Vol. 21 : Iss. 2 , Article 2.
Available at: https://scholar.rose-hulman.edu/rhumj/vol21/iss2/2

A Card Trick Based on Error-Correcting Codes

Cover Page Footnote

I would like to thank the McNair Scholars program at Cal Lutheran for providing a fellowship to conduct this research. In addition, I would like to thank my faculty mentor, Dr. John Villalpando for working with me throughout the research process.

ℝose-Hulman Undergraduate Mathematics Journal
Volume 21, Issue 2, 2020

A Card Trick Based on Error-Correcting Codes

By *Luis A. Perez*

Abstract. Error-correcting codes (ECC), found in coding theory, use methods to handle possible errors that may arise from electronic noise, to a scratch of a CD in a way where they are detected and corrected. Recently, ECC have gone beyond their traditional use. ECC can be used in applications from performing magic tricks to detecting and repairing mutations in DNA sequencing. This paper investigates an application of the Hamming Code, a type of ECC, in the form of a magic trick which uses Andy Liu's description of the Hamming Code through set theory and a known card trick. Finally, connections between this new card trick and the properties of the Hamming Code are explained.

1 Introduction

ECC are composed of error detection and correction processes. ECC are critical in today's world because they can reliably transmit information from source to receiver. The idea of guaranteeing reliable information through error-correcting codes is a concept that goes beyond traditional electronic data transmission. In general, error-correcting code concepts have been used in applications including magic tricks, strategies for games and DNA sequencing. The source and receiver are becoming more general as we are finding more applications where ECC can be used. This even includes performing card tricks. In the application of card tricks, ECC acts as a secret strategy for a magician and an apprentice that cannot be understood by the audience. We will later look into an example of a magic trick that uses error-correcting methods.

The goal of the paper is to explore a card trick, provide a newer version of the trick, and explain how the tricks use ECC concepts. The purpose of doing this is to highlight how ECC can be used outside of its traditional use such as data transmission. This goal is accomplished by reading about the card trick and understanding the ECC concepts used as well as thinking about how this trick can be improved and explain its restrictions through the ECC concepts involved. We first review a few definitions and define terms that will be used when presenting the magic trick.

Mathematics Subject Classification. 11A41
Keywords. Error-correcting codes, Parity, Modular Arithmetic, Magic Trick

Section 2 defines a few terms that will be used when presenting the magic trick. Section 3 briefly describes the previous works which lead to the development of the new card trick. Section 4 explores the parity card trick which is described by Bell, Witten and Fellows [2] and describes a new advanced version using ideas from this parity card trick and work done by Liu [3, "A magic trick"].

2 Key Definitions

Definition 2.1. *Binary* refers to the base-2 number system which only contains digits 0 and 1. We refer to a single binary digit as a *bit*. A bit string can then be defined as a collection of bits.

Example 2.2. 10101 is a bit string composed of five bits

We will later associate binary digits with colors when presenting the card trick.

Definition 2.3. *Parity bits*, or *check bits*, are defined as protection bits added to a binary string that are part of the error-correcting methodology.

The process of encoding a binary string determines how the parity bits will be chosen. We will later associate parity bits to parity cards when presenting the card trick.

Definition 2.4. The contents of a *codeword*, which is also a binary string, is the original binary string along with the parity bits that were added to it.

Example 2.5. Repetition codes take a binary string, say 101, and adds two copies of it, 101101, the parity bits, to form the codeword 101101101.

We will later see how codewords are associated to the rows and columns of a grid of cards.

Definition 2.6. *Error-correcting codes (ECC)* are companion procedures that assist a message in the process of data transmission [1]. We can generalize this definition to say that an ECC is a companion procedure which allows for the reliability of exchanging information between two parties. The first process is called *encoding* which determines how the parity bits will be chosen. The second process is called *decoding* which recovers the correct word from the codeword.

Later, we see the importance of encoding in the card trick and we get a sense of what the decoding process by the magician is like.

3 Background

Andy Liu's earliest work aims to describe a few error-correcting codes using set theory [1], including the repetition code, Rabenstein Code, and Hamming Code.

Liu's recent work draws on his previous work about the application of the Hamming Code method through set theory [3]. Of the two applications, the one that has most relevance to this research addresses a magic trick performed by an apprentice and a magician. The idea is that this duo is familiar with the communication scheme addressed in the structure of the Hamming Code through set theory in such a way where the apprentice acts as the "sabotager" in the transmission process in a strategic way. The receiver, the magician, can detect and correct the error to execute the magic trick. This work takes the functionality of the Hamming Code and extends its use beyond data transmission. Another application that we will be investigating is a card trick described by Bell, Witten and Fellows [2]. This card trick makes use of parity bits in order to find a card that the audience has selected to flip over. This card trick is further expanded using ideas from Liu's work to create a more advanced version.

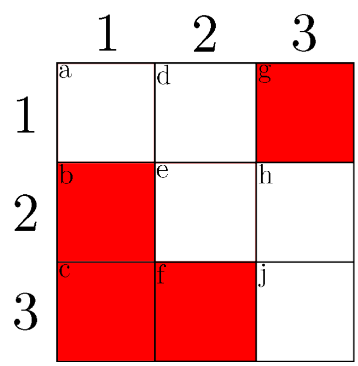
4 A Card Trick Based on ECC Concepts

We will notice that all of the factors that make these magic tricks work can be translated into the concepts that make up ECC.

4.1 The Parity Card Trick

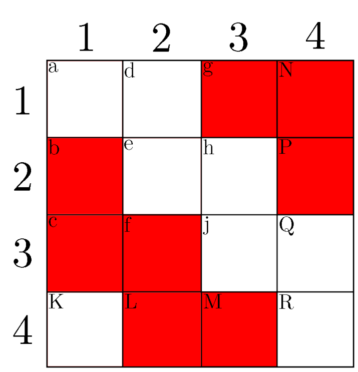
The Parity Card Trick features an interesting and simple view on error-correcting codes. The simple process explained in [2] provides an elementary understanding of what error detection and correction is all about. We take a look at this trick through modular arithmetic and then talk about other forms of the magic trick. This trick can be performed with anything that has two distinct sides such as cards, coins, books, etc. For the purpose of demonstration, we will use cards where one side is red and the other white. We now describe the trick in the case where the magician has 16 cards. Please note that only 16 cards are being used as a way to provide easy and simple visuals of the trick. With only 16 cards, a 4×4 grid can be easy memorized, taking away from the magic. However, this trick holds for any $m \times n$ grid for positive integers m, n so, for example, a 10×10 would be much harder for a magician to memorize. An explanation of this is covered in the summary. First, we break up the magic trick into different sections in order to highlight the main points.

4.1.1 Set-Up. A magician has 16 cards. The magician prompts the audience to arrange 9 of the 16 cards in a 3×3 grid with an assortment of face-up and face-down cards. We represent the colors in binary where red represents one and white represents 0. Suppose the audience arranges the nine cards as in the following figure:



We label each row and column as well as assign each of the nine cards a lower case letter; in this case, letters $a - j$, skipping letter i because it is similar to a 1.

4.1.2 Encoding: Adding Parity Cards. The magician continues the magic trick by placing the remaining seven of the sixteen cards, creating a 4×4 grid. The cards are strategically placed, and create a new column and row. After placing the remaining cards, we get the following 4×4 grid:



Note that the placement of the cards has the condition that they must form a new row and column. Therefore, there are four possible ways to add a new row and column (bottom/right, bottom/left, top/right, top/left). We assign upper case letters to these added cards; in this case, letters K – R, skipping letter O because it is similar to a 0.

These added cards are the parity bits. During the encoding process when using ECC in data transmission, the goal is to create a codeword which takes the message being sent and adds parity bits to secure the message. Similarly, the nine cards the audience arranged in the 3×3 grid act as the message and in the magician is now encoding the message by adding parity bits.

Introducing the need to place parity cards, we notice the following dependencies:

K: $\{a, b, c\}$; L: $\{d, e, f\}$; M: $\{d, e, f\}$; N: $\{a, d, g\}$; P: $\{b, e, h\}$; Q: $\{c, f, j\}$;
R: $\{\mathrm{K}, \mathrm{L}, \mathrm{M}, \mathrm{N}, \mathrm{P}, \mathrm{Q}, \mathrm{R}\}$.

These dependencies are defined above to generalize the idea of computing an even parity for a given row or column. In other words, we want to ensure that when adding a parity card to a respective row or column, the row or column will have an even number of a type of card; in this case, red. For example, when thinking about which side of the card (red or white) card K should be, we must first look at the color of cards a, b, c. If the sum of red cards in a, b, c is odd, then we want card K to be red. If the sum of red cards in a, b, c is even, then we want card K to be white.

We calculate the value/color of each parity card, that is, cards K – R, by performing modular arithmetic (modulo 2) with the sum of the cards that each parity card is dependent on. For this example, we compute the value of each parity card as follows:

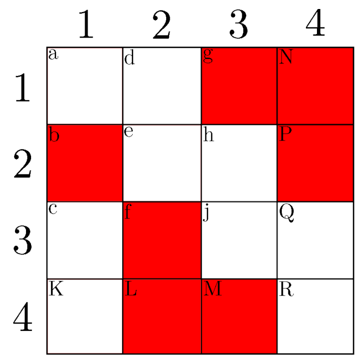
$$
\begin{aligned}
\mathrm{K} &= (a+b+c) \mod 2 & &= (0+1+1) \mod 2 & &\equiv 0 \Rightarrow \text{ white}\\
\mathrm{L} &= (d+e+f) \mod 2 & &= (0+0+1) \mod 2 & &\equiv 1 \Rightarrow \text{ red}\\
\mathrm{M} &= (g+h+j) \mod 2 & &= (1+0+0) \mod 2 & &\equiv 1 \Rightarrow \text{ red}\\
\mathrm{N} &= (a+d+g) \mod 2 & &= (0+0+1) \mod 2 & &\equiv 1 \Rightarrow \text{ red}\\
\mathrm{P} &= (b+e+h) \mod 2 & &= (1+0+0) \mod 2 & &\equiv 1 \Rightarrow \text{ red}\\
\mathrm{Q} &= (c+f+j) \mod 2 & &= (1+1+0) \mod 2 & &\equiv 0 \Rightarrow \text{ white}
\end{aligned}
$$

$$
\mathrm{R} = (\mathrm{K}+\mathrm{L}+\mathrm{M}+\mathrm{N}+\mathrm{P}+\mathrm{Q}+\mathrm{R}) \mod 2 = (0+1+1+1+1+0) \mod 2 \equiv 0 \Rightarrow \text{ white}
$$

The value computed by each parity card corresponds to the respective color. The lower case letter cards are now said to be “protected" from an error due to the placement of parity cards.

4.1.3 Introducing the Error.

Proceeding with the trick, the magician prompts the audience to arbitrarily flip one card after she/he has left the room. Note that when the audience flips the card, an error is being introduced. This error can be detected by checking for the validity of the parity cards as they have secured the message. Now that the magician has left the room, suppose the audience flips card (3,1) or card c as represented by the figure below.



4.1.4 Decoding: Error Detection. The magician comes back into the room after the audience has selected one card to flip. The main act is for the magician to accurately find the card that was flipped, since, after all, the magician was not in the room when the card was flipped. Here, we use “the error" interchangeably with the card the audience flipped. As briefly mentioned in the previous step, the way the error is detected requires re-computation of the parity card values in order to check for their validity.

Once again, we compute the values of the parity cards, keeping in mind that we already have values for them:

$$\begin{aligned}
\mathrm{K} = (a+b+c) \mod 2 &= (0+1+0) \mod 2 \equiv 1 \Rightarrow \text{ red } \neq \text{ white } \Rightarrow \text{ Invalid}\\
\mathrm{L} = (d+e+f) \mod 2 &= (0+0+1) \mod 2 \equiv 1 \Rightarrow \text{ red } = \text{ red } \Rightarrow \text{ Valid}\\
\mathrm{M} = (g+h+j) \mod 2 &= (1+0+0) \mod 2 \equiv 1 \Rightarrow \text{ red } = \text{ red } \Rightarrow \text{ Valid}\\
\mathrm{N} = (a+d+g) \mod 2 &= (0+0+1) \mod 2 \equiv 1 \Rightarrow \text{ red } = \text{ red } \Rightarrow \text{ Valid}\\
\mathrm{P} = (b+e+h) \mod 2 &= (1+0+0) \mod 2 \equiv 1 \Rightarrow \text{ red } = \text{ red } \Rightarrow \text{ Valid}\\
\mathrm{Q} = (c+f+j) \mod 2 &= (0+1+0) \mod 2 \equiv 1 \Rightarrow \text{ red } \neq \text{ white } \Rightarrow \text{ Invalid}
\end{aligned}$$

$$\mathrm{R} = (\mathrm{K}+\mathrm{L}+\mathrm{M}+\mathrm{N}+\mathrm{P}+\mathrm{Q}+\mathrm{R}) \mod 2 = (0+1+1+1+1+0) \mod 2 \equiv 0 \Rightarrow \text{ white } = \text{ white } \Rightarrow \text{ Valid}$$

When checking for the validity of the parity bits, we notice that parity cards K and Q are invalid because the computed value suggests that both should be red but they are white. Since we know there is only one possible mistake or one card flipped, we need to consider which card is able to cause parity cards K and Q to be valid. If we refer back to their dependencies, we notice that they both share a card of dependency, card c. Thus, if we were to flip c, all the parity cards would become valid so we conclude that card c was flipped.

4.1.5 Summary. When comparing this trick to the way we proceed in the encoding and decoding process in data transmission, the three by three grid created by the audience acts as the message attempted to be sent. The cards the magician adds act as the parity bits or check bits, securing the message. The encoded message is then the four by four grid which can be referred to as the codeword. When the audience flips a card, an error is introduced. Therefore, when the magician returns after a card has been flipped by the audience, the process of figuring out the flipped card can be referred to as decoding the codeword which amounts to detection of the error.

Notice that in order for an error to be detected and corrected, there must always be exactly two invalid parity cards when the parity cards are re-computed for validity. In fact, when the audience flips one card, it is guaranteed that two parity cards will be invalid. This is a result of the parity cards' dependence. In addition, these two invalid parity cards must be distributed in a way where one must be located on the row and the other on the column.

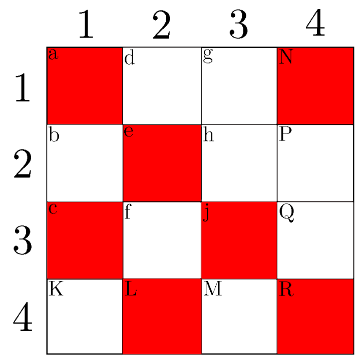
Another property of this magic trick is that it can be played with any number of cards as long as the card count meets the condition of being composite. This property is important to the trick as it is may seem trivial for a magician to perform with 16 cards. Further, the 16 cards were used to demonstrate the trick but in a show, the magician should use a large number of cards in order to make the trick look impressive. This property was found by noticing that in order to perform the magic trick, an $m \times n$ grid where $m, n \geq 1$ is needed which cannot be formed with a prime number of cards as an $m \times 1$ or $1 \times n$ grid would be formed. This is due to the prime number's inability to be divisible by other numbers except for itself and one.

4.2 A New Magic Trick

Given some inspiration from the magic trick by Bell, Witten and Fellows [2] which was previously discussed and the magic trick presented by Liu [3], a new version of the Parity Card Trick is created. Major differences in this parity card trick from the Bell, Witten and Fellows trick include an apprentice and a magician instead of only a magician, and having the audience select a card but not flip it as opposed to having the audience select a card and flip it. Also, the audience gets to choose the complete arrangement of the grid and the apprentice is responsible for flipping card(s) in order for the magician to correctly guess the card the audience selected. Finally, the magician does not witness anything as she/he leaves the room before the process starts. The end result is still the same, the trick ends with the magician coming back into the room and correctly picking the card the audience selected.

We describe the trick in the case where the magician has 16 cards. We break up the magic trick into different sections in order to highlight the main points.

4.2.1 Set-Up. The magician leaves the room while the apprentice prompts the audience to arbitrarily arrange 16 cards in a 4×4 grid. Suppose the audience arranges the cards as follows:



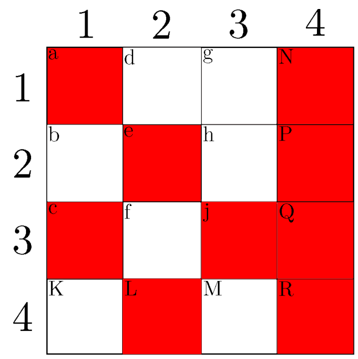
4.2.2 Introducing the Error. The apprentice prompts the audience to select a card but not flip it nor take it out of the grid. Note that the apprentice knows which card is selected. Let's assume that the audience selects card j or (3,3). We now have an arbitrary 4×4 grid with a target card, j. Now, the goal for the apprentice is to flip the necessary cards in order for the magician to guess which card the audience selected. In other words, the apprentice needs to strategically flips cards so that it is obvious to the magician where the "error" card is located. The apprentice will purposely validate and invalidate parity cards based on the card the audience selected.

4.2.3 Strategic Encoding. Since the audience has selected card j, the apprentice needs to flip cards so that all parity cards are valid except for Q and M, as these are the parity cards which include j in their dependence. We consider the validity of the check bits (referencing the figure above):

$$\begin{aligned}
\mathrm{K} &= (a+b+c) \mod 2 = (1+0+1) \mod 2 \equiv 0 \Rightarrow \text{ white } = \text{ white } \Rightarrow \text{ Valid}\\
\mathrm{L} &= (d+e+f) \mod 2 = (0+1+0) \mod 2 \equiv 1 \Rightarrow \text{ red } = \text{ red } \Rightarrow \text{ Valid}\\
\mathrm{M} &= (g+h+j) \mod 2 = (0+0+1) \mod 2 \equiv 1 \Rightarrow \text{ red } \neq \text{ white } \Rightarrow \text{ Invalid}\\
\mathrm{N} &= (a+d+g) \mod 2 = (1+0+0) \mod 2 \equiv 1 \Rightarrow \text{ red } = \text{ red } \Rightarrow \text{ Valid}\\
\mathrm{P} &= (b+e+h) \mod 2 = (0+1+0) \mod 2 \equiv 1 \Rightarrow \text{ red } \neq \text{ white } \Rightarrow \text{ Invalid}\\
\mathrm{Q} &= (c+f+j) \mod 2 = (1+0+1) \mod 2 \equiv 0 \Rightarrow \text{ white } = \text{ white } \Rightarrow \text{ Valid}
\end{aligned}$$

$$\mathrm{R} = (\mathrm{K+L+M+N+P+Q+R}) \mod 2 = (0{+}1{+}0{+}1{+}0{+}0{+}1) \mod 2 \equiv 1 \Rightarrow \text{ red } = \text{ red } \Rightarrow \text{ Valid}$$

The apprentice notices that parity cards M and P are false. Thus, the apprentice realizes that by flipping cards P and Q, all parity cards will be valid except for Q and M. This is represented in the following grid:



4.2.4 Decoding: Detecting the Error. The magician comes back into the room and sees the grid above. In order to complete the trick, the magician must select the card the audience selected, card j. Notice that, at this point, the grid is set so that, by checking for the validity of the parity cards, the magician will be able to correctly guess the card that was selected. So, checking for validity of the parity cards (referencing the grid above), we get:

$$\begin{aligned}
\mathrm{K} = (a+b+c) \mod 2 &= (1+0+1) \mod 2 \equiv 0 \Rightarrow \text{ white } = \text{ white } \Rightarrow \text{ Valid}\\
\mathrm{L} = (d+e+f) \mod 2 &= (0+1+0) \mod 2 \equiv 1 \Rightarrow \text{ red } = \text{ red } \Rightarrow \text{ Valid}\\
\mathrm{M} = (g+h+j) \mod 2 &= (0+0+1) \mod 2 \equiv 1 \Rightarrow \text{ red } \neq \text{ white } \Rightarrow \text{ Invalid}\\
\mathrm{N} = (a+d+g) \mod 2 &= (1+0+0) \mod 2 \equiv 1 \Rightarrow \text{ red } = \text{ red } \Rightarrow \text{ Valid}\\
\mathrm{P} = (b+e+h) \mod 2 &= (0+1+0) \mod 2 \equiv 1 \Rightarrow \text{ red } = \text{ red } \Rightarrow \text{ Valid}\\
\mathrm{Q} = (c+f+j) \mod 2 &= (1+0+1) \mod 2 \equiv 0 \Rightarrow \text{ white } \neq \text{ red } \Rightarrow \text{ Invalid}
\end{aligned}$$

$$\mathrm{R} = (\mathrm{K{+}L{+}M{+}N{+}P{+}Q{+}R}) \mod 2 = (0{+}1{+}0{+}1{+}1{+}1{+}1) \mod 2 \equiv 1 \Rightarrow \text{ red } = \text{ red } \Rightarrow \text{ Valid}$$

The magician notices that parity cards M and Q are invalid. Since cards M and Q both depend on card j, the magician will pick card j, the card the audience selected.

4.2.5 Summary. This new magic trick is more difficult because it requires the ability to problem solve under very little time as well as a solid undetected communication scheme between the apprentice and magician. What makes this trick more impressive is that the magician never sees what is going on. In addition, the card the audience selects is not flipped.

A thing to consider is when the apprentice flips the cards. There is more than one combination of cards which would make the magic trick work. However, for purposes of the trick, we are interested in the flipping the least amount of cards, so there is only one optimal combination. The least amount of cards we flip depends on the current validity of the parity cards and the cards we want to be valid and invalid.

References

[1] Alon N., Liu A. An Application of Set Theory to Coding Theory, *Mathematics Magazine*, Mathematical Association of America, Vol. 62, No. 40, Oct. 1989, pp. 233-237.

[2] Bell T., Witten I., Fellows M. Computer Science Unplugged: The Original Activities Book, *CS Unplugged*, Vol. 1, 1998, pp. 33-40.

[3] Liu A. Two Applications of a Hamming Code, *The College Mathematics Journal*, The Mathematical Association of America, Vol. 40, No. 1, Jan. 2009, pp. 2-5.

Luis A. Perez
California Lutheran University
`luisperez@callutheran.edu`