A Cryptographic Hash Function from Markoff Triples

Elena Fuchs[1], Kristin Lauter[2], Matthew Litman[1], and Austin Tran[1]

[1]Department of Mathematics, University of California, Davis
[2]Facebook AI Research, Seattle, WA

July 22, 2021

Abstract

Cryptographic hash functions from expander graphs were proposed by Charles, Goren, and Lauter in [**CGL**] based on the hardness of finding paths in the graph. In this paper, we propose a new candidate for a hash function based on the hardness of finding paths in the graph of Markoff triples modulo p. These graphs have been studied extensively in number theory and various other fields, and yet finding paths in the graphs remains difficult. We discuss the hardness of finding paths between points, based on the structure of the Markoff graphs. We investigate several possible avenues for attack and estimate their running time to be greater than $O(p)$. In particular, we analyze a recent groundbreaking proof in [**BGS1**] that such graphs are connected and discuss how this proof gives an algorithm for finding paths.

Keywords: Markoff triples, Cryptographic hash functions
MSC: 11T71, 94A60, 05C48

Contents

1 Introduction **2**
1.1 Markoff tree and graph . . . 2
1.2 Cryptographic hash function . . . 3
1.3 Avenues for attack . . . 4
1.4 Some background on Markoff triples . . . 5

2 Markoff triple hash function and data **6**
2.1 Cryptographic Heuristics . . . 6
2.2 Sampling . . . 9

3 Attack by Pathfinding via the Method of Bourgain-Gamburd-Sarnak **11**
3.1 Rotations . . . 11
3.2 The End Game . . . 16
3.3 The Middle Game and The Opening . . . 20

4 Attack by Lifting **21**

5 Other Possible Attacks and Future Avenues for Research **23**

1 Introduction

In this work, we introduce a proposal for a hash function based on the hardness of finding paths in the graph of Markoff triples, which we will define. The idea of using the hardness of path-finding in graphs to define cryptosystems was introduced at the NIST Hash function workshop in 2005 [**CGL**]. The paper [**CGL**] proposed two different candidate families of Ramanujan graphs: 1) LPS Cayley graphs, and 2) Supersingular Isogeny Graphs. The LPS-based hash function was attacked in two subsequent papers, which presented efficient algorithms to find collisions [**TZ**], and preimages [**PLQ**]. Path-finding in Supersingular Isogeny Graphs remains a hard problem in cryptography so far, and is the basis for the SIDH Key Exchange Protocol [**JFP**, **CFLMP**] in the third round of the NIST PQC competition.

In this paper, we propose to use graphs based on solutions to Markoff's equation to construct a new cryptographic hash function, and discuss why it appears that these graphs may be good candidates. Our main focus will be to evaluate the path-finding algorithm that can be extracted from the proof of Bourgain-Gamburd-Sarnak in [**BGS1**] that these graphs are connected in most cases, as this is currently the only certain way to find paths in these graphs in general. We will hence go into the details of the proof in [**BGS1**] and how it yields an algorithm to find paths, as well as explore some other potential attacks in Section 4 and 5.

1.1 Markoff tree and graph

Consider solutions in $(\mathbb{Z}_{\geq 0})^3 \backslash \{(0,0,0)\}$ to

$$x_1^2 + x_2^2 + x_3^2 - 3x_1x_2x_3 = 0. \tag{1}$$

Equation (1) is known as the *Markoff equation*, and its solutions are called *Markoff triples*, with the integers that occur as members of some triples known as Markoff numbers. As we discuss below, a lot is known about Markoff numbers, and one particularly useful observation, both to us and more generally in the arithmetic study of Markoff numbers, is that one can generate all such triples by considering the orbit of the group generated by the involutions

$$R_1(x_1, x_2, x_3) = (3x_2x_3 - x_1, x_2, x_3)$$

$$R_2(x_1, x_2, x_3) = (x_1, 3x_1x_3 - x_2, x_3)$$

$$R_3(x_1, x_2, x_3) = (x_1, x_2, 3x_1x_2 - x_3)$$

acting on the triple $(1, 1, 1)$ [**M1**],[**M2**]. In this way, one can view the set of Markoff triples as a tree as depicted in Figure 1. Note that the tree depicted in Figure 1 shows one of several very similar branches of the tree, the others are generated by acting on $(1, 1, 1)$ via R_1 and R_2, as well as letting one other involution act on the triple immediately adjacent to $(1, 1, 1)$ (in the figure, that other involution would be R_1 acting on $(1, 1, 2)$). Those branches will simply contain permutations of the triples shown in Figure 1.

Markoff triples first appeared in the literature in Markoff's master's thesis [**M1**], [**M2**] in the context of studying rational approximations via continued fractions. Markoff found that the sequence of Markoff numbers plays a big role in results that produce infinitely many irrationals α which admit a continued fraction convergent p/q such that

$$\left| \alpha - \frac{p}{q} \right| < \frac{m}{\sqrt{d}q^2}$$

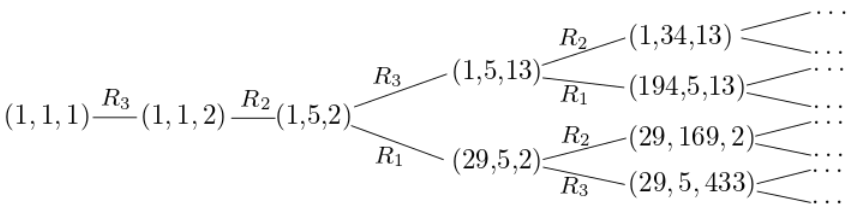

Figure 1: A branch of the Markoff tree generated by applying the involutions R_1, R_2, R_3 to the fundamental solution (1,1,1).

for appropriate values of m and d which are connected to the Markoff sequence. Moreover, he was able to determine exactly for which α the above bound is sharp. Markoff's work on the subject inspired a whole series of generalizations of his result which introduced novel techniques into the theory which are still used today, e.g. [**RS**]. Indeed, today the Markoff equation in (1) is of interest not only to those studying continued fractions but has become an important object in other branches of mathematics such as algebraic geometry and theoretical physics [**Ch**],[**LT**].

For cryptographic applications, we will consider what is, roughly speaking, the mod-p reduction of this tree, as well as a related graph where edges are defined slightly differently.

Specifically, let p be a (large) prime, and consider the set of nonzero solutions modulo p to equation (1). We call a solution (x_1, x_2, x_3) in $(\mathbb{F}_p)^3$ a *triple*, and each entry in the triple x_1, x_2, or x_3, a *coordinate*; so a coordinate is simply an element of $\mathbb{F}_p$.

We consider two graphs: G_p and $\hat{G}_p$. In both of these graphs, the vertices are comprised of nontrivial (we exclude $(0, 0, 0)$) solutions modulo p. In G_p, the edges are defined by the involutions R_1, R_2, R_3: two triples are connected by an edge if one of the three involutions takes one triple to the other. We will also refer to this graph as the involution graph. In $\hat{G}_p$, the edges are defined by *rotations* (see Section 3.1). Explicitly, they are given by

$$\mathrm{rot}_i = \tau_{i+1,i+2} \circ R_{i+1} \tag{2}$$

Here τ is a transposition of coordinates, and all index additions are done modulo 3. Two triples are connected by an edge in $\hat{G}_p$ if one of the 3 rotations takes one triple to the other. We refer to this as the rotation graph.

Our reason for considering G_p is that it is particularly convenient for setting up our hash function. It is the rotation graph $\hat{G}_p$, however, for which Bourgain-Gamburd-Sarnak prove connectivity and in fact give a path-finding algorithm. Notably, finding paths in the graph $\hat{G}_p$ is easily correlated to finding paths in G_p, and vice versa.

Specifically, one can check that, given three different indices $1 \leq i, j, k \leq 3$, and a triple (a, b, c), one has $R_i(a, b, c) = \tau_{j,i} R_j \tau_{k,j} R_k \tau_{i,k} R_i(a, b, c)$, so a path of length ℓ between two triples in G_p corresponds to a path of length 3ℓ in $\hat{G}_p$, whereas a path of length ℓ between two triples in $\hat{G}_p$ corresponds to a path of at length at least $\ell/3$, and possibly much longer, in G_p. In other words, an algorithm to find paths in $\hat{G}_p$ will find paths in the other graph in time not significantly shorter, and possibly much longer, time.

1.2 Cryptographic hash function

First we give a brief summary of the hash function we propose, with more details given in Section 2. We choose a large prime p: typical cryptographic size primes have at least 256 bits. There are some restrictions on the choice of p, see the discussion in the following section. We then label in G_p the edges corresponding to the involutions R_1, R_2, and R_3, respectively. The input to the hash function

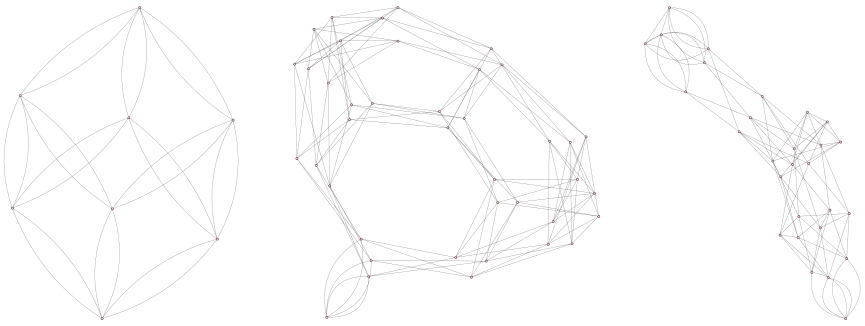

Figure 2: The Markoff mod-p graphs $\hat{G}_p$ for $p = 3$, 5, and 7.

is a bit string $b_0b_1b_2\ldots$. To compute the output of the hash function, start at a fixed vertex, such as the vertex $(1, 1, 1)$, and take a walk in the graph according to the directions in the bit string, reading the string bit by bit, one bit for each step of the walk. The output is the vertex at which this process ends.

The security of this hash function depends on the hardness of finding paths between two vertices in the graph G_p. This paper is concerned primarily with exploring potential avenues for attack and estimating their running times.

1.3 Avenues for attack

In Sections 3 and 4, we describe two potential attacks on our hash function. Both give ways of finding paths between two triples in the graph G_p.

The work of Bourgain, Gamburd, and Sarnak in [**BGS1**] gives a potential attack on the cryptosystem we have proposed, but we will show that its running time is heuristically $O(p)$. Specifically, in order to show that the graphs $\hat{G}_p$ are connected, their algorithm gives a way to construct a path between any two vertices in the Markoff graph mod p for most primes p. The very rough idea is to associate to every vertex a certain "order" that we describe in Section 3.1. Those vertices of maximal possible order are grouped into what Bourgain-Gamburd-Sarnak call *the cage*, which they prove is connected. They then show a way to connect any other vertex to the cage by walking along the graph while increasing the so-called order of the vertex until the cage is reached. We call this the BGS algorithm for finding paths in $\hat{G}_p$.

Not surprisingly, this is easiest to do when the order of the vertex in question is already quite large, and they call the process of passing from such a vertex to the cage the "End Game." Those points whose order is larger than a small power of p but not as large as the vertices involved in the End Game are treated separately in what Bourgain, Gamburd, and Sarnak call the "Middle Game." Finally, to move from vertices of very small order into the Middle Game and beyond, they employ rather technical methods in what they call the "Opening." Because it is central to understanding the potential attack that it gives on our cryptosystem, and because some of the details that are important to us are left to the reader in [**BGS1**], we carefully describe these three parts of Bourgain, Gamburd, and Sarnak's proof of Theorem 5 and then give a heuristic for the running time in Section 2.1. We show the following.

Proposition 1. *The length of the path given by the BGS algorithm is bounded from above by*

$$O(p\eta_p + 3p) = O\left(p\frac{\log p}{\log\log p}\right).$$

See section 2.1 for a definition of η_p, and the derivation. From this, we deduce the time complexity of the algorithm.

Proposition 2. *The time complexity of the BGS algorithm is at most*

$$O\left(p\frac{\log p}{\log\log p}\right).$$

We also provide data in Section 2.1 showing that these upper bounds are in fact of the right order, which supports the heuristics. We observe that the path is longest modulo primes p for which p^2-1 is *very smooth* (see discussion in Section 2.1). Note that, if a faster way of finding paths between vertices in G_p or $\hat{G}_p$ exists, this would be both of interest to the study of our hash function, and to the study of the arithmetic of Markoff numbers: it could provide a second proof of Theorem 5 that works even for those primes that Bourgain, Gamburd, and Sarnak could not handle.

Related to this first attack, we also present a seemingly simple attack which would use the fact that, as one walks without back-tracking along the Markoff tree depicted in Figure 1, the coordinates of the triple increase. This makes it trivial to find paths between vertices in this tree: simply walk from the two vertices in question along edges that decrease coordinates until one gets to $(1,1,1)$. If it were easily possible, given a vertex (x_1,x_2,x_3) in G_p to lift it to a vertex in the infinite Markoff tree which reduces to (x_1,x_2,x_3) modulo p, then one could simply connect the two vertices in question by connecting their lifts in the infinite tree, and then transferring this path back to G_p. However, in Section 4, where we describe this attack more carefully, we explain the obstacles to this approach. In particular, an efficient algorithm to lift would yield another proof of Theorem 5: an algorithm to lift any triple in G_p to one in the infinite tree (which is known to be connected) in fact shows that G_p itself, and, by the previous discussion of how the two graphs are related, $\hat{G}_p$ is connected. We conjecture the following.

Conjecture 3. *The length of a path found by lifting a triple in G_p to a triple in the Markoff tree over $\mathbb{Z}$ is at least $O(p)$ for most triples in G_p.*

Note that this does not take into account the difficulty of actually finding the lift. Hence the running time of this attack is likely comparable to the one based on the BGS algorithm.

1.4 Some background on Markoff triples

We now note a few important facts about the graphs G_p. First of all, it is known that $|G_p| = |\hat{G}_p| = p^2 + \left(\frac{-1}{p}\right)\cdot 3p$ if $p>3$, which is mentioned in [**dCM**] without proof. Here $\left(\frac{*}{*}\right)$ denotes the Legendre symbol. One way to prove this is to think of the left side of the Markoff equation as a quadratic form in one of the variables x_1,x_2,x_3, and then consider how many representations of 0 there are mod p, which is a well known problem. Furthermore, Meiri and Puder have proven the following.

Theorem 4 (Meiri, Puder [**MP**])**.** *Let G_p be as above and let Γ_p be the finite permutation group induced by the action of $\Gamma = \langle R_1, R_2, R_3\rangle$ on G_p. Then, outside a zero-density subset of all primes, the group Γ_p is either the full symmetric group or the alternating group on the vertices of G_p.*

They conjecture that this is in fact true for all primes $p \geq 5$. So we can compare walking along the graph G_p to generating elements of S_n and A_n (where $n = |G_p|$) with a given random generating set, which has been studied, for example in [**BH**].

The graphs G_p and $\hat{G}_p$ are now known to be connected for the majority of primes p. This was proven by Bourgain, Gamburd, and Sarnak in [**BGS1**] as a first step in studying the arithmetic

of Markoff triples (for example, the distribution of primes or numbers with a bounded number of prime factors among Markoff numbers). The structure of these graphs plays an important roll in sieving over Markoff triples, which is key in [**BGS2**]. Specifically, they show the following.

Theorem 5 (Bourgain, Gamburd, Sarnak [**BGS1**])**.** *For all primes $p \notin E$, where E is an exceptional set of primes, the graph $\hat{G}_p$ is connected. The set E is small: for any $\epsilon > 0$, the number of primes $p \leq T$ with $p \in E$ is at most T^ϵ for T large.*

Furthermore, they conjecture not only that G_p is connected for all primes p, but that in fact the family of graphs G_p where p is prime is an expander family. This is explored in [**dCM**], as we discuss in the following section. Theorem 5 is enough to show, as Bourgain, Gamburd, and Sarnak show in [**BGS2**], that the set of Markoff numbers contains infinitely many composite numbers, and in fact that almost all Markoff numbers are composite.

2 Markoff triple hash function and data

Recall that for a sufficiently large prime p, we can construct a hash function as follows. A fixed public initial vertex is specified, say $(1, 1, 1)$. Also choose an involution k; the choice of k is fixed but arbitrary. The edges of G_p are canonically labeled with 1, 2, or 3, corresponding to the three involutions R_1, R_2, R_3 respectively. Given a bit string of finite length as input, say $b_0 b_1 b_2 \ldots$, designate $c_0 = k$. Then, for $i > 0$, suppose c_{i-1} was the label of the previous edge, $c_i \in \{1, 2, 3\}$. Then we move along the edge

$$c_i = (c_{i-1} + b_i) \pmod 3 + 1$$

Note that doing this avoids substrings of the form $R_i R_i$, and so we avoid backtracking. The output of the hash function is the final vertex where the walk ends, after processing all the bits b_i in the string. Note that the initial bit string is not necessarily raw text or data, and will most likely be augmented with some compression function, such as the Merkle-Damgard construction.

For example, suppose we want to encode the binary message `10010001` in G_{13}. We choose $k = 0$ then apply the series of rotations

$$10011 \mapsto R_2 \circ R_3 \circ R_2 \circ R_1 \circ R_2(1, 1, 1) = (0, 5, 1)$$

We know that $|G_p| = O(p^2)$, so for the output space of this hash function to be comparable to say SHA-256, we would want to take $p \approx 2^{128}$. The security of this hash function depends upon the difficulty of path or cycle finding in G_p. That is, given x, $y \in G_p$, what is the time complexity of finding a path between x and y?

Note that if the starting vertex is $(1, 1, 1)$, the input string needs to be longer than $\log p$ so that the coordinates of the output start to wrap around modulo p. Otherwise a trivial lifting attack is possible. A better starting vertex v_0 can be obtained by taking a walk of length $\log(p)$ from $(1, 1, 1)$. In general we will assume that the length of the walk from v_0 is at least length $\log(p)$. In fact, [**BGS2**] has conjectured that the family G_p is an expander family; walks of length $O(\log p)$ are sufficient for mixing in expander graphs.

2.1 Cryptographic Heuristics

The theorems of Bourgain, Gamburd, and Sarnak, which we present in detail in Section 3 prove the correctness of the following path finding algorithm in $\hat{G}_p$ (under certain easy assumptions on p). This path finding algorithm uses a notion of "order" of a triple, coming from a certain rotation assigned to it (see Section 3.1 for the definition). The idea is then that there is a large connected

component of $\hat{G}_p$ consisting of triples of "maximal" order, and to connect any two triples one need only connect each of them to this large component, which Bourgain-Gamburd-Sarnak call the *cage*. One does this by walking along a specially concocted path in which the orders of the triples grow as one walks along it, until one reaches the cage. In other words, the algorithm runs as follows.

Suppose we want to connect two triples X and Y. We can do this in two steps:

1. First, if X or Y are not in the cage, then we want to connect them to the cage.

   Every triple X is part of special cycles in $\hat{G}_p$ which we describe in Section 3, called maximal orbits M_X of X. Bourgain-Gamburd-Sarnak show that the orbit M_X contains at least one point of higher order than X, call it X'. Then X' is connected to X, so replace X with X' and repeat the same argument. The order is guaranteed to increase each step, until eventually the order is maximal.

2. Now we can suppose X and Y are both in the cage. Then by Proposition 10 (Proposition 6 in [**BGS1**]), there exists a point Z in the cage such that $X - Z - Y$ is a valid path.

   In fact, we have an explicit way of finding Z. Since X or Y might have more than one maximal orbit, we search over all maximal orbits of X and Y and look for an intersection, which is guaranteed to exist. In the case that X and Y have the same singular maximal index, then we simply perform an appropriate transposition on either X or Y.

As noted in Proposition 1, we have an upper bound of

$$O(p\eta_p + 3p)$$

on the length of the path obtained using the BGS algorithm. We see this as follows. Each orbit has size $O(p)$, and the number of steps needed is bounded by $\sigma_0(p^2-1)$, where $\sigma_0(n) = \sum_{d|n} 1$ is the number of divisors of n. Call this value η_p. When lifting a triple X in $\hat{G}_p$ to a triple in $\mathbb{Z}$, the size of the coordinates of the lift in $\mathbb{Z}$ can be bound by $O(3^n)$, where n is the length of a path between X and $(1,1,1)$. We conjecture that the length of the shortest possible path should be $2\log p$, so the connecting path given by the BGS algorithm is not helpful in improving the bounds of the lift. It remains an important open question to tighten this bound.

As stated in Proposition 2, the above bound on the orbit size gives that the time complexity of the BGS algorithm is at most

$$O\left(p\frac{\log p}{\log\log p}\right).$$

We compute this bound by multiplying the size of an orbit by the possible number of orbits, using the asymptotic

$$\eta_p \sim \frac{\log(p^2-1)}{\log\log(p^2-1)}$$

which gives

$$O(p\eta_p) = O\left(p\frac{\log p}{\log\log p}\right)$$

This is an upper bound since it might be that X and Y are not in the cage, and in particular have minimal order. If X and Y are both in the cage, then path-finding in the cage has complexity $O(p)$.

We can imagine optimizing this algorithm by being greedier with the first step. Instead of looking at the entire orbit, as soon as we find *any* X' with order higher than X, we replace X with that X'. The algorithm is also guaranteed to work because the order is still guaranteed to increase at each step. If we assume this X' occurs uniformly randomly within the orbit, instead of looking

at p points in an orbit, we only look at $p/2$ points on average. The complexity of this modified algorithm is largely unchanged:

$$O\left(\frac{1}{2}p\frac{\log p}{\log\log p}+p\right)=O\left(p\frac{\log p}{\log\log p}\right)$$

This heuristic is supported by the data in Figure 3. Note that the relative scale for time taken is arbitrary; nevertheless we mention the specifications for reference. Calculations were done in SageMath 9.1, running on a quad-core i7-8550U CPU at 1.8 Ghz.

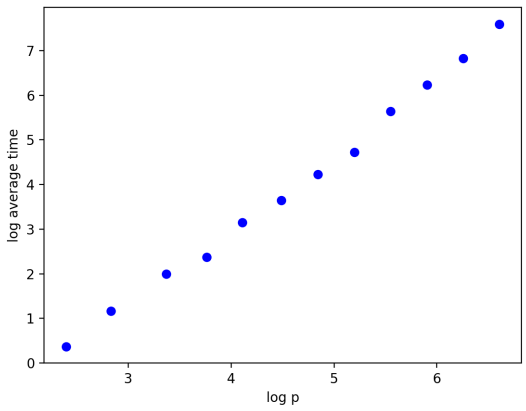

Figure 3: Plot of $\log p$ vs. log average time taken by the BGS algorithm, in seconds. Here we take primes $p \leq 739$. Time taken is in seconds, averaged over 10 trials for each p.

Now the complexity depends merely on the chance of a worst-case scenario, where the triple is not in the cage. This depends on a couple of factors: the proportional size of the cage, and the number of steps it could potentially take to connect any triple to the cage.

It turns out that both of these factors depend in turn on η_p. There is a correlation between η_p and the number of steps needed to connect a triple (not in the cage), as can be seen in Figure 4. Additionally, Figure 5 is supporting evidence that the size of cage also depends on η_p. The asymptotic behavior of this graph, as $\eta_p \to \infty$, is a relevant open question.

We would also like to see if the time taken to connect a point to the cage depends on η_p. A plot of this relationship can be seen in Figure 6, showing a strong correlation between η_p and the time taken to connect a point to the cage.

Concretely, [**BGS1**] only establishes that $\hat{G}_p$ is connected as long as p satisfies the following condition: for any y,

$$\sum_{d|p^2-1, d\in[(\log p)^{1/3}, y]} d^{2/3} < y$$

Therefore p is selected so that p^2-1 is not smooth. Not only does this guarantee connectedness, it also assists with the problem of short cycles. The length of any orbit must divide p^2-1, so avoiding small factors will also avoid small orbits. Fortunately, such primes are difficult to find, and thus easy to avoid. Different search methods have been proposed for finding smooth primes (for example, in the appendices of [**C**] and [**FKLPW**], the authors produce two separate approaches), all of which support the claim that finding such primes is a difficult task.

This is additional evidence that increasing η_p also increases the difficulty of path-finding. Thus we recommend that the security parameter be dependent on both the size of p as well as η_p.

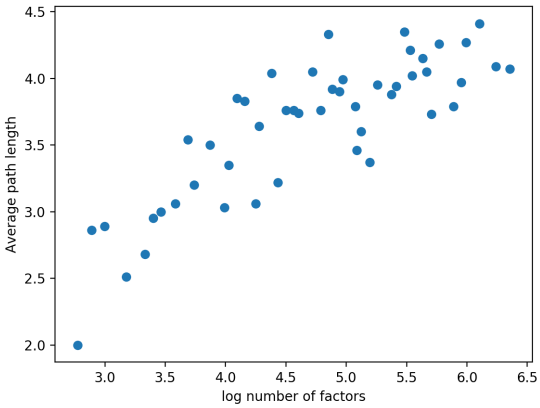

Figure 4: Plot of $\log \eta_p$ vs. average time taken by the BGS algorithm, in seconds. Here we take prime $p < 10000$ and $\eta_p < 6000$. Time taken is in seconds, averaged over 10 trials for each p.

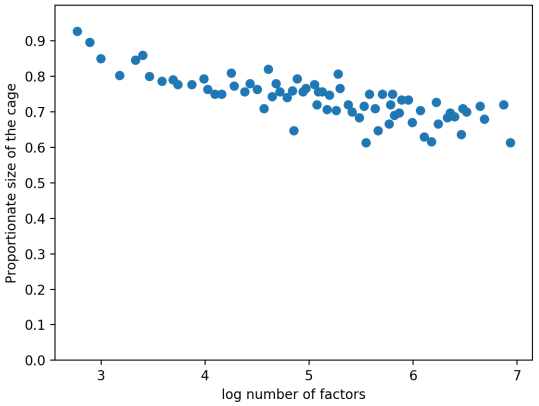

Figure 5: Plot of $\log \eta_p$ vs. proportion of all vertices in the graph $\hat{G}_p$ which are in the cage for primes $p < 100000$ with $\eta_p < 24000$.

2.2 Sampling

Without prior knowledge of the entirety of $\hat{G}_p$, how does one randomly sample a point from $\hat{G}_p$? One way would be the following. Start at a fixed point, say $(1, 1, 1)$ which is in $\hat{G}_p$ for all p. Then perform a random non-backtracking walk starting from $(1, 1, 1)$, of a large length l, the end of which is our sample. Below we see empirically that l does not affect the random distribution for sufficiently large l:

Of course, this method would be truly uniformly random if the family of graphs $\hat{G}_p$ were an expander family. We do have empirical evidence of this, as well as more compelling evidence from a paper of de Courcy-Ireland and Magee [**dCM**]. In particular, they state that G_p "resembles" a random graph, which is a start to examining the spectral gap for the adjacency matrices of the graphs G_p.

Specifically, de Courcy-Ireland and Magee show that the distribution of the eigenvalues of the adjacency matrix of a Markoff graph G_p asymptotically follows the Kesten-McKay law for the

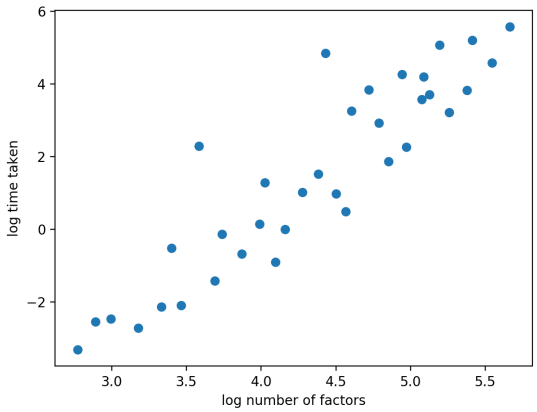

Figure 6: Plot of $\log \eta_p$ vs. time taken by the BGS algorithm to connect a point to the cage, in seconds. Here our primes p are taken such that $p < 2000$ and $\eta_p < 2000$. Time taken is in seconds, averaged over 10 trials for each p.

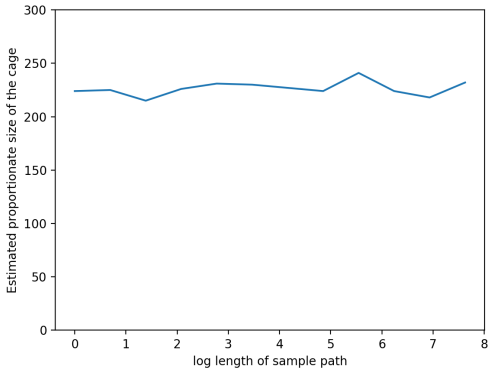

Figure 7: Graph of log length of a sample path vs. a sampled estimate of the size of the cage. Here p is fixed at 5851, and we performed 300 random walks of length l for each l. The data show how many of these 300 samples are in the cage.

distribution of eigenvalues of large randomly chosen 3-regular graphs. In general, the Kesten-McKay law says that, for a large random d-regular graph, the expected eigenvalue probability distribution is

$$\rho_d(\lambda) = \frac{d}{2\pi} \frac{\sqrt{4(d-1) - \lambda^2}}{d^2 - \lambda^2}$$

for $|\lambda| \leq 2\sqrt{d-1}$ and is 0 otherwise.

Let μ_p be the distribution of eigenvalues on G_p, which range from $[-3, 3]$:

$$\mu_p = \frac{1}{|G_p|} \sum \delta_{\lambda_j}$$

De Courcy-Ireland and Magee prove the following.

Theorem 6 ([**dCM**], Theorem 1.1)**.** *Given p, there exists a constant $L \sim \log p$ and a constant C,*

independent of p and L, such that

$$\int x^L d\mu_p = \int x^L \rho_3(x) dx + O(C^L/p)$$

However, as mentioned in [**dCM**], this distribution is not strong enough to show that the family of graphs G_p is an expander family. We would like the spectral gap to be nonzero, i.e. the number of eigenvalues in the interval $[3-\epsilon, 3]$ to be $O(1)$; the work in [**dCM**] only proves that this number is $O(p^2/\log p)$.

For a beautiful graphical comparison of the plot of this distribution to analogous plots of calculated eigenvalues for the Markoff surface mod p for $p = 83$ and 89, see Figure 1.1 of [**dCM**]. While not conclusive, this gives some indication that the family of Markoff graphs forms an expander family.

3 Attack by Pathfinding via the Method of Bourgain-Gamburd-Sarnak

In this section, we go through the key elements of the proof of Theorem 5, which is necessary for the analysis of how fast of a path-finding algorithm this produces in section 2.1.

3.1 Rotations

A key collection of tools in the proof of Theorem 5 are certain rotations that are associated to every triple in $\hat{G}_p$. In this section, we go over crucial results about these rotations.

Denote by τ_{ij} the transposition of the ith and jth coordinates.

Let $C_j(a)$ denote all triples for which the jth coordinate is equal to a.

Given a triple X, define a *rotation* function

$$\mathrm{rot}_{x_1}(X) = \tau_{23} \circ R_2(X) = (x_1, x_3, 3x_1x_3 - x_2)$$

Note that this function is easily extended to x_2 or x_3 by applying the appropriate permutation to $R_j(X)$ for the appropriate j.

Further note that without loss of generality we applied R_2 instead of R_3; we can simulate the latter by again applying the appropriate permutation to X. Thus define $\mathrm{rot}_{x_2}, \mathrm{rot}_{x_3}$ similarly.

Since rot_{x_1} fixes x_1, we can think of rot_{x_1} as a function in (x_2, x_3) on the plane defined by setting the first coordinate to be x_1:

$$\mathrm{rot}_{x_1}\begin{pmatrix} x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 3x_1 \end{pmatrix}\begin{pmatrix} x_2 \\ x_3 \end{pmatrix}$$

So we define the *rotation order* of x as the order of

$$\begin{pmatrix} 0 & 1 \\ -1 & x \end{pmatrix} \in \mathrm{SL}_2(\mathbb{F}_p)$$

Note here we are replacing $3x_1$ with x. For the remainder of this section, x will always denote $3x_1$.

The rotation order of a triple is then defined to be the maximum rotation order of its coordinates.

Iteratively applying one rotation to a triple X eventually returns one to X, and we call the set of all such points the *orbit* of that rotation.

The eigenvalues of the rotation matrix are $\frac{x \pm \sqrt{x^2-4}}{2}$, and so we separate cases depending on whether $\left(\frac{x^2-4}{p}\right) = \pm 1$.

- If $x \equiv \pm 2 \pmod{p}$, we say x is *parabolic*.
- If $\left(\frac{x^2-4}{p}\right) = 1$, then x is *hyperbolic*.
- If $\left(\frac{x^2-4}{p}\right) = -1$, then x is *elliptic*.

A triple is parabolic/hyperbolic/elliptic if its coordinate with maximal rotation order is parabolic/hyperbolic/elliptic. These suggestive names will begin to make more sense if $\hat{G}_p$ is pictured literally as a subset of Euclidean space.

To reiterate an above statement, if we were to say $12 \in \mathbb{F}_{17}$ is hyperbolic, we mean that the coordinate $x_1 = 4$ is hyperbolic.

Lemma 7 (Lemma 3 of [**BGS1**]). *Let x be parabolic, i.e. $x \equiv \pm 2 \pmod{p}$. If $p \equiv 3 \pmod{4}$, then $C_1(x)$ is empty (i.e. x does not appear in any triple in $\hat{G}_p$). If $p \equiv 1 \pmod{4}$, then*

$$C_1(2/3) = \left(\frac{2}{3}, t, t \pm \frac{2i}{3}\right)$$

$$C_1(-2/3) = \left(-\frac{2}{3}, t, -t \pm \frac{2i}{3}\right)$$

where $i^2 \equiv -1 \pmod{p}$ and t is any number $\pmod{p}$. So $C_1(x)$ is a pair of disjoint lines. Furthermore, the action of rot_x is explicitly given by

$$\mathrm{rot}_x\left(\left(\frac{2}{3}, t, t \pm \frac{2i}{3}\right)\right) = \left(\frac{2}{3}, t \pm \frac{2i}{3}, t \pm \frac{4i}{3}\right)$$

$$\mathrm{rot}_x\left(\left(-\frac{2}{3}, t, -t \pm \frac{2i}{3}\right)\right) = \left(-\frac{2}{3}, -t \pm \frac{2i}{3}, -t \mp \frac{4i}{3}\right)$$

So rot_2 fixes each line while rot_{-2} interchanges them.

Proof. Without loss of generality suppose $x_1 = \pm 2/3$. Then equation (1) reduces to

$$x_2^2 + x_3^2 + \frac{4}{9} \mp 2x_2x_3 \equiv 0 \pmod{p}$$

$$(x_2 \mp x_3)^2 \equiv -\frac{4}{9} \pmod{p}$$

So a solution exists if and only if $\left(\frac{-1}{p}\right) = 1$, which is equivalent to $p \equiv 1 \pmod{4}$.

Set $p \equiv 1 \pmod{4}$ and suppose $C_1(2/3) = (2/3, t, t + a)$. Then we have

$$t^2 + (t+a)^2 + \frac{4}{9} - 2t(t+a) \equiv 0 \pmod{p}$$

which reduces to $a^2 \equiv -4/9 \pmod{p}$ independent of t, which gives the desired result.

Similarly suppose $C_1(-2/3) = (-2/3, t, -t + a)$, which gives

$$t^2 + (-t+a)^2 + \frac{4}{9} + 2t(-t+a) \equiv 0 \pmod{p}$$

which again reduces to $a^2 \equiv -4/9 \pmod{p}$ which again gives the desired result.

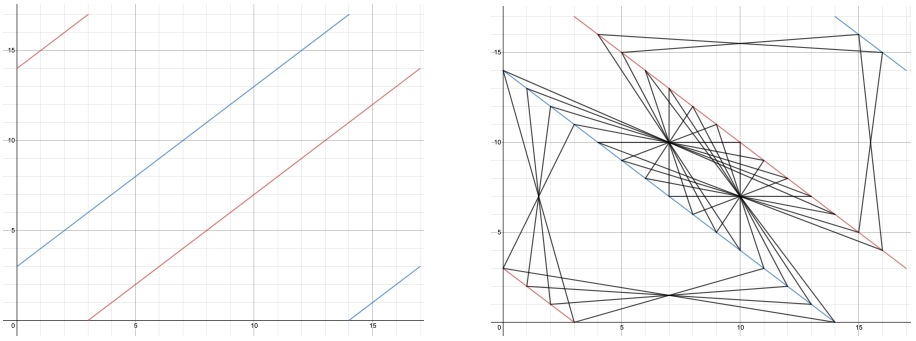

Figure 8: (a) Two lines in $C_1(12)$ fixed by rot_{12}. (b) Two lines in $C_1(5)$ interchanged by rot_5.

Now we can explicitly calculate

$$\mathrm{rot}_2\left(\left(\frac{2}{3}, t, t \pm \frac{2i}{3}\right)\right) = \left(\frac{2}{3}, t \pm \frac{2i}{3}, 3\frac{2}{3}\left(t \pm \frac{2i}{3}\right) - t\right) = \left(\frac{2}{3}, t \pm \frac{2i}{3}, t \pm \frac{4i}{3}\right)$$

$$\begin{aligned}\mathrm{rot}_{-2}\left(\left(-\frac{2}{3}, t, -t \pm \frac{2i}{3}\right)\right) &= \left(-\frac{2}{3}, -t \pm \frac{2i}{3}, -3\frac{2}{3}\left(-t \pm \frac{2i}{3}\right) - t\right) \\ &= \left(-\frac{2}{3}, -t \pm \frac{2i}{3}, -t \mp \frac{4i}{3}\right)\end{aligned}$$

as desired. □

Here is an example in G_{17}, where $17 \equiv 1 \pmod 4$, $2/3 \pmod p = 12$, and $i \equiv 4 \pmod{17}$.

Lemma 8. *If x is not parabolic, then we can write*

$$x = \chi + \chi^{-1}$$

where $\chi \in \mathbb{F}_p$ if x is hyperbolic, and $\chi \in \mathbb{F}_{p^2}$ if x is elliptic.

Proof. Suppose x is hyperbolic, i.e. $\left(\frac{x^2-4}{p}\right) = 1$. Then suppose $x^2 - 4 = r^2$. Set $\chi = (x + r)/2$, $\chi^{-1} = (x - r)/2$, and verify

$$\frac{x+r}{2}\frac{x-r}{2} = \frac{x^2 - r^2}{4} = 1$$

Then we have

$$r^2 = (\chi - \chi^{-1})^2 = \chi^2 + \chi^{-2} - 2 = (\chi + \chi^{-1})^2 - 4$$

as desired. Now similarly suppose x is elliptic. Then $x^2 - 4$ is not a residue in $\mathbb{F}_p$, but it is a residue in

$$\mathbb{F}_{p^2} \simeq \frac{\mathbb{F}_p[y]}{y^2 - (x^2 - 4)}$$

Set $x^2 - 4 = r^2$ where $r \in \mathbb{F}_{p^2}$ and repeat the above argument. □

Upon diagonalizing the rotation matrix rot_x, one arrives at

$$\begin{aligned}\mathrm{rot}_x &= \begin{pmatrix} 1 & 1 \\ \chi & \chi^{-1} \end{pmatrix} \begin{pmatrix} \chi & 0 \\ 0 & \chi^{-1} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ \chi & \chi^{-1} \end{pmatrix}^{-1} \\ &= (\chi^{-1} - \chi)^{-1} \begin{pmatrix} 1 & 1 \\ \chi & \chi^{-1} \end{pmatrix} \begin{pmatrix} \chi & 0 \\ 0 & \chi^{-1} \end{pmatrix} \begin{pmatrix} \chi^{-1} & -1 \\ -\chi & 1 \end{pmatrix}\end{aligned}$$

Thus

$$(\mathrm{rot}_x)^l = (\chi^{-1}-\chi)^{-1}\begin{pmatrix}1 & 1\\ \chi & \chi^{-1}\end{pmatrix}\begin{pmatrix}\chi^l & 0\\ 0 & \chi^{-l}\end{pmatrix}\begin{pmatrix}\chi^{-1} & -1\\ -\chi & 1\end{pmatrix}$$
$$= (\chi^{-1}-\chi)^{-1}\begin{pmatrix}\chi^{l-1}-\chi^{1-l} & -\chi^l+\chi^{-l}\\ \chi^l-\chi^{-l} & -\chi^{l+1}+\chi^{-l-1}\end{pmatrix}$$

If we consider $\chi^l = t$, where $t \in \langle\chi\rangle$, then we have

$$\langle \mathrm{rot}_x\rangle = \left\{(\chi^{-1}-\chi)^{-1}\begin{pmatrix}\chi^{-1}t-\chi t^{-1} & t^{-1}-t\\ t-t^{-1} & \chi^{-1}t^{-1}-\chi t\end{pmatrix} : t\in\langle\chi\rangle\right\}$$

Thus

$$C_1(x) = \left\{(\chi^{-1}-\chi)^{-1}\begin{pmatrix}\chi^{-1}t-\chi t^{-1} & t^{-1}-t\\ t-t^{-1} & \chi^{-1}t^{-1}-\chi t\end{pmatrix}\begin{pmatrix}x_2\\ x_3\end{pmatrix} : t\in\langle\chi\rangle\right\}$$
$$= (\chi-\chi^{-1})^{-1}\Big(t(x_3-\chi^{-1}x_2)+t^{-1}(\chi x_2-x_3), \tag{3}$$
$$t(\chi x_3-x_2)+t^{-1}(x_2-\chi^{-1}x_3)\Big) \tag{4}$$

again for $t \in \langle\chi\rangle$. Now we can rewrite the second coordinate as $at + bt^{-1}$ where

$$a = \frac{x_3-\chi^{-1}x_2}{\chi-\chi^{-1}}, \quad b = \frac{\chi x_2 - x_3}{\chi-\chi^{-1}}$$

Later we will need the fact that

$$ab = \frac{x_2x_3(\chi+\chi^{-1})-x_2^2-x_3^2}{(\chi-\chi^{-1})^2} = \frac{x^2}{(\chi-\chi^{-1})^2} = \left(\frac{\chi+\chi^{-1}}{\chi-\chi^{-1}}\right)^2 \neq 1 \tag{5}$$

Now we consider the cases of x hyperbolic or elliptic separately.

- For x hyperbolic: From equation (4), note that $a, b \in \mathbb{F}_p^*$, so substitute $t \mapsto ta^{-1}$ to see that

$$C_1(x) = \left\{\left(t+\frac{ab}{t}, \chi t+\frac{ab}{\chi t}\right) : t\in\mathbb{F}_p^*\right\}$$

Applying the rotation gives

$$\mathrm{rot}_x\left(t+\frac{ab}{t}, \chi t+\frac{ab}{\chi t}\right) = \left(\chi t+\frac{ab}{\chi t}, \chi^2 t+\frac{ab}{\chi^2 t}\right) \tag{6}$$

Since $t \in \mathbb{F}_p^*$, we see that $|C_1(x)| = p-1$. On the other hand, since x is hyperbolic, by Lemma 8, we can write

$$x = \rho^j + \rho^{-j}$$

where ρ is a primitive root of $\mathbb{F}_p$. Then if we iteratively apply rot_x, we cycle through $\frac{p-1}{j}$ elements in $C_1(x)$, i.e. the rotation order of rot_x is $\frac{p-1}{j}$ for some j.

An explicit example of this can be seen below in Figure 9 for the case of G_{17}:

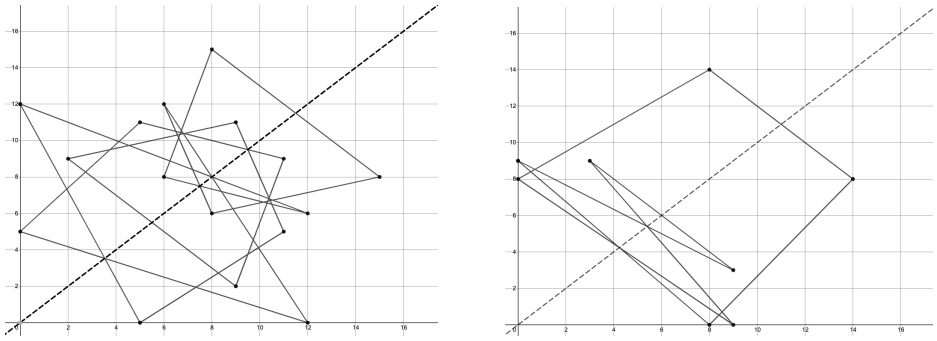

Figure 9: (a) A maximal order hyperbolic rotation rot_3 shown in the plane $C_1(3)$. (b) A hyperbolic rotation rot_2 of order $8 = \frac{p-1}{2}$ in $C_1(2)$. The line $x_2 = x_3$ is shown for symmetry.

- For x elliptic, the derivation is similar. We start by rewriting

$$x = \chi + \chi^{-1} = \nu + \nu^p$$

for $\nu \in \mathbb{F}_{p^2} - \mathbb{F}_p$. Then applying the rotation gives

$$\text{rot}_x\left(x, t, \frac{\kappa_x}{t}\right) = \left(x, t + \frac{\kappa_x}{t}, t\nu + \frac{\kappa_x}{t\nu}\right) \tag{7}$$

which implies

$$C_1(x) = \left\{\left(t + \frac{ab}{t}, \nu t + \frac{ab}{\nu t}\right) : t \in \mathbb{F}_{p^2}^*, \quad t^{p+1} = ab\right\}$$

where the latter requirement implies $t \in \mathbb{F}_{p^2} \backslash \mathbb{F}_p$, which in turn implies $|C_1(x)| = p + 1$. On the other hand, since x is elliptic, by Lemma 8, we can write

$$x = \xi^j + \xi^{-j}$$

where ξ is some element of $\mathbb{F}_{p^2}$. Explicitly, if γ is a generator of $(\mathbb{F}_{p^2})^\times$, then $\xi = \gamma^{p+1}$. So if we iteratively apply rot_x, we cycle through $\frac{p+1}{j}$ elements in $C_1(x)$, i.e. the rotation order of x_1 is $\frac{p+1}{j}$ for some j.

An explicit example of rotations for elliptic and hyperbolic elements in G_{17} can be seen in Figure 10.

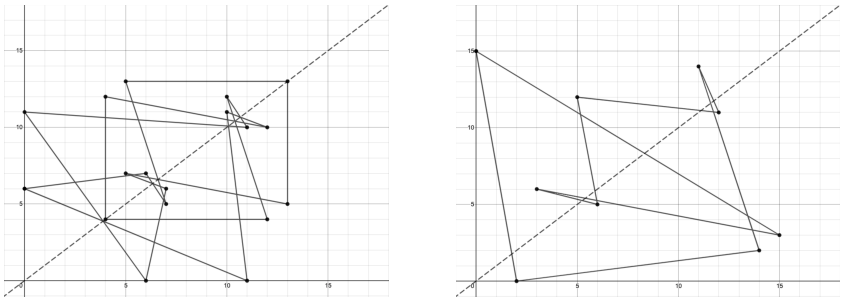

Figure 10: (a) A maximal order elliptic rotation rot_7 shown in the plane $C_1(7)$. (b) A hyperbolic rotation rot_8 of order $9 = \frac{p+1}{2}$ in $C_1(9)$.

From the above discussion, we have that x has a maximal rotation order of $p-1$ if x is hyperbolic, $p+1$ if it is elliptic, or $p, 2p$ if it is parabolic. If any of these cases applies to x, we say x is maximal hyperbolic/elliptic/parabolic respectively. A triple is maximal (hyperbolic/elliptic/parabolic) if one if its coordinates is with respect to its corresponding type (again remembering that $x = 3x_1$).

Note that the rotation order of a parabolic x is either p or $2p$. We consider both of these elements to be maximal parabolic. If a triple X contains either element, we can connect X to a triple containing any coordinate.

3.2 The End Game

We are now ready to delve into the BGS algorithm. In this section we aim to show that any element of order $p^{1/2+\delta}$ for $\delta > 0$ can be connected to a triple with a coordinate that is maximal with respect to its type. Later on we will show that every element in $\hat{G}_p$ can be connected to a triple of maximal order, and that such maximal triples themselves can be connected, implying the connectedness of $\hat{G}_p$.

Proposition 9 (Proposition 7 of [**BGS1**]). *Let X be a triple with rotation order at least $p^{1/2+\delta}$ for $\delta > 0$ fixed. Then X is connected to a maximal triple Y.*

As defined above, the triple X can be classified as hyperbolic, elliptic, or parabolic.

The parabolic case is trivial. As we discussed above, we can connect a parabolic triple to an arbitrary coordinate.

Let us first suppose the element is hyperbolic. Then applying (6) to X gives elements of the form

$$(x_1, \alpha_1 t + \alpha_2 t^{-1}, \alpha_3 t + \alpha_4 t^{-1})$$

Here $\alpha_i \in \mathbb{F}_p^*$ and $t \in H$, where H is some cyclic subgroup of $\mathbb{F}_p^*$. If we want to connect X to a maximal triple by iteratively applying (6), we would like the second coordinate to eventually take the form $\rho + \rho^{-1}$, where ρ is a primitive root of F_p^*. The latter is exactly the form of a maximal hyperbolic element.

So let $P(H)$ denote the number of solutions to

$$\alpha_1 t + \alpha_2 t^{-1} = \rho + \rho^{-1} \tag{8}$$

where ρ is a primitive root of $\mathbb{F}_p^*$.

On the other hand, let K be an arbitrary subgroup of $\mathbb{F}_p^*$. Now define $P(H, K)$ to be the number of solutions to (8) where we require $\rho \in K$ instead of ρ being a primitive root.

The subgroups H and K are determined by their indices in $\mathbb{F}_p^*$; set $d_K = (p-1)/|K|$ and $d_H = (p-1)/|H|$.

Now suppose (t, y) is a solution to

$$\alpha_1 t^{d_H} + \alpha_2 t^{-d_H} = y^{d_K} + y^{-d_K} \tag{9}$$

Then the map $(t, y) \mapsto (t^{d_H}, y^{d_K})$ sends solutions of (9) to solutions of (8); this map is $d_H d_K$ to 1.

Thus if $N(\alpha_1, \alpha_2)$ is the number of solutions to (9), then

$$P(H, K) = \frac{N(\alpha_1, \alpha_2)}{d_H d_K}$$

As shown by Lemma 8 of [**BGS1**], the curve

$$\alpha_1 t^{d_H} + \alpha_2 t^{-d_H} - y^{d_K} - y^{-d_K}$$

given by (9) is absolutely irreducible with genus $O(d_H d_K)$. Thus applying the Hasse-Weil bound for irreducible curves gives

$$N(\alpha_1, \alpha_2) = p + O(d_H d_K \sqrt{p})$$

which in turn gives

$$P(H, K) = \frac{p}{d_H d_K} + O(\sqrt{p}) \tag{10}$$

We now want to express $P(H)$ in terms of $P(H, K)$. We use inclusion/exclusion on K to eventually find all primitive roots. Let p_i be the distinct prime factors of $p-1$. Also let K_d be the subgroup of $\mathbb{F}_p^*$ of index d, e.g. $K_1 = \mathbb{F}_p^*$ and $K_{p-1} = \{1\}$. Then we have:

$$\begin{aligned} P(H) &= P(H, K_1) - \sum_i P(H, K_{p_i}) + \sum_{i,j} P(H, K_{p_i p_j}) - \cdots \\ &= \sum_{d|p-1} \mu(d) P(H, K_d) \end{aligned} \tag{11}$$

Plugging (10) into (11) gives

$$\begin{aligned} P(H) &= \sum_{d|p-1} \mu(d) \left(\frac{|H|}{d} + O(\sqrt{p}) \right) \\ &= \left(|H| \sum_{d|p-1} \frac{\mu(d)}{d} \right) + O(p^{1/2+\epsilon}) \\ &= \left(|H| \frac{\phi(p-1)}{p-1} \right) + O(p^{1/2+\epsilon}) \\ &\geq |H|(p-1)^{-\epsilon} + O(p^{1/2+\epsilon}) \end{aligned}$$

We assumed our initial triple X had order $\geq p^{1/2+\delta}$, i.e. $|H| \geq p^{1/2+\delta}$. Thus $P(H) > 1$ and so there exists at least one solution to equation (8). This implies that the orbit of this rotation contains a maximal triple and the hyperbolic case is handled.

The elliptic case is covered in detail in Section 3 of [**BGS1**]. However, as the technical details of their argument are not needed for the paper at hand, we omit them and move on to showing the collection of maximal elements is connected.

3.2.1 Connectedness of the Cage

The vertices in $\hat{G}_p$ corresponding to triples of maximal order form a connected component [**BGS1**]. Consider $C_j(\alpha) \cap C_k(\beta)$ with $j \neq k$, and without loss of generality let $j = 1, k = 2$. Also suppose $\alpha, \beta \neq 0, \pm 2/3$. Going forward we sometimes denote $C_1(\alpha)$ as $(\alpha, ?, ?)$ and $C_1(\alpha) \cap C_2(\beta)$ as $(\alpha, \beta, ?)$.

Then

$$|C_1(\alpha) \cap C_2(\beta)| = |(\alpha, \beta, ?)| = 0, 1, 2$$

In particular, the intersection consists of all γ such that $\alpha^2 + \beta^2 + \gamma^2 - 3\alpha\beta\gamma = 0$, which has a solution in γ if

$$\left(\frac{9\alpha^2\beta^2 - 4(\alpha^2 + \beta^2)}{p} \right) \geq 0$$

In particular

$$|(\alpha, \beta, ?)| = 1 + \left(\frac{9\alpha^2\beta^2 - 4(\alpha^2 + \beta^2)}{p} \right) \geq 0$$

So consider the *incidence graph* $I(p)$ of $\hat{G}_p$. The vertices of $I(p)$ are $C_j(\alpha)$ and the number of edges between $C_j(\alpha)$ and $C_k(\alpha)$ is $|(\alpha, \beta, ?)|$.

Proposition 10 (Proposition 6 of [**BGS1**])**.** *For $p > 10$, the incidence graph is connected and in fact has diameter 2.*

Proof. We want to connect $C_1(\alpha)$ and $C_2(\beta)$. Thus we want to find γ such that both $(\alpha, ?, \gamma)$ and $(?, \beta, \gamma)$ are nonempty. So suppose there is a point (α, l, γ); solve the quadratic in the second coordinate to see that we must have

$$9\alpha^2\gamma^2 - 4\alpha^2 - 4\gamma^2 = \lambda^2$$

for some λ. Similarly we must have that

$$9\beta^2\gamma^2 - 4\beta^2 - 4\gamma^2 = \mu^2$$

for some μ. Rearrange the two equations into the system

$$\begin{cases} (9\alpha^2 - 4)\gamma^2 - \lambda^2 = 4\alpha^2 \\ (9\beta^2 - 4)\gamma^2 - \mu^2 = 4\beta^2 \end{cases} \tag{12}$$

If $\alpha^2 = \beta^2$, then we just take $\lambda = \mu$, and we can reduce (12) to one equation and find an explicit value for γ. Otherwise (12) is an irreducible curve for which we know a solution in γ exists for $p > 10$. So the diameter of the incidence graph is at most 2. But of course $C_1(\alpha)$ is not connected to $C_1(\beta)$ if $\alpha \neq \beta$. Thus the diameter is precisely 2. □

Define the *cage* to be the subset of maximal triples. We claim the cage is connected, i.e. path-connected.

Suppose X is a maximal triple with maximal coordinate α, say $X = (\alpha, ?, ?)$. Suppose Y is a maximal triple with maximal coordinate β, say $Y = (?, \beta, ?)$. By Proposition 10, we know there exists a γ such that both $(\alpha, ?, \gamma)$ and $(?, \beta, \gamma)$ are nonempty. However, we need γ to have maximal order:

$$(\alpha, ?, ?) \xrightarrow{\alpha \text{ maximal}} (\alpha, ?, \gamma) \xrightarrow{\gamma \text{ maximal}} (?, \beta, \gamma) \xrightarrow{\beta \text{ maximal}} (?, \beta, ?)$$

The paper of Bourgain, Gamburd, and Sarnak [**BGS1**] finishes the proof to guarantee the existence of such a maximal γ. Thus the cage is connected. We now illuminate this approach through a concrete example.

3.2.2 Constructive Example

Let's now walk through a simple example to show how vertices are connected using the BGS algorithm. Take $p = 17$. We have the elements along with their order and type in the following table.

Element	Order	Type
0	4	parabolic
1	18	elliptic
2	8	hyperbolic
3	16	hyperbolic
4	16	hyperbolic
5	34	parabolic
6	6	elliptic
7	18	elliptic
8	9	elliptic
9	18	elliptic
10	9	elliptic
11	3	elliptic
12	17	parabolic
13	16	hyperbolic
14	16	hyperbolic
15	8	hyperbolic
16	9	elliptic

Consider the triple $X = (15, 0, 8) \in G_{17}$. This triple is not maximal, but it does have order $> p^{1/2+\delta}$. By Lemma 9, we should be able to connect X to the cage through rotations of its maximal element.

Since the coordinate 8 has the highest order, we consider rot_8 applied to X:

$$(15, 0, 8) \mapsto (0, 2, 8) \mapsto (2, 14, 8) \mapsto (14, 11, 8) \mapsto (11, 12, 8) \mapsto (12, 5, 8) \mapsto$$

$$(5, 6, 8) \mapsto (6, 3, 8) \mapsto (3, 15, 8) \mapsto (15, 0, 8)$$

which is just a shuffle of the coordinates

$$15 - 0 - 2 - 14 - 11 - 12 - 5 - 6 - 3 - 15$$

for which 14 and 3 are maximal hyperbolic, and 12 and 5 are maximal parabolic. Thus we can connect X to the cage in a number of ways.

A visual representation of this rotation within the plane $C_3(8)$ is given in Figure 11.

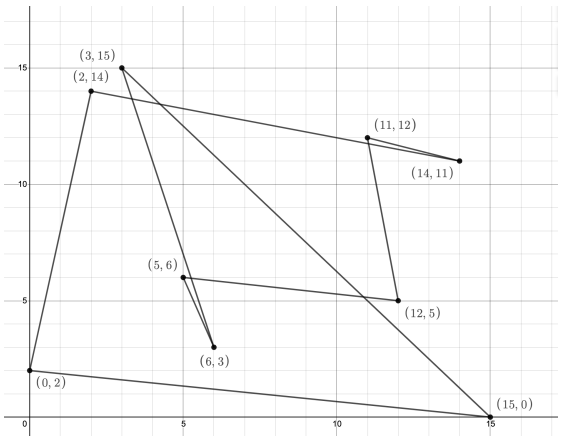

Figure 11: rot_8 applied to $(15, 0, 8)$ in the plane $C_3(8)$.

3.3 The Middle Game and The Opening

In this section we aim to show that any triple X of small order can be connected to the cage in a finite number of moves. By small order, we mean triples X whose order is p^{ε} (which we refer to as *The Middle Game*) or those whose order is less than $p < c$ for some constant c, i.e. points whose orders are uniformly bounded independent of p (which we refer to as *The Opening*).

We first handle the Middle Game in detail, and then outline how the Opening comes into play. In particular, we connect a triple of order p^{ε} to the cage by showing that one can connect it to a triple whose order is strictly greater than that of the original triple, and then iterate the process until we have a triple of order $p^{1/2+\delta}$ and we are in the End Game. This is done via the following procedure:

Define the *maximal orbit* M_X of a triple X as the orbit corresponding to the rotation of the maximal coordinate of X. So if X is a triple with order l, then $|M_X| = l$. Any orbit occurs with respect to either the first, second, or third coordinate; we call this number the index of the orbit.

1. Let $Y \in M_X$ and l_Y be the order of Y. Of course $l, l_Y \mid p^2 - 1$.

2. If $l_Y > l$, then necessarily the index of M_Y is not equal to the index of M_X. Then replace X with Y, thereby strictly increasing the order of X.

3. Otherwise, $l_Y \leq l$. Consider the sum

$$N_l = \sum_{l' \leq l} \#\{Y \in M_X : l_Y = l'\} \tag{13}$$

   If $N_l < l$, then there must be a point $Z \in M_X$ whose order is strictly greater than that of X. We can then replace X with Z and repeat this process until we arrive at an element with order at least $p^{1/2+\delta}$, which is a reduction to the endgame. This must happen because the order strictly increases at each step and $p^2 - 1$ has finitely many divisors.

Now we need to effectively bound N_l. As seen in the endgame, every Y (with order $l_Y \mid p^2 - 1$) in the maximal orbit of X corresponds to a solution of the equation:

$$\begin{cases} h_1 + \frac{\sigma}{h_1} = h_2 + \frac{1}{h_2} \\ h_1 \in H_1, h_2 \in H_2 \\ H_1, H_2 \text{ subgroups of } \mathbb{F}_p^* \text{ or } \mathbb{F}_{p^2}^* \\ \sigma \in \mathbb{F}_p \end{cases} \tag{14}$$

where $|H_1| = l$ and $|H_2| = l_Y$. Also, from equation (5), we have that $\sigma \neq 1$. So we see that in fact N_l denotes precisely the number of solutions to (14), so equivalently we want an upper bound on the number of solutions to (14). The following bound is derived in [**BGS1**], based off previous work of Bourgain (Proposition 2 in [**B10**]).

Proposition 11 (Proposition 10 of [**BGS1**])**.** *Given $\delta > 0$ there is $\tau < 1$ and C_τ depending on δ such that if $p^{\delta} < |H_1| < p^{1-\delta}$ then the number of solutions to (14) is at most $C_\tau |H_1|^{\tau}$.*

From this proposition, we simply deduce

$$N_l \leq C_\tau |H_1|^{\tau} = C_\tau l^{\tau}$$

which provides a necessary upper bound to the number of solutions of (14), as desired.

Thus any triple of order at least p^ϵ can be connected to the cage, and so all triples of order at least p^ϵ are connected. This algorithm is essential for our cryptographic constructions, and provides the backbone to the first step in connecting two triples X and Y as discussed in Section 2.1.

Next we consider the part of the BGS algorithm that is called "the Opening," in [**BGS1**]: that is, the rest of the points in $\hat{G}_p$ whose order is less than $p < c$ for some constant c, that is points whose orders are uniformly bounded.

In the Opening section of [**BGS1**], Bourgain, Gamburd, and Sarnak prove that one can connect triples with uniformly bounded orders to the cage to conclude that the Markoff graph mod p is connected; however their methods are non-constructive. To go about this, they look at the characteristic 0 case and show that there are no finite Γ-orbits. As this method is not needed in our cryptographic analysis of $\hat{G}_p$, we omit the technicalities and direct the interested reader to Section 5 of [**BGS1**] for a comprehensive analysis of the Opening.

The proof of Theorem 5 presented throughout Section 3 provides us with an algorithmic approach to finding paths in $\hat{G}_p$, thus establishing connectivity of $\hat{G}_p$. This method need not be optimal but the cryptographic analysis in Section 2.1 elucidates the strength of the cryptosystem against the BGS-style attack. We now look at another possible avenue of path-finding based off lifting solutions to $\mathbb{Z}$ and exploiting the structure of the Markoff tree.

4 Attack by Lifting

The main observation behind our plan of attack is the following lemma.

Lemma 12. *Let (x_1, x_2, x_3) be a Markoff triple in $\mathbb{Z}^3$ whose i-th coordinate x_i is maximal, and $x_i > 1$. Then applying R_i to the triple decreases the size of the i-th entry. Formally, suppose $|x_i| \geq |x_k|$ for all $1 \leq k \leq 3$ in the Markoff triple (x_1, x_2, x_3). Let $(x_{1,i}, x_{2,i}, x_{3,i})$ be the triple obtained from applying the i-th involution R_i to the triple:*

$$(x_{1,i}, x_{2,i}, x_{3,i}) := R_i(x_1, x_2, x_3).$$

Then $|x_{i,i}| < |x_i|$.

Proof. Let x_j, x_k be the other two coordinates of the triple (x_1, x_2, x_3) besides x_i. Note that, since $|x_i| > 1$, it is impossible for (x_i, x_j, x_k) to satisfy (1) if $|x_i| = |x_j| = |x_k|$. In fact, in this case we have that $|x_i|$ must be strictly larger than $|x_j|$ and $|x_k|$ in order for (1) to be true. Suppose further without loss of generality that $|x_j| \leq |x_k|$.

We have $x_{i,i} = 3x_jx_k - x_i$. If $x_i > 0$, then $x_jx_k > 0$ in order for (1) to be satisfied, and, again by (1) we have

$$3x_jx_k = (x_1^2 + x_2^2 + x_3^2)/x_i > x_i,$$

so that

$$|x_{i,i}| = |3x_jx_k - x_i| = 3x_jx_k - x_i.$$

Our goal is hence to show that $2x_i - 3x_jx_k > 0$. We have by (1) that

$$2x_i - 3x_jx_k = 2x_i - \frac{x_i^2 + x_j^2 + x_k^2}{x_i} = \frac{x_i^2 - x_j^2 - x_k^2}{x_i},$$

which, given that $x_i > 0$, is positive if and only if the numerator is positive. Rewrite the numerator as

$$x_i^2 + x_j^2 + x_k^2 - (2x_j^2 + 2x_k^2)$$

and compare with the left side of (1). We claim that $2x_j^2 + 2x_k^2 < 3x_ix_jx_k$, which would imply that the numerator above is positive as desired.

It remains to prove our claim. Given that $x_i > |x_j|$ and $|x_k| \geq |x_j|$, we have

$$3x_ix_jx_k > x_ix_jx_j + 2|x_k|x_jx_k \geq 2x_j^2 + 2x_k^2$$

as desired where the last inequality is true since $x_i \geq 2$ and $|x_j| \geq 1$. So, if $x_i > 0$ we are done.

If $x_i < 0$ the argument is nearly identical. We would have that $x_{i,i} < 0$ in that case, and so our goal would be to show that $-3x_jx_k + x_i < -x_i$, or that $2x_i - 3x_jx_k < 0$. Given that $(-x_i, -x_j, x_k)$ is a triple satisfying the properties in the first case above where $x_i > 0$, the argument above shows that $-2x_i + 3x_jx_k > 0$, which is exactly what we need. □

This lemma gives a very straightforward way of finding a path from any triple (x_1, x_2, x_3) in the tree to the triple that is the "origin," or $(1, 1, 1)$ in absolute value, which in turn gives a simple way of finding a path between any two vertices in the tree. Thus if triples can be efficiently lifted from the graph G_p to the tree, this algorithm gives a path-finding attack on the graph. The algorithm is as follows. Start with $W = I$, the identity.

1. If $(|x_1|, |x_2|, |x_3|) = (1, 1, 1)$ then we are done, and W is the word that describes the path from (x_1, x_2, x_3) to the origin. If not, determine i such that the i-th coordinate of $(|x_1|, |x_2|, |x_3|)$ is largest. Go to step 2.

2. Replace (x_1, x_2, x_3) with $R_i((x_1, x_2, x_3))$, replace W with WR_i, and go to step 1.

By the lemma, this algorithm will continuously decrease every largest coordinate in absolute value until each coordinate is 1 in absolute value. For example, for the triple $(29, -169, -14701)$ it gives

$$(29, -169, -14701) \xrightarrow{R_3} (29, -169, -2) \xrightarrow{R_2} (29, -5, -2) \xrightarrow{R_1} (1, -5, -2)$$

$$\xrightarrow{R_2} (1, -1, -2) \xrightarrow{R_3} (1, -1, -1).$$

Coming back to our problem of finding paths between two points in the graph G_p,, if our attacker is able to take a triple (x_1', x_2', x_3') which satisfies the Markoff equation modulo p and lift it to a solution (x_1, x_2, x_3) to the Markoff equation in $\mathbb{Z}$, then she need only run the algorithm above to find a path from (x_1, x_2, x_3) to the origin in which every coordinate is 1 in absolute value in order to find a path from (x_1', x_2', x_3') to the origin in G_p (it is the path corresponding to the same word as the one she will obtain from the above algorithm).

However, so far it appears that finding a Markoff triple that reduces to (x_1', x_2', x_3') modulo p is difficult for most candidate (x_1', x_2', x_3')'s. The reason for this is that, according to [**Z**], the number of Markoff triples in which the largest coordinate is at most T is asymptotic to $C(\log T)^2$ for some constant C, while the number of vertices in $G_p \sim p^2$. So in order to have a chance of covering all possible mod-p Markoff triples coming from G_p by Markoff triples over $\mathbb{Z}$, one must consider all those triples less than T where

$$C(\log T)^2 \geq p^2,$$

or, in other words, where T is of size roughly e^p. More likely, T will have to be much larger than that, since it is not at all true that all Markoff numbers less than T reduce to a different triple modulo p. Even with this estimate of e^p, however, one sees that the lifts will probably be very large (since p itself will be taken to be large), and certainly no straightforward search for a lift in $\mathbb{Z}$ will be computationally feasible.

Furthermore, even if one finds a lift, given that it will likely be of size e^p or larger, the path down to $(1,1,1)$ will be roughly of length p, as stated in Conjecture 3. This is similar to what one gets in the attack based on the proof of Bourgain-Gamburd-Sarnak that $\hat{G}_p$ is connected in [**BGS1**].

Constructing a collision attack from lifting is almost equivalent to path finding. If one has a method of efficiently finding lifts to $\mathbb{Z}$, two lifts of the same triple could result in two distinct paths between triples. Unless the two paths in $\mathbb{Z}$ overlap nontrivially, we would have a collision starting with $(0,0,0)$.

5 Other Possible Attacks and Future Avenues for Research

We note that the BGS algorithm can be slightly modified to search for collision resistance. Currently, the steps of the middle game are deterministic in connecting a triple to the cage; the rotations are always done on the maximal coordinate. For a collision attack, we would search for two distinct paths between points. So instead of always choosing the maximal coordinate, we can randomly choose coordinates instead (not necessarily uniformly). If we eventually arrive at the cage, then we have found another distinct path, since the cage is connected. Of course, there is no proof, other than empiricism, that any method other than choosing the maximal coordinate will succeed in a similar way.

Many potential attacks involve finding small cycles on G_p or $\hat{G}_p$, e.g. some adaptation of the Pollard rho algorithm. There are a number of reasons we believe such a study is unfruitful. A Pollard-style attempt would look for cycles by repeatedly applying a single involution. The construction of G_p means that such short cycles occur with vanishingly little frequency, as discussed in the Opening. In any case, such discrete logarithm attacks must involve at least $\Omega(\sqrt{p})$ group operations [**S**], which is not a significant improvement.

Nonetheless, it will certainly be important to understand better the distribution of cycle lengths in a graph G_p or $\hat{G}_p$. While it is known that small cycles in $\hat{G}_p$ exist, it is not known how common they are, and how likely one is to run into one in practice. Even less is known about the cycles in the graph G_p. This is a problem the authors hope to explore in a future paper.

In addition, it would be helpful to have a better picture of the size of an average lift of a Markoff triple mod p to one over $\mathbb{Z}$, so that we can further understand the potential for success of the lifting attack described in Section 4. This is currently being studied by the first-named author together with co-authors E. Bellah, S. Kim, D. Schindler, J. Sivaraman, and L. Ye.

References

[BH] Laszlo Babai, Thomas P. Hayes. The probability of generating the symmetric group when one of the generators is random. Publ. Math. Debrecen, **69** No. 3 (2006), pp. 271-280.

[B] Jean Bourgain. A modular Szemeredi-Trotter theorem for hyperbolas. arXiv:1208.4008.

[BGS1] Jean Bourgain, Alexander Gamburd, Peter Sarnak. Markoff Surfaces and Strong Approximation: 1. arXiv:1607.01530.

[BGS2] Jean Bourgain, Alexander Gamburd, Peter Sarnak. Markoff Surfaces and Strong Approximation. arXiv:1505.06411.

[CGL] Denis X. Charles, Eyal Z. Goren, and Kristin E. Lauter. Cryptographic hash functions from expander graphs. J. Cryptology, Vol. 22 (1), (2009) 93–113. eprint.iacr.org/2006/021

[Ch] William Chen, Nonabelian level structures, Nielsen equivalence, and Markoff triples, preprint (2021) https://static1.squarespace.com/static/59b0d0048419c2e19a207ba7/t/608608553348ca47ee4941e3/1619396694639/congruence.pdf.

[CFLMP] Anamaria Costache, Brooke Feigon, Kristin Lauter, Maike Massierer and Anna Puskas. Ramanujan graphs in cryptography. In: Research Directions in Number Theory: Women in Numbers IV, Association for Women in Mathematics Series, Vol. 19, pp. 1–40 (2019) Springer.

[C] Craig Costello. B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion, In: International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology – ASIACRYPT 2020 (2020), pp. 440-463.

[dCM] Matthew de Courcy-Ireland, Michael Magee. Kesten-McKay law for the Markoff surface mod p. arXiv:1811.00113.

[FKLPW] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: compact post-quantum signatures from quaternions and isogenies, In: International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology – ASIACRYPT 2020 (2020), pp. 64–93.

[JFP] David Jao, Luca De Feo, and Jérôme Plût, Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies, J. Math. Cryptol. 8 (2014), no. 3, pp. 209–247.

[LT] Oleg Lisovyy and Yuriy Tykhyy, Algebraic solutions of the sixth Painlevé equation, Journal of Geometry and Physics, Volume 85 (2014), pp. 124–163,

[M1] Andrey Markoff. Sur les formes quadratiques binaires indéfinies, Math. Ann. 15 (1879) 381–409.

[M2] Andrey Markoff. Sur les formes quadratiques binaires indéfinies, Math. Ann. 17 (1880) 379–399.

[MP] Chen Meiri, Doron Puder with an Appendix by Dan Carmon. The Markoff Group of Transformations in Prime and Composite Moduli. Duke Math J. **167** No. 14 (2018) pp. 2679–2720.

[PLQ] Christophe Petit, Kristin Lauter, and Jean-Jacques Quisquater. Full cryptanalysis of LPS and Morgenstern hash functions, Security and Cryptography for Networks 2008, pp. 263–277, Springer Berlin Heidelberg.

[RS] Michelle Rabideau and Ralf Schiffler. Continued fractions and orderings on the Markov Numbers. arXiv:1801.07155v2.

[S] Victor Shoup. Lower Bounds for Discrete Logarithms and Related Problems. EUROCRYPT 1997. Lecture Notes in Computer Science, vol 1233. Springer.

[TZ] Jean-Pierre Tillich and Gilles Zémor. Collisions for the LPS Expander Graph Hash Function, Advances in Cryptology - EUROCRYPT 2008, Lecture Notes in Computer Science, Vol 4965, pp. 254–269, Springer.

[Z] Don Zagier. Markoff numbers below a given bound, Mathematics of computation 39 No. 160 (1982) 709-723.