Spring 4-27-2018

Non-Commutative Zero-Knowledge Protocols

Bailey Stewart

Non-commutative Zero-Knowledge Protocols

Bailey J. Stewart

submitted in partial fulfillment of the requirements for Honors in Mathematics at the University of Mary Washington

Fredericksburg, Virginia

April 2018

This thesis by **Bailey J. Stewart** is accepted in its present form as satisfying the thesis requirement for Honors in Mathematics.

Date	Approved
_____	_____
	Randall D. Helmstutler, Ph.D. thesis advisor
_____	_____
	J. Larry Lehman, Ph.D. committee member
_____	_____
	Jennifer Magee, M.A. committee member

Contents

Abstract

We examine zero-knowledge protocols working in non-commutative structures. Specifically, we will discuss the advantages and disadvantages of using invertible elements and elements that have a square root. An adaptation of the zero-knowledge protocol will be presented for working in symmetric groups and monoids of endomorphisms and matrices. Additionally, the security behind these zero-knowledge protocols will be discussed as well as showing different scenarios where invertible elements and elements with squares are vulnerable to attacks.

1 Background

1.1 Overview

Throughout this paper, we will discuss the concepts behind zero-knowledge protocols as they pertain to cryptography. We will refer to the *honest prover* as the person holding a particular piece of information that should not be divulged, the *verifier* who does or does not believe a statement, the *trusted center* as the authority that establishes the setup requirements, and the *cheating prover* as the person trying to convince the verifier they are an honest prover. We will call the honest prover Peggy, the verifier Victor, the trusted center Trudy, and the cheating prover Eve. In the following section, we will examine the general setup and exchange on the original scheme written in $\mathbb{Z}_n$ developed by Uriel Feige, Amos Fiat, and Adi Shamir.

1.2 Zero-Knowledge Protocols

The Feige-Fiat-Shamir Identification Protocol, also known as the original zero-knowledge protocol (ZKP), was first published in 1988 in the Journal of Cryptology [1]. The main concept behind the ZKP is to allow a prover to prove something to a verifier while limiting the amount of information that is conveyed. For example, suppose Peggy lives in a red house in California and Victor lives in Virginia. Peggy will convince Victor that her house is red, without ever sending Victor a picture of her house. Peggy uses a ZKP to limit impersonation since anyone is able to send Victor a picture of a red house. Therefore, Victor and Peggy will trust each other and Eve will not have enough information to impersonate Peggy.

To begin, we will establish the general setup and exchange of this ZKP working in $\mathbb{Z}_n$ multiplicatively, where n is the product of two large prime numbers. Before the verification begins, Peggy and Victor agree on how many times, m, the process is repeated. Below the setup and exchange will be described for one round.

FFS ZKP: the setup.

1. Trudy chooses two large distinct prime numbers, p and q, and sets $n = pq$. Trudy publishes n to the general public, but does **NOT** publish p or q.
2. Peggy creates a secret number a and reduces it mod n.
3. Peggy computes b where $b \equiv a^2 \mod n$. When Peggy is ready to communicate with Victor she will send him the element b corresponding to her a.

   Note. Victor is not able to recover a by the difficulty in determining a modular square root in $\mathbb{Z}_n$ without knowledge of the factors of n.

FFS ZKP: the exchange.

1. Peggy chooses a random integer r and computes $s \equiv r^2 \bmod n$. She sends s to Victor.
2. Victor chooses a random bit t where $t \in \{0, 1\}$. Victor then sends his bit to Peggy.

   Note. Peggy will not know what t is until after she sends Victor her s.
3. Peggy computes $y \equiv ra^t \bmod n$. She sends y to Victor.
4. Victor confirms that $y^2 \equiv sb^t \bmod n$. Victor will only accept the response if $y^2 \equiv sb^t \bmod n$.

Note. Since $\mathbb{Z}_n$ is a commutative ring, we have $(ra)^2 = r^2a^2$. Substituting all of these expressions into the equation $y^2 \equiv sb^t$, we will either get:

$$(ra)^2 \equiv r^2a^2 \equiv sb^1 \text{ in the case } t = 1$$
$$r^2 \equiv sb^0 \text{ in the case } t = 0.$$

Therefore, Victor will accept that Peggy is a true prover if $y^2 \equiv sb^t$ for every round.

It is important to note the significance of selecting a random t. This step is the most vital since it will determine the value Victor expects to receive. It is essential for Victor to randomly choose this bit so Peggy does not always receive the same value for t. If Victor sends the same value every time, it will allow an impersonator to trick Victor into believing she is a valid prover with a success rate of 100%. This attack process will be discussed in Section 1.3. Below is an example demonstrating this ZKP in $\mathbb{Z}_n$. This paper will only demonstrate one round of the ZKP exchange, but recall that Peggy and Victor agree on m exchanges before the verification begins. The value for m can be unique for each verification between two different people.

Example 1.1. The following example demonstrates the general setup and exchange in $\mathbb{Z}_{327287}$.

1. Trudy selects $p = 509$ and $q = 643$ and calculates $n = 327287$. Trudy publishes $\mathbb{Z}_{327287}$ to Peggy and Victor.
2. Peggy creates a secret number $a = 58469$.
3. Peggy calculates $b \equiv a^2 \equiv 111246 \bmod 327287$. She sends Victor b because she is ready to communicate with him.

Next, Peggy and Victor participate in the exchange outlined above.

1. Peggy chooses $r = 41685$ and computes $s \equiv r^2 \equiv 72542 \bmod 327287$. Peggy sends s to Victor.
2. Victor chooses $t = 1$ and sends his bit to Peggy.
3. Peggy computes $y \equiv ra^t \equiv 41685 \cdot 58469^1 \equiv 301263 \bmod 327287$. Peggy sends y to Victor.
4. Victor checks that $y^2 \equiv sb^t$ and will believe Peggy if and only if they are equal. He computes

   $$y^2 \equiv 301263^2 \equiv 91773 \bmod 327287$$
   $$sb^t \equiv 72542 \cdot 111246 \equiv 91773 \bmod 327287.$$

   Victor believes Peggy since $y^2 \equiv sb^t$.

In the following example, we will notice the slight difference when Victor chooses $t = 0$. This process will reveal Peggy's secret r, but for each round of the exchange Peggy will choose a different r value. Victor will never be able to discover the secret value a Trudy creates because a is never isolated and cannot be discovered due to the hardness of finding a modular square root.

Example 1.2. This example uses the previous setup, but will demonstrate the difference when Victor chooses $t = 0$.

1. Peggy chooses $r = 96521$ and computes $s \equiv r^2 \equiv 78986 \bmod 327287$. Peggy sends s to Victor.
2. Victor chooses $t = 0$ and sends his bit to Peggy.
3. Peggy computes $y \equiv ra^t \equiv 96521 \cdot 58469^0 \equiv 96521 \bmod 327287$. Peggy sends $y = r$ to Victor.
4. Victor checks that $y^2 \equiv sb^t$ and will believe Peggy if they are equal. He computes:

$$y^2 \equiv 96521^2 \equiv 78986 \bmod 327287$$
$$sb^t \equiv 78986 \cdot 111246^0 \equiv 78986 \bmod 327287.$$

   Victor believes Peggy since $y^2 \equiv sb^t$.

In addition to not revealing any information, any strong ZKP must satisfy two more requirements: completeness and soundness. Since ZKPs must have these we are able to classify a ZKP under the category of an interactive proof system.

Definition 1.3. An *interactive proof system* is a system wherein two parties continuously exchange messages until the verifier has received an answer from a prover and is convinced that it is correct.

The exchange is repeated a certain number of times until the verifier is convinced he is communicating with an honest prover. Any practical ZKP would satisfy the following two properties, taken from [4]:

Completeness - For every $r \in \mathbb{Z}_n$, the verifier always accepts the outcome s after interacting with the prover on common input r.

Soundness - For some s that is not a square, the verifier accepts the outcome with a probability of at most $\frac{1}{2^m}$ after interacting with the prover on a common input.

This protocol is considered complete since an honest verifier will always be convinced of a true statement from an honest prover. Similarly, the exchange is sound since a cheating prover can convince an honest verifier that some false statement is actually true with only a small probability. The general setup of the ZKP only exists if one-way functions exist, such as squaring mod n. A cheating prover should not be able to obtain any additional information that is not already known as public.

1.3 Security and Attack

This zero-knowledge protocol is used today because of the main feature of the ZKP: the prover will not disclose any information to the verifier during the verification process. This concept provides a more secure way of communication. As discussed in the previous section, the properties of completeness and soundness convey the fact that a verifier will always accept the outcome when

communicating with an honest prover. Also, it is extremely unlikely that a cheating prover will be able to convince the verifier of a false statement.

Using these two conditions, we can determine the minimum number of rounds m needed to prove communication with an honest verifier. As discussed in the definition of soundness, the verifier will accept the false outcome with a probability of $\frac{1}{2^m}$ when completing m rounds of the protocol. For example, to show that there is less than a 1% chance that Victor verified a false statement, we must solve for m:

$$\begin{aligned}
\frac{1}{2^m} &\leq 0.01 \\
2^{-m} &\leq 0.01 \\
\ln(2^{-m}) &\leq \ln(0.01) \\
-m\ln(2) &\leq \ln(0.01) \\
-m &\leq \frac{\ln(0.01)}{\ln(2)} \\
m &\geq 6.64.
\end{aligned}$$

Therefore, after 7 rounds of the exchange, there is less than a 1% chance that Victor verified a false statement.

There is one known attack on this ZKP. In this attack, Eve is impersonating Peggy; Victor believes he is talking to Peggy. However, this attack is only effective 50% of the time. To successfully execute this attack, Eve must correctly predict the value Victor is going to choose for t. Remember, Victor does not choose t until after Eve sends Victor s.

If Eve believes Victor is going to choose $t = 0$, then she will follow the normal exchange. We recall that when $t = 0$, $y^2 = s$. However, if Eve changes her r, this will result in an s that will not equal y^2. If Eve believes Victor is going to choose $t = 1$ at the very beginning, then she will send Victor $s' = sb^{-1}$ instead of $s = r^2$.

This is the main reason why Victor must randomly choose his t value. If he always chooses $t = 1$, then Eve will send $s' = sb^{-1}$ for every round. If he always sends $t = 0$, then Eve will not change her value for s.

Example 1.4. For this example, we will use the same setup as in Example 1.1.

1. Eve chooses $r = 63251$ and computes $s \equiv r^2 \equiv 260000 \bmod 327287$. Eve believes $t = 1$ so she sends $s' = sb^{-1}$ to Victor instead, where $s' = 15189 \bmod 327287$. This value is disguised as r^2.

2. Victor chooses $t = 1$ and sends his bit to Peggy.

3. Since Victor chose $t = 1$, Eve will send Victor $y = r$.

4. Victor checks that $y^2 \equiv sb^t$. He computes:

$$\begin{aligned}
y^2 &\equiv 63251^2 \equiv 260000 \bmod 327287 \\
sb^t &\equiv 15189 \cdot 111246^1 \equiv 260000 \bmod 327287.
\end{aligned}$$

   Victor believes Eve is Peggy since $y^2 \equiv sb^t$.

Note. This exchange only worked since Eve correctly guessed that $t = 1$. This exchange would not have worked if Victor chose $t = 0$ because the following calculations would have been made:

$$y^2 \equiv 260000 \bmod 327287$$
$$sb^t \equiv 15189 \cdot 111246^0 \equiv 15189 \bmod 327287.$$

Victor would not believe Eve is Peggy because $y^2 \not\equiv sb^t$.

In the following example, Eve correctly guesses the value of t. As opposed to Example 1.4, Eve will choose $t = 0$ instead of 1. As discussed earlier in this section, Eve does not have to manipulate the exchange in any way if she correctly predicts $t = 0$.

Example 1.5. For this example, we will use the same setup as in Example 1.1 and a similar exchange as in Example 1.4.

1. Eve chooses $r = 63251$ and computes $s \equiv r^2 \equiv 260000 \bmod 327287$. Eve sends s to Victor.
2. Victor chooses $t = 0$ and sends his bit to Peggy.
3. Eve computes $y \equiv ra^t \equiv 63251 \cdot 58469^0 \equiv 63251 \bmod 327287$. Eve sends y to Victor.
4. Victor checks that $y^2 \equiv sb^t$:

$$y^2 \equiv 63251^2 \equiv 260000 \bmod 327287$$
$$sb^t \equiv 260000 \cdot 111246^0 \equiv 260000 \bmod 327287.$$

Victor believes Eve since $y^2 \equiv sb^t$.

Note. This exchange only works if Eve correctly guesses that $t = 0$. This exchange would not have worked if Victor chose $t = 1$. This is because $y^2 \equiv r^2 \equiv sb^0$.

We will now look at an example where Eve incorrectly guesses the value of t. Since she incorrectly guesses the value of t, Victor will know she is not an honest prover and will therefore discontinue his communication with her.

Example 1.6. For this example, we will use the same setup as in Example 1.1.

1. Eve chooses $r = 63251$ and computes $s \equiv r^2 = 260000 \bmod 327287$. However, Eve assumes that $t = 1$ so she sends $s' = sb^{-1}$ to Victor instead, where $s' = 15189 \bmod 327287$.
2. Victor chooses $t = 0$ and sends his bit to Peggy.
3. Eve guessed incorrectly. Therefore, no matter what value she sends for y, the statement will **NEVER** be verified. Eve sends $y = 63251$ to Victor.
4. Victor checks that $y^2 \equiv sb^t$:

$$y^2 \equiv 63251^2 \equiv 260000 \bmod 327287$$
$$sb^t \equiv 15189 \cdot 111246^0 \equiv 15189 \bmod 327287.$$

Victor does not believe Eve since $y^2 \not\equiv sb^t$. In this example, $y^2 \equiv r^2$, but $sb^t \equiv sb^{-1}b^0 \equiv sb^{-1}$. Thus, $r^2 \not\equiv sb^{-1}$ and Victor knows he is not talking to an honest prover.

This is the only known attack for this ZKP. However, this attack does not always work, because Eve has only a 50% chance per round of correctly predicting t. Therefore, the likelihood that Eve would be able to convince Victor that she is an honest prover is a different probability than that for Peggy in convincing Victor that she is an honest prover. Eve has a probability of convincing Victor she is an honest prover of only $\frac{1}{2^m}$. Peggy has a probability of 100% of convincing Victor she is an honest prover since she will never have to worry about guessing the value for t or solving the modular square root problem.

2 Zero-Knowledge Protocol over Symmetric Groups

Throughout this section and the rest of this paper, we will examine the setup and exchange of a zero-knowledge protocol in a non-commutative setting. Instead of focusing on commutative structures, we will shift our attention to non-abelian groups and eventually non-commutative monoids. Specifically, we will focus on symmetric groups, endomorphisms of finite sets, and matrices. We believe these three structures will potentially increase the security of our ZKP and therefore make it harder for Eve to impersonate an exchange.

2.1 Non-commutative ZKP over Symmetric Groups

The setup and exchange of ZKP over symmetric groups is very similar to the setup and exchange of ZKP in $\mathbb{Z}_n$. We recall that the symmetric group S_n is non-abelian when $n \geq 3$; however, for this exchange to work, Peggy must make some choices that depend on commutativity, though the overall structure is still non-commutative.

ZKP over S_n: the setup.

1. Peggy and Victor agree upon an S_n for fixed n.
2. Peggy creates a secret permutation a.
3. Peggy computes b where $b = a^2$. When Peggy is ready to communicate with Victor she will send him the permutation b corresponding to her a.

ZKP over S_n: the exchange.

1. Peggy chooses a random permutation r in S_n that commutes with a and computes $s = r^2$. She sends s to Victor.
2. Victor chooses a random bit t where $t \in \{0, 1\}$. Victor then sends his bit to Peggy.
3. Peggy computes $y = ra^t$. She sends y to Victor.
4. Victor confirms that $y^2 = sb^t$. Victor will only accept the response if $y^2 = sb^t$.

Note. Notice here that r and a must commute with one another. This is the most important rule when working with this ZKP. The group does not need to be commutative itself, but the private elements a and r must commute with one another. They must commute with one another in order for the math to work and for the corresponding answer to equal y^2.

Theorem 2.1. *Victor will always correctly verify an exchange over S_n when a and r commute with one another.*

Proof. For Victor to always correctly verify an exchange he must confirm that $y^2 = sb^t$. We know that $y^2 = (ra^t)^2$. This expression will simplify to ra^tra^t. Peggy will always create an r that commutes with a, so the expression can be rearranged to equal rra^ta^t. Now we can write the like terms with exponents to equal r^2a^{2t} which we know will either equal sb when $t = 1$ or s when $t = 0$. □

When working in symmetric groups, there is an easy way to create commuting elements; however, this technique leaves room for Eve to easily impersonate Peggy. This technique revolves around splitting the set $\{1, 2, \ldots, n\}$ into two equal halves, typically $\{1, 2, \ldots, \frac{n}{2}\}$ and $\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n\}$. When Peggy originally creates her secret permutation a she will only transform the first half of the set, while leaving the second half fixed. Then, when Peggy is creating her random permutation r she only transforms the second half of the set while leaving the first half fixed. These two permutations will always commute with one another. Peggy could also randomly choose half of the elements to permute for a and the other half to permute for r. This process will also satisfy $ra = ar$.

Example 2.2. The following example demonstrates how to create two permutations that commute in S_8.

1. Peggy creates the secret permutation a by manipulating the values for 1-4 and fixing the values for 5-8, for example $a = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 4 & 1 & 3 & 5 & 6 & 7 & 8 \end{pmatrix}$.

2. Peggy creates the random permutation r by manipulating the values for 5-8 and fixing the values for 1-4, for example $r = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 8 & 7 & 6 & 5 \end{pmatrix}$.

3. Consequently, the resulting products ra and ar will equal:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 8 & 7 & 6 & 5 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 4 & 1 & 3 & 5 & 6 & 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 4 & 1 & 3 & 8 & 7 & 6 & 5 \end{pmatrix}$$
$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 4 & 1 & 3 & 5 & 6 & 7 & 8 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 8 & 7 & 6 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 4 & 1 & 3 & 8 & 7 & 6 & 5 \end{pmatrix}.$$

   Peggy will have access to both of these values so she can begin communicating with Victor. We will call this the *splitting method*.

Another way to find commuting elements is by brute force. Although this method is more tedious, it adds a level of security since it can be harder for Eve to find elements that commute with a given permutation. After Peggy finds two elements that commute, the exchange in this ZKP will directly resemble the ZKP from Section 1.2. In the example below, we will work with two elements that were found to commute by brute force.

Example 2.3. The following example demonstrates the general setup and exchange in S_6. In this example, Victor will choose $t = 1$, although Peggy is not aware of this until after she sends Victor her permutation s.

1. Peggy and Victor agree to work in S_6.

2. Peggy creates the secret permutation $a = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 3 & 6 & 2 & 1 & 5 \end{pmatrix}$.

3. Peggy calculates $b = a^2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 3 & 6 & 2 & 1 & 5 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 3 & 6 & 2 & 1 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 6 & 5 & 3 & 4 & 1 \end{pmatrix}$. Peggy will send Victor b when she is ready to communicate with him.

Next, Peggy and Victor participate in the exchange outlined above.

1. Peggy chooses $r = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 1 & 4 & 5 & 3 & 2 \end{pmatrix}$ (confirming that $ra = ar$) and computes

$$s = r^2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 1 & 4 & 5 & 3 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 1 & 4 & 5 & 3 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 6 & 5 & 3 & 4 & 1 \end{pmatrix}.$$

   Peggy sends s to Victor.

2. Victor chooses $t = 1$ and sends his bit to Peggy.

3. Peggy computes $y = ra^1 = ra$:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 1 & 4 & 5 & 3 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 3 & 6 & 2 & 1 & 5 \end{pmatrix}^1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 4 & 2 & 1 & 6 & 3 \end{pmatrix}.$$

   Peggy sends y to Victor.

4. Victor must check to see if $y^2 = sb^1$. If they do equal one another, then Victor will believe that he is talking to Peggy. Victor computes the following:

$$y^2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 4 & 2 & 1 & 6 & 3 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 4 & 2 & 1 & 6 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 1 & 4 & 5 & 3 & 2 \end{pmatrix}$$
$$sb^t = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 6 & 5 & 3 & 4 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 6 & 5 & 3 & 4 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 1 & 4 & 5 & 3 & 2 \end{pmatrix}.$$

   Victor believes Peggy since $y^2 = sb^t$.

The security behind this ZKP in S_n relies on the hardness of finding a root in S_n and finding an element r that commutes with a. However, if Eve is able to find any element r that commutes with a she is able to successfully impersonate Peggy.

Theorem 2.4. *Eve is able to impersonate Peggy if she can always efficiently find any square root of b. She does not need to find the correct square root, only a root that equals b when squared.*

Below is a table showing whether a parameter is public or private. This table will be a helpful reminder as to how we establish an attack working in the ZKP for S_n.

parameter	public	private
a		✓
b	✓	
r		✓
s	✓	
t	✓	
y	✓	
y^2	✓	

From the table, we recall that a and r are the only private elements in this ZKP. We also recall that $b = a^2$ and $s = r^2$. In the general setup discussed in Section 1.2, we recall that it is quite difficult to find a modular square root. However, there is a known method for finding a root if a permutation is a square. From here, if Eve is able to find an element r that commutes with the root a she is able to successfully impersonate Peggy. Eve will always be able to create such an r by using a power of a.

Proof. For Victor to believe he is talking to an honest prover, he must be able to verify that $y^2 = (ra^t)^2$. During the exchange process, Eve will first find a permutation r that commutes with a. From here, Victor will either choose $t = 0$ or $t = 1$. He will always be able to successfully verify that he is talking to an honest prover since $(ra^t)^2 = r^2a^{2t}$ which will always simplify to sb^t. Hence, if Eve is able to find any root of b, when that permutation is squared it will always equal b. We previously showed why a and r must commute with one another. Therefore, if Eve is able to find any root of b and can find an element r that commutes with this root, she will always be able to impersonate Eve. □

Unfortunately, if Eve is able to find a square root and an element r that commutes with this square root, she can impersonate Peggy with a success rate of 100%. We will establish a criteria for finding a root of a square permutation in the next section.

2.2 Square Roots of Permutations

In this section, we will show the process for finding a root of a permutation. We will discuss disjoint cycles and how the cycle structure will lead to a procedure to find a root. We will refer to the length of a permutation with the variable q. We will also discuss even and odd cycles.

Proposition 2.5. *If α and β are disjoint cycles then $\alpha\beta = \beta\alpha$.*

Since a permutation has a unique representation as a product of disjoint cycles, we can use the product of disjoint cycles to determine if a permutation will have a root or not. In this ZKP, we can guarantee that we will always be working with square permutations since b is the result of a^2; however, it is always safe to confirm the criteria.

Theorem 2.6. *A permutation has a square root if and only if the number of disjoint cycles of the same even length in its cycle decomposition is even.*

Proof. This is required because the square of a cycle with even length $2q$ will split the cycle into two even length disjoint cycles of length q. On the other hand, the square of a cycle with odd length will still be a cycle of the same length. It is important to pay special attention to a square of a cycle with even length. After decomposing a permutation into its disjoint cycles, if there is not an even number of disjoint cycles of the same even length, then the permutation cannot be a square. □

Example 2.7. Suppose a permutation in S_8 is decomposed into the following disjoint cycles:

$$\begin{pmatrix}1 & 3 & 7\end{pmatrix}\begin{pmatrix}2 & 4\end{pmatrix}\begin{pmatrix}5 & 8\end{pmatrix}.$$

This permutation has one cycle of length three, two cycles of length two, and one cycle of length one. Therefore, this permutation is a square since it has an even number of cycles of length two.

Thus for a permutation to be a square, it must have an even number of cycles of the same even length; it does not matter the number of odd cycles since every odd cycle will always be a square. In the example below we will not have a square permutation.

Example 2.8. Suppose a permutation in S_{10} is decomposed into the following disjoint cycles:

$$\begin{pmatrix}1 & 4 & 8 & 5\end{pmatrix}\begin{pmatrix}2 & 10 & 3 & 7\end{pmatrix}\begin{pmatrix}6 & 9\end{pmatrix}.$$

This permutation has two cycles of length four, and one cycle of length two. Therefore, this permutation is not a square since it does not have an even number of cycles of length two.

Using the two previous results, we can establish a theorem and a technique for finding a square root for permutations.

Theorem 2.9. *Let σ be a permutation in S_n which has a square root. To find a square root of σ, the first step is to decompose σ into its unique product of disjoint cycles. The next step is to separate the cycles into groups of same length cycles. Finally, the root will be constructed by using the process below for odd and even length cycles.*

1. Decompose the permutation into its product of disjoint cycles.
2. Check to make sure all cycles of any even length occur in an even number. That is, make sure there is an even number of length two cycles, an even number of length four cycles, etc.
3. For the even cycles, we take the even number of cycles of the same even length and group them together in pairs. Suppose there are two cycles c and d of even length q:

$$c = \begin{pmatrix}e_1 & e_2 & e_3 & ... & e_{q-1} & e_q\end{pmatrix}$$
$$d = \begin{pmatrix}f_1 & f_2 & f_3 & ... & f_{q-1} & f_q\end{pmatrix}.$$

   Then we define

$$\sqrt{c \mid d} = \begin{pmatrix}e_1 & f_1 & e_2 & f_2 & ... & e_q & f_q\end{pmatrix}.$$

4. For every odd cycle $c = \begin{pmatrix}e_1 & e_2 & e_3 & ... & e_{q-1} & e_q\end{pmatrix}$ the formula for finding a root is as follows:

$$\sqrt{c} = \begin{pmatrix}e_1 & e_{(q+3)/2} & e_2 & e_{(q+5)/2} & ... & e_q & e_{(q+1)/2}\end{pmatrix}.$$

5. Finally, we have our root by combining all of the resulting cycles together.

It is important to note that the resulting square root will not have the same number of disjoint cycles as the original permutation. This is mainly because the result of squaring a cycle of even length w is two cycles each of length $\frac{1}{2}w$. It is also important to note that if an element goes to itself, e.g. $\sigma(3) = 3$, then the element has a root where 3 goes to itself.

Below we will examine a case where Eve impersonates Peggy and convinces Victor she is an honest prover. Eve is able to accomplish this because she uses the above procedure for finding a square root and then she is able to find a permutation that commutes with her newly discovered square root.

Example 2.10. Peggy and Victor agree upon working in S_8. Peggy is ready to start communicating with Victor so she sends him $b = \begin{pmatrix}1 & 2 & 3 & 4 & 5 & 6 & 7 & 8\\ 7 & 6 & 8 & 3 & 2 & 5 & 4 & 1\end{pmatrix}$. Eve sees this information and decides she would like to impersonate Peggy so she begins communicating with Victor. She must act quickly and find a square root before Peggy continues her communication with Victor. First, she finds a square root for b using the procedure we just described.

1. This permutation decomposes into the following product of disjoint cycles:

$$b = \begin{pmatrix} 1 & 7 & 4 & 3 & 8 \end{pmatrix} \begin{pmatrix} 2 & 6 & 5 \end{pmatrix}.$$

2. Since both of these disjoint cycles are of odd length, Eve recalls the formula for finding a root of an odd length cycle: $\sqrt{c} = \begin{pmatrix} e_1 & e_{(q+3)/2} & e_2 & e_{(q+5)/2} & \dots & e_q & e_{(q+1)/2} \end{pmatrix}$. The first cycle has $q = 5$ and will have the following root:

$$\begin{pmatrix} 1 & 3 & 7 & 8 & 4 \end{pmatrix}.$$

   The second cycle has $q = 3$ and will have this root:

$$\begin{pmatrix} 2 & 5 & 6 \end{pmatrix}.$$

3. Finally, Eve finds the final root by taking the disjoint root cycles and writing them as one product. Here is the root permutation Eve finds when given b:

$$a' = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 5 & 7 & 1 & 6 & 2 & 8 & 4 \end{pmatrix}.$$

4. Using a brute force attack, Eve is able to find a permutation that commutes with a'. She finds $r = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 5 & 1 & 8 & 6 & 2 & 3 & 7 \end{pmatrix}$. She double checks to make sure $a'r = ra'$.

5. Eve squares r and sends Victor s:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 5 & 1 & 8 & 6 & 2 & 3 & 7 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 5 & 1 & 8 & 6 & 2 & 3 & 7 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & 6 & 4 & 7 & 2 & 5 & 1 & 3 \end{pmatrix} = s.$$

6. Victor thinks he is talking to Peggy, and he sends Eve $t = 1$.

7. Eve calculates $y = ra^1$ and sends this value to Victor. She gets

$$y = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 6 & 3 & 4 & 2 & 5 & 7 & 8 \end{pmatrix}$$

   and sends this to Victor.

8. Victor confirms that $y^2 = sb^1$.

$$\text{He calculates } y^2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 5 & 3 & 4 & 6 & 2 & 7 & 8 \end{pmatrix}.$$

$$\text{He calculates } sb = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 5 & 3 & 4 & 6 & 2 & 7 & 8 \end{pmatrix}.$$

   Victor believes he is talking to Peggy since $y^2 = sb$. He will continue communicating with "Peggy" because he is very confident that he is talking to an honest prover.

2.3 Vulnerabilities of Using Symmetric Groups

This zero-knowledge protocol using symmetric groups has some advantages and disadvantages. At the beginning of this section, we discussed the differences of working in $\mathbb{Z}_n$ versus S_n. We know that $\mathbb{Z}_n$ is commutative while S_n is a non-abelian group. The difference between working in commutative versus non-commutative structures may add a layer of security that benefits the non-abelian groups.

We discussed at the beginning of this section the procedure for creating a ZKP with symmetric groups. The main difference between the ZKP in $\mathbb{Z}_n$ and the ZKP in S_n is Peggy is always able to find an r that commutes with a in $\mathbb{Z}_n$ since this ZKP is working in a commutative setting. This is where the layer of added security may be found, since Peggy has to work to find an r that will commute with a in S_n. This is a property that will be found when establishing a ZKP for any non-abelian group. As previously shown, there are two methods that Peggy can use to find a commuting element: splitting a permutation into two halves or by brute force. There are other methods that can be used to find a commuting element, such as raising a permutation to some power. We will briefly discuss this method at the end of this section.

Each of these methods has their own disadvantages. The splitting method is open to an attack that could be used with the square root attack described in the previous section. If Peggy creates her a value by using the splitting method, then the corresponding public value b will also only have half the elements fixed and it will be the same half.

Example 2.11. Suppose Peggy and Victor have agreed to work in S_8. Peggy has decided to create her value for a by manipulating the values for 1-4 and fixing the values for 5-8. Suppose that

$$a = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 1 & 2 & 3 & 5 & 6 & 7 & 8 \end{pmatrix}.$$

The corresponding permutation $b = a^2$ will be:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 1 & 2 & 3 & 5 & 6 & 7 & 8 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 1 & 2 & 3 & 5 & 6 & 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 1 & 2 & 5 & 6 & 7 & 8 \end{pmatrix}.$$

From here, Eve is able to find a square root using the square root attack. Eve is also able to find a commuting permutation r much more easily because the resulting permutation will have fixed values for 1-4 and manipulated values for 5-8. This process as a whole will take less time for Eve to execute because it will take less time to find a square root and the process of finding a commuting permutation r is much simpler.

Example 2.12. Suppose that Peggy sends Victor $b = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 1 & 2 & 5 & 6 & 7 & 8 \end{pmatrix}$. If Eve would like to impersonate Peggy she must find a root for b and a commuting element r for the root a'. To find a root for b, she will decompose b into its product of disjoint cycles and then use the procedure discussed in Section 2.2. Eve finds that $a' = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 1 & 5 & 6 & 7 & 8 \end{pmatrix}$. Eve must now find an r that commutes with a'. This will be easy for her to do because she notices that Peggy used the splitting method to create her a. Eve creates r by manipulating the values for 5-8 and fixing the values for 1-4. She decides on $r = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 8 & 5 & 6 & 7 \end{pmatrix}$. This process did not take Eve very long and now she can successfully impersonate Peggy.

The splitting method is not as safe to use because it will make impersonating Peggy very feasible and not time consuming. The brute force method is safer to use because manipulating all of the values for a permutation makes it harder to find a commuting permutation. On the flip side, sometimes it is too difficult or time consuming to find a commuting permutation this way. If Peggy is not able to find a commuting permutation then she will have to restart the communication with Victor which will make him slightly skeptical.

There is another method to finding a commuting permutation, but this method could also leave Victor doubtful. When Eve finds a square root a an easy way to find a commuting r is to raise a to some power. This can be suspicious to Victor if he realizes Eve is using this procedure.

Example 2.13. Suppose Eve has found $a = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 5 & 7 & 1 & 6 & 2 & 8 & 4 \end{pmatrix}$. From here, if Eve calculates powers a^k she can easily choose a commuting permutation. We show a table of a^2 through a^5 and their permutations below.

Power of a	Resulting permutation
a^2	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 7 & 6 & 8 & 3 & 2 & 5 & 4 & 1 \end{pmatrix}$
a^3	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & 2 & 4 & 7 & 5 & 6 & 1 & 3 \end{pmatrix}$
a^4	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 5 & 1 & 8 & 6 & 2 & 3 & 7 \end{pmatrix}$
a^5	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 6 & 3 & 4 & 2 & 5 & 7 & 8 \end{pmatrix}$

Eve should be careful and choose a power of a where the next power does not have many elements going to themselves. This is because Victor could become distrustful about the validity of such a permutation.

In addition to the disadvantages of the splitting method and brute force, we found an attack for finding a square root of a permutation. As stated previously, if Eve is able to find a square root, not necessarily the root that Peggy used, she will always be able to successfully impersonate Peggy when communicating with Victor. The only problem with this attack is Eve has to quickly find a square root and a commuting permutation and must start her communication with Victor before Peggy does. However, this is a problem for most attacks where time is an issue; for example, with the man in the middle attack.

3 Zero-Knowledge Protocols over Monoids

Throughout the rest of this paper we will focus our attention on creating ZKPs for monoids. We will specifically examine ZKPs for endomorphisms of a finite set and $n \times n$ matrices over finite fields. First we will begin by discussing monoids and some benefits and special characteristics that arise when working with monoids instead of groups.

3.1 Monoids

Based on the evidence presented for the disadvantages of the ZKP working with symmetric groups, we can conclude that this ZKP is not safe to use. We were able to create an attack on this ZKP by targeting the vulnerability of squares in S_n. We can conclude that developing a ZKP for S_n leaves

room for impersonation. To create less vulnerable ZKPs we could shift our attention to structures that are not only non-abelian, but also not groups.

Definition 3.1. A *monoid* is a set M with a binary operation $*$ such that the operation is closed, is associative, and contains an identity element e.

Some known monoids include, but are not limited to: the set $\mathbb{R}$, the set $\mathbb{Z}$, the set $\mathbb{Q}$, and the set $\mathbb{C}$ which can all be formed under addition or multiplication. The set of all $n \times n$ matrices can be formed over a ring with either matrix addition or multiplication. The set of endomorphisms can be formed under composition. Our basic contention is the following:

Claim 3.2. *It is safer to use monoids instead of groups.*

This is because there is no easy way to manipulate the exchange by multiplying an element with its inverse to result in the identity, which would simplify the math. Working with non-invertible elements will also prevent a cheating prover from recovering an original element since there is no known easy method for finding a root of a non-invertible element.

Example 3.3. We have already shown the process for finding a root while working in S_n. We were able to do this because we could decompose the permutation into its product of disjoint cycles. If we are working with non-invertible functions, we would not necessarily be able to do this because we could not decompose it. Suppose that $b = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 5 & 6 & 8 & 3 & 5 & 1 & 4 \end{pmatrix}$. We could attempt to decompose this function, but not every grouping will be a cycle. For instance, with b as above one would try:

$$\begin{pmatrix} 4 & 8 \end{pmatrix} \begin{bmatrix} 7 & 1 & 2 & 5 & 3 & 6 & 5 \end{bmatrix}.$$

The first grouping is a cycle; however, the second grouping is not a cycle since 5 appears twice and the second time 5 appears it will not continue the cycle structure. Therefore, since this is not a product of disjoint cycles, we cannot find a root using the procedure from Section 2.2; however, there could possibly be another method.

Proposition 3.4. *Every group is a monoid.*

A group G is a set that is closed under an operation $*$, in which the operation is associative, contains an identity element, and every element has an inverse. Therefore, a group is a special type of monoid where every element is invertible. It is important to state that since every group is a monoid, a monoid can be thought of as a group but without the inverse axiom.

Example 3.5. The set $\mathbb{Z}_n$ under multiplication is a good example of a monoid. This works because in $\mathbb{Z}_n$ not every element has an inverse, yet the operation is still associative and has an identity. This monoid is the basis of the original ZKP.

Definition 3.6. Let M be a monoid. If for $x \in M$ there is some $y \in M$ so that $x * y = e = y * x$ then we say x is a *unit* and y is its *inverse*.

In **ALL** instances, we can construct a group from a monoid. Using this definition, we can create the simplest group formed from a monoid by selecting only elements that are units in M.

Theorem 3.7. *If M is a monoid, then $M^* = \{x \in M \mid x \text{ is a unit}\}$ is a group under the operation of M.*

Proof. Assume M is a monoid. Let $x, y \in M^*$. Then, x has an inverse x^{-1} and y has an inverse y^{-1}. We claim that $y^{-1}x^{-1}$ is the inverse to xy. We must show the products $(y^{-1}x^{-1})(xy)$ and $(xy)(y^{-1}x^{-1})$ will both result in e. Considering the first product, we have:

$$\begin{aligned}(y^{-1}x^{-1})(xy) &= y^{-1}ey \\ &= y^{-1}y \\ &= e.\end{aligned}$$

The other is proven similarly:

$$\begin{aligned}(xy)(y^{-1}x^{-1}) &= xex^{-1} \\ &= xx^{-1} \\ &= e.\end{aligned}$$

Hence, $xy \in M^*$. Therefore, since the associativity and identity axioms hold from the definition of a monoid and every element x is invertible in M^*, M^* is a group. □

Corollary 3.8. *Every ring R with identity gives a monoid under multiplication. Then R^* is known as the group of units of the ring R.*

3.2 Endomorphisms

First, we will look at a class of monoids known as *endomorphism monoids*.

Definition 3.9. The *endomorphism monoid* E_n is the set of all functions $f : \{1, 2, ..., n\} \to \{1, 2, ..., n\}$ under composition.

When creating the ZKP over endomorphisms we will construct the protocol the exact same way that we established the ZKP over symmetric groups.

ZKP over E_n: the setup.

1. Peggy and Victor agree upon an E_n.
2. Peggy creates a secret element a.
3. Peggy computes b where $b = a^2$. When Peggy is ready to communicate with Victor she will send him the element b corresponding to her a.

ZKP over E_n: the exchange.

1. Peggy chooses a random element r in E_n that commutes with a and computes $s = r^2$. She sends s to Victor.
2. Victor chooses a random bit t where $t \in \{0, 1\}$. Victor then sends his bit to Peggy.
3. Peggy computes $y = ra^t$. She sends y to Victor.
4. Victor confirms that $y^2 = sb^t$. Victor will only accept the response if $y^2 = sb^t$.

Note. Notice here that r and a must commute with one another.

As discussed in Section 2.2 we exhibited an attack on the ZKP over symmetric groups. There is a similar attack for the ZKP over endomorphisms. This attack will also revolve around finding a square root for an endomorphism. The attack we displayed in the previous section worked because everything was invertible. This is not the case here. We did a literature search on this problem for endomorphisms and found nothing that was simple and concise. The only result we found was non-trivial: the paper [2] describes a technique for finding a square root of any endomorphism.

Similar to the attack for the ZKP over symmetric groups, Eve will only be able to execute it if she is able to find a commuting element r. Eve is still able to find a commuting element by raising the element to some power; however, this trick does not necessarily work that well in endomorphisms because the powers will eventually become fixed. If Eve is looking for a non-fixed element she will have a harder time finding a commuting element in a monoid since elements of a monoid are not necessarily invertible.

Example 3.10. Suppose Eve is trying to find a commuting element for $\begin{pmatrix}1&2&3&4&5&6&7&8\\3&4&6&2&3&6&7&2\end{pmatrix}$. She is not able to invert this transformation because this transformation is not one-to-one and is not onto. She is also not able to use arbitrary powers of this transformation, because they will eventually alternate between two transformations, as in:

$$a^2 = \begin{pmatrix}1&2&3&4&5&6&7&8\\6&2&6&4&6&6&7&4\end{pmatrix}$$
$$a^3 = \begin{pmatrix}1&2&3&4&5&6&7&8\\6&4&6&2&6&6&7&2\end{pmatrix}$$
$$a^4 = \begin{pmatrix}1&2&3&4&5&6&7&8\\6&2&6&4&6&6&7&4\end{pmatrix}$$
$$a^5 = \begin{pmatrix}1&2&3&4&5&6&7&8\\6&4&6&2&6&6&7&2\end{pmatrix}.$$

Here we can tell that all odd powers are the same as are all even powers. Eve would not be able to use any of these powers because this could be suspicious to Victor.

3.3 Matrices

The last monoid we will examine is $n \times n$ matrices defined under matrix multiplication. We will focus most of our attention on $M_2(\mathbb{Z}_2)$ which will contain all invertible and non-invertible matrices. This is the smallest possible monoid of matrices to analyze and only contains 16 different matrices. The setup and procedure for the ZKP over matrices is identical to the setup and procedure for endomorphisms. We will establish the setup and exchange for the monoid of all matrices $M_n(\mathbb{Z}_p)$ for any n and prime p.

ZKP over $M_n(\mathbb{Z}_p)$: the setup.

1. Peggy and Victor agree upon a positive integer n and prime p.
2. Peggy creates a secret matrix a.
3. Peggy computes b where $b = a^2$. When Peggy is ready to communicate with Victor she will send him the matrix b corresponding to her a.

ZKP over $M_n(\mathbb{Z}_p)$: the exchange.

1. Peggy chooses a random matrix r in $M_n(\mathbb{Z}_p)$ that commutes with a and computes $s = r^2$. She sends s to Victor.
2. Victor chooses a random bit t where $t \in \{0, 1\}$. Victor then sends his bit to Peggy.
3. Peggy computes $y = ra^t$. She sends y to Victor.
4. Victor confirms that $y^2 = sb^t$. Victor will only accept the response if $y^2 = sb^t$.

Note. Notice here that r and a must commute with one another.

We may potentially exploit this procedure by analyzing the different square roots produced by looking at $M_2(\mathbb{Z}_2)$. We stated earlier that we will look at both the invertible and non-invertible matrices. There are 16 different matrices in $M_2(\mathbb{Z}_2)$; however, there are only 10 different squares. We will list all of the squared matrices below and we will also include the corresponding matrices that are roots.

Number	Squared matrix	Invertible?	Root(s)
1	$\begin{bmatrix}0&0\\0&0\end{bmatrix}$	no	$\begin{bmatrix}0&0\\0&0\end{bmatrix}, \begin{bmatrix}0&0\\1&0\end{bmatrix}, \begin{bmatrix}0&1\\0&0\end{bmatrix}, \begin{bmatrix}1&1\\1&1\end{bmatrix}$
2	$\begin{bmatrix}0&0\\0&1\end{bmatrix}$	no	$\begin{bmatrix}0&0\\0&1\end{bmatrix}$
3	$\begin{bmatrix}0&0\\1&1\end{bmatrix}$	no	$\begin{bmatrix}0&0\\1&1\end{bmatrix}$
4	$\begin{bmatrix}0&1\\0&1\end{bmatrix}$	no	$\begin{bmatrix}0&1\\0&1\end{bmatrix}$
5	$\begin{bmatrix}1&0\\0&1\end{bmatrix}$	yes	$\begin{bmatrix}0&1\\1&0\end{bmatrix}, \begin{bmatrix}1&0\\0&1\end{bmatrix}, \begin{bmatrix}1&0\\1&1\end{bmatrix}, \begin{bmatrix}1&1\\0&1\end{bmatrix}$
6	$\begin{bmatrix}0&1\\1&1\end{bmatrix}$	yes	$\begin{bmatrix}0&1\\1&1\end{bmatrix}$
7	$\begin{bmatrix}1&0\\0&0\end{bmatrix}$	no	$\begin{bmatrix}1&0\\0&0\end{bmatrix}$
8	$\begin{bmatrix}1&0\\1&0\end{bmatrix}$	no	$\begin{bmatrix}1&0\\1&0\end{bmatrix}$
9	$\begin{bmatrix}1&1\\0&0\end{bmatrix}$	no	$\begin{bmatrix}1&1\\0&0\end{bmatrix}$
10	$\begin{bmatrix}1&1\\1&0\end{bmatrix}$	yes	$\begin{bmatrix}1&1\\1&0\end{bmatrix}$

In $M_2(\mathbb{Z}_2)$ there are 10 possible square matrices; three of these matrices are invertible while the other seven are non-invertible. There is a known formula for finding the number of possible invertible matrices for a given p when working in $M_n(\mathbb{Z}_p)$.

Theorem 3.11. *The number of different invertible matrices in $M_n(\mathbb{Z}_p)$ is given by*

$$\prod_{k=1}^{n} p^n - p^{k-1}.$$

As we showed earlier, there are three unique invertible squared matrices and six different invertible root matrices. We could also calculate this using the above formula:

$$\begin{aligned}\prod_{k=1}^{2} p^2 - p^{k-1} &= (p^2-1)(p^2-p)\\ &= (2^2-1)(2^2-2)\\ &= 6.\end{aligned}$$

We have already calculated this by showing all of the possible matrices for $M_2(\mathbb{Z}_2)$. This is helpful when working in $M_n(\mathbb{Z}_p)$; however we are only able to determine the number of different invertible root matrices and not the number of unique squares.

Theorem 3.12. *The number of non-invertible square matrices in $M_n(\mathbb{Z}_p)$ is given by*

$$p^{n^2} - \prod_{k=1}^{n} p^n - p^{k-1}.$$

Note that we want this number to be as big as possible for securing our ZKP over $M_n(\mathbb{Z}_p)$.

Example 3.13. Suppose we were working in $M_2(\mathbb{Z}_5)$. We can calculate the number of invertible square matrices by using our formula:

$$\begin{aligned}\prod_{k=1}^{2} p^2 - p^{k-1} &= (p^2-1)(p^2-p)\\ &= (5^2-1)(5^2-5)\\ &= 480.\end{aligned}$$

We also know that there are 625 possible matrices in $M_2(\mathbb{Z}_5)$. Therefore, we know that there are 145 square matrices that are non-invertible. However, we do not know how many unique squares these matrices produce. Thus, if Eve is given a non-invertible square matrix, she could have a very difficult time finding a root for the given square. For some numerical evidence on the "rarity" of invertible matrices, the following table counts the number of invertible matrices in $M_n(\mathbb{Z}_2)$ relative to the size of the entire monoid.

n	Number of matrices	Number of invertible matrices	Percentage of invertible
2	16	6	37.5%
3	512	168	32.8%
4	65,536	20,160	30.8%
5	33,554,432	9,999,360	29.8%

As shown in the table above, the percentage of invertible elements stays relatively small as n increases. As we have explained before we do not know how many unique squares these invertible elements yield. Therefore, working in $M_n(\mathbb{Z}_2)$ might be a viable option since there may be fewer invertible matrices.

We did a literature search on the problem of finding a square root of an invertible matrix over a finite field. The *only* result we found was the paper [3]. Here the author gives only a count for the number of square roots of the identity matrix: the results do not hold for arbitrary squares, nor does the author give a procedure for producing a square root. Therefore, this appears to be the most secure ZKP we have created and analyzed in this thesis.

It is safe for us to conclude that using monoids instead of groups may potentially add a layer of security. Although the endomorphism monoids E_n we examined had a known method for finding a square root, the solution is non-trivial and its complexity unclear. As we proved, it is much easier for Eve to impersonate Peggy when they agree to work in S_n. We are unsure how hard it would be for Eve to impersonate Peggy when working in an arbitrary monoid, but based on our literature searches it seems likely to be more difficult than when working with symmetric groups.

References

[1] Uriel Feige, Amos Fiat, and Adi Shamir, *Zero-knowledge proofs of identity*, Journal of Cryptology **1** (1988), no. 2, 77–94.

[2] Peter M Higgins, *A method for constructing square roots in finite full transformation semigroups*, Canad. Math. Bull **29** (1986), no. 3, 344–351.

[3] John H. Hodges, *The matrix equation $X^2 - I = 0$ over a finite field*, Amer. Math. Monthly **65** (1958), 518–520.

[4] Austin Mohr, *A Survey of Zero-Knowledge Proofs with Applications to Cryptography*, `http://www.austinmohr.com/Work_files/zkp.pdf`, 2007, accessed 23-April-2018.