Alternative Generators of the Zémor-Tillich Hash Function: A Quest for Freedom in Projective Linear Groups

Hayley Tomkins

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies in partial fulfillment of the requirements for the degree of
Master of Science in Mathematics[1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1]The M.Sc. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Abstract

Introduced in 1994, the celebrated Zémor-Tillich hash function over $\mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$ is mathematically elegant, efficient, and now finally broken (Grassl et. al, 2011). Yet, with a new choice of generators Tillich and Zémor's construction over $\mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$ still remains of interest; looking for generators in $\mathrm{GL}_2\big(\mathbb{F}_{p^n}\big)$ seems almost untouched. Here, we present a large class of generators to choose from, using a novel theorem that outlines conditions under which a set of matrices in $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ generates a free group, and whose proof is an interesting application of Tits' "Ping-Pong Lemma" (1972). The hash functions we form from this theorem are secure against known attacks, and simultaneously preserve many of the desired features of the Zémor-Tillich hash function. In particular, our hash functions retain the *small modifications property* that Zémor-Tillich was known for: for some δ, alterations of less than δ bits will necessarily change the hash value.

Dedications

To my guys, the green-eyed and the green.

Acknowledgements

I offer my utmost gratitude to my supervisors, Dr. Monica Nevins and Dr. Hadi Salmasian, for their exceptional mentorship and immense enthusiasm. Thank you as well to my examiners Dr. Inna Bumagin and Dr. Mike Newman for their careful reading and motivating questions, and to my family and friends for their support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Hash functions are an essential part of many cryptographic schemes, principally as a method of message authentication and modification detection. In [52] and [53] Gilles Zémor introduced the idea of building cryptographic hash functions from Cayley graphs of large girths. In these constructions, notions such as collision, second preimage, and preimage resistance were all easily restated as mathematical problems that were believed to be hard. Zémor's hash functions also had the unusual and valuable property that any small modification of a message would necessarily change its hash value. Though Zémor's initial choice of group and generators from which to construct such a hash function was soon broken [48], the construction formed the base for many subsequent hash functions, including our hash function of interest, the Zémor-Tillich hash function.

From its introduction in 1994, the Zémor-Tillich hash function [49] was well-received for many reasons. Remarkably, the Zémor-Tillich hash function was comparable in computation speed to current cryptographic standards. In a world where computational speed is usually sacrificed for having security based on some mathematical

problem, or vice versa, it is rare to satisfy both these properties. Further, the Zémor-Tillich hash function retained the property that made Zémor's original construction so appealing: small modifications of messages are detected. Moreover, it had a real-life application: in [42] Quisquator and Joye showed the Zémor-Tillich hash function would be ideal for authenticating video sequences. There were many other reasons Tillich and Zémor's construction was so appealing, for example, it could be computed in parallel and was scalable, meaning we could control the size of the output. As message lengths increased, the distribution of all possible hash values was shown to approach the uniform distribution. We present Tillich and Zémor's construction here (Definition 3.1.1).

Let $r_n(x)$ be an irreducible polynomial of degree n over $\mathbb{F}_2$, $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/\langle r_n(x)\rangle$, and $G = \mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$. Further, set

$$A := \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix}, \qquad B := \begin{pmatrix} x & x+1 \\ 1 & 1 \end{pmatrix}$$

where x is a root of r_n. Then, given $m = m_1 \dots m_k \in \{0,1\}^*$ we define the Zémor-Tillich hash function to be $H(m) := H(m_1) \cdots H(m_k)$ where $H(m_i) = A$ if $m_i = 0$ and $H(m_i) = B$ if $m_i = 1$.

Many initial attacks on the Zémor-Tillich hash function were proposed; however, each was specific to the choice of polynomial used to define the finite field $\mathbb{F}_{2^n}$, and consequently were easily avoidable. The first attack defined independently of this choice was by Geiselmann [16], but was unrealistic in practice as the collisions it produced contained long strings of ones and zeros. It was not until 2011 that a feasible attack was found on Tillich and Zémor's construction [18]. We will see in Chapter 3 that this attack was very specific to both the characteristic of the underlying finite field, as well as Tillich and Zémor's choice of generators. This left an open question: what

choices of generating matrices and underlying finite fields would produce, based on Tillich and Zémor's initial construction, a hash function which retained the strengths of the Zémor-Tillich hash function but was more robust to attacks?

While some alternative generators for the Zémor-Tillich hash function have been suggested, few have been presented with thorough analysis, and almost all suggestions are in $\mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$. Expanding the search for alternative generators to characteristic p, an odd prime, or the larger group $\mathrm{GL}_2\big(\mathbb{F}\big)$, where $\mathbb{F}$ is a finite field, appears to be almost entirely new and is the aim of this thesis. For what follows, let p be a prime and $\mathbb{F}_q = \mathbb{F}_p[x]/\langle r_n(x)\rangle$, where $r_n(x)$ is an irreducible polynomial over $\mathbb{F}_p$.

The crux of this thesis is Theorem 4.4.2, the Free Generators Theorem, which presents a general method for forming free generators of free subgroups of $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ and $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$, where $\mathbb{F}_p((x))$ is the field of formal Laurent series over $\mathbb{F}_p$. For A, B with polynomial entries, and generators of a free group in $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$, we take the generators of a hash function to be the images of A and B when we project their entries to $\mathbb{F}_q$ under the quotient by $\langle r_n(x)\rangle$.

For polynomial choices of A and B, this construction allows us to preserve the small modifications property. Namely, since A and B are free generators, any two distinct words in $\{A, B\}^*$ must have distinct products M, $\tilde{M}$ in $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$. If the entries of M and $\tilde{M}$ are each polynomials of degree less than n, the images of M and $\tilde{M}$ in $\mathbb{F}_q$ must remain distinct.

The idea of the Free Generators Theorem is to construct two cyclic matrix groups $\langle A\rangle$ and $\langle B\rangle$ acting on a space X, together with associated disjoint sets P_A and P_B in X, such that any element of $\langle A\rangle$ maps everything outside of P_A into P_A and that any element of $\langle B\rangle$ maps everything outside of P_B into P_B. Thus, any nontrivial word w in $\{A, B\}$ must necessarily map a point outside of $P_A \cup P_B$ to either P_A or P_B, and

thus cannot be identity. This argument is a well-known result of Tits, and is often referred to as the "Pong-Pong Lemma" [51].

In [5] Breuillard and Gelander considered Tits' result for projective linear groups over local fields acting on the associated projective space by equipping this space with a distance. In this setting, Breuillard and Gelander show that Tits' conditions can be satisfied by finding group elements which map everything far enough from a specified "repulsing point" to within a certain distance of a specified "attracting point". This idea is the inspiration for our work, in which we consider $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ as acting on the one dimensional projective space over $\mathbb{F}_p((x))$, $\mathbb{P}^1$, and discuss this projective space in some detail. However, while Breuillard and Gelander used this argument to find dense subgroups of real Lie groups, our application is much different. In particular, our detailed consideration of these ideas in terms of $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$, in addition to applying these ideas to explicitly construct an infinite class of pairs of matrices in $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ that generate a free group, are both new.

This thesis is organized as follows. In Chapter 2, we provide the necessary mathematical background for this thesis as well as the motivation for Tillich and Zémor's initial construction. In Chapter 3 we formally present and look closely at the Zémor-Tillich hash function, in particular the attributes that to us creates so much potential for application. As well, in Chapter 3 we explore known attacks on the Zémor-Tillich hash function, to which we would ultimately like our constructions to be resistant.

In Chapter 4 we first define a metric over $\mathbb{P}^1$ and present numerous properties of this metric which will aid both in our proof and hopefully the reader's understanding of the metric itself. We then present and prove the Free Generators Theorem, which details explicit and easily satisfiable conditions for which two matrices in $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ will generate a free subgroup. In Chapter 5 we present explicit choices of polynomial generators using the results of Chapter 4, and explore useful properties and security

of the associated hash functions to these choices. In addition, we discuss further directions we hope to take these explorations and possible extensions of the Free Generators Theorem. In particular, we show that the proof of the Free Generators Theorem also applies to choices of more than two generators.

Some additional results are included in the appendices. In Appendix A, we will consider some computational results pertaining to these explicit generator choices. In Appendix B, we consider another approach to finding alternative generators.

The original contributions in this thesis include the second half of Chapter 4 and all of Chapter 5, including the main theorem and suggestions of new generators.

Chapter 2

Hash Functions and Mathematical Background

This chapter is comprised of three self-contained sections, each providing some necessary background before we formally present the Zémor-Tillich hash function. In the first section, we define hash functions and some related terms. In the second section, we present some algebraic notions such as general and special linear groups over a ring, their projective counterparts, and finite fields. The last section is devoted to providing some of the history of the Zémor-Tillich hash function that motivated its construction.

2.1 Hash functions

Given a set, often referred to in this setting as an **alphabet**, we define a concatenation of elements in that set as a **word** or a **string**. The number of elements concatenated is referred to as the length of the word. The notation $\{0,1\}^k$ for some $k \in \mathbb{N}_0$ means

the set of all words in $\{0,1\}$ of length k, while $\{0,1\}^*$ is the set of all words of finite length. When we are working over $\{0,1\}$, a string is commonly referred to as a **binary string**, or a **bitstring**.

A **hash function** is a function $h : \{0,1\}^* \to \{0,1\}^N$ for some $N \geq 1$ such that $h(m)$ is easy to compute [27]. The resulting string $h(m)$ is often referred to as the **hash value** of the bitstring m. The desired properties of hash functions range greatly depending on how they are being used. Hash functions are usually required at minimum to be **collision resistant**, meaning that it is hard to find two distinct messages, m and m' such that $h(m) = h(m')$. Possessing **preimage resistance** means that given an output g, it is infeasible to find an element $m \in \{0,1\}^*$ such that $h(m) = g$. Another notion is **second preimage resistance**, meaning given an m, it is difficult to find a different m' such that $h(m) = h(m')$. Note that collision resistance implies second preimage resistance; if it is infeasible to find any two messages with the same hash value, it is infeasible to do so with one of these messages fixed. For this reason collision resistance is often referred to as **strong collision resistance**, while second preimage resistance is often referred to as **weak collision resistance** [27]. Common applications of hash functions include message authentication, to ensure that any alteration to a document after signing can be detected, and storing passwords.

2.2 Mathematical background

The Zémor-Tillich hash function we will discuss, and its extensions, are defined in matrix groups over finite fields. Here, we will present some notation and mathematical background that will be of use.

Definition 2.2.1. *Let R be a commutative ring. We define the general linear group*

of degree k over R, $\mathrm{GL}_k(R)$, to be

$$\mathrm{GL}_k(R) := \{g \in \mathrm{M}_{k\times k}(R) \mid \det(g) \in R^\times\}.$$

That is, $\mathrm{GL}_k(R)$ is the group of invertible $k \times k$ matrices over R. Similarly, we define the special linear group of degree k over R, $\mathrm{SL}_k(R)$, to be the subgroup of matrices with $\det = 1$, that is

$$\mathrm{SL}_k(R) := \{g \in \mathrm{M}_{k\times k}(R) \mid \det(g) = 1\}.$$

We will also consider the projective general linear group, which we define as follows.

Definition 2.2.2. *Let R be a commutative ring. The projective general linear group of degree k over R is*

$$\mathrm{PGL}_k(R) := \mathrm{GL}_k(R)/\mathrm{Z}$$

where Z is the centre of $\mathrm{GL}_k(R)$, and consists of all scalar matrices. Elements of $\mathrm{PGL}_k(R)$ are thus cosets $g\mathrm{Z}$ with $g \in \mathrm{GL}_k(R)$. Similarly, we define

$$\mathrm{PSL}_k(R) := \mathrm{SL}_k(R)/\tilde{\mathrm{Z}}$$

where $\tilde{\mathrm{Z}}$ is the centre of $\mathrm{SL}_k(R)$, that is $\tilde{\mathrm{Z}} = \{rI \ : \ r^k = 1\}$.

Since $\tilde{\mathrm{Z}} = \mathrm{Z} \cap \mathrm{SL}_k(R)$, $\mathrm{PSL}_k(R)$ can be identified as a subgroup of $\mathrm{PGL}_k(R)$. The relations between the groups defined in Definiton 2.2.1 and Defintion 2.2.2 are shown in Figure 2.1.

The following lemma highlights the distinction between $\mathrm{PGL}_k(R)$ and $\mathrm{PSL}_k(R)$ in the case $k = 2$, and will be useful later.

Lemma 2.2.3. *Let $g\mathrm{Z}$ be an element of $\mathrm{PGL}_2(R)$. Then $g\tilde{\mathrm{Z}}$ is the image of an*

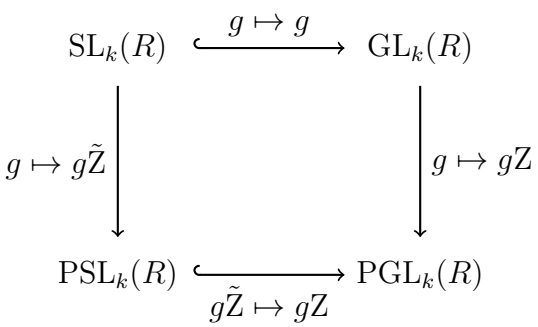

Figure 2.1: The relations between the the (projective) linear groups over a ring R.

element $\tilde{g}\tilde{\mathrm{Z}}$ *in* $\mathrm{PSL}_2\big(R\big)$ *if and only if* $\det(g)$ *is in the set* $\{x^2 \,:\, x \in R^\times\}$.

Proof: We first note that if $g\tilde{\mathrm{Z}} \in \mathrm{PSL}_2\big(R\big)$ then $g \in \mathrm{SL}_2\big(R\big)$ so $\det(g) = 1$, which is a square. On the other hand, if we know $\det(g) = h^2$ for some $h \in R^\times$, we can write

$$g\mathrm{Z} = \frac{1}{h} g \begin{bmatrix} h & 0 \\ 0 & h \end{bmatrix} \mathrm{Z} = \frac{1}{h} g\mathrm{Z}.$$

We see $\tilde{g} = \frac{1}{h}g$ has determinant 1, so $\frac{1}{h}g\tilde{\mathrm{Z}} \in \mathrm{PSL}_2\big(R\big)$ and $\tilde{g}Z = gZ$. □

Let p be a prime and $\mathbb{F}_p$ be the finite field with p elements. Further, let $r_n(x)$ be an irreducible polynomial over $\mathbb{F}_p$ of degree n, and $\langle r_n(x)\rangle$ be the ideal generated by $r_n(x)$ inside $\mathbb{F}_p[x]$. We will denote by $\mathbb{F}_q$ the finite field with $q = p^n$ elements which can be identified with $\mathbb{F}_q \cong \mathbb{F}_p[x]/\langle r_n(x)\rangle$. For $g \in \mathbb{F}_q$, we denote the order of g by $\mathrm{ord}(g)$. In Chapter 4, we will also introduce and discuss $\mathbb{F}_p((x))$, the field of formal Laurent series over $\mathbb{F}_p$. We also introduce the following notation.

Definition 2.2.4. *Let* $M \in \mathrm{M}_{n\times n}\big(R\big)$. *We then define* $\deg(M)$ *to be the maximum degree of the entries of* M.

In what follows, we often mainly consider linear groups over $R = \mathbb{F}_q$ or $R = \mathbb{F}_p((x))$ with $k = 2$. In the case $R = \mathbb{F}_q$ and $k = 2$, we have $Z = \big\{rI \,:\, r \in \mathbb{F}_q^\times\big\}$. As well, in

the case p is odd we see that $\tilde{\mathrm{Z}} = \{I, -I\}$, while if $p = 2$ we have $\tilde{\mathrm{Z}} = \{I\}$.

Further, we have that $|\mathrm{GL}_2(\mathbb{F}_q)| = (q^2 - q)(q^2 - 1)$ and $|\mathrm{SL}_2(\mathbb{F}_q)| = q(q^2 - 1)$. Since $|Z| = |\mathbb{F}_q^\times| = q - 1$, we have $|\mathrm{PGL}_2(\mathbb{F}_q)| = q(q^2 - 1)$. As well, considering $|\tilde{\mathrm{Z}}|$, we see that $|\mathrm{PSL}_2(\mathbb{F}_q)| = |\mathrm{PGL}_2(\mathbb{F}_q)|$ if $p = 2$ and $|\mathrm{PSL}_2(\mathbb{F}_q)| = \frac{1}{2}|\mathrm{PGL}_2(\mathbb{F}_q)|$ if p is odd.

2.3 Cayley graphs and Cayley hashes

In the early 1990s, Zémor [52] presented the idea of constructing hash functions from Cayley graphs. Zémor's approach came from a desire to satisfy the following small modification property that Godlewski and Camion had introduced in [17].

Property 2.3.1. [Small Modifications Property] There exists a $d \in \mathbb{N}_0$ such that if m' is any modification of m affecting fewer than d consecutive bits, then $h(m) \neq h(m')$.

Before presenting Zémor's idea, we need to present some background information on Cayley graphs.

Definition 2.3.2. *Given a group G and generating set S of G, the directed Cayley graph $C(G, S)$ is the graph with vertices the elements of G and a directed edge from w to v if and only if $v = sw$ for some $s \in S$.*

We will make use of the following two notions regarding Cayley graphs, as defined in [53].

Definition 2.3.3. *The directed girth of a graph G is the largest integer δ such that for any distinct vertices v, w there exists at most one path from v to w of length less than δ.*

We note that directed girth of a graph is different than the girth of a graph, which is defined as the length of the shortest cycle.

Definition 2.3.4. *The diameter of a graph* Γ *is the largest distance* $d(v, w)$ *between any two vertices* v *and* w *of* Γ.

With these notions in place, we now present Zémor's construction.

Definition 2.3.5. *Given a finite group* G *and two generators* A *and* B*, we define the associated hash function* H *as follows: For any message* $m_1 \dots m_k \in \{0,1\}^*$*,* $H(m_1 \dots m_k)$ *is the computed product* $H(m_1) \cdots H(m_k) \in G$*, where* $H(0) = A$ *and* $H(1) = B$.

In practice, one chooses an embedding of the finite group G into $\{0,1\}^N$ for some $N \sim \log_2(|G|)$. With this, the hash values lie in $\{0,1\}^N$ as is required, but we will consider the hash values as elements of G for discussion purposes.

The relation between the small modifications property and hash functions from Cayley graphs is now apparent. Namely, let G be a group with generating set $S = \{A, B\}$ and H be the associated hash function. If the directed girth of $C(G, S)$ is δ, then two messages of length less than δ cannot form a collision in H. This was precisely the motivation that led Zémor in 1991 to present the idea of using hash functions over the group $\mathrm{SL}_2(\mathbb{F}_p)$, based on previous uses of this group to produce Cayley graphs with large girths [52].

In particular, in [52] Zémor initially proposed the hash function H_1, which was defined as in Definition 2.3.5 with $G = \mathrm{SL}_2(\mathbb{F}_p)$ for p a large prime, and generators

$$A_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad B_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

To make trivial collisions of the form $A^{\mathrm{ord}(A)} = I$ infeasible, Tillich and Zémor suggested that p be approximately 150 bits in length [48].

In addition to picking generating sets that produced Cayley graphs with large girths, Zémor [52] also hoped to choose other generating sets whose Cayley graphs had small diameters. The motivation for this is as follows: given an element g in G the distance from I to g is bounded by the diameter. Thus, given a small diameter, every element of G is the image of a bitstring of a practical length. In particular Zémor included computing the diameter of a resulting Cayley graph in several early works [52], [53].

However, in [48] Tillich and Zémor showed that by essentially using the Euclidean algorithm, one could find a factorization of any matrix in $\mathrm{SL}_2(\mathbb{Z})$ with entries in $\mathbb{N}$ as a product of A_1s and B_1s. Thus one could produce collisions by finding a matrix of such a form that was congruent to the identity mod p. The reason this construction was subject to such an attack was that the choice of generators A_1 and B_1 generated too large of a subgroup of $\mathrm{SL}_2(\mathbb{Z})$. Attacks that use this kind of weakness are known as **density attacks**.

Tillich and Zémor's work prompted the construction of numerous other hash functions based on Cayley graphs, which are by some referred to as *Cayley hashes*. Formally, Petit, Lauter, and Quisquater describe Cayley hashes as "efficient and provably secure hash functions constructed from the Cayley graphs of (projective) linear groups" [35].

The most widely studied of these constructions was due to Tillich and Zémor themselves; the Zémor-Tillich hash function, which we present as the true starting point for this work in the next section. Before doing so, we mention some other directions of hash functions based on Cayley graphs.

With the density attack in mind, Zémor aimed to choose generating sets with the property that given any two matrices in $\mathrm{SL}_2(\mathbb{Z})$ congruent mod p, the probability

that one of these matrices is a product of As and Bs is small [53]. In particular, Zémor noted the sets $S_2 = \{A_1^2, B_1^2\}$ and $S_3 = \{A_1, A_1B_1\}$ have this property [53]. In [48], Tillich and Zémor also suggested other generators, such as using the sets $S = \{A_1^k, B_1^k\}$ for some $k > 1$. In [6], Bromberg, Shpilrain, and Vdovina further investigated this suggestion, and in particular the cases $k = 2$ and $k = 3$.

Inspired by the Zémor-Tillich hash function and Zémor's original construction, Cayley hashes from expander graphs have also been of great interest lately. Aiming to be resistant to previous attacks, such as small order and density attacks, Charles, Lauter and Goren suggested using the family of Ramanujan graphs over $\mathrm{PSL}_2(\mathbb{F}_p)$ introduced by Lubotzsky, Phillips, and Sarnak (LPS) [7]. Petit, Lauter and Quisquater suggested another construction based on expander graphs, called the Morgenstern hash function, which took values in $\mathrm{PSL}_2(\mathbb{F}_{2^n})$, but was also constructible over $\mathrm{PSL}_2(\mathbb{F}_q)$ [35].

In [50] Tillich and Zémor presented an attack based on the LPS hash function which was later extended to a preimage finding algorithm by Petit, Lauter, and Quisquater in [36]. Petit, Lauter, and Quisquater also adapted the attack to the Morgenstern hash function [36]. More recently, in [20] and [21] Jo suggests using Chiu's cubic Ramanujan graphs over $\mathrm{PSL}_2(\mathbb{F}_p)$, but immediately finds a corresponding attack.

We note these expander graph constructions were different from Tillich and Zémor's construction as they used undirected Cayley graphs, and were further based on quaternion-like algebras. Also, we note that another construction using Pizer graphs over elliptic curves was given in [7], and with, the LPS construction, was proposed for a patent [25].

The specific hash functions in this section will not be prominent in what follows, but we hope this discussion exhibits the interest and scope of such constructions.

Chapter 3

The Zémor-Tillich hash function

Motivated by Section 2.3, in this section we present our main starting point: the Zémor-Tillich hash function. Here, we discuss its definition and the many desirable properties it possesses. We then discuss attacks on Zémor-Tillich, focusing on the palindrome attack by Grassl, Ilić, Magliveras, and Steinwandt [18] which finally disqualified Tillich and Zémor's original choice of generators.

3.1 Definition and primary properties

In 1994, Tillich and Zémor [49] presented a new hash function based on Zémor's previous construction. We define the Zémor-Tillich hash function as follows.

Definition 3.1.1. *Let $r_n(x)$ be an irreducible polynomial of degree n over $\mathbb{F}_2$, let $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/\langle r_n(x)\rangle$ and let $G = \mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$. Further, set*

$$A := \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix}, \qquad B := \begin{pmatrix} x & x+1 \\ 1 & 1 \end{pmatrix}$$

where x is a root of r_n. Then, the Zémor-Tillich hash function is the associated hash function H as specified in Definition 2.3.5.

Using a well-known result of Dickson classifying the subgroups of $\mathrm{PSL}_2\big(\mathbb{F}_q\big)$, and its exposition in [47], Tillich and Zémor showed that $\langle A, B\rangle = \mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$, thus providing $2^n(2^{2n}-1)$ possible hash values.

As well, Tillich and Zémor presented the following the following result [49, Theorem 3.6], which suggests that the Zémor-Tillich hash function is not susceptible to density attacks (see Section 2.3).

Theorem 3.1.2 (Tillich and Zémor)**.** *For $k \in \mathbb{N}$ set*

$$\mathcal{S}_k = \left\{M \in \mathrm{M}_{2\times 2}\big(\mathbb{F}_2[x]\big) \ : \ \deg(M) \leq k\right\}.$$

Then, viewing A and B as elements of $\mathrm{M}_{2\times 2}\big(\mathbb{F}_2[x]\big)$,

$$\frac{|\{M \in \mathcal{S}_k \ : \ M \text{ is a word in } A \text{ and } B\}|}{|\mathcal{S}_k|} = \mathcal{O}(1/k).$$

The construction in Definition 3.1.1 also preserves the small modifications property, namely we have the following [49, Lemma 3.5].

Lemma 3.1.3. *Suppose that m, m' are bitstrings in $\{0,1\}^*$ such that $H(m) = H(m')$. Then at least one of m, m' must have length $\geq n$.*

Proof: We give a sketch of the proof here. A more general result is shown in detail in Appendix B.

Using induction it is straightforward to show that a product $m_1 \cdots m_k$ in A and B

over $\mathrm{M}_{2\times 2}\big(\mathbb{F}_2[x]\big)$ will be of the form

$$\begin{pmatrix} f_k & g_{k-1} \\ f_{k-1} & g_{k-2} \end{pmatrix} \text{ if } m_k = A \quad \text{and} \quad \begin{pmatrix} f_k & g_k \\ f_{k-1} & g_{k-1} \end{pmatrix} \text{ if } m_k = B$$

where each f_i and g_i is a polynomial, and the subscript i indicates the degree. This form is preserved for hash values bitstrings of length less than n when we quotient by $\langle r_n(x)\rangle$, thus ensuring different hash values in $\mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$. □

As well, Tillich and Zémor found that the Cayley graph associated to $G = \mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$ and $S = \{A, B\}$ satisfies the following proposition [49, Prop. 2.3].

Proposition 3.1.4. *Let $C(G, S)$ be a Cayley graph such that the greatest common divisor of its cycle lengths equals 1. Then the associated hash function of G, S is such that the distribution of hashed values of strings of length k tends to the uniform distribution as k tends to infinity.*

Finding a collision in the Zémor-Tillich hash function is equivalent to the problem of finding two products of reasonable length in $\{A, B\}$ that are equal in G, which in mathematics is sometimes referred to as the "balance problem". Further, second preimage and preimage resistance are respectively reduced to the so-called "representation problem" and "factorization problem" [38]. These are believed to be hard mathematical problems for many classes of groups [49].

In addition, Tillich and Zémor's hash function has many of the properties that made Zémor's original construction of considerable interest. As we are working over $\mathbb{F}_{2^n}$, computing a hashed value requires only simple operations which are easily implemented in software or hardware. Further, as it is the case that $H(xy) = H(x)H(y)$, it is possible to improve the speed of the Zémor-Tillich hash function by using parallelization and precomputations [1].

Originally, some modifications of the Zémor-Tillich hash function were suggested ([41], [34] and [40]) to improve its feasibility. In [41] Petit, Veyrat-Charvillon, and Quisquater optimized the computation time of the Zémor-Tillich hash function to within 25 times that of SHA−256, which was an industry standard. De Meulenaer, Petit, and Quisquater noted this was faster than the computation time of almost all other provably secure hash functions at the time [11]. Further in [11] the authors performed an extensive analysis that showed a modified version of the Zémor-Tillich hash function was comparable in speed to many other cryptographic standards at the time.

3.2 Attacks on the Zémor-Tillich hash function

In this section we present a summary of attacks on the Zémor-Tillich hash function. We outline the attacks that run in subexponential time, and how to prevent these attacks, in Table 3.1 below. We describe each of these in detail in sections 3.2.1 to 3.2.3, and 3.2.5, below.

3.2.1 Short relations attack

The first attack on the Zémor-Tillich hash function was proposed by Charnes and Pieprzyk [8]. The authors use the idea that for certain choices of $r_n(x)$, the element A has small order, and therefore short relations in $\{A, B, A^{-1}, B^{-1}\}$ will yield collisions of practical lengths. Specifically, the authors find and suggest using the relation $A^{-1}BA^{-1}B^2A^{-1}B = BA^{-1}B$. They then find a choice of polynomial for $n = 131$ such that the order of A is 263.

Attack	Resolution
Short relations attack [8] **:** $r_n(x)$ such that A or B has small order, produce short relations	very unlikely for random choice of $r_n(x)$ [1], carefully choose $r_n(x)$
Small order attack [46] **:** with decomposable choice of $r_n(x)$ find short products in $\{A, B\}$ with small order	unlikely for random choice of $r_n(x)$ [43] , choose n prime [46] or increase n to ~ 1024 [35]
Embedding attack [16] **:** powers of A (or B) are $\mu A + \lambda I$ with some conditions, use short product of these and solve linear system	no known prevention, but collisions produced are not practical [16]
Palindrome attack [18]**:** hash of palindromic string gives entries expressible by recursive relation in x and $x+1$, produce relations using maximal length Euclidean algorithm	choose new A and B, or add redundancy [37]

Table 3.1: Summary of known subexponential attacks on the Zémor-Tillich hash function and how to prevent against these attacks.

Before going further, we remark that the relation given by Charnes and Piepryzk is in fact more complex than necessary. For example, Abdukhalikov and Kim note [1] that the shorter relation $A^{-1}B = B^{-1}A$ also holds. Having a small order of A (or B) also yields trivial collisions of the form $A^{\mathrm{ord}(A)} = I$.

Still, the short relations attack cautions us that $r_n(x)$ should be chosen so that the orders of A and B are large. A natural question is: how hard is it to choose $r_n(x)$ in this way? In [1] Abdukhalikov and Kim show that a choice of $r_n(x)$ such that the orders of A and B are small is rare. Indeed, for $n = 131$ the probability that the orders of A and B are smaller than $M = 10^6$ is less than 10^{-27} [1].

3.2.2 Small order attack

The idea of attacking the Zémor-Tillich hashing scheme by finding elements of small order was further explored by Steinwandt, Grassl, Geiselmann, and Beth [46]. Rather than considering when A and B have small orders, the authors search for short bit-strings that hash to elements of small order. Using this idea, the authors find a collision of length 387 for $n = 140$.

However, this attack also depended on the choice of $r_n(x)$, and it was shown this attack can be avoided simply by choosing n to be prime. Further, in [43] the authors show that the probability a random choice of $r_n(x)$ will allow such an attack approaches 0 as n increases. In [35] the authors recommend increasing n to around 1024 to protect against the small order attack as the attack uses an "exponentially long exhaustive search".

Steinwandt et. al. [46] also show that for the parameters suggested by Tillich and Zémor, constructing a "trapdoor attack" is fairly easy. To construct such attack, one computes $H(m)$ over $\mathrm{M}_{2\times 2}\big(\mathbb{F}_2[x]\big)$ for a sufficiently large message m and chooses the defining irreducible polynomial $r_n(x)$ to be the an irreducible polynomial dividing the greatest common divisor of the four polynomial entries of $H(m) - I$.

3.2.3 Embedding attack

The first attack on the Zémor-Tillich hash function that was independent of the choice of irreducible polynomial $r_n(x)$ was proposed by Geiselmann [16] (we note also an alternative exposition of Geiselmann's embedding attack in [43]).

Geiselmann's idea is to embed $\langle A\rangle$, $\langle B\rangle$ into the subalgebras of $\mathrm{M}_{2\times 2}\big(\mathbb{F}_{2^n}\big)$ generated by A and B. By the Cayley-Hamilton theorem, A is the root of a polynomial $f(x)$

over $\mathbb{F}_{2^n}$, so we see that any element of the form $a_0 + a_1 A + a_2 A^2 + \ldots$, where $a_i \in \mathbb{F}_{2^n}$ for each $i \in \mathbb{N}_0$, can be written as

$$\mu A + \lambda I \tag{3.2.1}$$

for some $\mu, \lambda \in \mathbb{F}_{2^n}$. In particular, notice that if a matrix in $\mathrm{M}_{2\times 2}\big(\mathbb{F}_{2^n}\big)$ is a power of A, it must be of the form in (3.2.1). Geiselmann shows that an element of the form (3.2.1) is a power of A if and only if it has determinant 1 and satisfies a second condition that depends on the reducibility of $f(x)$: namely that either $(\mu A + \lambda I)^{\mathrm{ord}(A)} = I$ or $(\lambda\beta + \mu)^{\mathrm{ord}(A)} = 1$, where $\mathrm{ord}(A)$ is the order of A and β is a root of $f(x)$. An analogous set of conditions is made for B.

To find a collision, one chooses an element $C \in \mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$ and creates a system of equations by forming a short product of alternating terms of the form either $\mu A + \lambda I$ or $\mu B + \lambda I$ with $\mu, \lambda \in \mathbb{F}_{2^n}$. By setting this equal to C, this creates four equations in the μs and λs from the entries of the matrices on either side of the equality, as well as an additional equation for each alternating term to force a determinant of 1. One then solves this system and checks the second condition holds for each alternating term. If this is indeed the case, we have obtained that C is equal to a short sequence of alternating powers of A and B. Once such a system is found, the powers are calculated using a discrete log, yielding an attack. By choosing C to be a power of A or B, we thus obtain a short relation.

Geiselmann implemented his embedding attack in AXIOM [19] for $\mathbb{F}_{2^{21}}$ with defining polynomial $x^{21} + x^2 + 1$, and was able to find many collisions. One of the smaller of these collisions was

$$A^{79670} = B^3 \cdot A^7 \cdot B^{69216} \cdot A \cdot B^{88234}. \tag{3.2.2}$$

Noting that in this case, $\mathrm{ord}(A) = 2097151$ and $\mathrm{ord}(B) = 699051$, these collisions are

shorter than the trivial ones, or ones using short identities in A, B and their inverses as in [8].

We note this attack is efficient, but not practical. The collision in (3.2.2) cannot be considered functional as an attack because it contains such long strings of ones and zeros. Geiselmann noted this, as well as that this is a common property among the collisions he found [16].

3.2.4 General attacks

In [40], Petit, Quisquater, Tillich and Zémor optimize the birthday attack on the Zémor-Tillich hash function (see Appendix A) and extend this to a preimage attack. This attack had a time complexity of $\mathcal{O}(2^{n/2})$, and was noted to be practical for up to n of about 130, suggesting that n should be chosen larger than the Tillich and Zémor's original suggestion of $130 \leq n \leq 170$.

In [33], Petit finds heuristic methods to produce, from an arbitrary pair of matrices $\{A, B\}$ generating $\mathrm{SL}_2\left(\mathbb{F}_{2^n}\right)$, a set $\mathcal{S}$ containing one symmetric matrix and a subgroup of orthogonal matrices such that factoring over $\mathcal{S}$ would allow factoring over $\{A, B\}$. In [14], Faugère, Perret, and Petit use this to produce by conjugation a set of matrices of the form

$$E(t) = \begin{pmatrix} t & 1 \\ 1 & 0 \end{pmatrix}, \quad t \in \mathbb{F}_2[x], \tag{3.2.3}$$

again dependent $\{A, B\}$, to which the factorization problem is further reduced. Faugère et. al. then give an algorithm for factoring over matrices of the form (3.2.3) that improves on the time estimates in [40] for large values of n, but as a trade off produces impractically long factorizations. This idea does not appear to have been pursued further, and does not seem easily generalizable to characteristic $p > 2$.

In [30] Mullan presents some meet-in-the-middle attacks, each with time complexity $\mathcal{O}(\sqrt{q})$ and expected collision length $\mathcal{O}(2(\log q)^2)$. Mullan further showed that her algorithms could be modified in the case $p = 2$ to shorten the length of collisions produced to $\mathcal{O}(4n)$, and noted this approach could also work for Petit et. al.'s birthday attack approach [30]. In [30] Mullan also presents a summary table of the complexity and expected collision lengths of these meet-in-the-middle attacks, Petit et. al's optimized birthday attack and Geiselmann's embedding attack.

In [31], Mullan and Tsaban present a generic attack, which shortens the expected collision length considerably, providing collisions of length $\mathcal{O}(n)$ in the case $p = 2$, and $\mathcal{O}((\log q)^2/\log(\log q))$ in general [31].

3.2.5 Palindrome attack

In [18], Grassl, Ilić, Magliveras, and Steinwandt present a palindrome attack on the Zémor-Tillich hash function that is both independent of the choice of defining irreducible polynomial $r_n(x)$ and produces collisions of reasonable length. The palindrome attack also applied to the other proposed versions of Zémor-Tillich in [41], [34] and [40].

To find collisions in the Zémor-Tillich hash function, Grassl et. al. start by showing that any collision found using the generators $\{A', B'\} = \{A^{-1}AA, A^{-1}BA\}$ occurs if and only if the same message produces a collision in the original Zémor-Tillich generators. We note here that

$$A' = \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix} \quad B' = \begin{pmatrix} x+1 & 1 \\ 1 & 0 \end{pmatrix}. \tag{3.2.4}$$

We will denote the associated hash function of $\{A', B'\}$ by H'. Grassl et. al. show the following result [18, Corollary 1].

Proposition 3.2.1 (Grassl et. al.). *If $v \in \{0,1\}^*$ is a palindrome of even length, then*

$$A'H'(v)A' + B'H'(v)B' = \begin{pmatrix} a^2 & a^2 \\ a^2 & 0 \end{pmatrix} \tag{3.2.5}$$

for some $a \in \mathbb{F}_2[x]$ with $\deg(a) = |v|/2$.

We see that if $a \equiv 0 \mod r_n(x)$, then we have $H'(1v1) = H'(0v0)$, and thus have obtained a collision. To produce such an a, Grassl et. al. show the following result [18, Corollary 2].

Proposition 3.2.2 (Grassl et. al.). *If $v = b_n \dots b_1 b_1 \dots b_n \in \{0,1\}^*$ is a palindrome of length $2n$ then for each i the square root a_i of the upper left entry of $H'(b_i \dots b_1 b_1 \dots b_i)$ is given by*

$$a_i = \begin{cases} 1 & \text{if} \quad i = 0; \\ x + b_1 + 1 & \text{if} \quad i = 1; \\ (x + b_i)a_{i-1} + a_{i-2} & \text{if} \quad 1 < i \leq n. \end{cases}$$

The last line of this recursion may be familiar; in the Euclidean algorithm this is precisely the formula for r_i, with quotient $q_i = x + b_i$, and previous remainders $r_{i-1} = a_{i-1}$ and $r_{i-2} = a_{i-2}$. We saw that having $a \equiv 0 \mod r_n(x)$ is sufficient to form a collision in H'. Connecting the two, a collision can be found by having $a_n = r_n(x)$, which in turn can be produced by finding a polynomial, $h(x)$, such that applying the Euclidean algorithm for factoring polynomials to $(r_n(x), h(x))$ produces a chain of polynomial quotients $q_n, \dots, q_1$ in $\{x, x+1\}$. In [29], Mesirov and Sweet guarantee, and show how to find, such a polynomial $h(x)$. To obtain a factorization of $r_n(x)$, we

let $b_i = 0$ if $q_i = x$ and $b_i = 1$ if $q_i = x + 1$ (note that from Proposition 3.2.2, we see that the the opposite choice is made for the case $i = 1$). Then $H'(b_m \dots b_1 b_1 \dots b_m)$ gives us a matrix of the form in (3.2.5) with $a^2 = (r_n(x))^2 \equiv 0$. Thus, by the above argument, we have obtained a collision $H'(1v1) = H'(0v0)$.

We note that by selecting generators of the form $E(t_A)$, $E(t_B)$ as in (3.2.3) for some t_A, $t_B \in \mathbb{F}_2[x]$, Proposition 3.2.1 could be easily extended. Still, forming a collision would require obtaining only t_A and t_B as quotients from the Euclidean Algorithm, for which no such method is known.

Grassl et. al. also note that given a collision of the form $H'(1v1) = H'(0v0)$, the known identity $A^{-1}B = B^{-1}A$ gives that $H'(0v1) = H'(1v0)$ as well. Further, the palindrome attack was extended by Petit and Quisquater in [37] to a second preimage alogrithm and separately to a preimage algorithm that factorizes any element in $\mathrm{SL}_2\left(\mathbb{F}_{2^n}\right)$ over A and B using a set of matrices with precomputed preimages.

The palindrome attack is very dependent on using characteristic 2. In particular, in [30, Lemma 3.1(b)] Mullan shows that if $\mathbb{F}_q$ does not have characteristic 2 then $H(0v0) = H(1v1)$ cannot hold for any v.

To avoid this attack, a few alternative choice of generators over $\mathrm{SL}_2\left(\mathbb{F}_{2^n}\right)$ have been loosely proposed. For instance, in [37] Petit and Quisquater proposed that replacing A with $\begin{pmatrix} x^2 & 1 \\ 1 & 0 \end{pmatrix}$, A^2, or A^3 could be enough to prevent Grassl et. al's attack. A similar suggestion was also presented as an open challenge in [38].

Introducing some simple redundancy is believed to be enough to regain security in Zémor-Tillich against the palindrome attack and its preimage attack extension [37], as well as in other Cayley hashes constructions such as the Morgenstern hash function [36]. In [35], Petit et. al. also suggested introducing some redundancy to remove

malleability from Cayley hash functions; that is, the property that given $H(m_1)$ and $H(m_2)$, one can easily compute $H(m_1 m_2)$. One idea we had was to create some more localized redundancy, such as defining $h(mb_\ell m') := H(m)H(b_\ell)^2 H(m')$. To our knowledge, this has not been considered in the literature, and we do not explore it further here.

3.3 Outlook for the Zémor-Tillich hash function

It is important to note that the palindrome attack that finally broke the Zémor-Tillich hash function was highly dependent on the form of the generators chosen. The underlying structure of hash function constructions over $\mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$ and similar groups is increasingly believed to be a secure model [38]. In particular, for other choices of generators in $\mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$, as well as extensions over $\mathbb{F}_q$, Tillich and Zémor's construction appears to still be considered viable [39].

For example, in [1], Abdukhalikov and Kim introduce a natural extension of the Zémor-Tillich hash function to $\mathbb{F}_q \cong \mathbb{F}_p[x]/\langle r_n(x)\rangle$ for $p > 2$, by choosing the generators

$$A := \begin{pmatrix} x & -1 \\ 1 & 0 \end{pmatrix} \quad B := \begin{pmatrix} x & x-1 \\ 1 & 1 \end{pmatrix} \tag{3.3.1}$$

where again x is a root of r_n, and the natural image of x in $\mathbb{F}_q$, and showed that A and B generated all of $\mathrm{SL}_2\big(\mathbb{F}_{p^n}\big)$. No other suggestions of generators over $\mathrm{SL}_2\big(\mathbb{F}_q\big)$ are known.

We will show that no practical general attacks are known in $\mathrm{SL}_2\big(\mathbb{F}_q\big)$, and further that we can easily prevent against the short relations and small order attacks, as well as Geiselmann's embedding attack. Extending to $\mathrm{GL}_2\big(\mathbb{F}_q\big)$ would introduce yet another element of complexity for finding collisions. Moreover, expanding our search

for generators to $\mathrm{GL}_2\big(\mathbb{F}_q\big)$ seems new in itself. As in some initial suggestions in $\mathrm{SL}_2\big(\mathbb{F}_{2^n}\big)$, we would also like to consider some choices of generators with quadratic and higher degree entries. These arguments in conjunction with the many compelling properties of Tillich and Zémor's construction, motivate our search for generators over $\mathrm{GL}_2\big(\mathbb{F}_q\big)$.

Chapter 4

Free Generators Theorem

Following Zémor's original motivation, we use as a starting point the desire to preserve the property of the Zémor-Tillich hash function that small modifications of text are detected. Our idea of doing this is to choose A and B to be free generators of a free subgroup of $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ with polynomial entries, and then take as generators of a hash function the images of A and B in $\mathrm{GL}_2\big(\mathbb{F}_q\big)$ when we quotient by an irreducible polynomial $r_n(x)$ such that $\mathbb{F}_q \cong \mathbb{F}_p[x]/\langle r_n(x)\rangle$. Supposing $\deg(A)$ and $\deg(B)$ are small, we can see, and will later show formally, that this will guarantee collisions cannot occur before a certain length.

The main result of this chapter, the Free Generators Theorem (Theorem 4.4.2), provides an abundant source of pairs of matrices generating a free subgroup of $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$. The idea behind the Free Generators Theorem will be to use Tits' "Ping-Pong" Lemma (Proposition 4.5.1). Our argument is inspired by Breuillard and Gelander's [5] consideration of this argument for projective linear groups over local fields. In particular, Breuillard and Gelander [5] consider such groups as acting on an associated projective space, and show free groups can be found using group elements

which map points a sufficient distance from a specified *repulsing point* close to a specified *attracting point*. With this in mind, we construct free generators A, B in $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ by considering the argument over $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$. Again, we emphasize that our applications of this idea in this chapter are new.

4.1 The projective space $\mathbb{P}^1$ and matrices over $\mathbb{F}_p((x))$

Let p be a prime, $\mathbb{F}_p$ be the field with p elements, and $\mathbb{F}_p((x))$ be the field of formal Laurent series over $\mathbb{F}_p$. The elements of $\mathbb{F}_p((x))$ are series of the form

$$g(x) = \sum_{k=m}^{\infty} g_k x^k$$

for $g_i \in \mathbb{F}_p$ and $m \in \mathbb{Z}$. Because we want to see elements of $\mathbb{F}_p((x))$ as elements of an abstract field, not as functions, we will write g for an element $g(x) \in \mathbb{F}_p((x))$. With this notation, we will use g^{-1} to mean the multiplicative inverse of $g \in \mathbb{F}_p((x))^\times$, not $g^{-1}(x)$.

We denote the valuation $v(g)$ of an element $g \in \mathbb{F}_p((x))$ as above in the standard way, namely

$$v(g) = \begin{cases} \min\{k \,:\, g_k \neq 0\} & \text{if } g \neq 0; \\ \infty & \text{if } g = 0. \end{cases}$$

With this, we define the absolute value as $|g| = p^{-v(g)}$. For instance, the element $f = x^3 + x^6$ would have $|f| = p^{-3}$, while $g = x^{-5} + x^{-2}$ would have $|g| = p^5$. This absolute value is multiplicative, that is $|fg| = |f||g|$ for any f, $g \in \mathbb{F}_p((x))$. As well, this absolute value is non-Archimedean, meaning it satisfies the ultrametric or

non-Archimedean triangle inequality

$$|g+h| \leq \max\{|g|, |h|\} \tag{4.1.1}$$

for all g, $h \in \mathbb{F}_p((x))$. To see this, notice that for $g, h \in \mathbb{F}_p((x))$ the smallest index for which the Laurent series of $g+h$ could have a nonzero term cannot be strictly less than $\min\{v(g), v(h)\}$.

We denote the ring of integers of $\mathbb{F}_p((x))$ by $\mathcal{O}$. We note that

$$\mathcal{O} = \mathbb{F}_p[[x]] = \{g \in \mathbb{F}_p((x)) \ : \ v(g) \geq 0\}.$$

Further, the multiplicative group of $\mathcal{O}$ is

$$\mathcal{O}^\times = \{g \in \mathbb{F}_p((x)) \ : \ v(g) = 0\}.$$

As well, we define

$$V = \{(u_1, u_2) \ : \ u_1, u_2 \in \mathbb{F}_p((x))\}.$$

With this, we define the 1-dimensional projective space over $\mathbb{F}_p((x))$ by

$$\mathbb{P}^1 = \mathbb{P}^1(\mathbb{F}_p((x))) := (V \backslash \{0\})/\sim$$

where $\sim$ is the equivalence relation $(u_1, u_2) \sim (v_1, v_2)$ if there exists a $k \in \mathbb{F}_p((x))^\times$ such that $(u_1, u_2) = (kv_1, kv_2)$.

In particular, for $(u_1, u_2) \neq (0, 0)$ we define $[u] = [u_1 : u_2] \in \mathbb{P}^1$ to be the equivalence class

$$\Big\{k(u_1, u_2) \in V \ : \ k \in \mathbb{F}_p((x))^\times\Big\}.$$

Remark 4.1.1. The equivalence classes $[1:0]$ and $[0:1]$ will be denoted by $[e_1]$ and

$[e_2]$ respectively. Note that $[f : g] = [1 : gf^{-1}]$ for $f \neq 0$.

We will consider $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ and its subgroups as acting on V by matrix multiplication on the left, that is for $g = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \in \mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ and $u = (u_1, u_2) \in V$ we have

$$g \cdot u = \begin{pmatrix} a_{11}u_1 + a_{12}u_2 \\ a_{21}u_1 + a_{22}u_2 \end{pmatrix}.$$

This action factors to an action of $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ and its subgroup on $\mathbb{P}^1$ by $g \cdot [u] = [g \cdot u]$. In particular, the following lemma shows that $\mathrm{GL}_2\big(\mathcal{O}\big)$ acts transitively on $\mathbb{P}^1$.

Lemma 4.1.2. *Suppose that $[v] \in \mathbb{P}^1$. Then there exists an element $g \in \mathrm{GL}_2\big(\mathcal{O}\big)$ such that $g \cdot [v] = [e_2]$.*

Proof: Let $v = (v_1, v_2) \in V$ and assume $v \neq (0, 0)$.

Notice if $\max\{|v_1|, |v_2|\} = |v_1| = p^\ell$, then $|x^\ell v_1| = 1$, so $x^\ell v_1 \in \mathcal{O}^\times$, and $|x^\ell v_2| \leq 1$, so $x^\ell v_2 \in \mathcal{O}$. Further $[v] = [x^\ell v] = [x^\ell v_1 : x^\ell v_2]$. Likewise, if we instead have $\max\{|v_1|, |v_2|\} = |v_2| = p^\ell$, then $|x^\ell v_1| \leq 1$, $|x^\ell v_2| = 1$ and $[v] = [x^\ell v] = [x^\ell v_1 : x^\ell v_2]$. We can thus assume that there exists a representative v of $[v]$ such that $[v] = [\alpha : \beta]$ for some $\alpha, \beta \in \mathcal{O}$, and further that either $\alpha \in \mathcal{O}^\times$ or $\beta \in \mathcal{O}^\times$ or both.

If $\alpha \in \mathcal{O}^\times$ then

$$g = \begin{pmatrix} \beta & -\alpha \\ \alpha^{-1} & 0 \end{pmatrix} \in \mathrm{GL}_2\big(\mathcal{O}\big)$$

gives $g \cdot [v] = [e_2]$. If $\beta \in \mathcal{O}^\times$ then let

$$g = \begin{pmatrix} -\beta & \alpha \\ 0 & \beta^{-1} \end{pmatrix} \in \mathrm{GL}_2\big(\mathcal{O}\big).$$

We then obtain $g \cdot [v] = [e_2]$. □

We recall that elements of $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ are cosets of the form gZ where Z is the centre of $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ and $g \in \mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$. For ease of notation, for an element $gZ \in \mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ where

$$g = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \text{we will write} \quad gZ = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

Since Z acts trivially on $\mathbb{P}^1$, we can consider $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ and its subgroups as acting on $\mathbb{P}^1$ in the same way as $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$, that is, by left matrix multiplication.

4.2 A metric on $\mathbb{P}^1$

Our main theorem in this chapter will use neighbourhoods of points in $\mathbb{P}^1$, and so in particular we need a notion of a distance between two points in $\mathbb{P}^1$. In this section, we will define a distance on $\mathbb{P}^1$ and show, as the section title suggests, that this distance is in fact a metric. To start with, we define a norm on V, which is in some contexts referred to as the sup-norm or the ∞-norm.

Definition 4.2.1. *For $(u_1, u_2) \in V$ the norm of (u_1, u_2) is*

$$||(u_1, u_2)|| = \max\{|u_1|, |u_2|\}.$$

For any $v = (v_1, v_2) \in V$ with v nonzero, we can see that $|v_1|$ and $|v_2|$ must both be a power of p or zero, so it follows that $||v||$ must be a power of p. Interestingly, we will show in the next lemma that $||v||$ is invariant under the action of elements of $\mathrm{GL}_2\big(\mathcal{O}\big)$ on V.

Lemma 4.2.2. *Let $g \in \mathrm{GL}_2(\mathcal{O})$. Then for any $u \in V$ we have*

$$||g \cdot u|| = ||u||.$$

Proof: Let $g = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \in \mathrm{GL}_2(\mathcal{O})$ and let $u = (u_1, u_2) \in V$.

We see $g \cdot u = (a_{11}u_1 + a_{12}u_2, a_{21}u_1 + a_{22}u_2)$. By the non-Archimedean triangle inequality (4.1.1) we have

$$\begin{aligned} ||g \cdot u|| &= \max\{|a_{11}u_1 + a_{12}u_2|, |a_{21}u_1 + a_{22}u_2|\} \\ &\leq \max\{|a_{11}u_1|, |a_{12}u_2|, |a_{21}u_1|, |a_{22}u_2|\}. \end{aligned} \tag{4.2.1}$$

We observe for $a \in \mathcal{O}$, we have that $|a| \leq 1$, so $|af| \leq |f|$ for any $f \in \mathbb{F}_p((x))$. Since $g \in \mathrm{GL}_2(\mathcal{O})$, (4.2.1) therefore gives

$$||g \cdot u|| \leq \max\{|u_1|, |u_2|\} = ||u||.$$

As $g \in \mathrm{GL}_2(\mathcal{O})$, we know that $g^{-1} \in \mathrm{GL}_2(\mathcal{O})$ as well. Thus by an identical argument to above, we know $||g^{-1} \cdot v|| \leq ||v||$ for all $v \in V$, in particular $v = g \cdot u$. Thus we have

$$||u|| = \left|\left|g^{-1}g \cdot u\right|\right| \leq ||g \cdot u|| \leq ||u||.$$

□

The following definition of distance is taken from Breuillard and Gelander [5].

Definition 4.2.3. *Let $[u], [v] \in \mathbb{P}^1$ be such that $[u] = [u_1 : u_2]$ and $[v] = [v_1 : v_2]$.*

Then the distance between $[u]$ *and* $[v]$ *is*

$$d([u],[v]) = \frac{||u \wedge v||}{||u||\,||v||} = \frac{|u_1v_2 - u_2v_1|}{\max\{|u_1|,|u_2|\}\max\{|v_1|,|v_2|\}}.$$

In our case the alternating tensor product $u \wedge v$ is one-dimensional, and we take the absolute value as its norm.

We note that $d([u],[v])$ is independent of the choice of representatives of the classes $[u]$ and $[v]$. Further, just as our norm is invariant under the action of elements of $\mathrm{GL}_2(\mathcal{O})$ on V, our distance is invariant under the action of elements of $\mathrm{GL}_2(\mathcal{O})$ on $\mathbb{P}^1$. That is, elements of $\mathrm{GL}_2(\mathcal{O})$ act by isometries on $\mathbb{P}^1$.

Lemma 4.2.4. *Let* $g \in \mathrm{GL}_2(\mathcal{O})$. *Then for any* $[u]$, $[v] \in \mathbb{P}^1$ *we have*

$$d([u],[v]) = d(g\cdot[u], g\cdot[v]).$$

Proof: As in the proof of Lemma 4.2.2, let $g = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \in \mathrm{GL}_2(\mathcal{O})$ and let $[u] = [u_1 : u_2]$, $[v] = [v_1 : v_2] \in \mathbb{P}^1$.

We see $g\cdot[u] = [a_{11}u_1 + a_{12}u_2 : a_{21}u_1 + a_{22}u_2]$ and $g\cdot[v] = [a_{11}v_1 + a_{12}v_2 : a_{21}v_1 + a_{22}v_2]$. Further, we note that since $g \in \mathrm{GL}_2(\mathcal{O})$, $\det(g) \in \mathcal{O}^\times$, thus $|\det(g)| = 1$. Therefore we have

$$\begin{aligned}
||g\cdot u \wedge g\cdot v|| &= |(a_{11}u_1 + a_{12}u_2)(a_{21}v_1 + a_{22}v_2) - (a_{21}u_1 + a_{22}u_2)(a_{11}v_1 + a_{12}v_2)| \\
&= |a_{11}a_{21}u_1v_1 + a_{11}a_{22}u_1v_2 + a_{12}a_{21}u_2v_1 + a_{12}a_{22}u_2v_2 \\
&\quad -a_{21}a_{11}u_1v_1 - a_{21}a_{12}u_1v_2 - a_{22}a_{11}u_2v_1 - a_{22}a_{12}u_2v_2| \\
&= |(a_{11}a_{22} - a_{12}a_{21})(u_1v_2 - u_2v_1)| \\
&= |\det(g)||u_1v_2 - u_2v_1| = ||u \wedge v||.
\end{aligned}$$

So $||g \cdot u \wedge g \cdot v|| = ||u \wedge v||$. As well, by Lemma 4.2.2 we have that $||g \cdot u|| = ||u||$ and $||g \cdot v|| = ||v||$. Thus, with Definition 4.2.3, we have

$$d([u],[v]) = \frac{||u \wedge v||}{||u||\,||v||} = \frac{||g \cdot u \wedge g \cdot v||}{||g \cdot u||\,||g \cdot v||} = d(g \cdot [u], g \cdot [v]).$$

□

Having our distance invariant under $\mathrm{GL}_2(\mathcal{O})$ will be extremely useful in conjunction with Lemma 4.1.2 for the remaining proofs in this section.

Another property of our distance is that it is an ultra-metric, as defined below. This is expected since our absolute value is non-Archimedean.

Definition 4.2.5. *Let S be a set of points. A function $D : S \times S \to \mathbb{R}$ is an ultra-metric if it is a metric and satisfies the ultrametric or strong triangle inequality*

$$D([u],[w]) \leq \max\{D([u],[v]), D([v],[w])\}$$

for any $[u]$, $[v]$, $[w] \in S$.

We note here that D is a metric if it is non-negative, symmetric, satisfies the triangle inequality, and takes value zero when, and only when $[u] = [v]$.

Lemma 4.2.6. *The function $d : \mathbb{P}^1 \times \mathbb{P}^1 \to \mathbb{R}$ defined in Definition 4.2.3 is an ultra-metric.*

Proof: By Definition 4.2.3, we see that d is non-negative and symmetric. Given $[u],[v],[w] \in \mathbb{P}^1$, we need to show that

$$d([u],[v]) = 0 \text{ if and only if } [u] = [v] \tag{4.2.2}$$

and that

$$d([u],[w]) \leq \max\{d([u],[v]), d([v],[w])\}. \tag{4.2.3}$$

We note that if the strong triangle inequality (4.2.3) holds, the triangle inequality must hold too. Thus by Definition 4.2.5, showing the relations (4.2.2) and (4.2.3) hold is sufficient to conclude d is a metric, and in particular an ultra-metric.

By Lemma 4.1.2 and Lemma 4.2.4 we may assume that $[v] = e_2$. Suppose also that $[u] = [a : b]$ and $[w] = [c : d]$.

To see that (4.2.2) holds, suppose first that $[u] = [v]$. Then by Definition 4.2.3 we have $d([u],[v]) = 0$. On the other hand, if $d([u],[v]) = 0$ we see

$$d([u],[v]) = 0 \iff d([a:b],[0:1]) = 0 \iff \frac{|a|}{\max\{|a|,|b|\}} = 0 \iff |a| = 0.$$

Thus $a = 0$, and therefore $[u] = [a : b] = [0 : b] = [v]$. So we see that (4.2.2) holds.

We now have left to prove that (4.2.3) holds. By Definition 4.2.3, the inequality (4.2.3) holds if and only if

$$\frac{|ad-bc|}{\max\{|a|,|b|\}\max\{|c|,|d|\}} \leq \max\left\{\frac{|a|}{\max\{|a|,|b|\}}, \frac{|c|}{\max\{|c|,|d|\}}\right\}. \tag{4.2.4}$$

Multiplying both sides of (4.2.4) by $\max\{|a|,|b|\}\max\{|c|,|d|\}$, we have that (4.2.3) holds if and only if

$$|ad-bc| \leq \max\{|a|\max\{|c|,|d|\}, |c|\max\{|a|,|b|\}\} = \max\{|ac|,|bc|,|ad|\}. \tag{4.2.5}$$

By the non-Archimedean triangle inequality (4.1.1) we have $|ad-bc| \leq \max\{|ad|,|bc|\}$, and we observe $\max\{|ad|,|bc|\} \leq \max\{|ad|,|bc|,|ad|\}$ holds trivially. Therefore equation (4.2.5) holds. □

The next lemma gives a useful property of d.

Lemma 4.2.7. *Let* $[u], [v] \in \mathbb{P}^1$. *Then* $0 \leq d([u],[v]) \leq 1$. *Further if* $d([u],[v]) \neq 0$ *then* $d([u],[v]) = \frac{1}{p^d}$ *for some* $d \in \mathbb{N}_0$.

Proof: Let $[u]$, $[v] \in \mathbb{P}^1$. By Lemma 4.1.2 and Lemma 4.2.4 we may assume that $[v] = e_2$. Suppose also that $[u] = [a : b]$. Then

$$d([u],[v]) = d([a:b],[0:1]) = \frac{|a|}{\max\{|a|,|b|\}} \leq 1. \tag{4.2.6}$$

We note that as the value that $|\cdot|$ assumes is always zero or a power of p, (4.2.6) shows that $d([u],[v])$ must be zero or a nonpositive power of p. □

4.3 Neighbourhoods in $\mathbb{P}^1$

Before going further it is useful to visualize how distance and neighbourhoods look in $\mathbb{P}^1$ with respect to our metric. To do so, we prove some results about neighbourhoods in $\mathbb{P}^1$ that will also be of use in the proof of our main theorem. We will first define what we mean by the ϵ-neighbourhood of an element of $\mathbb{P}^1$.

Definition 4.3.1. *For* $\epsilon > 0$, *the set*

$$N([u],\epsilon) = \left\{[v] \in \mathbb{P}^1 \ : \ d([u],[v]) \leq \epsilon\right\}$$

is the ϵ*-neighbourhood of* $[u]$ *in* $\mathbb{P}^1$. *Further, we will call* ϵ *its radius.*

We observe that by Lemma 4.2.7, for any $\epsilon > 0$ there exists a $d \in \mathbb{N}_0$ such that $N([u],\epsilon) = N\left([u], \frac{1}{p^d}\right)$.

We also recall that from Remark 4.1.1 we know any element of $\mathbb{P}^1\backslash\{[e_2]\}$ has a representative of the form $[1:h]$. Thus to see what a neighbourhood of an element of $\mathbb{P}^1$ looks like, we need only consider elements of the form $[1:h]$ and $[e_2]$.

Proposition 4.3.2. *Suppose that* $[1:h] \in \mathbb{P}^1$ *and let* $d \in \mathbb{N}_0$. *Then*

$$N\left([1:h], \frac{1}{p^{d+1}}\right) = \begin{cases} \{[1:g] \in \mathbb{P}^1 \ : \ |g-h| \leq \frac{1}{p^{d+1}}\} & \text{if } |h| \leq 1, \\ \{[1:g] \in \mathbb{P}^1 \ : \ |g-h| \leq |h|^2 p^{-(d+1)}\} & \text{if } 1 < |h| \leq p^d \\ \{[1:g] \in \mathbb{P}^1 \ : \ |g| > p^d\} \cup \{[e_2]\} & \text{if } |h| > p^d. \end{cases} \tag{4.3.1}$$

Further,

$$N\left([e_2], \frac{1}{p^{d+1}}\right) = \{[1:g] \in \mathbb{P}^1 \ : \ |g| > p^d\} \cup \{[e_2]\}. \tag{4.3.2}$$

Remark 4.3.3. We remark that in some cases, such as the proof of Lemma 4.3.5, it may be more useful to consider $N\left([h:1], \frac{1}{p^{d+1}}\right)$. In this respect, we note by Lemma 4.2.4 we have that $d([u],[v])$ is invariant under multiplication by $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. This allows us to replace $[1:h]$ with $[h:1]$ and $[1:g]$ with $[g:1]$ in Proposition 4.3.2 when needed.

We notice the $\frac{1}{p^{d+1}}$-neighbourhoods of $[e_2]$ and $[1:h]$ are the same for $|h| > p^d$. We can see this makes sense intuitively, as $[1:h] = [\frac{1}{h}:1]$, and for $|h|$ large the point $[\frac{1}{h}:1]$ is close to $[e_2] = [0:1]$. Indeed, suppose $[1:h] \in \mathbb{P}^1$ such that $|h| > p^d$, and let $[u] \in \mathbb{P}^1$. By Lemma 4.2.7 we have that $|h| \geq p^{d+1}$ must hold. Thus Definition 4.2.3 gives

$$d([e_2],[1:h]) = \frac{1}{|h|} \leq \frac{1}{p^{d+1}}.$$

If $d([1:h],[u]) \leq \frac{1}{p^{d+1}}$, by the strong triangle inequality (4.2.3) we know

$$d([e_2],[u]) \leq \max\{d([e_2],[1:h]), d([1:h],[u])\} = \frac{1}{p^{d+1}}.$$

Similarly, if $d([e_2],[u]) \leq \frac{1}{p^{d+1}}$, we know

$$d([1:h],[u]) \leq \max\{d([1:h],[e_2]), d([e_2],[u])\} = \frac{1}{p^{d+1}}.$$

From these observations, we know that (4.3.2) follows from the last case (4.3.1).

We prove Proposition 4.3.2 as a sequence of three lemmas — Lemma 4.3.4, Lemma 4.3.5, and Lemma 4.3.6 — below.

Lemma 4.3.4. *Suppose that* $[1:h] \in \mathbb{P}^1$ *is such that* $|h| \leq 1$ *and let* $d \in \mathbb{N}_0$. *Then*

$$N\left([1:h], \frac{1}{p^{d+1}}\right) = \left\{[1:g] \in \mathbb{P}^1 \ : \ |g-h| \leq \frac{1}{p^{d+1}}\right\}.$$

Proof: We note by the comments following Remark 4.3.3 that $[e_2] \notin N\left([1:h], \frac{1}{p^{d+1}}\right)$ since $|h| \leq 1 \leq p^d$. Thus

$$N\left([1:h], \frac{1}{p^{d+1}}\right) = \left\{[1:g] \ : \ d([1:h],[1:g]) \leq \frac{1}{p^{d+1}}\right\}.$$

Further, by our hypothesis $\max\{1,|h|\} = 1$, so Definition 4.2.3 gives

$$d([1:h],[1:g]) = \frac{|g-h|}{\max\{1,|g|\}}.$$

Suppose that $|g| > 1$. Then $|g-h| = |g|$, so $[1:g]$ does not satisfy $|g-h| \leq \frac{1}{p^{d+1}}$. On the other hand, we observe $\max\{1,|g|\} = |g|$. With $|g-h| = |g|$ this implies that

$$d([1:h],[1:g]) = \frac{|g|}{1 \cdot |g|} = 1,$$

so $[1:g] \notin N\left([1:h], \frac{1}{p^{d+1}}\right)$.

We therefore suppose that $|g| \leq 1$ holds. We then have that $d([1:h],[1:g]) = |g-h|$,

and the result is immediate.

□

Lemma 4.3.5. *Suppose that* $[1:h] \in \mathbb{P}^1$ *let* $d \in \mathbb{N}_0$. *If* $|h| > p^d$ *then*

$$N\left([1:h], \frac{1}{p^{d+1}}\right) = \{[1:g] \in \mathbb{P}^1 \ : \ |g| > p^d\}$$

Proof: We first note that as $|h| > p^d$, $h \neq 0$ so $[1:h] = [h^{-1}:1]$, and we may instead consider $N\left([h^{-1}:1], \frac{1}{p^{d+1}}\right)$. Since $|h| > p^d$ we have $|h^{-1}| < \frac{1}{p^d}$, or equivalently $|h^{-1}| \leq \frac{1}{p^{d+1}}$. By Lemma 4.3.4 and Remark 4.3.3 we have that

$$N\left([h^{-1}:1], \frac{1}{p^{d+1}}\right) = \left\{[g:1] \in \mathbb{P}^1 \ : \ |g - h^{-1}| \leq \frac{1}{p^{d+1}}\right\}. \tag{4.3.3}$$

We notice that since $|h^{-1}| \leq \frac{1}{p^{d+1}}$, if $|g| \geq \frac{1}{p^d}$ we have $|g - h^{-1}| = |g| \geq \frac{1}{p^d}$, and so $[g:1]$ is not in this set. We can therefore see that $|g - h^{-1}| \leq \frac{1}{p^{d+1}}$ if and only if $|g| < \frac{1}{p^d}$. This gives that

$$\begin{aligned}\left\{[g:1] \in \mathbb{P}^1 \ : \ |g - h^{-1}| \leq \frac{1}{p^{d+1}}\right\} &= \left\{[g:1] \in \mathbb{P}^1 \ : \ |g| < \frac{1}{p^d}\right\} \\ &= \left\{[1:g^{-1}] \in \mathbb{P}^1 \ : \ |g^{-1}| > p^d\right\} \cup \{[e_2]\}.\end{aligned}$$

Combining this with (4.3.3), we have our result. □

Lemma 4.3.6. *Suppose that* $[1:h] \in \mathbb{P}^1$ *and let* $d \in \mathbb{N}_0$. *If* $1 < |h| \leq p^d$ *then*

$$N\left([1:h], \frac{1}{p^{d+1}}\right) = \{[1:g] \in \mathbb{P}^1 \ : \ |g - h| \leq |h|^2 p^{-(d+1)}.\}$$

Proof: As in the proof of Lemma 4.3.4, we note that by the comments following Remark 4.3.3, we have that $[e_2] \notin N\left([1:h], \frac{1}{p^{d+1}}\right)$ since $|h| \leq p^d$.

Now consider $[1:g] \in \mathbb{P}^1$. We have

$$d([1:h],[1:g]) = \frac{|g-h|}{|h|\max\{1,|g|\}}.$$

First note that if $|g| \leq 1$ then $d([1:h],[1:g]) = 1$, and so $[1:g] \notin N\left([1:h], \frac{1}{p^{d+1}}\right)$. Thus we consider $|g| > 1$, so

$$d([1:h],[1:g]) = \frac{|g-h|}{|h||g|}. \tag{4.3.4}$$

We see that if $|g| \neq |h|$

$$d([1:h],[1:g]) = \frac{\max\{|g|,|h|\}}{|h||g|} = \frac{1}{\min\{|g|,|h|\}}.$$

Since $|h| \leq p^d$ we know $\min\{|g|,|h|\} \leq p^d$ as well, which gives $d([1:h],[1:g]) \geq \frac{1}{p^d}$, a contradiction. Thus $[1:g] \in N\left([1:h], \frac{1}{p^{d+1}}\right)$ implies that $|g| = |h|$ must hold.

On the other hand, suppose that $|g-h| \leq |h|^2 p^{-(d+1)}$. Notice that if $|g| \neq |h|$ then $|g-h| = \max\{|g|,|h|\}$, so in particular, $|h| \leq |g-h|$. Since by assumption we have $|h| \leq p^d$, this gives that if $|g| \neq |h|$ then

$$|h| \leq |g-h| \leq |h|^2 p^{-(d+1)} \leq |h| p^d p^{-(d+1)} = |h|/p$$

which is a contradiction. Thus $|g-h| \leq |h|^2 p^{-(d+1)}$ also implies that $|g| = |h|$ must hold.

Assuming $|g| = |h|$, (4.3.4) implies that

$$d([1:h],[1:g]) \leq \frac{1}{p^{d+1}} \iff \frac{|g-h|}{|h|^2} \leq \frac{1}{p^{d+1}} \iff |g-h| \leq |h|^2 p^{-(d+1)}.$$

□

To end this section, we present a pleasing result about neighbourhoods in $\mathbb{P}^1$.

Proposition 4.3.7. *For each $d \in \mathbb{N}_0$ there exist $p^d(p+1)$ disjoint neighbourhoods of radius $\frac{1}{p^{d+1}}$ such that for any point $[u] \in \mathbb{P}^1$, $N\left([u], \frac{1}{p^{d+1}}\right)$ is precisely one of these neighbourhoods. In particular, these neighbourhoods cover $\mathbb{P}^1$.*

Proof: Fix $d \in \mathbb{N}_0$. We first consider $[1 : h] \in \mathbb{P}^1$ such that $|h| \leq 1$ and $h = h_0 + h_1 x + \ldots$. By Proposition 4.3.2, we have that

$$N\left([1:h], \frac{1}{p^{d+1}}\right) = \left\{[1:g] \ : \ |g-h| \leq \frac{1}{p^{d+1}}\right\}.$$

We see that $|g-h| \leq \frac{1}{p^{d+1}}$ holds precisely when all terms of degree less than x^{d+1} in the Laurent series of $g-h$ are zero, so $g = h_0 + h_1 x + \ldots + h_d x^d + r$ for some $r \in x^{d+1}\mathcal{O}$ must hold. (This statement is made more explicit later in Lemma 5.1.1.)

With this in mind, for each $a_0, \ldots, a_d \in \mathbb{F}_p$ we consider the set

$$S = \left\{[1:g] \ : \ g = a_0 + a_1 x + \ldots + a_d x^d + r, r \in x^{d+1}\mathcal{O}\right\}. \tag{4.3.5}$$

For any point $[1 : h] \in \mathbb{P}^1$ such that $|h| \leq 1$, $N\left([1:h], \frac{1}{p^{d+1}}\right)$ is precisely the set S with $a_0 = h_0, \ldots, a_d = h_d$. Further, we observe that for any element $[1, g] \in S$, $N\left([1:g], \frac{1}{p^{d+1}}\right) = S$. We note there are p^{d+1} distinct choices of $a_0, a_1, \ldots, a_d \in \mathbb{F}_p$, so there are p^{d+1} different sets of the form of (4.3.5).

Now suppose that $[1 : h] \in \mathbb{P}^1$ is such that $1 < |h| \leq p^d$. We then have that $\frac{1}{p^d} \leq |h^{-1}| < 1$. By Remark 4.3.3 we therefore see

$$N\left([1:h], \frac{1}{p^{d+1}}\right) = N\left([h^{-1}:1], \frac{1}{p^{d+1}}\right) = \left\{[g^{-1}:1] \in \mathbb{P}^1 : |g^{-1} - h^{-1}| \leq \frac{1}{p^{d+1}}\right\}.$$

Considering this, we look at the set

$$\left\{[g^{-1}:1] \ : \ g^{-1} = a_1 x + \ldots + a_d x^d + r, r \in x^{d+1}\mathcal{O}\right\} \tag{4.3.6}$$

for each possible choice of $a_1, a_2 \ldots a_d \in \mathbb{F}_p$ such that $a_1, a_2, \ldots, a_d$ are not all zero. We note there are $p^d - 1$ such sets, and, as before, for any point $[1:h] \in \mathbb{P}^1$ such that $1 < |h| \leq p^d$, $N\left([1:h], \frac{1}{p^{d+1}}\right)$ is the set given by (4.3.6) with $a_1 = \tilde{h}_1, \ldots, a_d = \tilde{h}_d$, where $h^{-1} = \sum_{k=1}^{\infty} \tilde{h}_k x^k$.

Lastly, we see by Proposition 4.3.2 that if $[u] \in \mathbb{P}^1$ is such that $[u] = [e_2]$ or $[1:h]$ for some $|h| > p^d$, we have $N\left([u], \frac{1}{p^{d+1}}\right)$ is precisely the set

$$\left\{[1:g] \in \mathbb{P}^1 : |g| > p^d\right\} \cup \{[e_2]\}. \tag{4.3.7}$$

We see for any $[u] \in \mathbb{P}^1$, there exists a set S of the form (4.3.5), (4.3.6), or (4.3.7), such that $[u] \in S$ and $N\left([u], \frac{1}{p^{d+1}}\right) = S$. Further, as there are p^{d+1} neighbourhoods of form (4.3.5), $p^d - 1$ neighbourhoods of form (4.3.6), and one neighbourhood of form (4.3.7), we have $1 + p^{d+1} + (p^d - 1) = p^d(p+1)$ neighbourhoods. That these sets are disjoint is clear from construction. □

Proposition 4.3.7 will be used many times throughout the remainder of this chapter, but is also in itself interesting; for each choice of $d \in \mathbb{N}_0$ our projective space $\mathbb{P}^1$ is partitioned into $p^d(p+1)$ sets, each of which contains points that are within distance $\frac{1}{p^{d+1}}$ from each other, but of distance greater than $\frac{1}{p^{d+1}}$ from points in other sets. Further, we see that we can obtain the neighbourhoods of radius $\frac{1}{p^{d+2}}$ by partitioning each of the previous $p^d(p+1)$ neighbourhoods of radius $\frac{1}{p^{d+1}}$ into p new ones.

For $[u] \in \mathbb{P}^1$ such that $[u] = [e_2]$ or $[u] = [1:h]$ with $|h| \leq 1$ or $|h| > p^d$, the proof of Proposition 4.3.7 also explicitly states what the $\frac{1}{p^{d+1}}$-neighbourhood of $[u]$ should

look like. While it is not needed for what follows, using similar methods to those we used above we can also find the explicit form of the $\frac{1}{p^{d+1}}$-neighbourhood of $[u]$ when $[u] = [1 : h] \in \mathbb{P}^1$ with $1 < |h| \leq p^d$. Namely, suppose that $|h| = p^{d-\ell}$ such that $0 \leq \ell < d$ and $h = h_{-(d-\ell)}x^{-(d-\ell)} + h_{-(d-\ell-1)}x^{-(d-\ell-1)} + \ldots$. It can then be shown that $N\left([u] : \frac{1}{p^{d+1}}\right)$ is

$$\left\{[1 : g] \ : \ g = h_{-(d-\ell)}x^{-(d-\ell)} + \ldots + h_{-(d-2\ell)}x^{-(d-2\ell)} + r, r \in x^{-(d-2\ell)+1}\mathcal{O}\right\}.$$

4.4 Statement of the Free Generators Theorem

We wish to find conditions for which elements A, $B \in \mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ will freely generate a free subgroup of $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$. To do this, we will consider the action of A and B on $\mathbb{P}^1$ by considering the action of the preimages of A and B in $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ on $\mathbb{P}^1$.

To see how a general matrix $M \in \mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ will act on $\mathbb{P}^1$, we notice that if M has eigenvalues that are far apart in absolute value, then M should map $\mathbb{P}^1$ towards the projective class of the eigenvector with the larger corresponding eigenvalue. We note that as every scalar multiple of an eigenvector is also an eigenvector, we can consider the eigenvectors of $M \in \mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ as elements of $\mathbb{P}^1$. Further, the eigenvectors of kM are the same for any $k \in \mathbb{F}_p((x))$, and become the fixed points of the image of M in $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$. For this reason, we will abuse notation slightly, and refer to the fixed points of a matrix $A \in \mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ as the "eigenvectors" of A.

With this in mind, we first investigate what a general form for A and B, in terms of the eigenvectors and eigenvalues of their preimages, will look like.

Lemma 4.4.1. *Let $\tilde{A}$, $\tilde{B} \in \mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$. Suppose that $\tilde{A}$ has distinct eigenvectors $[a : c]$, $[1 : b] \in \mathbb{P}^1$ with corresponding eigenvalues g, $h \in \mathbb{F}_p((x))$ and that $\tilde{B}$ has*

distinct eigenvectors $[1:\tilde{a}]$, $[1:\tilde{b}] \in \mathbb{P}^1$ *with corresponding eigenvalues* $\tilde{g}$, $\tilde{h} \in \mathbb{F}_p((x))$. *Then the respective images of* $\tilde{A}$ *and* $\tilde{B}$ *in* $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ *are*

$$A = \begin{bmatrix} ab - cf & a(f-1) \\ cb(1-f) & abf - c \end{bmatrix} \quad \textit{and} \quad B = \begin{bmatrix} \tilde{b} - \tilde{a}\tilde{f} & \tilde{f} - 1 \\ \tilde{a}\tilde{b}(1-\tilde{f}) & \tilde{b}\tilde{f} - \tilde{a} \end{bmatrix} \tag{4.4.1}$$

where $f = \frac{h}{g}$, $\tilde{f} = \frac{\tilde{h}}{\tilde{g}} \in \mathbb{F}_p((x))$.

Proof: We show this for $\tilde{A}$, as the proof for $\tilde{B}$ is identical. By linear algebra

$$A = \begin{bmatrix} a & 1 \\ c & b \end{bmatrix} \begin{bmatrix} g & 0 \\ 0 & h \end{bmatrix} \begin{bmatrix} b & -1 \\ -c & a \end{bmatrix} \in \mathrm{PGL}_2\big(\mathbb{F}_p((x))\big).$$

Where we note the last matrix has been scaled by $ab - c$, which is nonzero as $[a:c]$ and $[1:b]$ are necessarily distinct.

Since we are working over $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ we note the eigenvalues of $\tilde{A}$ are nonzero, and in particular $g \neq 0$. Thus, up to $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ we can scale by $\frac{1}{g}$, giving

$$A = \begin{bmatrix} a & 1 \\ c & b \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & f \end{bmatrix} \begin{bmatrix} b & -1 \\ -c & a \end{bmatrix} = \begin{bmatrix} ab - cf & a(f-1) \\ cb(1-f) & abf - c \end{bmatrix}$$

where $f = \frac{h}{g}$. □

We note that, by Remark 4.1.1, we can assume four distinct elements of $\mathbb{P}^1$ are of the form $[a:c]$, $[1:b]$, $[1:\tilde{a}]$, and $[1:\tilde{b}]$ for some a, b, c, $\tilde{a}$, $\tilde{b} \in \mathbb{F}_p((x))$. Thus, for elements A and B in $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$, Lemma 4.4.1 is as general as possible.

Lemma 4.4.1 simplifies our main argument greatly; to find conditions for which A, B generate a free subgroup of $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ it is sufficient to consider A and B of

the form in equation (4.4.1). We now state our main theorem, the proof of which is given in Section 4.5.

Theorem 4.4.2 (Free Generators Theorem)**.** *Let p be a prime and let $d \in \mathbb{N}_0$. Suppose there exist $a, b, c, \tilde{a}, \tilde{b} \in \mathbb{F}_p((x))$, $f, \tilde{f} \in \mathbb{F}_p((x))^\times$, such that*

Ξ_1 : $d([u],[v]) > \frac{1}{p^{d+1}}$ *for each pair of* $[u],[v]$ *in* $\big\{[a:c],[1:b],[1:\tilde{a}],[1:\tilde{b}]\big\}$

Ξ_2 : $\min\{|f|,|f^{-1}|\} \leq \frac{1}{p^{2d+1}}$, *and* $\min\{|\tilde{f}|,|\tilde{f}^{-1}| \leq \frac{1}{p^{2d+1}}\}$

Ξ_3 : *There exists* $[z] \in \mathbb{P}^1$ *such that* $d([z],[u]) > \frac{1}{p^{d+1}}$ *for each* $[u]$ *in* $\big\{[a:c],[1:b],[1:\tilde{a}],[1:\tilde{b}]\big\}$.

Then the matrices

$$A = \begin{bmatrix} ab - cf & a(f-1) \\ cb(1-f) & abf - c \end{bmatrix} \text{ and } B = \begin{bmatrix} \tilde{b} - \tilde{a}\tilde{f} & \tilde{f} - 1 \\ \tilde{a}\tilde{b}(1-\tilde{f}) & \tilde{b}\tilde{f} - \tilde{a} \end{bmatrix} \tag{4.4.2}$$

generate a free group in $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$. *In particular, any inverse images of A, B in* $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ *also generate a free group.*

Definition 4.4.3. *Given a prime p, we define $\mathfrak{S}_p$ to be the set of pairs of matrices $S = (\tilde{A}, \tilde{B})$ such that $\tilde{A}$, $\tilde{B} \in \mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$ are preimages of matrices A, B in* $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ *given by Theorem 4.4.2. Further, we will write $\mathfrak{S}$ when p is clear from context.*

Proposition 4.4.4. *Let p be a prime and let $d \in \mathbb{N}_0$, such that $d > 0$ if $p = 2$. Then there exists a, b, c, $\tilde{a}$, $\tilde{b}$, f, $\tilde{f}$ such that conditions Ξ_1 and Ξ_2 of our Free Generators Theorem are satisfied.*

Proof: We first note that to satisfy Ξ_2, we need only take f and $\tilde{f}$ to be sufficiently small elements of $\mathbb{F}_p((x))$. For instance we can take $f = \tilde{f} = x^{2d+1}$.

If $p \neq 2$, we see that any pair of the vectors $[0:1]$, $[1:1]$, $[1:-1]$, and $[1:0]$ are each distance 1 apart, and note $1 > \frac{1}{p^{d+1}}$ for $p > 2$, $d \geq 0$.

If $p = 2$, we see that any pair of the vectors $[0:1]$, $[1:1]$, $[1:x]$, and $[1:0]$ at least a distance of $\frac{1}{2}$ apart, and note $\frac{1}{2} > \frac{1}{p^{d+1}}$ since $p = 2$ and $d > 0$. □

Proposition 4.4.5. *Let p be a prime and let $d \in \mathbb{N}_0$, such that $d > 0$ if $p = 2$ or $p = 3$. Further, let a, b, c, $\tilde{a}$, and $\tilde{b} \in \mathbb{F}_p((x))$ satisfy condition Ξ_1 of Theorem 4.4.2. Then there exists $[z] \in \mathbb{P}^1$ that satisfies condition Ξ_3 of Theorem 4.4.2.*

Proof: By Proposition 4.3.7, we see that each of the $[a:c], [1:b], [1:\tilde{a}]$ and $[1:\tilde{b}]$ must each have $\frac{1}{p^{d+1}}$-neighbourhood equal to one of the neighbourhoods given. If $d = 0$ and $p \geq 5$ we see that $p^d(p+1) = p + 1 > 4$. Similarly, if $d > 0$ we see $p^d(p+1) > 4$ for any prime p. Therefore in each case we know by Proposition 4.3.7 that there exists at least one non-empty $\frac{1}{p^{d+1}}$ neighbourhood, N, disjoint from the $\frac{1}{p^{d+1}}$-neighbourhoods of $[a:c], [1:b], [1:\tilde{a}]$ and $[1:\tilde{b}]$, and we can take $[z]$ to be any element in N. □

Remark 4.4.6. We note that because of Proposition 4.3.7, condition Ξ_1 implies that the $\frac{1}{p^{d+1}}$-neighbourhoods are necessarily distinct. This will be important for the proof of the Free Generators Theorem.

Remark 4.4.7. We note that we cannot choose $d = 0$ when $p = 2$ as it would be impossible to satisfy condition Ξ_1, since if $d = 0$ there are only 3 distinct $\frac{1}{p^{d+1}}$-neighbourhoods in $\mathbb{P}^1$.

In Section 4.7, we will show in the case $p = 3$ and $d = 0$ such a point $[z]$ does not exist, and that in this case conditions Ξ_1 and Ξ_2 of the Free Generators Theorem are sufficient for A and B to generate a free subgroup of $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ as desired.

Remark 4.4.8. Note that as a result of Proposition 4.3.2, condition Ξ_1 implies that at most one of b, $\tilde{a}$, $\tilde{b}$ can have absolute value greater than p^d, and none can if $|c| > p^d|a|$.

4.5 Proof of the Free Generators Theorem

The proof of Theorem 4.4.2 will use the following proposition owing to Jacques Tits [51, Prop 1.1], known as the Ping-Pong Lemma.

Proposition 4.5.1 (Ping-Pong Lemma)**.** *Let P be a set, I an index set, G a group acting on P, $(G_i)_{i\in I}$ a family of subgroups generating G, $(P_i)_{i\in I}$ a family of subsets of P and $[z]$ a point of $P \setminus \bigcup_{i\in I} P_i$. Assume that for all $i, j \in I$ with $i \neq j$ and all $g \in G_i \setminus \{1\}$, one has $g(P_j \cup \{[z]\}) \subset P_i$. Then G is the free product of the subgroups G_i $(i \in I)$.*

Here we will take $P = \mathbb{P}^1$, $I = \{A, B\}$, $G_A = \langle A\rangle$ and $G_B = \langle B\rangle$, and G to be the subgroup of $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ generated by G_A and G_B.

Let $\tilde{A}$ and $\tilde{B}$ be respective preimages of A and B in $\mathrm{GL}_2\big(\mathbb{F}_q\big)$. In Lemma 4.4.1 we saw that $\tilde{A}$ has eigenvectors $[a : c]$ and $[1 : b]$ with (up to a scalar multiple) respective eigenvalues g and $h \in \mathbb{F}_p((x))$ such that $\frac{h}{g} = f$. Correspondingly, $\tilde{B}$ has eigenvectors $[1 : \tilde{a}]$ and $[1 : \tilde{b}]$ with respective eigenvalues $\tilde{g}$ and $\tilde{h} \in \mathbb{F}_p((x))$ such that $\frac{\tilde{h}}{\tilde{g}} = \tilde{f}$.

By condition Ξ_2 we know that either $|f|$ or $|f^{-1}| \leq \frac{1}{p^{2d+1}}$, and $|\tilde{f}|$ or $|\tilde{f}^{-1}| \leq \frac{1}{p^{2d+1}}$. For our argument we will assume that $|f|$, $|\tilde{f}| \leq \frac{1}{p^{2d+1}}$. To see that we are able to make this assumption, notice that if A and B generate a free group in $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$, then so do A^{-1} and B. Further, notice that by the construction in Lemma 4.4.1, A^{-1} is obtained from A by replacing f with f^{-1}.

As we are assuming the case that $|f| \leq \frac{1}{p^{2d+1}}$, we have that the eigenvalue corresponding to $[a : c]$ is large in absolute value compared to the corresponding eigenvalue of $[1 : b]$, and so A will map elements of $\mathbb{P}^1$ towards $[a : c]$. An analogous observation can be made for B, as we are assuming $|\tilde{f}| \leq \frac{1}{p^{2d+1}}$. That is, B will map elements of $\mathbb{P}^1$ towards $[1 : \tilde{b}]$.

With this in mind we will define

$$N_{[a:c]} = N\left([a : c], \frac{1}{p^{d+1}}\right) \quad \text{and} \quad N_{[1:b]} = N\left([1 : b], \frac{1}{p^{d+1}}\right).$$

Similarly, we will define

$$N_{[1:\tilde{a}]} = N\left([1 : \tilde{a}], \frac{1}{p^{d+1}}\right) \quad \text{and} \quad N_{[1:\tilde{b}]} = N\left([1 : \tilde{b}], \frac{1}{p^{d+1}}\right).$$

With this intuition, we choose

$$P_A = N_{[a:c]} \cup N_{[1:b]} \quad \text{and} \quad P_B = N_{[1:\tilde{a}]} \cup N_{[1:\tilde{b}]}. \tag{4.5.1}$$

We also need a point $[z] \in \mathbb{P}^1$ such that $[z] \in \mathbb{P}^1 \setminus (P_A \cup P_B)$. The existence of such a point is guaranteed by condition Ξ_3 of Theorem 4.4.2.

Figure 4.1 allows us to visualize the neighbourhoods composing P_A and P_B as subsets of $\mathbb{P}^1$; however, we caution this image is of $\mathbb{P}^1(\mathbb{R})$ and not of our non-Archimedian setting.

To show that the conditions of Proposition 4.5.1 are satisfied, and thus prove Theorem 4.4.2, we need to show that for all $g \in G_A \setminus \{1\}$, $g(P_B \cup \{[z]\}) \subset P_A$ and for all $h \in G_B \setminus \{1\}$, $h(P_A \cup \{[z]\}) \subset P_B$. We notice that if we replace $[a : c]$ with $[1 : \tilde{a}]$, $[1 : b]$ with $[1 : \tilde{b}]$, and f with $\tilde{f}$, we obtain B from A. As B has the same form of A,

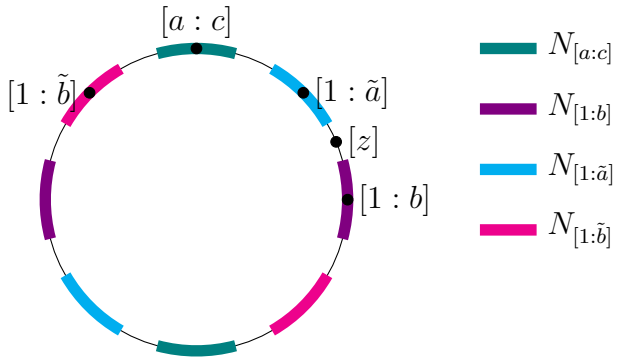

Figure 4.1: A visual representation of the $\frac{1}{p^{d+1}}$-neighbourhoods of the eigenvectors of A and B and the point $[z] \in \mathbb{P}^1$. To satisfy conditions Ξ_1 and Ξ_2 of Theorem 4.4.2 these neigbourhoods must be disjoint and the point $[z]$ must lie outside each neighbourhood.

we thus need only show that for any $g \in G_A \setminus \{1\}$ we have $g(P_B \cup \{[z]\}) \subset P_A$.

We will show something stronger. More specifically, we will show in Lemma 4.6.1 and Proposition 4.6.3 that

$$A\left(\mathbb{P}^1 \setminus N_{[1:b]}\right) \subseteq N_{[a:c]} \tag{4.5.2}$$

and in Corollary 4.6.4 that

$$A^{-1}\left(\mathbb{P}^1 \setminus N_{[a:c]}\right) \subseteq N_{[1:b]}. \tag{4.5.3}$$

These mappings are shown in Figure 4.2.

Assume for now that (4.5.2) and (4.5.3) hold; their proofs are deferred to Section 4.6. We now show that equations (4.5.2) and (4.5.3) are sufficient to prove that for any $g \in G_A \setminus \{1\}$ we have $g(P_B \cup \{[z]\}) \subset P_A$, which will complete the proof of Theorem 4.4.2.

Suppose that $g \in G_A \setminus \{1\}$. Then either $g = A^k$ or A^{-k} for some $k \in \mathbb{N}$. Since the proof is identical, we assume without loss of generality that $g = A^k$. We proceed by induction on k.

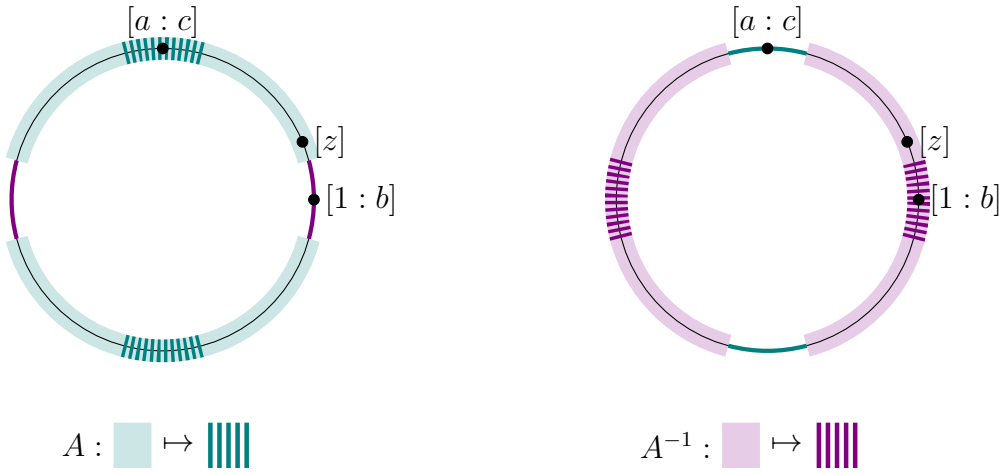

Figure 4.2: A visual representation of the action of A (left figure) and the action of A^{-1} (right figure) on $\mathbb{P}^1$. We see that A maps $\mathbb{P}^1 \setminus N_{[1:b]}$ to $N_{[a:c]}$ and that A^{-1} maps $\mathbb{P}^1 \setminus N_{[a:c]}$ to $N_{[1:b]}$.

Note that by condition Ξ_3, we have that $[z] \in \mathbb{P}^1 \setminus N_{[1:b]}$. As well, we recall from Remark 4.4.6 that condition Ξ_1 implies that $P_B = N_{[1:\tilde{a}]} \cup N_{[1:\tilde{b}]} \subseteq \mathbb{P}^1 \setminus N_{[1:b]}$. By (4.5.2) we have that A maps $\mathbb{P}^1 \setminus N_{[1:b]}$ into $N_{[a:c]}$. Thus we have $A(P_B \cup \{[z]\}) \subset N_{[a:c]}$.

Now suppose that $A^{k-1}(P_B \cup \{[z]\}) \subset N_{[a:c]}$. Again, from Remark 4.4.6, condition Ξ_1 of Theorem 4.4.2 implies that $N_{[a:c]} \subset \mathbb{P}^1 \setminus N_{[1:b]}$, so $A^{k-1}(P_B \cup \{[z]\}) \subset \mathbb{P}^1 \setminus N_{[1:b]}$. Thus by (4.5.2) we have $A^k(P_B \cup \{[z]\}) = A(A^{k-1}(P_B \cup \{[z]\})) \subset N_{[a:c]} \subset P_A$, as required.

4.6 Proofs of (4.5.2) and (4.5.3)

In this section we will show that $A\left(\mathbb{P}^1 \setminus N_{[1:b]}\right) \subseteq N_{[a:c]}$ and $A^{-1}\left(\mathbb{P}^1 \setminus N_{[a:c]}\right) \subseteq N_{[1:b]}$ hold, thus completing the proof of the Free Generators Theorem. We first note by Proposition 4.3.2 the point $[e_2]$ is in $\mathbb{P}^1 \setminus N_{[1:b]}$ precisely when $|b| \leq p^d$. With this in mind we have the following lemma.

Lemma 4.6.1. *Suppose* $A \in \mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ *is as in Theorem 4.4.2 such that* $|b| \leq p^d$. *Then* $A \cdot [e_2] \in N_{[a:c]}$.

Proof: We see

$$A \cdot [e_2] = \begin{bmatrix} ab - cf & a(f-1) \\ cb(1-f) & abf - c \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a(f-1) \\ abf - c \end{bmatrix}.$$

We have, noting that $|f| \leq \frac{1}{p^{2d+1}} < 1$,

$$\begin{aligned} d([a:c], A \cdot [e_2]) &= d([a:c], [a(f-1) : abf - c]) \\ &= \frac{|a(abf-c) - ca(f-1)|}{\max\{|a|,|c|\}\max\{|a(f-1)|, |abf-c|\}} \\ &= \frac{|af(ab-c)|}{\max\{|a|,|c|\}\max\{|a|, |abf-c|\}} \\ &\leq \frac{|a||f|\max\{|ab|,|c|\}}{\max\{|a|,|c|\}\max\{|a|,|abf-c|\}}. \end{aligned}$$

Since $|b| \leq p^d$ and $|f| \leq \frac{1}{p^{2d+1}}$ we know $|bf| \leq \frac{1}{p^{d+1}}$. This implies that $|abf| < |a|$. Thus $\max\{|a|, |abf - c|\} = \max\{|a|, |c|\}$. So

$$d([a:c], A \cdot [e_2]) \leq \frac{|a||f|\max\{|ab|,|c|\}}{\max\{|a|^2, |c|^2\}}.$$

Recalling that $|bf| \leq \frac{1}{p^{d+1}}$, we see

$$|a||f||ab| \leq |a|^2|bf| \leq \frac{\max\{|a|^2,|c|^2\}}{p^{d+1}}$$

and recalling $|f| \leq \frac{1}{p^{2d+1}}$ we have

$$|a||f||c| \leq \frac{\max\{|a|^2,|c|^2\}}{p^{2d+1}}.$$

Thus in either case $d([a:c], A \cdot [e_2]) \leq \frac{1}{p^{d+1}}$. □

In order to consider where A maps general elements of $\mathbb{P}^1 \setminus N\left([1:b], \frac{1}{p^{d+1}}\right)$ we will need the following lemma.

Lemma 4.6.2. *Suppose* $A \in \mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ *is as in Theorem 4.4.2, and* $[1:g] \in \mathbb{P}^1 \setminus N_{[1:b]}$. *Then*

$$\max\{|f(ag-c)-a(g-b)|, |bf(ag-c)-c(g-b)|\} = \max\{|a|,|c|\}|g-b|. \quad (4.6.1)$$

Proof: Suppose b is fixed and $[1:g] \in \mathbb{P}^1 \setminus N\left([1:b], \frac{1}{p^{d+1}}\right)$. Proposition 4.3.2 gives us that g must satisfy the following.

(A) If $|b| \leq 1$ then $|g-b| \geq \frac{1}{p^d}$.

(B) If $1 < |b| \leq p^d$ then either $|g| \neq |b|$ or $|g-b| \geq |b|^2 p^{-d}$.

(C) If $|b| > p^d$ then $|g| \leq p^d$.

To prove our claim, we will first show that each of $|fg|$, $|f|$, $|bfg|$, $|bf|$ is strictly smaller than $|g-b|$. We consider the three cases above. We note that we frequently will use the fact that $|f| \leq p^{-2d-1} < p^{-d}$.

(A) Suppose that $|b| \leq 1$, so $|g-b| \geq \frac{1}{p^d}$. Since $|b| \leq 1$, we know $|bfg| \leq |fg|$ and $|bf| \leq |f|$. Further, we have $|f| < p^{-d} \leq |g-b|$, so $|f| < |g-b|$. We see that if $|g| \leq |b|$ then $|g| \leq 1$, since $|b| \leq 1$, so $|fg| \leq |f| < |g-b|$. If $|g| > |b|$, then we have $|fg| < |g| = |(g-b)|$, where the first equality holds since $|f| < 1$.

(B) Now suppose that $1 < |b| \leq p^d$. Since $|b| > 1$, we know $|fg| < |bfg|$ and $|f| < |bf|$. Note that because $|f| < \frac{1}{p^d}$ and $|b| < p^d$, we have $|bf| < 1$. If $|g| \neq |b|$ we see that $|g| \leq |g-b| = \max\{|g|,|b|\}$, so $|bfg| < |g| \leq |g-b|$ since $|bf| < 1$. As well, since $|b| > 1$, if $g \neq b$ then $|bf| < 1 < |g-b|$. If $|g| = |b|$ we note that $|g-b| \geq |b|^2 p^{-d}$ must hold, and, recalling that $|f| < p^{d-d}$, see $|bfg| = |b|^2|f| < |b|^2 p^{-d} \leq |g-b|$.

(C) Lastly, suppose that $|b| > p^d$, so $|g| \leq p^d$. Then again we have $|fg| < |bfg|$ and $|f| < |bf|$ since $|b| > 1$. As $|g| \leq p^d$, we have that $|g-b| = |b|$. Since $|f| < 1$ we know $|bf| < |b|$. Finally, as $|g| \leq p^d$ and $|f| < p^{-d}$ we have $|fg| < 1$, so $|bfg| < |b|$.

We now know that each of $|fg|$, $|f|$, $|bfg|$, $|bf|$ is strictly smaller than $|g-b|$, and will use this show that (4.6.1) holds. We first notice that if $|f(ag-c)-a(g-b)| = |a(g-b)|$ and $|bf(ag-c)-c(g-b)| = |c(g-b)|$ both hold, we are done. We thus consider when $|f(ag-c)-a(g-b)| \neq |a(g-b)|$, and note that for $|bf(ag-c)-c(g-b)| \neq |c(g-b)|$, the proof is identical.

Suppose $|f(ag-c)-a(g-b)| \neq |a(g-b)|$. Since $|afg| < |a(g-b)|$, we see that $|a(g-b)| \leq |fc|$ must hold. Since $|f| < |g-b|$, we know $|fc| < |c(g-b)|$. This implies that $|afg| < |a(g-b)| \leq |fc| < |c(g-b)|$, so $|f(ag-c)-a(g-b)| \leq |c(g-b)|$.

As well, we have that $|bf| < |g-b|$, thus $|cbf| < |c(g-b)|$. Since $|bfg| < |g-b|$, we have $|abfg| < |a(g-b)| < |c(g-b)|$. This implies that $|bf(ag-c)-c(g-b)| = |c(g-b)|$. □

Proposition 4.6.3. *Suppose $A \in \mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ is as in Theorem 4.4.2, and that $[1:g] \in \mathbb{P}^1 \setminus N_{[1:b]}$. Then $A \cdot [1:g] \in N_{[a:c]}$.*

Proof: We have

$$\begin{aligned} A \cdot [1:g] &= \begin{bmatrix} ab-cf & a(f-1) \\ cb(1-f) & abf-c \end{bmatrix} \begin{bmatrix} 1 \\ g \end{bmatrix} \\ &= \begin{bmatrix} f(ag-c)-a(g-b) \\ bf(ag-c)-c(g-b) \end{bmatrix}. \end{aligned}$$

So $d([a:c], A \cdot [1:g])$

$$= \frac{|a(bf(ag-c)-c(g-b))-c(f(ag-c)-a(g-b))|}{\max\{|a|,|c|\}\max\{|f(ag-c)-a(g-b)|, |bf(ag-c)-c(g-b)|\}}$$

$$= \frac{|(ab-c)f(ag-c)|}{\max\{|a|,|c|\}\max\{|f(ag-c)-a(g-b)|,|bf(ag-c)-c(g-b)|\}}. \quad (4.6.2)$$

By Lemma 4.6.2, we have that the denominator of (4.6.2) satisfies

$$\max\{|a|,|c|\}\max\{|f(ag-c)-a(g-b)|,|bf(ag-c)-c(g-b)|\} = \max\{|a|^2,|c|^2\}|g-b|.$$

Therefore we have that

$$d([a:c],A\cdot[1:g]) = \frac{|(ab-c)f(ag-c)|}{\max\{|a|^2,|c|^2\}|g-b|}.$$

Using the non-Archimedean triangle inequality, we see

$$|ab-c| \leq \max\{|ab|,|c|\} \leq \max\{|a|,|c|\}\max\{|1|,|b|\}.$$

Similarly, $|ag-c| \leq \max\{|a|,|c|\}\max\{|1|,|g|\}$.

Thus

$$d([a:c],A\cdot[1:g]) \leq \frac{|f|\max\{1,|g|\}\max\{1,|b|\}}{|g-b|}.$$

Finally, we notice that

$$\frac{\max\{1,|g|\}\max\{1,|b|\}}{|g-b|} = \left(d\left([1:b],[1:g]\right)\right)^{-1}.$$

By our hypothesis $d([1:b],[1:g]) > \frac{1}{p^{d+1}}$ so $\left(d\left([1:b],[1:g]\right)\right)^{-1} \leq p^d$. Therefore

$$d([a:c],A\cdot[1:g]) \leq |f|p^d \leq p^{d-(2d+1)} = \frac{1}{p^{d+1}}$$

as required. □

Recall we want to see where elements of $G_A = \langle A \rangle$ send $\mathbb{P}^1 \setminus P_A$ and $[z]$. We have shown in Lemma 4.6.1 and Proposition 4.6.3 that A sends elements of $\mathbb{P}^1 \setminus N_{[1:b]}$ to $N_{[a:c]}$, which will tell us information about A^k for $k \in \mathbb{N}$. We also need information about elements in G_A of the form A^{-k}.

Corollary 4.6.4. *Suppose A is as in Theorem 4.4.2. Then $A^{-1}\big(\mathbb{P}^1 \setminus N_{[a:c]}\big) \subset N_{[1:b]}$.*

Proof: Suppose that $A^{-1}\big(\mathbb{P}^1 \setminus N_{[a:c]}\big) \not\subset N_{[1:b]}$. Then there exists an element $[u] \in \mathbb{P}^1 \setminus N_{[a:c]}$ such that $A^{-1} \cdot [u] \in \mathbb{P}^1 \setminus N_{[1:b]}$. Since $A^{-1} \cdot [u] \in \mathbb{P}^1 \setminus N_{[1:b]}$, by Lemma 4.6.1 and Proposition 4.6.3 we have $A \cdot (A^{-1} \cdot [u]) = [u] \in N_{[a:c]}$. As we know $[u] \in \mathbb{P}^1 \setminus N_{[a:c]}$, this is a contradiction. □

4.7 Free Generators Theorem in the case $p = 3$

Suppose in the Free Generators Theorem we wish to take $p = 3$ and $|f| = \frac{1}{p}$ or $|f| = \frac{1}{p^2}$ for instance, so by Condition Ξ_2 of Theorem 4.4.2 we must necessarily take $d = 0$. We know that $N_{[a:c]}$, $N_{[1:b]}$, $N_{[1:\tilde{a}]}$, and $N_{[1:\tilde{b}]}$ must be precisely the four possible distinct neighbourhoods guarenteed by Proposition 4.3.7. Further, by Proposition 4.3.7 we know that these neighbourhoods cover $\mathbb{P}^1$, that is

$$N_{[a:c]} \cup N_{[a:b]} \cup N_{[1:\tilde{a}]} \cup N_{[1:\tilde{b}]} = \mathbb{P}^1.$$

Therefore it is impossible to find a point $[z] \in \mathbb{P}^1$ that satisfies condition Ξ_3 of Theorem 4.4.2.

Since no point $[z] \in \mathbb{P}^1$ exists in this special case with $d = 0$ and $p = 3$, we are not able to use the Ping-Pong Lemma. However, we have now proved Theorem 4.4.2 for the case $p = 3$ when $d = 1$, and we will use this to show Theorem 4.4.2 holds without

condition Ξ_3 for the case $p = 3$ and $d = 0$. The idea will be to map the matrices A and B in Theorem 4.4.2 to ones that satisfy the conditions of Theorem 4.4.2 with $d = 1$. We begin with two lemmas.

Lemma 4.7.1. *The map*

$$\begin{aligned} \Phi : \mathrm{GL}_2\big(\mathbb{F}_3((x))\big) &\to \mathrm{GL}_2\big(\mathbb{F}_3((x))\big) \\ f(x) &\to f(x^3) \end{aligned}$$

is an injective homomorphism.

Proof: To see that the map is injective, it suffices to note that the analogous map

$$\begin{aligned} \phi : \mathbb{F}_3((x)) &\to \mathbb{F}_3((x)) \\ f(x) &\to f(x^3) \end{aligned}$$

is a field homomorphism, and thus necessarily injective. □

The map ϕ will be used frequently in what follows, and in particular the relation that for any $h \in \mathbb{F}_3((x))$, we have

$$|\phi(h)| = |h|^3. \tag{4.7.1}$$

Lemma 4.7.2. *Suppose that* $[u] = [u_1 : u_2]$ *and* $[v] = [v_1 : v_2] \in \mathbb{P}^1$ *are such that* $d([u], [v]) = 1$. *Then*

$$d([\phi(u_1) : \phi(u_2)], [\phi(v_1) : \phi(v_2)]) = 1.$$

Proof: With (4.7.1), we see that

$$d([\phi(u_1) : \phi(u_2)], [\phi(v_1) : \phi(v_2)]) = \frac{|\phi(u_1v_2 - u_2v_2)|}{\max\{|\phi(u_1)|, |\phi(u_2)|\} \max\{|\phi(v_1)|, |\phi(v_2)|\}}$$

$$\begin{aligned} &= \frac{|u_1v_2 - u_2v_2|^3}{\max\{|u_1|^3, |u_2|^3\}\max\{|v_1|^3, |v_2|^3\}} \\ &= d([u],[v])^3 = 1^3 = 1, \end{aligned}$$

noting that the last line is by our hypothesis. □

Proposition 4.7.3. *Suppose that a, b, c, $\tilde{a}$, $\tilde{b}$, f and $\tilde{f}$ satisfy conditions Ξ_1 and Ξ_2 of Theorem 4.4.2 with $p = 3$, $d = 0$, and A and B are as given in Theorem 4.4.2. Then A and B generate a free group in $\mathrm{PGL}_2\big(\mathbb{F}_3((x))\big)$.*

Proof: We first observe that since $p = 3$ and $d = 0$, condition Ξ_1 and Lemma 4.2.7 imply that $d([u],[v]) = 1 > \frac{1}{3}$ for any pair $[u]$, $[v]$ in $\Big\{[a:c],[1:b],[1:\tilde{a}],[1:\tilde{b}]\Big\}$. Thus, by Lemma 4.7.2, we see that $d([u],[v]) = 1$ for any pair $[u]$, $[v]$ in

$$\Big\{[\phi(a):\phi(c)],[1:\phi(b)],[1:\phi(\tilde{a})],[1:\phi(\tilde{b})]\Big\}.$$

So the image of our eigenvectors, under the homomorphism ϕ, satisfy Ξ_1 for $d \geq 0$.

Further, since $d = 0$, condition Ξ_2 implies that $|f| \leq \frac{1}{3}$ and $|\tilde{f}| \leq \frac{1}{3}$. Consequently, with (4.7.1), we have that $|\phi(f)| = |\frac{1}{3}|^3 = \frac{1}{3^3}$ and $|\phi(\tilde{f})| = |\frac{1}{3}|^3 = \frac{1}{3^3}$. Therefore $\phi(f)$ and $\phi(\tilde{f})$ satisfy condition Ξ_2 of Theorem 4.4.2 with $d = 1$.

Finally, condition Ξ_3 is satisfied with $d = 1$ by Proposition 4.4.5.

Therefore $\phi(a)$, $\phi(b)$, $\phi(c)$, $\phi(\tilde{a})$, $\phi(\tilde{b})$, $\phi(f)$, and $\phi(\tilde{f})$ satisfy the conditions of Theorem 4.4.2 with $d = 1$. As well, the matrices given in Theorem 4.4.2 for these choices are precisely $\Phi(A)$ and $\Phi(B)$. Consequently, by Theorem 4.4.2, $\Phi(A)$ and $\Phi(B)$ generate a free group in $\mathrm{PGL}_2\big(\mathbb{F}_3((x))\big)$.

Since Φ is an injective homomorphism, any relation in A and B is necessarily a relation in $\Phi(A)$ and $\Phi(B)$. Thus since $\Phi(A)$ and $\Phi(B)$ generate a free group in $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$, it is the case that A and B must do so too. □

Remark 4.7.4. We note we will append to $\mathfrak{S}$ (as defined in 4.4.3) choices of (A, B) that satisfy Proposition 4.7.3. For the point of referring back to it, we will also consider Proposition 4.7.3 as part of the Free Generators Theorem.

Lastly, we conclude this section with a lemma that helps us visualize the neighbourhoods in the case $p = 3$ and $d = 0$.

Lemma 4.7.5. *Suppose $p = 3$ and $d = 0$, and that a, b, c, $\tilde{a}$, $\tilde{b} \in \mathbb{F}_3((x))$ satisfy condition Ξ_1 of Theorem 4.4.2 . Then either*

$$\left\{[a : c], \cdot[1 : b], \cdot[1 : \tilde{a}], \cdot[1 : \tilde{b}]\right\} = \{[0 : 1], [1 : r], [1 : 1 + s], [1 : 2 + t]\} \tag{4.7.2}$$

or

$$\left\{[a : c], \cdot[1 : b], \cdot[1 : \tilde{a}], \cdot[1 : \tilde{b}]\right\} = \{[1 : u], [1 : r], [1 : 1 + s], [1 : 2 + t]\} \tag{4.7.3}$$

for some r, s, t, $u \in \mathbb{F}_p((x))$ such that $|r|$, $|s|$, $|t| < 1$ and $|u| > 1$.

Proof: Condition Ξ_1 implies that the $\frac{1}{p^{d+1}}$-neighbourhoods of the eigenvectors, that is $N_{[a:c]}$, $N_{[1:b]}$, $N_{[1:\tilde{a}]}$, and $N_{[1:\tilde{b}]}$, must be disjoint. By Proposition 4.3.7, we know that there are precisely $3^0(3 + 1) = 4$ possible distinct neighbourhoods of radius $\frac{1}{p^{d+1}}$ in $\mathbb{P}^1$, and therefore these must correspond to $N_{[a:c]}$, $N_{[1:b]}$, $N_{[1:\tilde{a}]}$, and $N_{[1:\tilde{b}]}$. From equations (4.3.5) and (4.3.7) in the proof of Proposition 4.3.7 we see that our elements must be one of the forms given in (4.7.2) or (4.7.3). □

Chapter 5

Constructing Hash Functions

In the last chapter, we sought pairs of matrices over $\mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$ that were free generators of a free subgroup of $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$, and whose images in $\mathrm{GL}_2\big(\mathbb{F}_q\big)$ would serve as an alternative option for generators of the Zémor-Tillich hash function. Recalling that the preimages of any pair of matrices generating a free subgroup of $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ also generated a free subgroup of $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$, the Free Generators Theorem provides an extensive choice of such generators.

We recall $\mathfrak{S}$ is the set of pairs of generators produced by the Free Generators Theorem, with entries in $\mathbb{F}_p[x]$, as in Definition 4.4.3 and the later amendments in Remark 4.7.4. In this chapter we will formally show many choices of polynomial generators in $\mathfrak{S}$ satisfy Property 2.3.1. Further, we analyse security properties of the corresponding hash functions of generators given by the Free Generators Theorem. We will first consider some particular choices of polynomial generators. In what follows, we will mainly consider the case that p is odd.

We highlight that in the last chapter we used the notation A and B for matrices in $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ given by the Free Generators Theorem. In this chapter, we will

work primarily over $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$, and thus use the notation A and B for elements of $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$. For the class of A and B in $\mathrm{PGL}_2\big(\mathbb{F}_q\big)$ we will respectively write $[A]$ and $[B]$ unless stated otherwise.

5.1 Polynomial generators

Since in practice we would like to work with matrices in $\mathrm{GL}_2\big(\mathbb{F}_q\big)$, it makes sense to choose A and B so that they have polynomial entries over $\mathbb{F}_p$. We note that for A, $B \in \mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ with polynomial entries, if A and B generate a free subgroup in $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ they must generate a free monoid in $\mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$.

For $[1:h] \in \mathbb{P}^1$ such that $|h| \leq 1$ the condition in (4.3.1) that $|g-h| \leq \frac{1}{p^{d+1}}$ can be seen more intuitively in terms of congruences modulo powers of x, which we state as a lemma below.

Lemma 5.1.1. *For arbitrary $h, g \in \mathbb{F}_p((x))$ such that $h \in \mathcal{O}$ and $d \in \mathbb{N}_0$ we have $d([1:h],[1:g]) \leq \frac{1}{p^{d+1}}$ if and only if $g \equiv h \mod x^{d+1}$.*

Proof: By Proposition 4.3.2 we have that $d([1:h],[1:g]) \leq \frac{1}{p^{d+1}}$ if and only if $|g-h| \leq \frac{1}{p^{d+1}}$. By the definition of the norm, this occurs if and only if the smallest nonzero term of $g-h$ is of degree $d+1$ or greater, or equivalently when g and h are congruent modulo x^{d+1}. □

Remark 5.1.2. Suppose that our choices of b, $\tilde{a}$, and $\tilde{b}$ for Theorem 4.4.2 are in $\mathbb{F}_p[x]$. If $[a:c] = [0:1]$ then condition Ξ_1 holds if and only if no pair in $\left\{b, \tilde{a}, \tilde{b}\right\}$ is congruent mod x^{d+1}. Further, if $[a:c] = [1:c]$ then condition Ξ_1 holds if and only if no pair in $\left\{b, c, \tilde{a}, \tilde{b}\right\}$ is congruent mod x^{d+1}.

We also note that $f \in \mathbb{F}_p[x]$ satisfies condition Ξ_2 of the Free Generators Theorem if

and only if $f \equiv 0 \bmod x^{2d+1}$.

For p an odd prime, we now consider what some polynomial choices of $(A, B) \in \mathfrak{S}$ look like for some simple choices of eigenvectors $[a : c]$, $[1 : b]$, $[1 : \tilde{a}]$, and $[1 : \tilde{b}]$. Namely, we set $[a : c] = [0 : 1]$ and take $[1 : b]$, $[1 : \tilde{a}]$, and $[1 : \tilde{b}]$ to be $[1 : 0]$, $[1 : 1]$, and $[1 : -1]$ in some order. These possible choices of $[1 : b]$, $[1 : \tilde{a}]$, and $[1 : \tilde{b}]$ give us 6 different pairs (A, B), and are shown in Table 5.1. We let $G_1(f, \tilde{f}), \ldots, G_6(f, \tilde{f})$ be these choices of $\{A, B\}$.

Using Definition 4.2.3, we see that the elements $[0 : 1]$, $[1 : 0]$, $[1 : 1]$, and $[1 : -1] \in \mathbb{P}^1$ are all distance 1 apart, and so satisfy condition Ξ_1 of the Free Generators Theorem for any choice of $d \geq 0$.

To allow the most freedom in satisfying condition Ξ_2, we will choose $d = 0$. With this choice, any $f, \tilde{f} \in \mathbb{F}_p[x]$ such that f and $\tilde{f}$ have a zero constant term will satisfy condition Ξ_2.

For $p = 3$, the choices above therefore satisfy the conditions of Proposition 4.7.3. For $p > 3$, condition Ξ_3 is satisfied by Proposition 4.4.5, and so these choices satisfy the conditions of the Free Generators Theorem. We thus see that any of the pairs of matrices (A, B) in Table 5.1 generate a free subgroup in $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ for p an odd prime. We state this as Corollary 5.1.3 below.

$\{A, B\}$	A	B	$[a : c]$	$[1 : b]$	$[1 : \tilde{a}]$	$[1 : \tilde{b}]$
$G_1(f, \tilde{f})$	$\begin{pmatrix} f & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} \tilde{f}+1 & 1-\tilde{f} \\ 1-\tilde{f} & \tilde{f}+1 \end{pmatrix}$	$[0 : 1]$	$[1 : 0]$	$[1 : 1]$	$[1 : -1]$
$G_2(f, \tilde{f})$	$\begin{pmatrix} f & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} \tilde{f}+1 & \tilde{f}-1 \\ \tilde{f}-1 & \tilde{f}+1 \end{pmatrix}$	$[0 : 1]$	$[1 : 0]$	$[1 : -1]$	$[1 : 1]$
$G_3(f, \tilde{f})$	$\begin{pmatrix} f & 0 \\ f-1 & 1 \end{pmatrix}$	$\begin{pmatrix} \tilde{f} & \tilde{f}-1 \\ 0 & 1 \end{pmatrix}$	$[0 : 1]$	$[1 : 1]$	$[1 : -1]$	$[1 : 0]$
$G_4(f, \tilde{f})$	$\begin{pmatrix} f & 0 \\ 1-f & 1 \end{pmatrix}$	$\begin{pmatrix} \tilde{f} & 1-\tilde{f} \\ 0 & 1 \end{pmatrix}$	$[0 : 1]$	$[1 : -1]$	$[1 : 1]$	$[1 : 0]$
$G_5(f, \tilde{f})$	$\begin{pmatrix} f & 0 \\ f-1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1-\tilde{f} \\ 0 & \tilde{f} \end{pmatrix}$	$[0 : 1]$	$[1 : -1]$	$[1 : 0]$	$[1 : 1]$
$G_6(f, \tilde{f})$	$\begin{pmatrix} f & 0 \\ 1-f & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & \tilde{f}-1 \\ 0 & \tilde{f} \end{pmatrix}$	$[0 : 1]$	$[1 : 1]$	$[1 : 0]$	$[1 : -1]$

Table 5.1: The matrices A and B given by Theorem 4.4.2 for given choices of $[a : c]$, $[1 : b]$, $[1 : \tilde{a}]$, $[1 : \tilde{b}]$ with $d = 0$.

Note that when the choices of f and $\tilde{f}$ are not specified, or are irrelevant, we will write G_i for $G_i(f, \tilde{f})$ for convenience.

Corollary 5.1.3. *Let p be an odd prime, and $f, \tilde{f} \in \mathbb{F}_p[x]$ such that $|f|, |\tilde{f}| \leq 1/p$. Then for each for $1 \leq i \leq 6$ the pairs of matrices $G_i(f, \tilde{f})$ given in Table 5.1 generate a free group in $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$. In particular, $G_i(f, \tilde{f})$ also generates a free monoid in $\mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$.*

We notice that the choices in Table 5.1 do not satisfy the conditions of the Free Generators Theorem if $p = 2$. One may wonder if they still generate a free subgroup for an appropriate choice of f, $\tilde{f}$. We address this in the following statement.

Proposition 5.1.4. *With $p = 2$, the sets of matrices $G_1, \ldots, G_6$ given in Table 5.1*

do not generate free groups in $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ *for any* f, $\tilde{f}$.

Proof: Notice that with $p = 2$, for each of G_1 and G_2, we know $B \notin \mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$ as $\det B = 0$.

In Table 5.1 we see that for each of G_3, G_4, G_5, and G_6, the matrices A and B have a common eigenvector, namely $[1 : 1]$. Considering the eigenvectors of A as a basis $\mathcal{B} = \{[0 : 1], [1 : 1]\}$, we thus have that B is triangular with respect to the basis $\mathcal{B}$, while A is a diagonal matrix. Thus $\langle A, B\rangle$ is conjugate to a subgroup of the group of triangular matrices in $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$, which is solvable, and therefore contains no free subgroups. □

When $p = 2$, we see by Proposition 4.3.7 that to be able to take our four eigenvectors with disjoint $\frac{1}{p^{d+1}}$-neighbourhoods, and a point $[z]$ outside these neighbourhoods, we need $p^d(p+1) \geq 5$, so we must take $d \geq 1$. This implies that, to obtain polynomial generators, f and $\tilde{f}$ must be taken to be of degree at least 3, and one of the eigenvectors must have an entry of degree 1. Thus, the Free Generators Theorem yields at minimum generators of degree 4. As an example, we set $p = 2$, $d = 1$ and take $[a : c] = [0 : 1]$, $[1 : b] = [1 : 0]$, $[1 : \tilde{a}] = [1 : 1]$, $[1 : \tilde{b}] = [1 : x]$. Then, for any $f, \tilde{f} \in \mathbb{F}_2[x]$ such that $|f|$, $|\tilde{f}| \leq \frac{1}{2^3}$, the Free Generators Theorem gives that

$$A = \begin{pmatrix} f & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} \tilde{f} + x & \tilde{f} + 1 \\ x\tilde{f} + x & x\tilde{f} + 1 \end{pmatrix}$$

generate a free subgroup in $\mathrm{GL}_2\big(\mathbb{F}_2((x))\big)$. To obtain generators for $p = 2$ with entries of lower degree, we can extend Tillich and Zémor's original degree argument, as shown in Appendix B.

5.2 Constructing a hash function

With the idea of what some alternative choices of generators for the Zémor-Tillich hash function chosen from $\mathfrak{S}$ could look like, we now formally define our hash function. We first need the following map.

Definition 5.2.1. *Let p be a prime, $r_n(x)$ be an irreducible polynomial in $\mathbb{F}_p[x]$ of degree n, and $\mathbb{F}_q \cong \mathbb{F}_p[x]/\langle r_n(x)\rangle$. Set $\mathcal{D} := \{M \in \mathrm{M}_{2\times 2}(\mathbb{F}_p[x]) \,:\, r_n(x) \nmid \det(M)\}$. We define the projection map*

$$\pi_{r_n} : \mathcal{D} \to \mathrm{GL}_2(\mathbb{F}_q)$$

to be the map taking entries of a matrix to their projection in $\mathbb{F}_q$ under the quotient by $\langle r_n(x)\rangle$. For ease of notation, when $r_n(x)$ is not specified we will write π for π_{r_n}.

Remark 5.2.2. To form our hash function, we use this map to project a choice of generators $(A, B) \in \mathfrak{S}$ to elements of $\mathrm{GL}_2(\mathbb{F}_q)$. This is only possible if $\det(A)$ and $\det(B)$ are not divisible by $r_n(x)$. In Section 5.3, we will see that we will want to choose A, B so that the degrees of their entries are much smaller than n. Thus, $\deg(\det(A))$ and $\deg(\det(B))$ must be smaller than n, and so not divisible by $r_n(x)$.

We therefore assume π_{r_n} is defined for A and B from now forward.

We now formally define what we mean by the associated hash function for choices of $(A, B) \in \mathfrak{S}_p$, and $r_n(x)$ an irreducible polynomial in $\mathbb{F}_p[x]$, where we recall from Definition 4.4.3 that $\mathfrak{S}_p$ is the notation for $\mathfrak{S}$ given a specific choice of characteristic p.

Definition 5.2.3. *Let p be a prime. For A, $B \in \mathrm{GL}_2(\mathbb{F}_p((x)))$ and $r_n(x)$ an irreducible polynomial in $\mathbb{F}_p[x]$ of degree n, we define the associated hash function of*

(A, B) and $r_n(x)$ to be the hash function given by Definition 2.3.5 for $G = \mathrm{GL}_2(\mathbb{F}_q)$ and $\pi_{r_n(x)}(A)$, $\pi_{r_n(x)}(B)$, where $\mathbb{F}_q \cong \mathbb{F}_p[x]/\langle r_n(x)\rangle$ and $\pi_{r_n(x)}$ is as in Definition 5.2.1. We define $\mathfrak{H}(p, r_n(x))$ to be the set of all associated hash functions of (A, B) and $r_n(x)$ such that $(A, B) \in \mathfrak{S}_p$.

Note we will write $\mathfrak{H}$ in place of $\mathfrak{H}(p, r_n(x))$ when p and $r_n(x)$ are clear from context.

As mentioned after Definition 2.3.5, one chooses an embedding of $\mathrm{GL}_2(\mathbb{F}_q)$ into $\{0,1\}^N$ for some fixed N so that the hash values can be considered as bitstrings of fixed length. A small example of this is done in Appendix A. However, we will consider our hash values as elements of $\mathrm{GL}_2(\mathbb{F}_q)$.

We now see that the Free Generators Theorem creates infinitely many hash functions over $\mathrm{GL}_2(\mathbb{F}_q)$ as we vary p and n. A natural question is, which choices of these hash functions are the best? As well, we may wonder if these hash functions are resistant to the previous attacks on the Zémor-Tillich hash function and what properties they possess. For the remainder of this chapter, we provide some answers to these inquiries.

5.3 Properties of hash functions in $\mathfrak{H}$

We first investigate some properties of the elements in $\mathfrak{H}$. First, we show that if the degrees of the entries of A and B are small compared to n, we can indeed obtain the small modifications property as desired. We then show that given a relation in $\pi(A)$, $\pi(B)$ and their inverses, we can find a new choice of defining polynomial under which such a relation does not hold. We conclude by considering the group generated by $\langle \pi(A), \pi(B)\rangle$, and show that under certain easily satisfiable conditions we must obtain at least all of $\mathrm{PGL}_2(\mathbb{F}_q)$.

5.3.1 Small modifications property

Recall that, as in Tillich and Zémor's construction, one of the main goals of our construction is to preserve the small modifications property, Property 2.3.1.

With this in mind, suppose that A and B generate a free monoid in $\mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$, as in the case of the generators in Table 5.1, or any polynomial generators in $\mathfrak{S}$. We state the following result, part (a) of which is analogous to Lemma 3.1.3. Part (b) shows that our matrices actually satisfy a stronger property.

Proposition 5.3.1. *Let A, $B \in \mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$ such that A and B generate a free monoid in $\mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$, and let $r_n(x)$ be an irreducible polynomial in $\mathbb{F}_p[x]$. Let H be the associated hash function as in Definition 5.2.3 for (A, B) and $r_n(x)$. Further, suppose $\delta := \max\{\deg(A), \deg(B)\}$ and that $m \in \{0,1\}^r$ and $m' \in \{0,1\}^s$ are distinct bitstrings for some $0 \leq r, s < n/\delta$. Then*

(a) $H(m) \neq H(m')$.

(b) If $(A, B) \in \mathfrak{S}$ then $H(m) \neq kH(m')$ for any $k \in \mathbb{F}_q$ such that, viewing k as an element of $\mathbb{F}_p[x]$, $(\deg(k) + s\delta) < n$. In particular, $H(m) \neq kI$ for any $k \in \mathbb{F}_q$.

Proof: Let M and M' be the respective products yielding $H(m)$ and $H(m')$ in $\mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$ before they are projected into $\mathrm{GL}_2\big(\mathbb{F}_q\big)$, so that $\pi(M) = H(m)$ and $\pi(M') = H(m')$. We see the entries of M are of degree at most $r\delta$ and the entries of M' are of degree at most $s\delta$. Since $r, s < n/\delta$, we know that each of the entries of M and M' has degree less than n, and therefore $\pi(M) = \pi(M') \in \mathrm{GL}_2\big(\mathbb{F}_q\big)$ if and only if $M = M' \in \mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$.

(a) We know that M, $M' \in \langle A, B\rangle$, and by our hypothesis that A, B generate a free monoid in $\mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$. This implies that $M \neq M'$ since m and m' are distinct. Thus it is impossible that $\pi(M) = \pi(M')$ in $\mathrm{GL}_2\big(\mathbb{F}_q\big)$.

(b) Notice that since (A, B) in $\mathfrak{S}$, their images in $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ generate a free subgroup of $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$. This implies that $[M] \neq [M']$. We thus have that $M \neq \ell M'$ for any $\ell \in \mathbb{F}_p((x))$.

Suppose that $\pi(M) = k\pi(M')$ in $\mathrm{GL}_2\big(\mathbb{F}_q\big)$ for some $k \in \mathbb{F}_q$. Viewing k as an element of $\mathbb{F}_p[x]$, we then have that $M = kM' + r_n(x)T$ for some $T \in \mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$. By our hypothesis we know that $\deg(k) + \deg(M') < n$, so $\deg(kM') < n$. Since the entries of M each have degree less than n, this implies $T = 0$, thus $M = kM'$, which is a contradiction. □

Remark 5.3.2. We also note that part (b) of Proposition 5.3.1 implies that products in $\pi(A), \pi(B)$ of length less than n/δ cannot be the identity. In particular, $\pi(A)$ and $\pi(B)$ must have order at least n/δ.

We also have the following corollary.

Corollary 5.3.3. *Let $r_n(x)$ be an irreducible polynomial in $\mathbb{F}_p[x]$ and H be the associated hash function of $r_n(x)$ and one of the generator sets $G_1, \ldots, G_6$ from Table 5.1. If m and m' are bitstrings in $\{0,1\}^*$ such that $H(m) = H(m')$ then at least one of m, m' has length at least $n/\max\{\deg(f), \deg(\tilde{f})\}$.*

We note that part (a) of Proposition 5.3.1 is applicable to any choice of A and B that generate a free monoid in $\mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$, and that our construction of such a hash function could also extend to any such choices. Tillich and Zémor show their hash function generates a free monoid using a degree argument, Lemma 3.1.3. This argument easily extends to matrices with entries having monic polynomial entries and certain relations in the degrees of their entries. We present this extension in Appendix B; however, we will see that our method using the Free Generators Theorem provides many advantages to the method in Appendix B, and further generates many examples that would not be possible to generate using Tillich and Zémor's method.

As an example, we would not be able to obtain $G_1(f, \tilde{f}), \ldots, G_6(f, \tilde{f})$ for any choice of f and $\tilde{f}$.

5.3.2 Guarding against attacks using known identities

We would like to ensure $\pi(A)$ and $\pi(B)$ have large enough order to prevent collisions of the form $\pi(A)^{\mathrm{ord}(\pi(A))} = I$ and $\pi(B)^{\mathrm{ord}(\pi(B))} = I$, as well as Charnes and Piepryzk's short relations attack [8]. To do so, we see that if $\det(\pi(A))$ is a primitive root then $\mathrm{ord}\,(\pi(A)) \geq q - 1$, and note in [1], the authors consider order of at least $q - 1$.

However, preventing short relations of the form $W(\pi(A), \pi(B)) = kI$, where $k \in \mathbb{F}_q^\times$ and $W(\pi(A), \pi(B))$ is a nontrivial word in $\{\pi(A), \pi(B), \pi(A)^{-1}, \pi(B)^{-1}\}^*$, is also desirable. As in example, considering A and B as the Zémor-Tillich generators, Grassl et. al. [18] use the relation $B^{-1}A = A^{-1}B$ to find non-palindromic collisions from their original palindromic ones. We use this as motivation for the following proposition.

Proposition 5.3.4. *Let $(A, B) \in \mathfrak{S}_p$ and let $W(A, B) \in \{A, A^{-1}, B, B^{-1}\}^*$ be a nontrivial word. Then there exists a choice of $r_n(x)$ such that if $\mathbb{F}_q = \mathbb{F}_p[x]/\langle r_n(x)\rangle$ then*

$$W(\pi_{r_n}(A), \pi_{r_n}(B)) \neq kI \in \mathrm{GL}_2\big(\mathbb{F}_q\big)$$

for any $k \in \mathbb{F}_q^\times$.

Proof: Define $\phi := \det(AB) \in \mathbb{F}_p[x]$. Let $\mathbb{F}_p[x]_{1/\phi}$ be the localization of $\mathbb{F}_p[x]$ at ϕ, that is

$$\mathbb{F}_p[x]_{1/\phi} = \left\{ \frac{g}{\phi^m} \ : \ g \in \mathbb{F}_p[x], m \geq 0 \right\}.$$

Since $\frac{1}{\det A} = \frac{\det B}{\phi}$, we have $\frac{1}{\det A} \in \mathbb{F}_p[x]_{1/\phi}$. Similarly $\frac{1}{\det B} \in \mathbb{F}_p[x]_{1/\phi}$. As well,

we note that $\mathbb{F}_p[x]_{1/\phi}$ is contained in the fraction field $\mathbb{F}_p(x)$ and thus in $\mathbb{F}_p((x))$. Consequently, we have that $A,\, B \in \mathrm{GL}_2\big(\mathbb{F}_p[x]_{1/\phi}\big) \subset \mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$.

For any irreducible polynomial $r \in \mathbb{F}_p[x]$, we define the ideal

$$\mathfrak{p} := \langle r \rangle = \{rg \ : \ g \in \mathbb{F}_p[x]\}$$

and, assuming $r \nmid \phi$, let $\mathfrak{p}_{1/\phi}$ be the localization of $\mathfrak{p}$ at ϕ, that is the ideal of $\mathbb{F}_p[x]_{1/\phi}$

$$\mathfrak{p}_{1/\phi} = \left\{ \frac{rg}{\phi^m} \ : \ g \in \mathbb{F}_p[x], m \geq 0 \right\}.$$

Further, we consider the surjective homomorphism

$$\psi_r : \mathbb{F}_p[x]_{1/\phi} \to \mathbb{F}_q$$

induced by

$$x \mapsto \xi$$

where ξ is a root of r.

We see this homomorphism has kernel $\mathfrak{p}_{1/\phi}$. Thus by the first isomorphism theorem we have

$$\mathbb{F}_p[x]_{1/\phi}/\mathfrak{p}_{1/\phi} \cong \mathbb{F}_q.$$

Further, we observe that the natural images of A and $B \in \mathrm{GL}_2\big(\mathbb{F}_q\big)$ under this homomorphism are $\pi_r(A)$ and $\pi_r(B)$ respectively.

Since $A, B \in \mathfrak{S}$, the images of A, B in $\mathrm{PGL}_2\big(\mathbb{F}_p((x))\big)$ also generate a free group. This implies that no nontrivial word in $\{A, B, A^{-1}, B^{-1}\}$ can be I or any scalar multiple kI of I for any $k \in \mathbb{F}_q^{\times}$.

We thus observe that $W(\pi_r(A), \pi_r(B)) = kI$ in $\mathrm{GL}_2(\mathbb{F}_q)$ for some $k \in \mathbb{F}_q^\times$ if and only if, viewing k as an element of $\mathbb{F}_p[x] \subset \mathbb{F}_p[x]_{1/\phi}$,

$$W = \begin{bmatrix} k+\alpha & \beta \\ \gamma & k+\delta \end{bmatrix}$$

for some α, β, γ, $\delta \in \mathbb{F}_p[x]_{1/\phi}$ such that $\psi_r(\alpha) = \psi_r(\delta) = \psi_r(\beta) = \psi_r(\gamma) = 0$.

We thus may choose $r_n(x) \in \mathbb{F}_p[x]$ such that at least one of $\psi_{r_n}(\alpha)$, $\psi_{r_n}(\delta)$, $\psi_{r_n}(\beta)$, $\psi_{r_n}(\gamma)$ is nonzero. □

We note that $r_n(x)$ could also be chosen to simultaneously satisfy the last line of the above proof for each of multiple choices of words, thus allowing us to avoid a small set of relations. However, this proposition does not allow any way to prevent all small relations at once.

5.3.3 The size of the subgroup $\langle [\pi(A)], [\pi(B)] \rangle$

In [49], Tillich and Zémor showed that their choice of A and B generates $\mathrm{SL}_2(\mathbb{F}_{2^n})$. There are results in the literature asserting that for randomly chosen elements from $\mathrm{SL}_2(\mathbb{F}_{2^n})$ (or $\mathrm{PSL}_2(\mathbb{F}_q)$), the probability that they generate the whole group approaches 1 as the size of $\mathrm{SL}_2(\mathbb{F}_{2^n})$ approaches infinity [22], [26]. However, this does not apply to our case as we are taking elements from $\mathfrak{H}$, and thus not choosing A and B randomly.

A well-known result of Dickson [12] determines all possible subgroups of $\mathrm{PSL}_2(\mathbb{F}_q)$. As in [49], we will use its presentation by Suzuki in [47], who gives an elegant exposition of Dickson's proof. We also note a more detailed description of the subgroups can be found in [24]. Because we do not restrict our argument to specific choices of

generators, we will have to do a bit more work than was needed for the argument in [49].

In what follows, it is important to understand the relations between $\mathrm{GL}_2(\mathbb{F}_q)$, $\mathrm{SL}_2(\mathbb{F}_q)$, $\mathrm{PGL}_2(\mathbb{F}_q)$ and $\mathrm{PSL}_2(\mathbb{F}_q)$, which we presented in Section 2.2 and in particular Figure 2.1. We now present Dickson's result [47, Theorem 6.25]

Theorem 5.3.5. *Let p be a prime and $q = p^n$. A subgroup of $\mathrm{PSL}_2(\mathbb{F}_q)$ is isomorphic to one of the following groups:*

(a) a dihedral group or one of its subgroups;

(b) a group K of order $q(q-1)/d$ where $d = \gcd(2, q-1)$, and its subgroups, such that any Sylow p-subgroup of K is normal in K;

(c) the symmetric group on four elements S_4, or the alternating group on four or five elements, A_4 or A_5;

(d) $\mathrm{PSL}_2(\mathbb{F}_{p^\ell})$ or $\mathrm{PGL}_2(\mathbb{F}_{p^\ell})$ for some $\ell \mid n$.

We note that an embedding of $\mathrm{PGL}_2(\mathbb{F}_{p^\ell})$ into $\mathrm{PSL}_2(\mathbb{F}_q)$ is shown in [47, Theorem 6.25 part (x)].

Before using Theorem 5.3.5, we need the following lemma.

Lemma 5.3.6. *Let K be as in (b) of Theorem 5.3.5. Then K is necessarily conjugate to a subgroup of the Borel group*

$$\mathfrak{B} = \left\{ M \in \mathrm{PSL}_2(\mathbb{F}_q) \ : \ M \text{ is upper triangular} \right\}.$$

Proof: Let Q be a Sylow p-subgroup of K. Since we know that $|K|$ is $q(q-1)$ if $p = 2$ and $q(q-1)/2$ otherwise, this implies that $|Q| = q$, so Q is a Sylow p-subgroup

of $\mathrm{PSL}_2(\mathbb{F}_q)$ and $\mathrm{PGL}_2(\mathbb{F}_q)$ as well. We define

$$N = \left\{ \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \; : \; h \in \mathbb{F}_q \right\}.$$

As the order of N is q, and q is the highest power of p dividing $|\mathrm{PSL}_2(\mathbb{F}_q)|$, N is a Sylow p-subgroup of $\mathrm{PSL}_2(\mathbb{F}_q)$, so must be conjugate to Q. That is, $N = PQP^{-1}$ for some $P \in \mathrm{PSL}_2(\mathbb{F}_q)$. Since $Q \triangleleft K$, it follows that $N \triangleleft PKP^{-1}$. It is a well-known fact that the normalizer of N is $\mathfrak{B}$. So since $N \triangleleft PKP^{-1}$, $PKP^{-1} \subseteq \mathfrak{B}$ must hold. Thus K is contained in $P^{-1}\mathfrak{B}P$ for some $P \in \mathrm{PSL}_2(\mathbb{F}_q)$. □

For the following, we identify $\mathrm{PSL}_2(\mathbb{F}_q)$ as a subgroup of $\mathrm{PGL}_2(\mathbb{F}_q)$.

Proposition 5.3.7. *Let $(A, B) \in \mathfrak{S}$ and $\delta = \max\{\deg A, \deg B\}$. Assume that at least one of the following holds:*

(i) f or f' is a primitive root of $\mathbb{F}_q$, n is odd and $n/\delta > 5$ or

(ii) n is prime and $n/\delta > p(p^2 - 1)$.

Then $\mathrm{PSL}_2(\mathbb{F}_q) \subseteq \langle [\pi(A)], [\pi(B)] \rangle$.

These assumptions of Proposition 5.3.7 are in practice very reasonable. We note that we already would like to choose n, δ, such that n/δ is much larger than 5 so that our small modifications property given by Proposition 5.3.1 is practical. Further, we note that choosing n to be prime has already been suggested to prevent against the small order attack proposed in [46].

Proof: To simplify notation, we will write $\Pi(A)$ for $[\pi(A)] \in \mathrm{PGL}_2(\mathbb{F}_q)$ and $\Pi(B)$ for $[\pi(B)] \in \mathrm{PGL}_2(\mathbb{F}_q)$. We note that $p(p^2 - 1) > 5$ holds for all choices of p, so we can assume $n/\delta > 5$.

Let $G = \mathrm{PSL}_2(\mathbb{F}_q) \cap \langle \Pi(A), \Pi(B) \rangle$. We determine if $G = \mathrm{PSL}_2(\mathbb{F}_q)$, by ruling out all possible proper subgroups of $\mathrm{PSL}_2(\mathbb{F}_q)$ as presented in Theorem 5.3.5.

(a) We notice that the group $\langle \Pi(A)^2, \Pi(B)^2 \rangle$ is a subgroup of G. Thus, if G is a dihedral group, or a subgroup of a dihedral group, it is the case that $\langle \Pi(A)^2, \Pi(B)^2 \rangle$ must be as well. In particular, this means $\langle \Pi(A)^2, \Pi(B)^2 \rangle$ is either dihedral or cyclic. As $n/\delta > 4$, Remark 5.3.2 gives that $(\Pi(A)^2)^2 \neq I$ and $(\Pi(B)^2)^2) \neq I$, so this subgroup is not dihedral.

To see that $\langle \Pi(A)^2, \Pi(B)^2 \rangle$ is not cyclic, suppose, that $\Pi(A)^2\Pi(B)^2 = \Pi(B)^2\Pi(A)^2$. Then, we know that $\pi(A)^2\pi(B)^2 = k\pi(A)\pi(B)^2$ in $\mathrm{GL}_2(\mathbb{F}_q)$. Notice the determinants of $\pi(A)^2\pi(B)^2$ and $\pi(B)^2\pi(A)^2$ are equal, so $k = \pm 1 \in \mathbb{F}_q$ must hold. Since $n/\delta > 4$ and $\deg(k) = 0$, Proposition 5.3.1 part (b) implies that $\pi(A)^2\pi(B)^2 \neq k\pi(A)\pi(B)^2$ for $k = \pm 1$, so this is a contradiction. Therefore, $\Pi(A)^2$ and $\Pi(B)^2$ do not commute, so this subgroup is not cyclic.

(b) By Lemma 5.3.6, we know that K is contained in $P\mathfrak{B}P^{-1}$ for some $P \in \mathrm{PSL}_2(\mathbb{F}_q)$. Since K is conjugate to a subgroup of $\mathfrak{B}$, the upper triangular matrices, all elements share a common eigenvector. By construction, A and B have different eigenvectors, respectively $[a : c]$, $[1 : b]$ and $[1 : \tilde{a}]$, $[1 : \tilde{b}]$. Since $\delta < n$, we know that a, b, c, $\tilde{a}$, $\tilde{b}$ all must have entries of degree less than n. Thus, when we quotient by $r_n(x)$, the eigenvectors remain distinct. As a consequence G cannot be isomorphic to a subgroup of K.

(c) We note that any element in A_4 or S_4 has order at most 4, while any element in A_5 has order at most 5. In particular if $\langle \Pi(A), \Pi(B) \rangle$ were a subgroup of one of these, either $\Pi(A)^4 = I$ or $\Pi(A)^5 = I$ must hold. Thus, using Remark 5.3.2, G cannot be in case (c) since $n/\delta > 5$.

(d) To show that G is not a subgroup of the form $\mathrm{PSL}_2(\mathbb{F}_{p^\ell})$ or $\mathrm{PGL}_2(\mathbb{F}_{p^\ell})$, for some

$\ell \mid n$, we suggest two methods that are each sufficient on their own.

First, suppose that either f or f' is a primitive root of $\mathbb{F}_q$ and that n odd. We suppose without loss of generality that f is a primitive root. We recall that by construction (see Lemma 4.4.1) there is a $P \in \mathrm{PGL}_2(\mathbb{F}_q)$ such that

$$\Pi(A) = P \begin{bmatrix} 1 & 0 \\ 0 & f \end{bmatrix} P^{-1} \in \mathrm{PGL}_2(\mathbb{F}_q).$$

Thus, since f is a primitive root, $\Pi(A)$ has order $p^n - 1$.

Suppose that $G \leqslant \mathrm{PGL}_2(\mathbb{F}_{p^\ell})$ for some $\ell \mid n$. We saw that $|\mathrm{PGL}_2(\mathbb{F}_{p^\ell})| = p^\ell(p^{2\ell} - 1)$ (Section 2.2), thus we have that $\Pi(A)^{p^\ell(p^{2\ell}-1)} = 1$, so $p^n - 1 \mid p^\ell(p^{2\ell} - 1)$. We note that $\gcd(p^n - 1, p^n) = 1$, so $\gcd(p^n - 1, p^\ell) = 1$ must hold, thus $p^n - 1 \mid (p^{2\ell} - 1)$.

In particular this implies that $p^n - 1 \mid \gcd(p^n - 1, p^{2\ell} - 1)$. We note it is known that $\gcd(p^n - 1, p^{2\ell} - 1) = p^{\gcd(n,2\ell)} - 1$. Further, since n is odd, $\gcd(n, 2\ell) = \gcd(n, \ell)$. Combining these, we have $p^n - 1 \mid p^{\gcd(n,\ell)} - 1$, so in particular $\gcd(\ell, n) \geq n$. Since $\ell \mid n$, this implies $\gcd(\ell, n) = n$, so $n = \ell$ must hold.

Suppose now that n is prime, and ensure $n/\delta > p(p^2 - 1)$, noting neither f nor $\tilde{f}$ is now required to be a primitive root. Since n is prime, the only possible subgroups in (d) are those for $\ell = 1$, that is $\mathrm{PGL}_2(\mathbb{F}_p)$ and $\mathrm{PSL}_2(\mathbb{F}_p)$. In Section 2.2 we saw $|\mathrm{PGL}_2(\mathbb{F}_p)| = p(p^2 - 1)$, and $|\mathrm{PSL}_2(\mathbb{F}_p)| = p(p^2 - 1)$ if $p = 2$ and $p(p^2 - 1)/2$ if p is odd. Thus in particular $\Pi(A)^{p(p^2-1)} = I$, so by Remark 5.3.2, if $n/\delta > p(p^2 - 1)$ case (d) cannot hold. □

To ensure $[\pi(A)]$ and $[\pi(B)]$ generate $\mathrm{PGL}_2(\mathbb{F}_q)$, we state the following as a corollary of Lemma 2.2.3.

Proposition 5.3.8. *Let A and B be matrices in $\mathrm{PGL}_2(\mathbb{F}_q)$ such that, as in Propo-*

sition 5.3.7, $\mathrm{PSL}_2(\mathbb{F}_q) \subseteq \langle [\pi(A)], [\pi(B)] \rangle$. *Then if either* $\det \pi(A)$ *or* $\det \pi(B)$ *is not a square in* $\mathbb{F}_q$, $\langle [\pi(A)], [\pi(B)] \rangle = \mathrm{PGL}_2(\mathbb{F}_q)$.

Remark 5.3.9. Determining conditions for which $\langle \pi(A), \pi(B) \rangle$ generate all of $\mathrm{GL}_2(\mathbb{F}_q)$ is left for future work. One could potentially do this as an application of Aschbacher's Theorem [3] (we also note its exposition in [44]). We note in the small examples we have considered (see Appendix A), $\pi(A)$ and $\pi(B)$ have always generated $\mathrm{GL}_2(\mathbb{F}_q)$, provided that one of $\det A$ and $\det B$ is not a square mod q.

5.4 Choosing generators

Let $S = (A, B) \in \mathfrak{S}$, $\det(\pi(A)) = \alpha$, $\det(\pi(B)) = \beta$ and $\delta = \max\{\deg(A), \deg(B)\}$. Summarizing the results of Section 5.3, we would like to choose A, B such that

- n/δ is large;
- α, β are primitive roots;
- n is prime or n is odd and either f or f' is a primitive root of $\mathbb{F}_q$;
- α or β is not a square in $\mathbb{F}_q$.

We note here that the last condition is satisfied when p is odd if α and β are primitive roots. Namely, suppose that p is odd and $\alpha \equiv \gamma^2$ for some $\gamma \in \mathbb{F}_q$. Since $\gamma \in \mathbb{F}_q^\times$, we know $\gamma^{q-1} = 1$. It follows then that $\alpha^{\frac{q-1}{2}} = (\gamma^2)^{\frac{q-1}{2}} = 1$, so $\mathrm{ord}(\alpha) \mid \frac{q-1}{2}$, so α cannot be a primitive root.

For an example of generators with p odd, we can consider the pairs of matrices $G_i(f, \tilde{f})$ in Table 5.1. We see for each of these choices, $\det(A) = f$ and so we would choose f to be a primitive root of smallest possible degree, considered as an element of $\mathbb{F}_p((x))$.

Though finding a primitive root of small degree is not straightforward for a general case of finite field, we note that the maximum degree of such a primitive root is indeed bounded [45]. Further, certain choices of $r_n(x)$ could make this easier. For example, many packages, such as GAP [15] and Magma [4], use Conway polynomials for $r_n(x)$, which guarantee that x itself is indeed a primitive root.

We notice that for the choices $G_i(f, \tilde{f})$ in Table 5.1 $\det(B)$ is either $4\tilde{f}$ or $\tilde{f}$. To find an appropriate choice of $\tilde{f}$, we would thus search for a power of α, relatively prime to $p^n - 1$, that is an element of small degree in $\mathbb{F}_q$.

5.5 The effect of the determinant

In some initial implementations of the generators $G_i(f, \tilde{f})$ in Table 5.1 in GAP, we discovered that ill-informed choices of f and $\tilde{f}$ could leak information about the length of the original bitstring. In this section we explain these potential issues and propose two separate methods to prevent them.

For this section, we suppose that $\pi(A)$, $\pi(B) \in \mathrm{GL}_2(\mathbb{F}_q)$ such that $\det(\pi(A)) = \alpha$ and $\det(\pi(B)) = \beta$. Suppose also that $m \in \{0,1\}^\ell$ and $M = H(m) \in \mathrm{GL}_2(\mathbb{F}_q)$. Further, suppose m has ℓ_1 zeros and ℓ_2 ones, so $\ell_1 + \ell_2 = \ell$.

5.5.1 Attacks via the determinant

We first present some potential attacks that take advantage of any information leaked by the determinant. Many of these attacks will depend on being able to compute discrete logs in $\mathbb{F}_q$, which we discuss in Section 5.6. For $g, h \in \mathbb{F}_q^\times$ we will use the notation $\mathrm{dlog}_g(h)$ for the $k \in \{1, \ldots, q-1\}$ such that $g^k = h$.

Attack 1. *Determining ℓ when $\alpha = \beta$.*

When $\alpha = \beta$, $\text{dlog}_\alpha(\det(M)) \equiv \ell \bmod (p^n - 1)$, where we recall ℓ is our message length. We note that $p^n - 1$ is much larger than any message length since it is of cryptographic size. As $\ell \ll p^n - 1$, an attacker able to efficiently compute dlog_α can determine the length ℓ of the message.

Attack 2. *Gathering information about ℓ_1 or ℓ_2 when $\gcd(\text{ord}(\alpha), \text{ord}(\beta))$ is close to $\text{ord}(\alpha)$ or $\text{ord}(\beta)$ in size.*

We recall that ℓ_1 was the number of zeros in m, while ℓ_2 was the number of ones, so that $\ell_1 + \ell_2 = \ell$.

Suppose that $\text{ord}(\alpha) = s$ and $\text{ord}(\beta) = t$ with $gcd(s, t) = k$. We note s and t must divide the order of the group $\mathbb{F}_q^\times$, which is equal to $p^n - 1$.

Since $\det(M) = \alpha^{\ell_1}\beta^{\ell_2}$, we see that $\det(M)^s = \beta^{\ell_2 s}$. Thus $\det(M)^s = 1$ if and only if $t \mid \ell_2 s$, which is true if and only if $t/k \mid \ell_2$. Thus, an attacker can compute $\det(M)^s$ and can conclude the number of ones in m is divisible by t/k if $\det(M)^s = 1$ and not divisible by t/k if $\det(M)^s \neq 1$.

This could be problematic if $\gcd(s, t)$ is large in comparison to, but not equal to s or t. For instance, if β is a primitive root, so $t = p^n - 1$, and α has order $s = \frac{p^n-1}{2}$, then the attacker can deduce the parity of ℓ_2 by determining whether $\det(M)^s$ is 1 or not.

If α and β are primitive roots then we notice that this attack strategy does not yield any information.

Attack 3. *Determining ℓ_1 and ℓ_2 when α is a primitive root, and $\beta = \alpha^r$ for some r such that $\ell_1 < r$ and $\ell_1 + \ell_2 r < p^n - 1$.*

We have $\det M = \alpha^k$ for some k. This gives us $\alpha^k = \alpha^{\ell_1}\alpha^{r\ell_2}$, or equivalently $k = \ell_1 + r\ell_2 \bmod (p^n - 1)$. Since $\ell_1 + r\ell_2 < p^n - 1$, this means that $k = \ell_1 + r\ell_2$.

Since $\ell_1 < r$, dividing k by r gives $\ell_1/r + \ell_2$ where ℓ_2 is an integer and $\ell_1/r < 1$. Therefore if an attacker can efficiently compute dlog_α, so can recover k and r, then the attacker can compute $\lfloor \frac{k}{r} \rfloor$ to obtain ℓ_2. From this, and the relation $\ell_1/r + \ell_2 = k$, the attacker can also compute ℓ_1, and thus would know the number of ones and the number of zeros in m.

Remark 5.5.1. We note that the above attacks are prevented by choosing α to be a primitive root, and $\beta = \alpha^r$ for some r of size $\mathcal{O}(p^n/2)$ such that $\gcd(r, p^n - 1) = 1$, so β is a primitve root. Choosing α and β to be primitive roots was already suggested in Section 5.3.2. Further, two of these attacks depend on being able to calculate discrete logs in $\mathbb{F}_q$. The difficulty of this is discussed in Section 5.6.

Attack 4. *Determining the parity of ℓ or ℓ_1 when α is a primitive root.*

Since α is a primitive root, we know that $\beta = \alpha^r$ for some r. Note that we have $\det(M) = \alpha^k$ where $k = \ell_1 + r\ell_2$ where $\ell_2 = \ell - \ell_1$. This gives

$$k = r\ell + (1 - r)\ell_1 \mod p^n - 1. \tag{5.5.1}$$

Suppose that an attacker can efficiently compute dlog_α, so can recover k. Since we are assuming p is odd, we know that $p^n - 1$ is even. Therefore if r is even, we know that k is even precisely when ℓ_1 is even. If r odd, then $r - 1$ is even so we know that k is even precisely when ℓ is even.

Remark 5.5.2. Attack 4 is not limited to parity. Notice that in (5.5.1) if $s \in \mathbb{N}$ is such that $s \mid (p^n - 1)$ and $s \mid r$, we know that $s \mid k$ precisely when $s \mid \ell_1$. Similarly, if $s \in \mathbb{N}$ is such that $s \mid (p^n - 1)$ and $s \mid (r - 1)$, we know that $s \mid k$ precisely when

$s \mid \ell$. Note that if β is a primitive root, then $\gcd(r, p^n - 1) = 1$, so the first case will not occur.

In Section 5.5.3 and Section 5.5.4 we will see two methods to prevent Attacks 1 to 4. The method in Section 5.5.3 also prevents the determinant from affecting the distribution of hash values, which we first discuss in the next section.

5.5.2 Distribution of the determinant

We recall that the Zémor-Tillich hash function had the property that the distribution of hash values of messages of length ℓ approached the uniform distribution as ℓ approached infinity (see Section 3.1). In our construction, we would also like the hash values to exhibit the uniform distribution.

Suppose that α, β are primitive roots as suggested in Section 5.3.2 and that $\beta = \alpha^r$ for some $r > 1$ such that $\gcd(r, p^n - 1) = 1$. As well, suppose that m is a bitstring of length ℓ. Then, as above, we know $\det(H(m)) = \alpha^k$ where $k = \ell_1 + r\ell_2 \mod p^n - 1$ for some $\ell_1, \ell_2 \in \mathbb{N}_0$ such that $\ell_1 + \ell_2 = \ell$.

We observe that for a fixed ℓ, $|\{\ell_1, \ell_2 \in \mathbb{N}_0 \ : \ \ell_1 + \ell_2 = \ell\}| = \ell + 1$. Thus, there are only $\ell + 1$ possible values of the determinants of hashes of bitstrings of fixed length ℓ. We note in practice, ℓ will be much smaller than p^n. For example, a message of size $1GB$ has $\sim 2^{33}$ bits, whereas in Section 5.6, we will see that we want $p^n \sim 2^{512}$.

Consequently, in practice we know that the hash values of messages of length ℓ cannot be distributed uniformly among all possible determinants, as their determinants must be among a subset of at most $\ell + 1$ possible values of $\mathrm{GL}_2\big(\mathbb{F}_q\big)$. Further, notice that amongst the $\ell+1$ possible determinants, the determinants of hash values of bitstrings of length ℓ will not be distributed uniformly, as having $\ell_1 = 0$ is much more unlikely

than having ℓ_1 and ℓ_2 balanced.

Remark 5.5.3. We note that, assuming that p is odd, and r, ℓ fixed with r odd and ℓ even, the relation $k = r\ell + (1-r)\ell_1 \mod p^n - 1$ as in (5.5.1) implies that the possible determinants of hash values of messages of length ℓ can only be even powers of α. Again, more generally, if $s \in \mathbb{N}$ such that $s \mid (p^n - 1)$, $s \mid (r-1)$, and $s \mid \ell$, the possible determinants of hash values of messages of length ℓ can only be powers of α divisible by s.

5.5.3 Padding

One common option in cryptography for preventing attacks based on the weaknesses mentioned above is by padding messages with some bits to obscure the original determinant of the hash value. As an example, one of the most standard forms of padding is PKCS #5, which pads messages to be a multiple of a given block length based on the amount of padding needed [23].

For elements of $\mathfrak{H}$, we propose padding our messages as follows. Namely, suppose we wish to hash messages of length at most N. We could then pad our messages to bitstrings of length $2N$ with precisely N ones and N zeros. This would ensure all outputs had determinant $\alpha^{N(1+r)}$, thus completely eliminating the effect of the determinant on the security and the distribution. If we choose elements of $\mathfrak{H}$ such that $\pi(A)$, $\pi(B)$ generate all of $\mathrm{GL}_2(\mathbb{F}_q)$ (see Remark 5.3.9), the number of possible hash values would be $|\{M \in \mathrm{GL}_2(\mathbb{F}_q) \ : \det(M) = \alpha^{N(1+r)}\}| = q(q^2-1)$. We recall from Section 2.2 this is the same size as $\mathrm{SL}_2(\mathbb{F}_q)$, and so is comparable to Zémor-Tillich for $p = 2$ and the extension of Zémor-Tillich given in [1] (see Section 3.3) for p odd.

Note that in the case $\alpha = \beta$, any choice of padding to a fixed length would both

protect against the attacks in Section 5.5.1 and remove the distribution issues in Section 5.5.2.

5.5.4 Hash functions over $\mathrm{PGL}_2(\mathbb{F}_q)$

Another way to remove the risk of the determinant leaking information is to define our hash function over $\mathrm{PGL}_2(\mathbb{F}_q)$, rather than $\mathrm{GL}_2(\mathbb{F}_q)$. To do so, we would take our hash function H to be the associated hash function of the equivalence classes $\{[\pi(A)], [\pi(B)]\}$ and $G = \mathrm{PGL}_2(\mathbb{F}_q)$. We note that the work in Section 5.3.3 already gave precise conditions for which our matrices would generate all of $\mathrm{PGL}_2(\mathbb{F}_q)$.

To store elements of $\mathrm{PGL}_2(\mathbb{F}_q)$ as bitstrings, we would first need a canonical representation of an element of $\mathrm{PGL}_2(\mathbb{F}_q)$. In some initial implementations (see Appendix A), we store the matrix

$$\begin{bmatrix} t & u \\ v & w \end{bmatrix} \in \mathrm{PGL}_2(\mathbb{F}_q) \quad \text{as} \quad \begin{cases} \begin{pmatrix} 1 & u/t \\ v/t & w/t \end{pmatrix} & \text{if} \quad t \neq 0; \\ \begin{pmatrix} 0 & 1 \\ u/v & w/v \end{pmatrix} & \text{if} \quad t = 0. \end{cases}$$

A small example of this, and a subsequent embedding into $\{0,1\}^N$ for a fixed N, is done in Appendix A.

While this may seem a bit unfamiliar, Proposition 5.3.4 tells us that for any word in $W \in \{\pi(A), \pi(B), \pi(A)^{-1}, \pi(B)^{-1}\}$, we can find a choice of irreducible polynomial such that $W \neq sI \in \mathrm{GL}_2(\mathbb{F}_q)$ for any $s \in \mathbb{F}_q^\times$, which is equivalent to the statement that $[W] \neq [I] \in \mathrm{PGL}_2(\mathbb{F}_q)$. As well, the upcoming discussions in Section 5.6 and Section 5.7 seem to apply in this construction, indicating that all other properties and

notions of security for elements of $\mathfrak{H}$ will be retained when we project to $\mathrm{PGL}_2(\mathbb{F}_q)$.

Further, Proposition 5.3.1 part (b) gives us a partial version of the small modifications property when we project our matrices to elements of $\mathrm{PGL}_2(\mathbb{F}_q)$. For small examples we have computationally found that the small modifications property holds.

Constructing our hash function over $\mathrm{PGL}_2(\mathbb{F}_q)$ would completely obscure the determinant, and thus prevent Attacks 1, 2, 3, 4. However, we note this hash function still leaks a small amount of information.

Namely, suppose α, β are primitive roots, such that $\beta = \alpha^r$ for some $r > 1$ such that $\gcd(r, p^n - 1) = 1$. Let m be a message in $\{0,1\}^*$ of length ℓ, and ℓ_1 and ℓ_2 are as before. Then we know $\det H(m) = \alpha^k$, where, as in (5.5.1), $k = r\ell + (1-r)\ell_1 \mod p^n - 1$.

Again, we are assuming p is odd, so $p^n - 1$ is even. With $\gcd(p^n - 1, r) = 1$, this implies that r is odd. We notice that $\det H(m)$ is a square modulo $\mathbb{F}_q$ if and only if k is even, which in turn holds if and only if ℓ is even. In particular, if ℓ is even, this means that messages of length ℓ produce only elements of $\mathrm{PGL}_2(\mathbb{F}_q)$ with determinants that are squares in $\mathbb{F}_q$.

In this construction, we also do not expect a uniform distribution among messages of fixed lengths, as our last observation demonstrates. Again, this weakness is addressed by an appropriate padding.

5.6 Security

Here we consider which of the attacks on the Zémor-Tillich hash function discussed in Chapter 3 could be extended to hash functions in $\mathfrak{H}$.

In Chapter 3 we saw that Zémor-Tillich was believed to be resistant against density attacks, as a result of Theorem 3.1.2. In [49], Tillich and Zémor provide a sketch of a proof of Theorem 3.1.2, which is a counting argument. Though not done here, we believe this argument easily extends to elements of $\mathfrak{H}$ that satisfy the conditions in Section 5.3.3, and thus conjecture such elements of $\mathfrak{H}$ are also resistant to density attacks.

The first attack on Zémor-Tillich was Charnes and Piepryzk's short relations attack (Section 3.2.1). This attack depended on $r_n(x)$ being such that A or B had small order, and thus could easily extend to elements of $\mathrm{GL}_2(\mathbb{F}_q)$. However, the probability that a randomly chosen $r_n(x)$ yielded A and B with order large enough (at least $q-1$), is extremely high for Tillich and Zémor's generators, thus this attack was not a concern [1]. For elements in $\mathfrak{H}$, we saw in Section 5.3.2 that choosing $\det(A)$ and $\det(B)$ to be primitive roots will ensure A and B have order at least $q-1$.

In [43], Regenscheid shows that the small order attack presented by Steinwandt et. al. (Section 3.2.2) is also extendible to larger characteristics. The attack requires $r_n(x)$ to be decomposable, that is a composition of two nontrivial polynomials. However, Regenscheid shows that in $\mathbb{F}_2$ the probability that a randomly chosen irreducible polynomial of degree n is decomposable approaches 0 as n approaches infinity [43]. These methods extend to irreducible polynomials over $\mathbb{F}_p$. Further, Steinwandt et. al. noted that the small order attack could be avoided by simply choosing n prime [46].

In [36], Petit, Lauter, and Quisquater mention that the attack used on the Morgenstern hash function (Section 2.3) could possibly be used for the Zémor-Tillich hash function and similar construction by embedding the associated Cayley graphs into Morgenstern graphs. However, no concrete details were given and as far as we know this has not been explored further.

Geiselmann's embedding attack (Section 3.2.3) could also apply for generators over $\mathrm{SL}_2(\mathbb{F}_q)$ for a general choice of $q = p^n$. The alternative proof in [43] can be easily extended to the case $p > 2$ when the generators are diagonalizable, as in the case of elements $S \in \mathfrak{S}$. The computation time of this attack depends on the computation time of computing discrete logs in $\mathrm{GL}_2(\mathbb{F}_{2^n})$. Note that the discrete log problem in $\mathrm{GL}_2(\mathbb{F}_{p^n})$ can be reduced to the problem of computing discrete logs in $\mathbb{F}_{(p^n)^2}$ [28].

Computing discrete logs in $\mathbb{F}_{p^n}$ when $p = 2$ is made faster by applying Coppersmith's algorithm [9], which also applies in the case that p is fixed and n approaches infinity, and runs in subexponential time [10]. Currently for n prime Coppersmith's algorithm is still the fastest known algorithm for computing discrete logs in $\mathbb{F}_{p^n}$, and in $\mathbb{F}_{(p^n)^2}$ when p is not close to n in size [2]. In general, computing discrete logs in $\mathbb{F}_{p^n}$ with odd characteristic is considered to be more difficult [32]. Further, we recall that Geiselmann's algorithm produced only collisions containing long strings of ones and zeros, which would not be useful in practice.

Assuming that Mullan's attack in [31] is extendible to the case of $\mathrm{GL}_2(\mathbb{F}_q)$, this would therefore be the best known attack on the functions in $\mathfrak{H}$. We recall this attack had running time $\mathcal{O}(\sqrt{q})$ and produced collisions of length $\mathcal{O}((\log q)^2/\log(\log q))$ for a general characteristic p. In Appendix A, we discuss the small-space birthday attack, and present some early computational results. Other options for attacks on elements of $\mathfrak{H}$ are an optimization of the birthday attack as in [40] or an attack such as the meet-in-the-middle approach in [30]. Each of these are standard attacks that have no advantage, and may be even more difficult, for our hash functions.

As each of these general attacks runs in exponential time, choosing parameters comparable to those in current cryptographic standards is expected to be sufficient to provide a secure hash function. Currently, the NIST approved hash algorithms are SHA3-224, SHA3-256, SHA3-384, SHA3-512 [13]. The number associated to these

hash algorithms denotes their security strength in bits. For instance, SHA3-512 provides 512 bits of security against a preimage or second preimage finding algorithm, and 256 bits of security against a collision finding algorithm. Since the fastest known attacks have running time $\mathcal{O}(\sqrt{p^n})$, choosing $p^n \sim 2^{512}$ would provide equivalent security.

5.7 Practicality

We saw in Section 5.3.1 that for A and B of reasonably small degree, the hash functions in $\mathfrak{H}$ preserve the small modifications property. Further, we note that as our construction is a Cayley hash function it is naturally parallelizable. Elements of $\mathfrak{H}$ are also scalable, meaning we are able to control the size of the output. In Section 5.6 and Section 5.3 we saw that under certain easily satisfiable conditions our hash functions are secure against all previous efficient attacks on the Zémor-Tillich hash function, the density attack on Zémor's original construction in $\mathrm{SL}_2(\mathbb{F}_p)$, as well as any potential weaknesses from a badly chosen determinant. Thus for appropriate choices of parameters, our hash functions are collision, preimage and second preimage resistant.

Another desired property of a hash function is that with increasing message lengths, the distribution of hash values approaches the uniform distribution. Rigorously, one might check this property using Proposition 3.1.4. In practice, proposed hash functions are usually checked heuristically. In the case $p = 2$, one can test pseudorandomness using the Dieharder suite as was done with the variant of the Zémor-Tillich hash function proposed in [41]. As Cayley hashes produced in this way tend to distribute this property, we hypothesize this would be the case for hash functions in $\mathfrak{H}$, provided they are padded as suggested in Section 5.5.3. Further, we refer the reader to some

initial computations in Appendix A. However, this is something we would ideally like to argue more rigorously.

As well, we would like the hash functions in $\mathfrak{H}$ to be fast. Although the number of bit operations needed per multiplication depends on the matrices A, B, we expect for most choices this will be reasonable. Petit, Lauter and Quisquater [35] implemented Cayley hashes and compared them to SHA-256 both in hardware (in which they ran 5 times slower) and in software (in which they ran 400 times slower), and believed it was possible to improve these speeds, in particular by computing these functions in parallel. Because of this, we expect that in hardware our hash functions would be comparable in speed to current cryptographic standards. We note that the SHA family of hash functions is not parallelizable.

The speed and distribution of elements in $\mathfrak{H}$ are both properties we hope to study further in future work. We note in the literature very little is presented numerically on these points. We present some initial implementation code using GAP in Appendix A. In future work, we hope to extend our computational analysis of elements of $\mathfrak{H}$.

5.8 Future work

Let us recap our work in this chapter. Here, we proposed an abundant choice of generators to replace those in Tillich and Zémor's construction. To do so, we use the novel Free Generators Theorem from Chapter 4, which provides an infinite set of pairs A, B that generate a free group in $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$. We then project polynomial choices of such A and B, to $\mathrm{GL}_2\big(\mathbb{F}_q\big)$, where $\mathbb{F}_q \cong \mathbb{F}_p[x]/\langle r_n(x)\rangle$, using the projection map π defined by taking the quotient by $\langle r_n(x)\rangle$. The hash function is then as in Zémor-Tillich ; given a bitstring $m_1 \dots m_k$, $H(m_1 \dots m_k) = H(m_1) \cdots H(m_k)$, where $H(0) = \pi(A)$ and $H(1) = \pi(B)$. The set of all such hash functions was denoted by

$\mathfrak{H}$. We note that in Appendix B we saw this method both had many advantages over, and produced different matrices from, the natural extension of the degree argument used in the Zémor-Tillich hash function.

To prevent potential weaknesses from using a determinant, two strategies were proposed in Section 5.5: either padding elements of $\mathfrak{H}$ (Section 5.5.3), or projecting elements of $\mathfrak{H}$ to $\mathrm{PGL}_2(\mathbb{F}_q)$ (Section 5.5.4). More exploration into these ideas, including finding an choosing an optimal padding and determining whether the small modifications property holds in $\mathrm{PGL}_2(\mathbb{F}_q)$, is left to future work.

Our method also seems appropriate for several extensions, which we discuss here and hope to explore further in subsequent work.

To start with, the methods in Chapter 4 could be nontrivially extended to finding generators of free groups in $\mathrm{GL}_k(\mathbb{F}_p((x)))$ or $\mathrm{PGL}_k(\mathbb{F}_p((x)))$ for $k > 2$.

There are also some more immediate extensions. As the Free Generators Theorem produces not one, but many hash functions, we could also use the Free Generators Theorem to produce a keyed hash function, where the choice of hash function was pulled randomly from a subset of $\mathfrak{H}$ satisfying our desired conditions.

The Free Generators Theorem is also easily extendible to a larger number of generators, as we state below. This provides $k+1$ generators, allowing us to hash messages written in base $k+1$. For example, suppose that we instead wish to define a hash function that is base 3, rather than binary. In future work, we would like to further examine these base $k+1$ hash functions.

Proposition 5.8.1. *Let p be a prime, $d \in \mathbb{N}_0$ and $k \in \mathbb{N}$. Further, let a, b, c, $a_1, \ldots, a_k$, $b_1, \ldots, b_k, \in \mathbb{F}_p((x))$ and $f, f_1, \ldots, f_k \in \mathbb{F}_p((x))^*$, such that*

1. *there exists a $[z] \in \mathbb{P}^1$ such that $d([u],[v]) > \frac{1}{p^{d+1}}$ for each pair of $[u],[v]$ in*

$\{[a:c],[1:b]\} \bigcup \{[1:a_i],[1:b_i] \ : \ 1 \leq i \leq k\} \bigcup \{[z]\}$, *and*

2. $\min\{|f|,|f^{-1}|\} \leq \frac{1}{p^{2d+1}}$, *and* $\min\{|f_i|,|f_i^{-1}|\} \leq \frac{1}{p^{2d+1}}$ *for* $1 \leq i \leq k$.

Then the set of matrices

$$\left\{ A := \begin{pmatrix} ab-cf & a(f-1) \\ cb(1-f) & abf-c \end{pmatrix} \right\} \bigcup \left\{ B_i := \begin{pmatrix} b_i - a_i f_i & f_i - 1 \\ a_i b_i (1-f_i) & b_i f_i - a_i \end{pmatrix} \ : \ 1 \leq i \leq k \right\}$$

are free generators of a free group in $\mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$.

We note that, because of Proposition 4.3.7, these conditions can be satisfied when $p^d(p+1) \geq k+2$.

Proof: (Idea of proof): To see that this holds, we notice that A is precisely A from Theorem 4.4.2, and each B_i is of this form, taking $a=1$, $c=a_i$ and $b=b_i$. Therefore, we can again reduce the proof to showing nonidentity elements in $\langle A \rangle$ will map $\mathbb{P}^1 \setminus P_A$ into P_A, where P_A is as in the proof of Theorem 4.4.2. Our conditions ensure the neighbourhoods of our eigenvectors are again disjoint, and that f is sufficiently small or large in absolute value. Thus, Lemmas 4.6.1 and 4.6.3 will still apply, giving $A\left(\mathbb{P}^1 \setminus N_{[1:b]}\right) \subseteq N_{[a:c]}$ and $A\left(\mathbb{P}^1 \setminus N_{[a:c]}\right) \subseteq N_{[1:b]}$, as well as $B_i\left(\mathbb{P}^1 \setminus N_{[1:b_i]}\right) \subseteq N_{[1:a_i]}$ and $A\left(\mathbb{P}^1 \setminus N_{[1:a_i]}\right) \subseteq N_{[1:b_i]}$ for each $1 \leq i \leq k$. The proof will follow identically. □

We note that the asymmetry of the generator A corresponds to the asymmetry of the description of points in $\mathbb{P}^1$; we can view $\mathbb{P}^1$ as $\{[1:g] \ : \ g \in \mathbb{F}_p((x))\} \cup \{[0:1]\}$. The matrix A allows for the choice of $[0:1]$ as an eigenvector, and thus provides a slightly more general statement than having only elements of the form B_i.

We believe another straightforward extension would be applying the ideas of the proof of the Free Generators Theorem to produce free generators of general linear groups over other local fields, such as $\mathbb{Q}_p$, by equipping it with an analogous distance.

Appendix A

Implementation

In this section we show some of the code we used to implement our hash function in the software GAP [15].

We first set up p, n, and our finite field $\mathbb{F}_q$. In this code, we note `a` represents the image of x when we project from $\mathbb{F}_p[x]$ to $\mathbb{F}_q$ under the quotient modulo $\langle r_n(x) \rangle$. GAP uses Conway polynomials to choose $r_n(x)$, which guarantees a is both a root of $r_n(x)$ and a primitive root of $\mathbb{F}_q$. We note this implementation in GAP is only possible for small values of n. For instance, choosing $p^n = 3^{11}$ is already too large.

We then define our choices of A and B by telling GAP the rows of A and B and using the preset functions `ConvertToVectorRep` and `ConvertToMatrixRep` to tell GAP to treat A and B as matrices over the finite field, thus speeding up computations.

We then set up the function `Bin` which finds the binary representation of an integer.

```
p:=3; n:=5;
a:= Z(p^n);
```

```
vA1:=[ a, 0*a];; vA2:=[0*a, a^0];;
vB1:=[-a^2+a^0, a^0+a^2];; vB2:=[a^0+a^2, -a^2+a^0];;

ConvertToVectorRep(vA1, p^n);; ConvertToVectorRep(vA2, p^n);;
ConvertToVectorRep(vB1, p^n);; ConvertToVectorRep(vB2, p^n);;
A:=[vA1, vA2];; B:=[vB1, vB2];;
ConvertToMatrixRep(A, p^n);; ConvertToMatrixRep(B, p^n);;

Bin:=function(x)
local bnry, lg, midx, idx1;
bnry:=[];
if x=0 then
bnry:=[0];
else
lg:=LogInt(x, 2);
if x>lg then
lg:=lg+1;
fi;
midx:=x;
for idx1 in [1..lg] do
if midx<2^(lg-idx1) then
Append(bnry, [0]);
else
Append(bnry, [1]);
midx:=midx-2^(lg-idx1);
fi;
od;
fi;
return bnry;
end;
```

For $m \in \{0,1\}^*$, we recall that elements in $\mathfrak{H}$ (see Definition 5.2.3) give matrices in $\mathrm{GL}_2\big(\mathbb{F}_q\big)$. In practice, we need to then map these matrices in $\mathrm{GL}_2\big(\mathbb{F}_q\big)$ to bitstrings of a fixed length.

To do so, we first note that since a is a primitive root, matrices in $\mathrm{GL}_2\big(\mathbb{F}_q\big)$ will have entries either 0 or a^r for some $r \in \{0, \ldots, p^n - 2\}$. Further, GAP stores elements of $\mathbb{F}_q^\times$ as these powers of a. We thus store each entry a^r as the binary representation of

r and store 0 as the binary representation of $p^n - 1$, each padded with zeros to be of length $\lceil \log_2(p^n) \rceil$. This process produces fixed outputs of length $4\lceil \log_2(p^n) \rceil$.

As an example, with $p = 3$, $n = 5$, we have $\lceil \log_2(3^5) \rceil = 8$, so our hash function produces bitstrings of fixed length 32. Given A and B as set up in the code above, the matrix

$$AB := \begin{bmatrix} a^{196} & a^{47} \\ a^{46} & a^{195} \end{bmatrix} \in \mathrm{GL}_2\big(\mathbb{F}_q\big)$$

is stored as

$$[\underbrace{1,1,0,0,0,1,0,0}_{196},\underbrace{0,0,1,0,1,1,1,1}_{47},\underbrace{0,0,1,0,1,1,1,0}_{46},\underbrace{1,1,0,0,0,0,1,1}_{195}]$$

We have included our code to do this below. Here `bitsize` is $\lceil \log_2(3^5) \rceil$. Given an element $g \in \mathbb{F}_q$ the function `WhatPow` outputs $p^n - 1$ if $g = 0$, and for $g = a^r$ finds r. Then, `BinRep` finds the binary representation of $r \in \{0, \ldots, p^n - 1\}$ as a bitstring of fixed length `bitsize`. Lastly, `MatrixBinRep` takes a matrix in $\mathrm{GL}_2\big(\mathbb{F}_q\big)$ and outputs it as a bitstring of length $4 * bitsize$.

```
if p=2 then
bitsize:=LogInt(p^n,2);
else
bitsize:=LogInt(p^n,2)+1;
fi;

blength:=bitsize;

BinRep:=function(x)
local bnrp, idx, padlh, pad;
pad:=[];
bnrp:=Bin(x);
padlh:=blength-Length(bnrp);
for idx in [1..padlh] do
```

```
Append(pad, [0]);
od;
Append(pad, bnrp);
return pad;
end;

WhatPow:=function(x)
if x=0*a then
return p^n-1;
else
return LogFFE(x, a);
fi;
end;


MatrixBinRep:=function(x)
local bnrep;
bnrep:=BinRep(WhatPow(x[1][1]));
Append(bnrep, BinRep(WhatPow(x[1][2])));
Append(bnrep, BinRep(WhatPow(x[2][1])));
Append(bnrep, BinRep(WhatPow(x[2][2])));
return bnrep;
end;
```

We are now ready to define our hash function H.

```
H:= function(x)
local lh, idx3, hsh;
lh := Length(x);
if x[1]=0 then
hsh:=A;
else
hsh:=B;
fi;
for idx3 in [2..lh] do
if x[idx3]=0 then
hsh:=hsh*A;
else
hsh:=hsh*B;
fi;
```

```
od;
return MatrixBinRep(hsh);
end;
```

A.1 Small space birthday attack

One of the ways we've tested our hash function for small values of p and n is by implementing a small space birthday attack. We summarize the idea of this attack, as it is presented in [23], below. For more details, we direct the reader to [23].

Suppose that our hash function outputs bitstrings in of length ℓ. The **birthday attack** is a well-known idea; by computing the hash values of $\ell/2$ random bitstrings, we expect two of these hash values to be the same with a probability of approximately $1/2$.

A significant disadvantage of the birthday attack is that one must store each of these random bitstrings as well as their hashes. To remove the need for so much memory, one can use the **small space birthday attack**. The small space birthday attack is expected to work in the same time as the birthday attack, but uses negligible memory.

The **small space birthday attack** is as follows. We begin with a random bitstring x_0 of length not equal to ℓ. At each step, we calculate

$$x_i := H(x_{i-1}) \quad \text{and} \quad x_{2i} := H(H(x_{2(i-1)}))$$

and store only x_{i-1}, x_i, $x_{2(i-1)}$, and x_{2i}. Notice we then have $x_i = H^i(x)$, where H^i denotes i successive applications of H.

Each step, we check if $x_i = x_{2i}$. If it is the case that $x_I = x_{2I}$ at some index I, we know a collision exists, however, it is possible that $x_{I-1} = x_{2I-1}$ may also hold. Since

x_0 is a bitstring of length different from $x_I = H^I(x)$, we know that in particular at some index J, we will find a point where $x_{J+1} = x_{J+I+1}$, but $x_J \neq x_{J+I}$. We thus compute $H(x_i)$ and $H(x_{i+I})$ until we find this index J.

To understand why this works, notice that at each step when we check $x_i = x_{2i}$, we are also checking $x_j = x_{j+i}$ for any $j \leq i$.

Below, we show our code for the small space birthday attack, provided the previous code has already been implemented. We will first need to set up a random bitstring x_0 of some length not equal to ℓ. In this this example we take the length of x_0 to be 17. We also need to initialize our source for random integers. The suggested way of doing this on GAP is the first two lines of code below, where we pick the seed to be some arbitrarily chosen integer, such as the time. We note this initialization only has to be done once per session.

```
seed:=6;
mysource:=RandomSource(IsMersenneTwister, seed);

x0length:=17;
x0:=Random(mysource, 0,2^(x0length)-1);

blength:=x0length;
x0bin:=BinRep(x0);
blength:=bitsize;

x:=x0bin; x2:=x0bin;
for i in [1..2^(20)] do
x:=H(x);
x2:=H(H(x2));
if x=x2 then
xJ:=x0bin; xJplusI:=x;
for j in [1 .. i] do
xJnew:=H(xJ);
xJplusInew:=H(xJplusI);
if xJnew=xJplusInew then
```

```
Print(xJ, " "); Print(xJplusI, " ");
break;
else
xJ:=xJnew;
xJplusI:=xJplusInew;
fi;
od;
break;
fi;
od;
```

A.2 The decreasing size of $H^k(S)$

When testing small examples of the small space birthday attack, one notices immediately that the index I is often a multiple of a few fixed values. We note that this is expected, which we explain below.

Let ℓ be the length of output of H, and $S \subset \{0,1\}^\ell$ be the set of all possible hash values of H. For $k \in \mathbb{N}$ define $H^k(S) = \{H^k(m) \ : \ m \in S\}$ where $H^0 := S$. For any element $H^k(m) \in H^k(S)$, we have $H^k(m) = H^{k-1}(H(m)) \in H^{k-1}(S)$, since by definition $H(m) \in S$. Thus

$$S \supseteq H(S) \supseteq H^2(S) \supset H^3(S) \supseteq \ldots.$$

At each step, the size of $H^k(S)$ must either decrease or stay the same. Thus, at some point we must have $H^k(S) = H^{k-1}(S)$. This forms a bijection on $H^{k-1}(S)$, that is, a permutation of the elements of $H^{k-1}(S)$. The orbits in this permutation become the possible values of the divisors I, as we have that our collision is of the form $H(x_j) = H(x_{I+j})$. This idea is behind many cycle detection algorithms.

For each step that we do not have $H^k(S) = H^{k-1}(S)$, we can also measure the size to

which we expect our set to reduce. Namely, suppose that a hash function exhibits the uniform distribution, and that there are N possible hash values, $h_1, \ldots, h_N$. Further, suppose that we have K bitstrings we are hashing. The expected number of hash values we obtain from these K bitstrings is $\sum_i^N E(\delta_i)$ where δ_i is 1 if h_i is obtained, and 0 otherwise. For each δ_i, this means $E(\delta_i)$ is the probability that at least one of the K bit strings hashes to x_i, which under the assumption of uniform distribution is $1 - (\frac{N-1}{N})^k$. Thus, the expected number of hash values we obtain from these K bitstrings is

$$\sum_i^N \left(1 - \left(\frac{N-1}{N}\right)^K\right) = N\left(1 - \left(\frac{N-1}{N}\right)^K\right). \tag{A.2.1}$$

A.3 Some preliminary results

We conclude this appendix with some results that were computed in GAP. In what follows, we let $p^n = 3^3$, and use the preset choice of $r_n(x) = x^3 - x + 1$ in GAP, so that the image of x in $\mathbb{F}_q$ is itself a primitive root. We notice that, in the case $p = 3$, $\det(A) = f$ and $\det(B) = \tilde{f}$ for each $G_i(f, \tilde{f})$ in Table 5.1. With this in mind, our choices of generators were $G_1(-x^2, x)$, $G_3(x, -x^2)$, $G_3(x, -x^2+x)$, $G_3(x, -x^2-x)$; we note that $-x^2 = x^{15}$, $-x^2+x = x^{17}$, $-x^2-x = x^{23}$. Since $p^n - 1 = 26$, $\det(\pi(A))$ and $\det(\pi(B))$ are distinct primitive roots. Further, we have checked computationally that in each of these cases $\langle \pi(A), \pi(B) \rangle = \mathrm{GL}_2\big(\mathbb{F}_{3^3}\big)$. This also gives that $[\pi(A)]$ and $[\pi(B)]$ generate all of $\mathrm{PGL}_2\big(\mathbb{F}_{3^3}\big)$ (we note our examples are too small to apply Proposition 5.3.7).

We first consider implementing the above elements of $\mathfrak{H}$ as hash functions over $\mathrm{PGL}_2\big(\mathbb{F}_q\big)$. To do so, we use the canonical representation given in Section 5.5.4, which gives outputs of length $\ell = 3\lceil \log_2(3^3) \rceil + 1 = 16$.

Let $S' := \{M \in S \ : \ \det(M) \text{ is a square in } \mathbb{F}_q\}$, where as above $S \subset \{0,1\}^\ell$ is the set of all possible hash values of H. Note that $|S'| = |\frac{1}{2}\mathrm{PGL}_2(\mathbb{F}_{3^3})| = 9828$ (see Section 5.5.4). Further, as for each of our choices $\det(B)$ is an odd power of $\det(A)$, and our hash values are bitstrings of even length, they must themselves hash to elements of $\mathrm{PGL}_2(\mathbb{F}_q)$ with determinant a square in $\mathbb{F}_q$ (see Section 5.5.4). Thus, we see that $H^k(S) \subset S'$ for each $k \geq 1$.

In Figure A.1, we compare the size of $H^{k-1}(S')$ for $1 \leq k \leq 25$ for implementations of our choices of G_i as generators of hash functions over $\mathrm{PGL}_2(\mathbb{F}_{3^3})$ compared to the expected model given uniform distribution (A.2.1).

We see that Figure A.1 shows that our small examples have close fit to the expected model under the hypothesis of uniform distribution, especially in the cases of $G_3(x, -x^2 + x)$ and $G_3(x, -x^2 - x)$.

We note that for the four examples shown in Figure A.1, where $n = 3$ and $\delta = 2$, we have found that the earliest collisions have at least one bitstring of length either 6 or 8, depending on G_i and f, $\tilde{f}$. We note that if the the small modifications property holds over $\mathrm{PGL}_2(\mathbb{F}_q)$ (see Section 5.5.4), we expect two distinct bitstrings cannot yield a collision unless one has length at least n/δ.

Next we implement the hash function in $\mathrm{GL}_2(\mathbb{F}_{3^3})$, but with padding, as suggested in Section 5.5.3. We have chosen a simple padding that has the property that padded messages will have the same number of ones and zeros, as suggested in Section 5.5.3. Namely, we see that the output of the unpadded construction is $4\lceil \log_2(3^3) \rceil = 20$. When measuring the size of $H^k(U)$, we thus are hashing bitstrings of length 20, since we are taking elements from $H^k(U) \subset S$. We can therefore pad these inputs to bitstrings of length 40 in the following way.

Let $m = m_1 \dots m_{20}$ be a bitstring of length 20. We define the padded bitstring of m

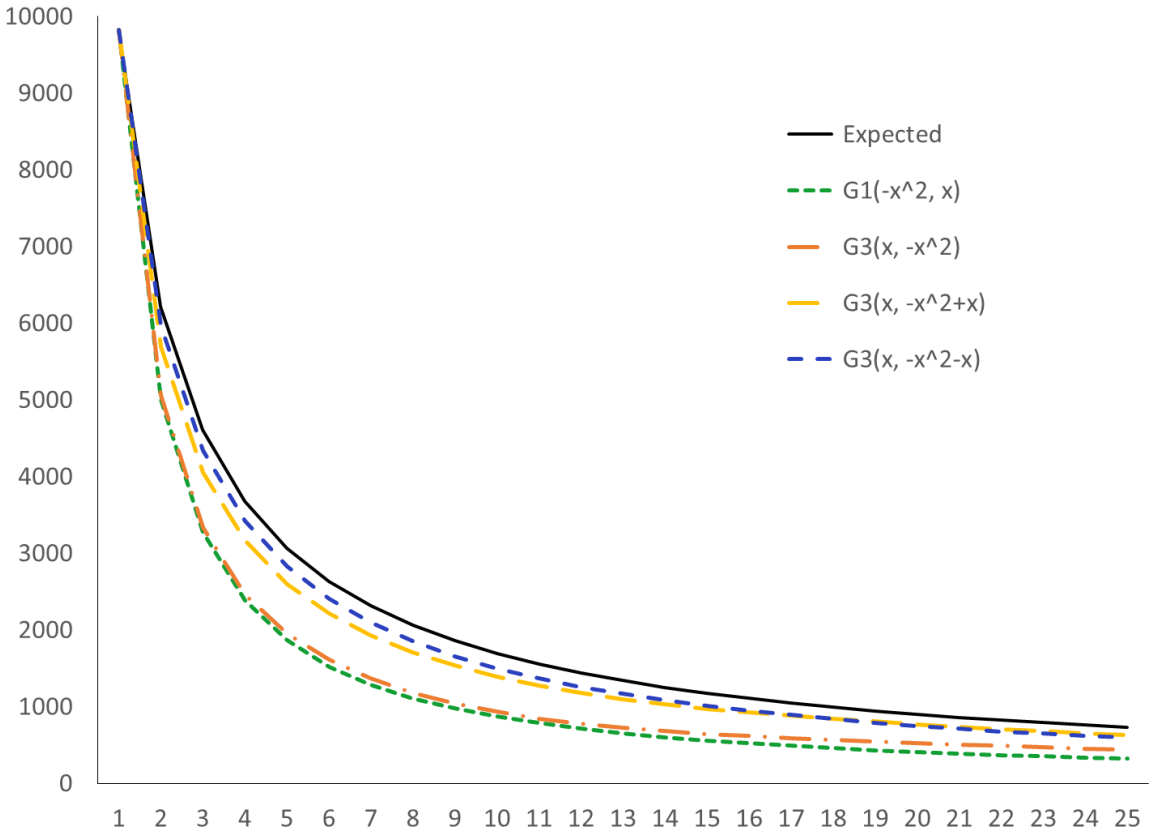

Figure A.1: Comparision of the size of $H^{k-1}(S')$ for $1 \leq k \leq 25$ between $G_1(-x^2, x)$, $G_3(x, -x^2)$, $G_3(x, -x^2 + x)$, $G_3(x, -x^2 - x)$, and the expected model (A.2.1), where S' is the set of possible hash values with square determinant for each choice of generators, $G = \mathbb{F}_{3^3}$, and the number of possible hash values of messages of even length is $N = |\frac{1}{2}\mathrm{PGL}_2(\mathbb{F}_{3^3})| = 9828$.

to be the bitstring $m_1 \dots m_{20}\tilde{m}_1 \dots \tilde{m}_{20}$ where $\tilde{m}_i = 1$ if $m_i = 0$ and $\tilde{m}_i = 0$ if $m_i = 1$ for $1 \leq i \leq 20$. For example, the bitstring 10011 00001 00011 00000 is padded to the bitstring 10011 00001 00011 00000 01100 11110 11100 11111. We note that the hashes of messages padded in this way all have the same determinant.

In Figure A.2 we compare the size of $H^{k-1}(U)$ for $1 \leq k \leq 25$ for implementations of our choices of padded G_i as generators of hash functions over $\mathrm{GL}_2(\mathbb{F}_{3^3})$ to the unpadded extended Zémor-Tillich (3.3.1) (which we recall generated $\mathrm{SL}_2(\mathbb{F}_{3^3})$ [1]) and the expected model given uniform distribution (A.2.1) for U a subset of $10,000$

possible hash values. Note U is different in each case, as it is randomly chosen and dependent on our choice of generators. We note from Section 5.5.3 we expect the number of possible hash values to be $N = |\mathrm{SL}_2(\mathbb{F}_{3^3})| = 19656$.

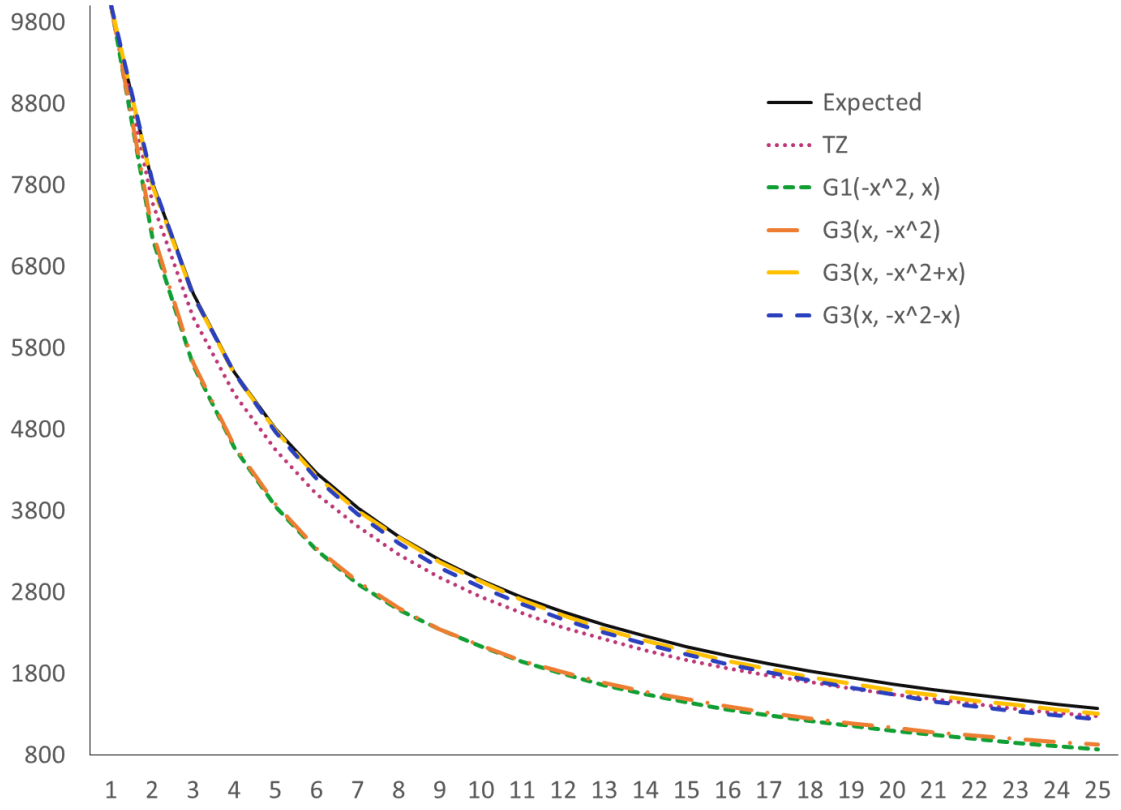

Figure A.2: Comparison of the the size of $H^{k-1}(U)$ for $\leq k \leq 25$ between padded implementations of $G_1(-x^2, x)$, $G_3(x, -x^2)$, $G_3(x, -x^2 + x)$, $G_3(x, -x^2 - x)$, unpadded extended Zémor-Tillich (3.3.1), and the expected model (A.2.1), where U is a set of $10,000$ possible hash values, $G = \mathbb{F}_{3^3}$, and the number of possible hash values is $N = |\mathrm{SL}_2(\mathbb{F}_{3^3})| = 19656$.

Figure A.2 shows that with padding, our small examples closely resemble the expected model under the hypothesis of uniform distribution, and again have an especially close fit for $G_3(x, -x^2 + x)$ and $G_3(x, -x^2 - x)$. In particular, Figure A.2 supports our hypothesis from Section 5.7 that the padded hash values of elements of $\mathfrak{H}$ will follow the uniform distribution.

Appendix B

Generating a free monoid

Another natural way of finding alternative generators of the Zémor-Tillich hash function that satisfy the small modifications property is to extend Tillich and Zémor's degree argument in Lemma 3.1.3. In this section, we present such an extension.

Lemma B.0.1. *Suppose*

$$A = \begin{pmatrix} f_a & f_b \\ f_c & f_d \end{pmatrix} \qquad B = \begin{pmatrix} g_{a'} & g_{b'} \\ g_{c'} & g_{d'} \end{pmatrix} \tag{B.0.1}$$

are matrices in $\mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$ *such that* f_i, g_i *are a monic polynomials in* $\mathbb{F}_p(x)$ *of degree* i, *and* $a, b, c, d, b', c', d' \in \mathbb{N}_0$ *such that* $a = a'$, $a > b, c > d$, *and* $a > b', c' > d'$.

Let $V = v_1 \cdots v_k$ *for* $k \geq 2$ *where each* $v_i \in \{A, B\}$. *Then* V *is of the the form*

$$V = \begin{pmatrix} t_{ak} & t_{a(k-1)+b} \\ t_{a(k-1)+c} & t_{a(k-2)+b+c} \end{pmatrix} \text{ if } v_1 = A \text{ and } v_k = A$$

$$V = \begin{pmatrix} t_{ak} & t_{a(k-1)+b'} \\ t_{a(k-1)+c} & t_{a(k-2)+b'+c} \end{pmatrix} \text{ if } v_1 = A \text{ and } v_k = B$$

$$V = \begin{pmatrix} t_{ak} & t_{a(k-1)+b} \\ t_{a(k-1)+c'} & t_{a(k-2)+b+c'} \end{pmatrix} \text{ if } v_1 = B \text{ and } v_k = A$$
$$V = \begin{pmatrix} t_{ak} & t_{a(k-1)+b'} \\ t_{a(k-1)+c'} & t_{a(k-2)+b'+c'} \end{pmatrix} \text{ if } v_1 = B \text{ and } v_k = B$$

where t_i denotes some monic polynomial in $\mathbb{F}_p[x]$ of degree i.

We note that certain sets of these strict inequalities can be relaxed. As an example, we could use $a \geq c, c'$, and $b \geq d, d'$. As another example, one could allow for instance $a \geq b$ and $b \geq d$, $a \geq b'$ and $b' \geq d'$, $c \geq d$ and $a \geq c$, or $c' \geq d'$ and $a \geq c'$. We leave the details of this, as it is not crucial.

Further, for A, B as in Lemma B.0.1, an identical argument shows that Corollary B.0.1 holds if instead $d = d'$ and $d > c, b > a$, $d' > c', b' > a'$. Such a result could also be seen by conjugating A and B by $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Proof: We proceed by induction on k. For $k = 2$ we see by direct calculation that

$$AA = \begin{pmatrix} f_a^2 + f_b f_c & f_a f_b + f_b f_d \\ f_c f_a + f_d f_c & f_c f_b + f_d f_d \end{pmatrix} \quad AB = \begin{pmatrix} f_a g_a + f_b g_{c'} & f_a g_{b'} + f_b g_{d'} \\ f_c g_a + f_d g_{c'} & f_c g_{b'} + f_d g_{d'} \end{pmatrix}$$
$$BA = \begin{pmatrix} g_a f_a + g_{b'} f_c & g_a f_b + g_{b'} f_d \\ g_{c'} f_a + g_{d'} f_c & g_{c'} f_b + g_{d'} f_d \end{pmatrix} \quad BB = \begin{pmatrix} g_a^2 + g_{b'} g_{c'} & g_a g_{b'} + g_{b'} g_{d'} \\ g_{c'} ga + g_{d'} g_{c'} & g_{c'} g_{b'} + g_{d'} g_{d'} \end{pmatrix}.$$

Considering the relations $a > b, c > d$, and $a > b', c' > d'$ we have our result.

Consider $V = v_1 \cdots v_{k+1}$. Since the proof is identical, we suppose $v_{k+1} = A$. Then by

our inductive hypothesis

$$v_2....v_{k+1} = \begin{pmatrix} t_{ak} & t_{a(k-1)+b} \\ t_{a(k-1)+l} & t_{a(k-2)+b+l} \end{pmatrix}$$

where l is either c or c'.

We see that if $v_1 = A$ then

$$V = A \cdot \begin{pmatrix} t_{ak} & t_{a(k-1)+b} \\ t_{a(k-1)+l} & t_{a(k-2)+b+l} \end{pmatrix} = \begin{pmatrix} f_a t_{ak} + f_b t_{a(k-1)+l} & f_a t_{a(k-1)+b} + f_b t_{a(k-2)+b+l} \\ f_c t_{ak} + f_d t_{a(k-1)+l} & f_c t_{a(k-1)+b} + f_d t_{a(k-2)+b+l} \end{pmatrix}$$

whereas if $v_1 = B$ then

$$V = B \cdot \begin{pmatrix} t_{ak} & t_{a(k-1)+b} \\ t_{a(k-1)+l} & t_{a(k-2)+b+l} \end{pmatrix} = \begin{pmatrix} g_a t_{ak} + g_{b'} t_{a(k-1)+l} & g_a t_{a(k-1)+b} + g_{b'} t_{a(k-2)+b+l} \\ g_{c'} t_{ak} + g_{d'} t_{a(k-1)+l} & g_{c'} t_{a(k-1)+b} + g_{d'} t_{a(k-2)+b+l} \end{pmatrix}.$$

Again, considering the relations $a > b > c > d$, and $a > b' > c' > d'$ we have our result. □

From this we obtain the following corollary.

Corollary B.0.2. *Let A, B be as in Lemma B.0.1, such that A, $B \in \mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$. Suppose either $b \neq b'$ or $c \neq c'$, and $V = v_1 \cdots v_r$ and $W = w_1 \cdots w_s \in \mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$, where v_i, $w_j \in \{A, B\}$. We then have that $V = W$ if and only if $r = s$ and $v_i = w_i$ for $1 \leq i \leq r$.*

Proof: Suppose $V = W$. Without loss of generality suppose $b \neq b'$.

From Lemma B.0.1 we know the $(1,1)-$entry of V is of degree ra and the $(1,1)-$entry of W is of degree sa. Since $V = W$ this implies that $r = s$.

Now from Lemma B.0.1 we know the $(1,2)$-entry of V (resp. W) is of degree $a(k-1)+b$

if $v_r = A$ (resp. $w_r = A$) and $a(k-1)+b'$ if $v_r = B$ (resp. $w_r = B$). Since $b \neq b'$ this implies that $v_r = w_r$.

Thus we have that $v_1...v_{r-1} = w_1...w_{r-1}$, and repeating the argument above shows that $v_i = w_i$ for $1 \leq i \leq r-1$. □

Note that Corollary B.0.2 implies in particular that A and B as in Lemma B.0.1 generate a free monoid in $\mathrm{M}_{2\times 2}\big(\mathbb{F}_p[x]\big)$. With Proposition 5.3.1, part (a), Corollary B.0.2 thus gives the following result.

Corollary B.0.3. *Let p be a prime, $r_n(x)$ be an irreducible polynomial over $\mathbb{F}_p$, and A, B be as in Lemma B.0.1 such that A, $B \in \mathrm{GL}_2\big(\mathbb{F}_p((x))\big)$. Suppose that H is the associated hash function of (A, B) and $r_n(x)$ as in Definition 5.2.3 and m, m' are distinct bitstrings in $\{0,1\}^*$ of lengths r and s, such that ar, $as < n$. Then $H(m) \neq H(m')$.*

One can check that Corollary B.0.3 applies to Tillich and Zémor's original choice of generators, as well as the alternative choices proposed at the end of Section 3.2.5. As well, for the case $p = 2$ we note this method provides generators of lower degree than we were able to obtain in Section 5.1.

However, our method of choosing generators using the Free Generators Theorem, Theorem 4.4.2, is still preferable in many ways. For example, in Section 5.3 we found precise conditions on our parameters for generating a large enough set of hash values, and saw that the freeness allows for us to prevent against attacks using short relations. Our theorem further allows more freedom in the choice of generators, which we believe would make the corresponding hash function less susceptible to attacks, such as the palindrome attack, which are dependent on the structure of the generators themselves.

Moreover, the Free Generators Theorem provides many choices of generators that could not be produced using the degree argument, such as $G_1(f, \tilde{f}), \ldots, G_6(f, \tilde{f})$ for any choice of A, B.

Bibliography

[1] Kanat Abdukhalikov and Chul Kim. On the security of the hashing scheme based on SL_2. In *Fast Software Encryption*, pages 93–102. Springer, 1998.

[2] Gora Adj, Alfred Menezes, Thomaz Oliveira, and Francisco Rodríguez-Henríquez. Computing discrete logarithms using Joux's algorithm. *ACM Comm. Computer Algebra*, 49(2):60, 2015.

[3] Michael Aschbacher. On the maximal subgroups of the finite classical groups. *Inventiones mathematicae*, 76(3):469–514, 1984.

[4] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[5] Emmanuel Breuillard and Tsachik Gelander. On dense free subgroups of Lie groups. *Journal of Algebra*, 261(2):448–467, 2003.

[6] Lisa Bromberg, Vladimir Shpilrain, and Alina Vdovina. Navigating in the Cayley graph of $SL_2(\mathbb{F}_p)$ and applications to hashing. *Semigroup Forum*, 94(2):314–324, 2017.

[7] Denis X Charles, Kristin E Lauter, and Eyal Z Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, 2009.

[8] Chris Charnes and Josef Pieprzyk. Attacking the SL_2 hashing scheme. *Advances in Cryptology ASIACRYPT'94*, pages 322–330, 1995.

[9] Don Coppersmith. Fast evaluation of logarithms in fields of characteristic two. *IEEE transactions on information theory*, 30(4):587–594, 1984.

[10] Don Coppersmith, Andrew M Odlzyko, and Richard Schroeppel. Discrete logarithms in GF(p). *Algorithmica*, 1(1-4):1–15, 1986.

[11] Giacomo De Meulenaer, Christophe Petit, and Jean-Jacques Quisquater. Hardware implementations of a variant of the Zémor-Tillich hash function: Can a provably secure hash function be very efficient? *IACR Cryptology ePrint Archive*, 2009:229, 2009.

[12] Leonard Eugene Dickson. *Linear groups with an exposition of the Galois field theory*. Dover, 1958.

[13] Morris J Dworkin. SHA-3 standard: Permutation-based hash and extendable-output functions. Technical report, NIST - FIPS, 2015. (Accessed from https://www.nist.gov/publications/sha-3-standard-permutation-based-hash-and-extendable-output-functions, June 2018).

[14] Jean-Charles Faugere, Ludovic Perret, Christophe Petit, and Guénaël Renault. New subexponential algorithms for factoring in $SL(2, \mathbb{F}_{2^n})$. *Preprint*, 2011. (Accessed from https://www.cs.bham.ac.uk/ petitcz/, May 2017).

[15] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.9.1*, 2018. (Accessed from https://www.gap-system.org/, June 2018).

[16] Willi Geiselmann. A note on the hash function of Tillich and Zémor. In *Cryptography and coding (Cirencester, 1995)*, volume 1025 of *Lecture Notes in Comput. Sci.*, pages 257–263. Springer, Berlin, 1995.

[17] Ph. Godlewski and P. Camion. Manipulations and errors, detection and localization. In *Advances in cryptology—EUROCRYPT '88 (Davos, 1988)*, volume 330 of *Lecture Notes in Comput. Sci.*, pages 97–106. Springer, Berlin, 1988.

[18] Markus Grassl, Ivana Ilić, Spyros Magliveras, and Rainer Steinwandt. Cryptanalysis of the Tillich-Zémor hash function. *Journal of Cryptology*, 24(1):148–156, 2011.

[19] Richard D Jenks and Robert S Sutor. *AXIOM: The Scientific Computation System*. New York: Springer-Verlag, 1992.

[20] Hyungrok Jo. *Cryptanalysis on Hash Functions Based on Ramanujan Graphs*. PhD thesis, Kyushu University, 2017.

[21] Hyungrok Jo. Hash functions based on Ramanujan graphs. In *Mathematical Modelling for Next-Generation Cryptography*, pages 63–79. Springer, 2018.

[22] William M Kantor and Alexander Lubotzky. The probability of generating a finite classical group. *Geometriae Dedicata*, 36(1):67–87, 1990.

[23] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014.

[24] Oliver H King. The subgroup structure of finite classical groups in terms of geometric configurations. *Surveys in combinatorics*, 5:29–56, 2005.

[25] Kristin E Lauter, Denis X Charles, and Eyal Zvi Goren. Hash function constructions from expander graphs, June 3 2008. US Patent 7,382,876.

[26] Martin W. Liebeck and Aner Shalev. The probability of generating a finite simple group. *Geom. Dedicata*, 56(1):103–113, 1995.

[27] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[28] Alfred J. Menezes and Yi-Hong Wu. The discrete logarithm problem in GL(n, q). *Ars Combin.*, 47:23–32, 1997.

[29] Jill P Mesirov and Melvin M Sweet. Continued fraction expansions of rational expressions with irreducible denominators in characteristic 2. *Journal of Number Theory*, 27(2):144–148, 1987.

[30] Ciaran Mullan. *Some Results in Group-based crytography*. PhD thesis, University of London, 2011.

[31] Ciaran Mullan and Boaz Tsaban. SL_2 homomorphic hash functions: Worst case to average case reduction and short collision search. *Designs, Codes and Cryptography*, 81(1):83–107, 2016.

[32] Andrew M Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 224–314. Springer, 1984.

[33] Christophe Petit. Towards factoring in SL_2. *Designs, Codes and Cryptography*, pages 1–23, 2014.

[34] Christophe Petit, Giacomo de Meulenaer, and Jean-Jacques Quisquater. ZesT: an all-purpose hash function based on Zémor-Tillich, 2009. (Accessed from https://www.cs.bham.ac.uk/ petitcz/, September 2016).

[35] Christophe Petit, KE Lauter, and Jean-Jacques Quisquater. Cayley hashes: A class of efficient graph-based hash functions. *Preprint*, 2007. (Accessed from https://www.cs.bham.ac.uk/ petitcz/, May 2017).

[36] Christophe Petit, Kristin Lauter, and Jean-Jacques Quisquater. Full cryptanalysis of LPS and Morgenstern hash functions. In *International Conference on Security and Cryptography for Networks*, pages 263–277. Springer, 2008.

[37] Christophe Petit and Jean-Jacques Quisquater. Preimages for the Tillich-Zémor hash function. In *International Workshop on Selected Areas in Cryptography*, pages 282–301. Springer, 2010.

[38] Christophe Petit and Jean-Jacques Quisquater. Rubik's for cryptographers. *Notices Amer. Math. Soc.*, 60(6):733–740, 2013.

[39] Christophe Petit and Jean-Jacques Quisquater. Cryptographic hash functions and expander graphs: The end of the story? In *The New Codebreakers. Lecture Notes in Computer Science*, volume 9100, pages 304–311. Springer, 2016.

[40] Christophe Petit, Jean-Jacques Quisquater, Jean-Pierre Tillich, and Gilles Zémor. Hard and easy components of collision search in the Zémor-Tillich hash function: New attacks and reduced variants with equivalent security. In *Cryptographers Track at the RSA Conference*, pages 182–194. Springer, 2009.

[41] Christophe Petit, Nicolas Veyrat-Charvillon, and Jean-Jacques Quisquater. Efficiency and pseudo-randomness of a variant of Zémor-Tillich hash function. In *Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on*, pages 906–909. IEEE, 2008.

[42] Jean-Jacques Quisquater and Marc Joye. Authentication of sequences with the SL_2 hash function: Application to video sequences. *Journal of computer security*, 5(3):213–223, 1997.

[43] Andrew Richard Regenscheid. *An algebraic hash function based on* SL_2. PhD thesis, Iowa State University, 2007.

[44] Colva M Roney-Dougal. Conjugacy of subgroups of the general linear group. *Experimental Mathematics*, 13(2):151–163, 2004.

[45] Victor Shoup. Searching for primitive roots in finite fields. In *Proceedings of the twenty-second annual ACM symposium on theory of computing*, pages 546–554. ACM, 1990.

[46] Rainer Steinwandt, Markus Grassl, Willi Geiselmann, and Thomas Beth. Weaknesses in the $\mathrm{SL}_2(\mathbb{F}_{2^n})$ hashing scheme. In *Annual International Cryptology Conference*, pages 287–299. Springer, 2000.

[47] Michio Suzuki. *Group theory, Volume I.* Springer-Verlag, New York, 1982.

[48] Jean-Pierre Tillich and Gilles Zémor. Group-theoretic hash functions. *Algebraic Coding*, pages 90–110, 1994.

[49] Jean-Pierre Tillich and Gilles Zémor. Hashing with SL_2. In *Annual International Cryptology Conference*, pages 40–49. Springer, 1994.

[50] Jean-Pierre Tillich and Gilles Zémor. Collisions for the LPS expander graph hash function. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 254–269. Springer, 2008.

[51] Jacques Tits. Free subgroups in linear groups. *Journal of Algebra*, 20(2):250–270, 1972.

[52] Gilles Zémor. Hash functions and graphs with large girths. In *Advances in Cryptology EUROCRYPT91*, pages 508–511. Springer, 1991.

[53] Gilles Zémor. Hash functions and Cayley graphs. *Designs, Codes and Cryptography*, 4(3):381–394, 1994.