Cryptographic hash functions from expander graphs

Denis Charles[1], Eyal Goren[2], and Kristin Lauter[1]

[1] Microsoft Research, Redmond, WA 98052, USA.
[2] McGill University, Montréal, Canada H3A 2K6

Abstract. We propose constructing provable collision resistant hash functions from expander graphs. As examples, we investigate two specific families of optimal expander graphs for provable hash function constructions: the families of Ramanujan graphs constructed by Lubotzky-Phillips-Sarnak and Pizer respectively. When the hash function is constructed from one of Pizer's Ramanujan graphs, (the set of supersingular elliptic curves over $\mathbb{F}_{p^2}$ with ℓ-isogenies, ℓ a prime different from p), then collision resistance follows from hardness of computing isogenies between supersingular elliptic curves. We estimate the cost per bit to compute these hash functions, and we implement our hash function for several members of the LPS graph family and give actual timings.

1 Introduction

With the untimely demise of SHA-1, NIST is soliciting proposals for new cryptographic hash functions to standardize. The goal is to construct an efficiently computable hash function which is collision resistant. We call it a *provable hash* if to compute a collision is to solve some other well-known hard problem such as factoring or discrete log, for example as in the scheme proposed in [4]. We propose constructing provable cryptographic hash functions from expander graphs. Expander graphs are graphs in which the neighbor set of any "not too large" subset of vertices contains many new vertices. This property of expander graphs leads to other interesting properties, one important example being the rapid mixing of Markov chains on expanders. In our construction the input to the hash function is used as directions for walking around a graph, and the ending vertex is the output of the hash function. Our construction can be applied to any expander graph, but we give here two families of optimal expander graphs, and investigate the efficiency and collision resistance properties of these two families. The two families are the Ramanujan graphs constructed by Pizer and Lubotzky-Phillips-Sarnak (LPS) respectively. Ramanujan graphs are optimal expander graphs, in a technical sense (see section 2), and thus have excellent mixing properties. For these two families, the collision resistance follows from arithmetic properties of the graphs' constructions.

When constructing a hash function from the Ramanujan graph of supersingular elliptic curves over $\mathbb{F}_{p^2}$ with ℓ-isogenies, ℓ a prime different from p, as in Pizer ([13]), finding collisions is at least as hard as computing isogenies between supersingular elliptic curves. This is believed to be a very difficult problem (see Section 6 below), and the best known algorithm currently known solves the problem in $O(\sqrt{p}\log^2 p)$ time. Thus we propose to set p to be a 256-bit prime, to get 128 bits of security from the resulting hash function.

To compute the hash function from Pizer's graph when $\ell = 2$ requires roughly $2\log(p)$ field multiplications per bit of input to the hash function. This is roughly the same efficiency as a provable hash based on the ECDLP, and relatively inefficient compared to the provable hash function [4], but our construction has the advantage that the output of our hash function is $\log(p)$ bits, and efficiency may be improved with optimizations.

Hash functions from LPS graphs are more efficient to compute than those from Pizer's graphs. Applying our construction gives a hash function similar to the one proposed by Zémor and Tillich [18], [19]. Finding collisions reduces to a another problem which is also believed to be difficult (see Section 7). To compute the hash function requires only 7 field multiplications per bit of input, but the field size may need to be bigger (1024 bit prime p instead of 256 bits, for example), and the output is $4\log(p)$ bits. We have implemented this hash function for primes of varying size and we give unoptimized timings in Section 7.

These hash functions may be too inefficient to be applied in all situations, but would be appropriate for some protocols where a secure hash function is needed and other operations are on the same order of magnitude. This is the case, for example, for public key cryptographic protocols such as authenticated key exchange. An important property of our hash functions is that the hard mathematical problem underlying the collision resistance appears to be independent from other known hard problems such as factoring and ECDLP (elliptic curve discrete logarithm problem). For the Pizer graph, the hard mathematical problem is finding an isogeny between two given supersingular elliptic curves, and we explain in Section 6 how this problem is related to the problem of finding lattice vectors of a given norm. For the LPS graphs, the underlying hard problem is a representation problem in group theory.

2 Background and Definitions

Hash functions. A hash function maps bit strings of some finite length to bit strings of some fixed finite length, and must be easy to compute. We are concerned in this paper with unkeyed hash functions which are collision resistant. Unkeyed hash functions do not require a secret key to compute the output.

Elliptic curves. Let p be a prime greater than 3 and q a power of p. An *elliptic curve* E over the field $\mathbb{F}_q$ of q elements can be given by a *Weierstrass equation*

$$E\colon\ y^2 = x^3 + ax + b, \qquad a, b \in \mathbb{F}_q,$$

where the polynomial $x^3 + ax + b$ has no repeated roots. One adds a "point at infinity" 0_E, which, when the curve is given in projective space as $y^2z = x^3 + axz^2 + bz^3$, is the point $(0\colon 1\colon 0)$. There is a group structure on an elliptic curve, given by polynomial equations, such that for every finite extension $\mathbb{F}_{q^r}$ the $\mathbb{F}_{q^r}$-rational points of E, $E(\mathbb{F}_{q^r}) := \{(x, y) : y^2 = x^3 + ax + b, x, y \in \mathbb{F}_{q^r}\} \cup \{0_E\}$, are an abelian group. Given two elliptic curves E_1, E_2 over $\mathbb{F}_q$, a *homomorphism* $f\colon E_1 \to E_2$ is a morphism of algebraic curves (i.e., a polynomial map) that respect the group laws. A non-zero homomorphism is called an *isogeny*. An isogeny is automatically surjective and has a finite kernel whose cardinality is called the *degree of the isogeny*. For example, for any positive integer n, the multiplication-by-n map $[n] : E \to E$ is an isogeny of degree n^2. If p does not divide n, then $\ker[n] \cong (\mathbb{Z}/n\mathbb{Z}) \times (\mathbb{Z}/n\mathbb{Z})$. In particular, if $\ell \neq p$ is a prime, there are precisely $\ell + 1$ subgroups of order ℓ. Another example is the Frobenius morphism. Let $E/\mathbb{F}_q$ be given by the equation $y^2 = x^3 + ax + b$, then the elliptic curve $E^{(p)}$ is the curve given by the equation $y^2 = x^3 + a^p x + b^p$. There is a canonical isogeny $Fr : E \to E^{(p)}$ given by $(x, y) \mapsto (x^p, y^p)$. The degree of this isogeny is p.

The *j-invariant* of E is the quantity $1728\frac{4a^3}{4a^3+27b^2}$. Two elliptic curves over $\mathbb{F}_q$ are isomorphic over a finite extension $\mathbb{F}_{q^r}$ if and only if they have the same j-invariant. Given an element $j \in \mathbb{F}_q$, there is an elliptic curve E over $\mathbb{F}_q$ with $j(E) = j$. For example, one may take $E\colon\ y^2 = x^3 + \frac{3j}{1728-j}x + \frac{2j}{1728-j}$ ($y^2 = x^3 + x$ if $j = 1728$ and $y^2 = x^3 + 1$ if $j = 0$).

An elliptic curve E over $\mathbb{F}_q$ is called *supersingular* if for every finite extension $\mathbb{F}_{q^r}$ there are no point in $E(\mathbb{F}_{q^r})$ of exact order p. The j-invariants of supersingular elliptic curves are called *supersingular j-invariants*. They all lie in $\mathbb{F}_{p^2}$, in particular there are finitely many such j-invariants.

For more on elliptic curves and the various characterizations of supersingular elliptic curves see [16].

Expander graphs. Let $G = (V, E)$ be a graph with vertex set V and edge set E. We will deal with undirected graphs, and say a graph is k-regular if each vertex has k edges coming out of it. An expander graph with N vertices has *expansion constant* $c > 0$ if for any subset $U \subset V$ of size $|U| \leq \frac{N}{2}$, the boundary $\Gamma(U)$ of U (which is all neighbors of U minus all elements of U) has size $|\Gamma(U)| \geq c|U|$. An alternate definition of the expansion constant requires that for any subset $U \subset V$, the boundary union all elements of U has size satisfying:

$$|\Gamma(U) \cup U\ | \geq \min\{(1 + c)|U|, \frac{N}{2} + 1\}.$$

It follows from the second definition that any expander graph is connected (see [5] for more background on expander graphs).

There is also an algebraic way to define the expansion property of a graph. The adjacency matrix of an undirected graph is symmetric, and therefore all its eigenvalues are real. For a connected graph, G, the largest eigenvalue is k, and all others are strictly smaller ([5, Lecture 9, Fact 5.6, 5.7]). Order the eigenvalues as follows:

$$k > \mu_1 \geq \mu_2 \geq \cdots \geq \mu_{N-1}.$$

Then the expansion constant c can be expressed in terms of the eigenvalues as follows: ([2])

$$c \geq \frac{2(k - \mu_1)}{3k - 2\mu_1}.$$

Therefore, the smaller the eigenvalue μ_1, the better the expansion constant. A theorem of Alon-Boppana says that for an infinite family X_m of connected, k-regular graphs, with the number of vertices in the graphs tending to infinity, that $\liminf \mu_1(X_m) \geq 2\sqrt{k-1}$. This motivates the definition of a *Ramanujan graph*, a k-regular connected graph which satisfies $\mu_1 \leq 2\sqrt{k-1}$. A family of k-regular Ramanujan graphs is optimal with respect to the size of μ_1.

3 Construction of a hash function from an expander graph

The use of expander graphs to produce pseudo-random behaviour is well-known to complexity theorists. The idea here is to use expander graphs to produce hash functions which are collision-resistant. We give two examples of such graphs in the following sections.

Roughly speaking, the input to the hash is used as directions for walking around a graph (*without backtracking*), and the output of the hash function is the ending vertex of the walk. For a fixed hash function, the walk starts at a fixed vertex in the given graph. A family of hash functions can be defined by allowing the starting vertex to vary. We execute a walk on a k-regular expander graph by converting the input to the hash function to a base-$(k-1)$ number whose digits then dictate which edge to take at each step. Starting at the first vertex, each step of the walk chooses an edge emanating from that vertex to follow to get to the next vertex. At each step in the walk, the choice of the edge to follow is determined by the next digit of the (converted) input. We do not allow backtracking in the walk, so only $k-1$ choices for the next edge are allowed at each step.

A random walk on an expander graph mixes very fast so the output of the hash function will be uniform provided the input was uniformly random. The output of a random walk on an expander graph with N vertices tends to the uniform distribution after $O(\log(N))$ steps. More quantitatively: Define a sequence of random variables $X_0, X_1, \cdots, X_\ell$, where X_i is defined to be the label of the vertex at the i-th step of a random walk on an expander graph on N vertices. Then for every δ there is an $\ell = O\big(\log(1/\delta)\big)$ such that for every vertex v

$$\left| \Pr[X_\ell = v] - \frac{1}{N} \right| < \delta.$$

The constant implied by the O-notation does not depend on the size of the graph. Thus, the observation made earlier follows, for instance, by setting $\delta = 1/N^2$. One can look at [5, Lecture 10, Theorem 6] for a proof.

4 Pizer's Ramanujan graphs

We refer the readers to [16, Ch. 3,5] for the relevant background on elliptic curves over finite fields.

The graphs. We first define the family of graphs ([13]). Let p and ℓ be two distinct prime numbers. Define the graph $G(p, \ell)$ to have vertex set, V, the set of supersingular elliptic curves over the finite field $\mathbb{F}_{p^2}$. Recall that an elliptic curve over a finite field of characteristic p is *supersingular* if it has no p-torsion over any extension field. Elliptic curves which are not supersingular are called *ordinary*. The property of being supersingular can be recognized from the Weierstrass equation of the curve [16, Chapter 5, Thm 4.1] or from its zeta function. Furthermore, supersingular elliptic curves are all defined over $\mathbb{F}_{p^2}$.

We label vertices with their j-invariants, which can be computed directly from the curve equation and are a priori elements of $\mathbb{F}_{p^2}$. The number of vertices of $G(p,\ell)$ is $\lfloor \frac{p}{12} \rfloor + \epsilon$, where $\epsilon \in \{0,1,2\}$ depending on the congruence class of p modulo 12 (loc. cit). Later, we will impose $p \equiv 1 \pmod{12}$, in which case $\epsilon = 0$. Since there are roughly $p/12$ distinct j-invariants, we will choose a linear congruential function to map j-invariants from $\mathbb{F}_{p^2}$ injectively into $\mathbb{F}_p$ for the output of the hash function. Thus the output of the hash funtion will be just $\log(p)$ bits. We propose to use a graph of cryptographic size $p \approx 2^{256}$.

The edge set is as follows: Given a supersingular j-invariant, j_1, choose an elliptic curve E_1 with $j(E_1) = j_1$ and a subgroup $H_1 \subseteq E_1$ of order ℓ. Connect j_1 to $j_2 := j(E_2)$ where E_2 is the elliptic curve E_1/H_1. A priori, since there are $\ell + 1$ subgroups of order ℓ this gives a directed $\ell + 1$-regular graph. However, if $p \equiv 1 \mod 12$ then the graph can be made into an undirected graph as follows: For each subgroup $H_1 \subseteq E_1$ of order ℓ, there is a canonical choice of subgroup $H_2 \subseteq E_2$ (of order ℓ) such that $E_2/H_2 \cong E_1$. Thus, we can identity the edges associated to H_1 with the edge associated to H_2. For a more explicit description of the graph (and how to compute it) see below.

The Ramanujan property of this graph follows from the fact that the adjacency matrix (called the Brandt matrix) gives the action of the ℓ^{th} Hecke operator on the space of weight 2 cusp forms of level p. So the bound on the eigenvalues follows from the corresponding result for modular forms (the Ramanujan-Petersson conjecture proven by Eichler and Shimura in this case).

Walking around the graph. For C a subgroup of the group of points on an elliptic curve E, Vélu in [17] gives explicit formulas for determining the equation of the isogeny $E \to E/C$ and the Weierstrass equation of the curve E/C. We give the formulas when ℓ is an*odd* prime. Let E be given by the equation

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

We define the following two functions in $\mathbb{F}_q(E)$. For $Q = (x,y)$ a point on $E - \{\mathcal{O}\}$, define

$$\begin{aligned} g^x(Q) &= 3x^2 + 2a_2x + a_4 - a_1y \\ g^y(Q) &= -2y - a_1x - a_3, \end{aligned}$$

and set

$$\begin{aligned} t(Q) &= 2g^x(Q) - a_1g^y(Q) \\ u(Q) &= (g^y(Q))^2 \\ t &= \sum_{Q \in (C - \{\mathcal{O}\})} t(Q) \\ w &= \sum_{Q \in (C - \{\mathcal{O}\})} (u(Q) + x(Q)t(Q)). \end{aligned}$$

Then the curve E/C is given by the equation

$$Y^2 + A_1XY + A_3Y = X^3 + A_2X^2 + A_4X + A_6$$

where

$$\begin{aligned} &A_1 = a_1, A_2 = a_2, A_3 = a_3, \\ &A_4 = a_4 - 5t, A_6 = a_6 - (a_1^2 + 4a_2)t - 7w. \end{aligned}$$

From the Weierstrass equation of E/C we can easily determine the j-invariant of E/C. We apply Vélu's formulas for subgroups of order ℓ, and it is clear that this procedure can be done using $O(\ell)$ elliptic curve operations for each of the $\ell + 1$ groups of order ℓ.

5 Efficiency

Here are the steps to compute the output of the hash function when using supersingular elliptic curves and 2-isogenies (i.e., $\ell = 2$). Since there are 3 edges emanating from each vertex, and no backtracking is allowed in a walk, from each vertex, there are two choices of which edge to follow next, and this can be determined by 1 bit as follows. Start at a vertex E_1. Subgroups of E_1 of order 2 are each given by a single two-torsion point on the elliptic curve $E_1 : y^2 = f(x)$. The 3 non-trivial 2-torsion points are $P_i = (x_i, 0)$, where the cubic $f(x)$ factors as

$$(x - x_1)(x - x_2)(x - x_3)$$

over an extension field of degree at most 2. As an example, when computing the isogeny ϕ which corresponds to taking the quotient by $\langle P_1 \rangle$, both of the other 2-torsion points are mapped to the same 2-torsion point $\phi(P_2) = \phi(P_3)$ on the isogenous elliptic curve, E_2. In turn, the isogeny which corresponds to taking the quotient of E_2 by the subgroup generated by $\phi(P_2)$ is the dual isogeny $\hat{\phi}$ and leads back to E_1. So to choose the next step from E_2, it suffices to choose between the two other 2-torsion subgroups different from $\langle \phi(P_2) \rangle$. An efficient way to determine the 2 new 2-torsion points on E_2 is to keep $\tilde{x_1}$, the x-coordinate of $\phi(P_1)$, and to factor $(x - \tilde{x_1})$ out of the new cubic $f_2(x)$, leaving a quadratic to be factored. The roots of the quadratic can be ordered according to some convention, and one bit suffices to choose between them for the next step in the walk. So if the input bit length is n, then the hash function takes a walk of length n steps.

Using the Vélu's formulas [17] one calculates that if E is given by $y^2 = x^3 + a_4 x + a_6$ and the 2-torsion point Q is $(\alpha, 0)$ then the elliptic curve $E/\langle Q \rangle$ can be given by the equation

$$y^2 = x^3 - (4a_4 + 15\alpha^2)x + (8a_6 - 14\alpha^3).$$

Furthermore, the equation for the isogeny is

$$(x, y) \mapsto \left(x + \frac{(3\alpha^2 + a_4)}{x - \alpha}, y - \frac{(3\alpha^2 + a_4)y}{(x - \alpha)^2} \right).$$

So summarizing, each vertex corresponds to an elliptic curve E_i given by an equation $y^2 = f_i(x)$, where $f_i(x)$ is a cubic. To compute the 2-torsion subgroups at each step, factor the cubic $f_i(x)$. At each step, calculate the 2-torsion by keeping the image of the other 2-torsion point (not used to quotient by), and then factoring the quadratic. After ordering, choose which one to quotient by and apply Vélu's formulas (field operations in $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$).

Cost per bit of input to the hash function:

1. Find the 2-torsion:
   a. Apply the isogeny from the previous step to one point: 7 field multiplications.
   b. Factor out the linear factor from the cubic $f_i(x)$: one field inversion.
   c. Factor the quadratic by completing the square and taking a square root: roughly $(3/2)\log(p)$ field multiplications plus a field inversion if $p \equiv 3 \pmod 4$. If $p \not\equiv 3 \mod 4$, then one can do this with $2 \log p$ multiplications in a residue ring of $\mathbb{F}_p[x]$ (Cippola's method). The construction of the residue ring requires $\log p$ random bits.
2. Order the 2-torsion.
3. Use Vélu to obtain the equation of the next elliptic curve: 9 field multiplications.

In addition, at the first vertex, the cubic defining the curve must be factored, and at the last step, computing the j-invariant requires several field multiplications and 1 field inversion.

An estimate of total cost can be made by estimating a field inversion as 5 field multiplications (and as usual not counting field additions). Here we did not distinguish which field multiplications occur in $\mathbb{F}_p$ and which occur in $\mathbb{F}_{p^2}$, but that is at most a factor of 2 difference. Also, the above is not optimized, so there may be better ways to do some of the steps.

Summary of efficiency of the hash function under these assumptions: cost per bit in terms of field multiplications is roughly $2\log(p)$.

6 Collision resistance

Definition. A hash function h is said to be *collision resistant* if it is computationally infeasible to find two distinct inputs, x, y, which hash to the same output $h(x) = h(y)$. This property is also called *strong collision resistance.*

Definition. A hash function h is said to be *preimage resistant* if, given any output of h (for which a corresponding input is not known), it is computationally infeasible to find an input, x, which hashes to that output. A hash function with this property is also called *one way.*

We will relate the collision resistance and preimage resistance properties of the hash function to the following mathematical problems, and then argue why these problems are hard.

Notation: Let h_i denote the hash function defined by letting the starting vertex for the walk be the supersingular elliptic curve E_i.

Problem 1. Produce a pair of supersingular elliptic curves over $\mathbb{F}_{p^2}$, E_1 and E_2, and two distinct isogenies of degree ℓ^n between them, $f_1 : E_1 \to E_2$, $f_2 : E_1 \to E_2$.

Problem 2. Given E, a supersingular elliptic curve over $\mathbb{F}_{p^2}$, find an endomorphism $f : E \to E$ of degree ℓ^{2n} that is not the multiplication by ℓ^n map.

In the above problems, by the phrase "find an isogeny" we mean a polynomial time procedure that given a point P evaluates the isogeny at that point.

Theorem 1. Finding a collision in the hash function h_i implies a solution to Problem 1 with $E_1 = E_i$, and a solution to Problem 2 with $E = E_i$.

Proof: Finding a collision for a hash function in this family amounts to finding two distinct paths between two vertices. For the hash function h_i, the first vertex $E_1 = E_i$. Assuming the hash function takes inputs of a fixed bit length, the paths must also have the same length. Finding two distinct paths in the graph from the vertex $E_1 = E_i$ to the vertex E_2 allows one to construct two distinct isogenies $\phi_1 : E_1 \to E_2$ and $\phi_2 : E_1 \to E_2$, $\phi_1 \neq \phi_2$, via composition of isogenies, where $E_1 = E_i$ and E_2 are supersingular elliptic curves over $\mathbb{F}_{p^2}$. Furthermore, the length constraint on the paths implies that $\deg \phi_1 = \deg \phi_2$, and the fact that the edges of the graph are ℓ-isogenies means that the degree of the two isogenies must be of the same ℓ-power degree. Taking the dual of ϕ_2, we get an isogeny $\hat{\phi_2} : E_2 \to E_1$. Now $\phi_1 \circ \hat{\phi_2} : E_1 \to E_1$ is an endomorphism of the elliptic curve E_1 of degree ℓ^{2n} for some n. This endomorphism cannot be the multiplication by ℓ^n map (which also has degree ℓ^{2n}), since $\phi_2 \neq \phi_1$. In other words, a collision also leads to a cycle[1] of even length in the graph. Thus, explicitly finding a collision in this hash function allows one to find two isogenies of the same ℓ-power degree between a pair of supersingular elliptic curves, and to find an ℓ^{2n}-degree endomorphism of a given supersingular elliptic curve $E = E_1 = E_i$ that is not the multiplication by ℓ^n map. In both cases, given a path or a cycle in the graph one can evaluate the isogeny by composing the isogenies along the path. Each step of the composition can be done efficiently by evaluating the isogeny via Vélu's formulas.□

Problem 3. Given E_1 and E_2, two supersingular elliptic curves over $\mathbb{F}_{p^2}$, find an isogeny $f : E_1 \to E_2$ of degree ℓ^n between them.

Theorem 2. Finding preimages for the hash function h_i implies a solution to Problem 3 with $E_1 = E_i$.

Proof: Given an output y to the hash function h_i, let E_2 be the supersingular elliptic curve over $\mathbb{F}_{p^2}$ whose j-invariant corresponds to y. To find an input x, such that $h_i(x) = y$,is to find a path in the graph of ℓ-isogenies from $E_1 = E_i$ to E_2. If the hash function takes inputs of length n, then for $\ell = 2$, the length of the path must be n, and thus the isogeny f must have degree ℓ^n. For general primes ℓ, the length of the path will be roughly $\frac{n}{\log_2(\ell)}$. □

Remark. Note the following relationships between these problems. As observed in Theorem 1, finding a collision implies a solution to Problem 1 and a solution to Problem 2. In the opposite direction, if a solution to Problem 2 is given in "factored" form, then it also implies a solution to Problem 1 and the ability to produce a collision. That is, if a cycle in the graph is found, written in "factored" form as a

[1] We use the term cycle rather loosely here, as we allow a cycle to intersect itself.

sequential list of vertices, it can be used to create two distinct paths between two vertices by following the cycle in two different directions until the paths meet. The path can be converted into an isogeny with $O(\ell \log^2(p))$ amount of work at each step. However, if a solution to Problem 1 or 2 is given as an isogeny or an endomorphism, specified either by a recipe for evaluation or by a subgroup to quotient by (the size of the subgroup ℓ^n would presumably be too large to make this practical), then it is not clear how to decompose the isogeny or endomorphism into the successive steps in the graph that would produce a collision. See the paragraph on factoring isogenies below. Note that the same is true for the equivalence of Problem 3 with preimage finding. If a solution to Problem 3 is given in terms of a path in the graph, then it can be used to find preimages.

Note also that a solution to Problem 3 implies a solution to Problem 1. This follows from the fact that a solver for Problem 3 can be used to solve Problem 2 by first taking a random walk on the graph with endpoints E_1 and E_2, and then asking the Problem 3-solver for another path between them. If the two paths are the same, repeat. Since the graph is an expander there are many distinct paths between any two vertices. The first path was chosen at random, and consequently, the probability that the Problem 3-solver produced the same path is low. Thus with high probability we will get two distinct paths from E_1 to E_2 and hence get a solution for Problem 1. In other words there is a probabilistic polynomial time reduction from Problem 1 to Problem 3. This is natural given Theorems 1 and 2, since a preimage finder can also be used to produce collisions.

A note on factoring isogenies: In the last paragraph we encountered the problem of writing an isogeny $f : E_0 \to E_n$ of degree ℓ^n as a composition of isogenies $\phi_n \circ \phi_{n-1} \circ \cdots \phi_1$ where $\deg \phi_i = \ell$. One might be tempted to use Corollary III.4.11 of [16] to solve this problem. The result states that an isogeny $f : E \to E'$ factors as

$$\begin{tikzcd} E \arrow[rr, "f"] \arrow[dr, "\phi"'] & & E' \\ & E_1 \arrow[ur, "\lambda"'] & \end{tikzcd}$$

if and only if $\ker \phi \subseteq \ker f$. For instance, one can use this criterion to find the first step in the "factorization" of the isogeny as follows: Given $f : E_0 \to E_n$, f factors as $f' \circ \phi_1$ iff $\ker \phi_1 \subseteq \ker f$. This can be checked by taking a point, P (say), that generates the subgroup $\ker \phi_1$ and checking whether $f(P)$ is the identity on E_n. Doing this for each of the $\ell + 1$ possibilities for the subgroup $\ker \phi_1$ we can identify the first step of the factorization. A problem arises with this approach if one carries it to subsequent steps of the factorization. Consider the second step of this process: one needs to check for each possible isogeny $\phi_2 : E_1 \to E_2$, whether $\ker \phi_2 \circ \phi_1 \subseteq \ker f$. Since $\deg \phi_2 \circ \phi_1 = \ell^2$, we know that $\ker \phi_2 \circ \phi_1 \subseteq E_0[\ell^2]$ the ℓ^2-torsion points on E_0. Furthermore, we know that $\ker \phi_1 \subseteq \ker \phi_2 \circ \phi_1$. Given that $E_0[\ell^2] \cong \mathbb{Z}/\ell^2\mathbb{Z} \times \mathbb{Z}/\ell^2\mathbb{Z}$, this means we have to find a $P \in E_0[\ell^2]$ of exact order ℓ^2 such that $\phi_1(P)$ lies in $\ker \phi_2$. Continuing this way, one would need to find points P in $E_0[\ell^k]$ of exact order ℓ^k. The problem is that such points in $E_0[\ell^k]$ are defined over large degree extensions of the field that E_0 is defined over. In general, this degree could be as large as ℓ^k and the finite field would have p^{ℓ^k} elements. Thus, even if f is of degree ℓ^n where n is $O(\log p)$ this approach becomes infeasible. As a consequence, obtaining a converse to Theorem 1 (turning a solution to Problem 1 or 2 into a procedure for finding hash collisions) seems unlikely.

Hardness of Problem 3 (Preimage resistance)
Since walks on an optimal expander graph quickly approximate the uniform distribution, we can argue heuristically that a Pollard-rho type attack on Problem 3 would succeed in time proportional to the square-root of the graph size, i.e. for the graph $G(p, \ell)$, in time $O(\sqrt{p} \log^2 p)$. Such an attack would not always find a path of the correct length, however. This appears to be the best attack known on any of these problems.

Problem 3 was introduced in [8], where it was argued that the problem is hard in both the ordinary and the supersingular cases. In [8], Galbraith gives an algorithm to find an isogeny between two given ordinary, isogenous elliptic curves which runs in time $O(p^{3/2} \log(p))$ assuming the Riemann hypothesis for imaginary quadratic fields. He notes that a similar algorithm to solve the same problem for supersingular elliptic curves runs in time $O(p \log(p))$. The ordinary case can also be described in another language as solving a discrete log problem in orders of class groups of imaginary quadratic number fields, which has been well-studied. Although subexponential index calculus methods apply ([9]), taking quadratic orders with large discriminant makes the problem as hard as factoring integers of that size ([10]). Note the difference between the ECDLP situation and here: problems on supersingular elliptic curves are not necessarily easier than the corresponding problem on ordinary elliptic curves. In fact, for our problem, there is no class group to work in for the supersingular case, and the degree map is a rank 4 quadratic form instead of rank 2 (see the Hardness of Problem 1).

Hardness of Problems 1 and 2 (Collision resistance)
To find a cycle in the graph is to solve Problem 2, so first of all, we will ensure that our graph has no short cycles (i.e. has large girth). We will put restrictions on the congruence class of the prime p to ensure that there are no short cycles in the graph as follows.

Translation into the language of quadratic forms. The problem of finding isogenies can be translated into the language of representation of numbers by quadratic forms. As explained in the proof of Theorem 1, finding two distinct isogenies ϕ_1, ϕ_2 between two elliptic curves E_1 and E_2 of degree ℓ^n leads to an endomorphism of degree ℓ^{2n} of E_1 that is not the multiplication by ℓ^n map. The degree map is a rank 4 positive definite quadratic form, which can also be described as the Norm map on a maximal order in a quaternion algebra. The endomorphism ring (over $\overline{\mathbb{F}}_p$) of a supersingular elliptic curve is isomorphic to a maximal order in the quaternion algebra $B = B_{p,\infty}$ over $\mathbb{Q}$ ramified only at p and ∞ ([16, Chapter 5, Theorem 3.1]). The maximal order is a rank 4 $\mathbb{Z}-$lattice. The existence of an endomorphism of degree ℓ^{2n} implies the existence of a *non-trivial* representation (i.e., not as the norm of ℓ^n) of the number ℓ^{2n} by the quadratic form that is the norm form on the lattice. Note though, that the best known algorithms for determining the endomorphism ring of a supersingular elliptic curve as a maximal order in B are exponential in p ([3]). Thus the process of translating the problem of finding cycles to the language of quadratic forms seems to be computationally hard in itself.

Ensuring that $G_{p,\ell}$ has no small cycles. We can use the machinery introduced above to efficiently find cases where $G(p, \ell)$ has no small cycles. By choosing p carefully relative to ℓ we can ensure that there are no cycles of length n for n in a given interval $[0, S]$. A non-trivial cycle of length $2n$ in the graph of ℓ-isogenies implies that the norm form of some maximal order in B represents ℓ^{2n} in a non-trivial way. If the cycle corresponds to an element x of norm ℓ^{2n} then that implies that the quadratic polynomial $X^2 - \mathrm{Tr}(x)X + \mathrm{Norm}(x)$ is irreducible, and so that p is ramified or inert in the field defined by the polynomial. To illustrate this, take $\ell = 2$ and $n = 1$. Then we consider $X^2 - \mathrm{Tr}(x)X + 4$. Since $b^2 - 4ac < 0$, the trace must satisfy $\mathrm{Tr}(x) \in \{-3, -2, -1, 0, 1, 2, 3\}$, so the field determined by the polynomial is $\mathbb{Q}(\sqrt{-1})$, $\mathbb{Q}(\sqrt{-3})$, $\mathbb{Q}(\sqrt{-7})$, or $\mathbb{Q}(\sqrt{-15})$. One then just needs to make sure p splits in all these fields, which by quadratic reciprocity is a congruence condition. So in this example it is enough that $p \equiv 1 \pmod 4$, $p \equiv 1 \pmod 3$, $p \equiv 1 \pmod 7$, and $p \equiv 1 \pmod 5$, so if p is congruent to 1 modulo $3 \cdot 4 \cdot 5 \cdot 7 = 420$ then there are no cycles of length 2. This idea can be applied in general to make sure there are no short cycles in the graph.

Choosing an appropriate starting vertex. One can apply the idea in the previous paragraph in a different way to exclude short cycles by choosing ℓ, p *and* the starting vertex carefully. We illustrate this when $p \equiv 3 \mod 4$ for ease of exposition even though we restrict to the case where $p \equiv 1 \mod 12$ for our construction. Proposition 5.2 of [12] then tells us that one maximal order, $\mathfrak{m}$ (say) in $B_{p,\infty}$ has $\mathbb{Z}$-basis given by

$$\frac{1}{2}(1 + \jmath), \frac{1}{2}(\imath + k), \jmath, k$$

where $\imath^2 = -1$, $\jmath^2 = -p$ and $\imath\jmath = -\jmath\imath = k$. Let $\ell \equiv 3 \mod 4$ and take a supersingular elliptic curve E that has $\mathrm{End}(E) = \mathfrak{m}$ as the starting vertex of the walk in the hash function. Now there is a cycle of length $2t$ starting from E in the graph $G(p, \ell)$ iff there is an endormorphism $x \in \mathfrak{m}$ such that $N(x) = \ell^{2t}$ (here $N(x)$ is the norm of x). Suppose $x = a\frac{1}{2}(1 + \jmath) + b\frac{1}{2}(\imath + k) + c\jmath + dk$, where $a, b, c, d \in \mathbb{Z}$ then its norm

$$N(x) = \frac{a^2}{4} + \left(\frac{a}{2} + c\right)^2 p + \frac{b^2}{4} + \left(d + \frac{b}{2}\right)^2 p.$$

Suppose $N(x) = \ell^{2t}$, $\left(\frac{a}{2} + c\right) = 0$ and $\left(\frac{b}{2} + d\right) = 0$, this would mean that

$$\frac{a^2}{4} + \frac{b^2}{4} = \ell^{2t}.$$

This implies that a and b are even and that $\ell^{2t} = r^2 + s^2$, for some integers r and s. Since $\ell \equiv 3 \mod 4$ the only way ℓ^{2t} can be written as the sum of two squares is $\left(\pm\ell^t\right)^2 + 0^2$ and $0^2 + \left(\pm\ell^t\right)^2$. Thus we must have $s = 0$ and $r = \pm\ell^t$ or the other way around. Thus the only endomorphisms of norm ℓ^{2t} are the trivial ones, multiplication by $\pm\ell^t$ or multiplication by $\pm\ell^t$ composed with the automorphism $\imath$, provided our assumption that $\left(\frac{a}{2} + c\right) = 0$ and $\left(\frac{b}{2} + d\right) = 0$ was true. Hence any non-trivial endomorphisms must violate this assumption. If $x = a\frac{1}{2}(1+\jmath)+b\frac{1}{2}(\imath+k)+c\jmath+dk$ is such that either $\left(\frac{a}{2} + c\right) \neq 0$ or $\left(\frac{b}{2} + d\right) \neq 0$, then $N(x) \geq \frac{p}{4}$. Thus if $N(x) = \ell^{2t}$ then we must have $t \geq \frac{1}{2}\log_\ell(p/4)$. If ℓ is fixed, this gives us a lower bound of $\Omega(\log p)$ for the size of the smallest cycle starting from the vertex E. This gives a lower bound of $\frac{1}{2}\sqrt{p}$ for the degree of any non-trivial endomorphism. Next, we give some reasons why we believe that finding such high degree endomorphisms is a hard problem.

We illustrate another example by looking at the case when $p \equiv 1 \mod 8$. Here we have one maximal order, $\mathfrak{m}$ (say) in $B_{p,\infty}$ whose $\mathbb{Z}$-basis is given by (see Proposition 5.2 of [12])

$$\frac{1}{2}(1+\jmath), \frac{1}{2}(\imath+k), \frac{1}{q}(\jmath+ak), k$$

where $\imath^2 = -1, \jmath^2 = -p$, $q \equiv 3 \mod 4$ is a prime such that $\left(\frac{p}{q}\right) = -1$ and a is an integer such that $q|(a^2p+1)$. Let $q \equiv 3 \mod 4$ be a small prime and then pick a prime $p \equiv 1 \mod 24$ such that $\left(\frac{p}{q}\right) = -1$. We claim that the graph $G(p,\ell)$ for $\ell \equiv 3 \mod 4$ cannot have small cycles starting from any vertex representing a supersingular elliptic curve with endomorphism ring $\mathfrak{m}$. Indeed, a cycle of length $2t$ gives rise to an endomorphism x of E whose norm is ℓ^{2t}. This means that if $x = \frac{r}{2}(1+\jmath) + \frac{s}{2}(\imath+k) + \frac{t}{q}(\jmath+ak) + uk$ (where $r, s, t, u \in \mathbb{Z}$), then its quarternionic norm

$$N(x) = \frac{r^2}{4} + \frac{s^2}{4} + p\left(\frac{r}{2} + \frac{t}{q}\right)^2 + p\left(\frac{s}{2} + \frac{ta}{q} + u\right)^2 = \ell^{2t}.$$

Suppose $\left(\frac{r}{2} + \frac{t}{q}\right) = 0$ and $\left(\frac{s}{2} + \frac{ta}{q} + u\right) = 0$ then $\ell^{2t} = \frac{r^2}{4} + \frac{s^2}{4}$, but $\ell \equiv 3 \mod 4$ all such endomorphisms are the trivial ones coming from multiplication by ℓ^t or $\imath\ell^t$ (recall that $\imath$ is an automorphism). Thus, we must have either $\left(\frac{r}{2} + \frac{t}{q}\right) \neq 0$ or $\left(\frac{s}{2} + \frac{ta}{q} + u\right) \neq 0$. In either case, $N(x) \geq \frac{1}{4q^2}p$. Thus $t \gg \log_\ell p$ if q is fixed. This means that there are no non-trivial endomorphisms of degree $< \frac{1}{4q^2}p$. Thus finding such high degree endomorphisms is likely to be hard.

If the graph $G(p,\ell)$ does not have small cycles then the best known attack is the Pollard-rho attack which will find a cycle in expected time $O(\sqrt{p}\log^2 p)$. Thus taking $p \approx 2^{256}$ would give roughly 128 bits of security against this attack.

Timings for the Hash function based on the Pizer graph. We implemented our hash function to find the actual performance of the hash function. Our results are given below. For a prime p of 192-bits and $\ell = 2$, the time per step of the walk (which is also the time per input bit) is 3.9×10^{-5} secs. This translates to a hashing bandwidth of about 25.6 Kbps. For a prime p of 256-bits, the time per input bit is 7.6×10^{-5} secs or 13.1 Kbps. The implementation was done in C, and the computer on which the timings were taken was an 64-bit AMD Opteron 252 2.6Ghz machine.

7 LPS Ramanujan graphs

An alternative to using the graph $G(p,\ell)$ is to use the Lubotzky-Phillips-Sarnak expander graph ([11]). We describe that graph below. Let ℓ and p be two distinct primes, with ℓ a small prime and p relatively large. We also assume that p and ℓ are such that $\ell \equiv 1 \pmod 4$ and ℓ is a quadratic residue $\pmod p$ (this is the case if $\ell^{(p-1)/2} \equiv 1 \pmod p$). We denote the LPS graph, with parameters ℓ and p, by $X_{\ell,p}$. We define the vertices and edges that make up the graph $X_{\ell,p}$ next. The vertices of $X_{\ell,p}$ are the matrices in $PSL(2, \mathbb{F}_p)$, i.e. the invertible 2×2 matrices with entries in $\mathbb{F}_p$ that have determinant 1 together with the equivalence relation $A = -A$ for any matrix A. Given a 2×2 matrix A with determinant 1, our name for the vertex will be the 4-tuple of entries of A or those of $-A$ depending on which is lexicographically smaller in the usual ordering of the set $\{0, \ldots, p-1\}^4$. We describe the edges that make up the graph next. A matrix A is connected to the matrices gA where the g's are the following explicitly defined matrices. Let i be an integer satisfying $i^2 \equiv -1 \pmod{p}$. There are exactly $8(\ell+1)$ solutions (g_0, g_1, g_2, g_3) to the equation

$$g_0^2 + g_1^2 + g_2^2 + g_3^2 = \ell.$$

Among these there are exactly $\ell + 1$ with $g_0 > 0$ and odd and g_j even for $j = 1, 2, 3$. To each such (g_0, g_1, g_2, g_3) we associate the matrix

$$g = \begin{pmatrix} g_0 + ig_1 & g_2 + ig_3 \\ -g_2 + ig_3 & g_0 - ig_1 \end{pmatrix}.$$

This gives us a set S of $\ell + 1$ matrices in $PGL(2, \mathbb{F}_p)$, but their determinants are squares modulo p and hence they lie in the index 2 subgroup of $PGL(2, \mathbb{F}_p)$ namely, $PSL(2, \mathbb{F}_p)$. It is a fact that if g is in S then so is g^{-1}. Furthermore, since ℓ is small, the set of matrices in S can be found by exhaustive search very quickly. The graph $X_{\ell,p}$ has $p(p^2-1)/2$ vertices and is $\ell + 1$-regular.

This is an example of a Cayley graph. Given a group G and a subset $G_1 \subseteq G$ (normally a generating set) one constructs a graph whose nodes are the elements of G and for every $g \in G_1$ the nodes x, y have an edge corresponding to g if $x = gy$ or $y = gx$. This graph is related to the graphs proposed in the construction of hash functions by Zémor and Tillich [19], with a different choice for the set G_1 (see Section 9 below).

Collision resistance. Finding a collision is equivalent to explicitly calculating the product of generators giving a cycle on the graph. In Sarnak, ([15, §3.4.1]), one finds that the calculation of the girth amounts to finding the minimal t such that ℓ^t is represented by the quadratic form

$$g_0^2 + 4p^2 g_1^2 + 4p^2 g_2^2 + 4p^2 g_3^2$$

subject to the condition that at least one of g_1, g_2, g_3 is not zero. The argument there shows that $t \geq 2 \log_\ell p$. Thus the girth of the LPS graph is at least $2 \log_\ell p$. Since finding the minimal cycle as a product solves the representability problem in $O(t)$ operations and provides an explicit solution, the problem of calculating the minimal cycle cannot be easier than the representability problem, which is considered hard. We remark (loc. cit. §3.3) that the girth of the LPS graph is essentially optimal; for example, it is larger than the girth of a random graph, and in loc. cit. is claimed to be the (asymptotically) largest known. Thus, one does not expect the problem of finding a shortest cycle in the LPS graphs to be easier than the problem for a general homogeneous ℓ-regular graph, which is widely agreed to be hard. To support this, the arguments sketched in ([19] §2.3) to argue that it is hard to find collisions for their hash function also apply to our construction with the LPS graph.

Timings for the hash function based on the LPS graphs. Our implementation of the hash function based on the LPS graph (with $\ell = 5$) takes 1.6×10^{-5} seconds per step of the walk for a prime p of 1024-bits. At each step of the walk $\log_2 \ell$ bits of the input are consumed and so this translates to a hashing bandwidth of $\frac{\log_2 \ell}{1.6 \times 10^{-5}} \approx 145$ Kbps. The machine running the code was the same as before. One disadvantage seems to be that 4 elements of $\mathbb{F}_p$ take $4 \log p$ bits to represent, and if $\log p$ is about 1024, then the output size is too long. For a 192-bit prime p, one step of the walk requires 1.04×10^{-6} seconds. In terms of bandwidth this is about 2.23 Mbps (again with $\ell = 5$). More generally, one step of the walk on this graph costs 8 field multiplications (or 7 if we use Strassen's method), so estimating the time required to do a field multiplication as α gives a direct estimate of the time required to compute the hash per bit of input as $\frac{8\alpha}{\log_2 \ell}$. One can decrease the computational cost per bit at the expense of storing a larger table (of size $\ell + 1$) of generators for the graph. But, if the table is too large then one will have to account for the memory access cost in the analysis.

8 Generic attacks on expander graph based hash functions

Our purpose in this section is to explain a certain generic method of attacks on the collision resistance property of hash functions constructed out of expander graphs in the manner discussed in this paper. Let G be a connected graph and let $w = (v_0; E_1, E_2, \ldots, E_n)$ be a walk in G, with initial vertex v_0 and edges E_i. Let v_n denote the vertex where the walk terminates. Let f be an automorphism of the graph G. We assume that an adversary $\mathcal{A}$ knows f and that the computation of f on any vertex and edge is "fast". Thus, applying f, $\mathcal{A}$ can easily find $f(w) = (f(v); f(E_1), \ldots, f(E_n))$. If, on the average, the distance in G between v and $f(v)$ is small enough then $\mathcal{A}$ is likely to find a walk $w_{v_0,f(v_0)}$ between v_0 and $f(v_0)$ and a walk $w_{v_n,f(v_n)}$ between v_n and $f(v_n)$ by brute force search. The walks $(w_{v_0,f(v_0)}|f(w))$ and $(w|w_{v_n,f(v_n)}))$ are two walks of the same length with the same initial and final vertices. Thus $\mathcal{A}$ can find two different inputs to the hash function hashing to the same value. Alternately, the walk $(w_{v_0,f(v_0)}|f(w)|f(w_{v_n,f(v_n)}))$ represents another input (of different length, usually) hashing to the same value as w. We call such an attack a *generic attack*.

One can easily provide examples of good expanders with an involution f such that the distance between any v and $f(v)$ is one. Indeed, given a good expander graph $H = (V_H, E_H)$ let $G = (V_G, E_G)$ be its extended double cover: if $V_H = \{v_1, \ldots, v_n\}$ then $V_G = \{x_1, \ldots, x_n, y_1, \ldots, y_n\}$ and x_i, y_j are adjacent if $i = j$, or $v_i v_j \in E_H$. This is a connected graph with the involution $f(x_i) = y_i$.

We next discuss our examples of the supersingular graphs and the LPS graphs and explain why the generic attack method fails.

Supersingular graphs. Let p, ℓ be primes, $\ell \equiv 1 \mod 12$, and $G = G(p, \ell)$ be the supersingular graph as in section 4. The only obvious automorphism of G we have is the Frobenius automorphism Fr, sending a supersingular j-invariant j_1 to j_1^p. It also acts on the edges: if H is a subgroup of order ℓ of a supersingular elliptic curve E_1 with $j(E_1) = j_1$ then $Fr(H)$ is a subgroup of order ℓ of $E_1^{(p)}$. The number of fixed points of Fr is the number of supersingular j-invariants defined over $\mathbb{F}_p$, whose order of magnitude is the class number of $\mathbb{Q}(\sqrt{-p})$, which is asymptotically $\sqrt{p}$ if $p \equiv 3 \mod 4$ and $2\sqrt{p}$ otherwise. More generally, we have the following lemma.

Lemma. Let i be a non-negative integer. The number $\alpha(i)$ of supersingular j-invariants such that $\mathrm{dist}_G(j, j^p) \leq i$ is the number of pairs (E, g) consisting of a supersingular elliptic curve E and an endomorphism g of E or degree $p\ell^j$, $j \leq i$, up to isomorphism. Assume that $j \leq \log_\ell(p/4)$ then

$$\alpha(i) = \ell^{i/2} O(\sqrt{p}).$$

Proof: Given an isogeny $h\colon E^{(p)} \to E$ of degree ℓ^j, $j \leq i$, let $g = Fr \circ h$ be the endomorphism of E of degree $p \cdot \deg(g)$. Conversely, an endomorphism g of order $p\ell^j, j \leq i$ can be factored uniquely as a composition, up to automorphisms,

$$E \xrightarrow{Fr} E^{(p)} \xrightarrow{h} E,$$

where the order of h is ℓ^j. We note that to give a pair (E, g) is equivalent to giving a supersingular elliptic curve E and an embedding of the ring $\mathcal{O}_{a,j} := \mathbb{Z}[x]/(x^2 + ax + p\ell^j) \hookrightarrow \mathrm{End}(E)$. For such an embedding to exist we must have that p does not split in the quotient field $K_{a,j}$ of $\mathcal{O}_{a,j}$ and that $K_{a,j}$ is a quadratic imaginary field. Since we have $x^2 + ax + p\ell^j = x(x+a) \mod p$, for p not to split we must have $p|a$, while the second condition is simply that $a^2 < 4p\ell^j$. Note that if $4\ell^j \leq p$, as we now assume, this forces a to be zero. Thus, we need to consider pairs consisting a supersingular elliptic curve and an embedding $\mathcal{O}_j := \mathcal{O}_{0,j} = \mathbb{Z}[x]/(x^2 + p\ell^j) \hookrightarrow \mathrm{End}(E)$. Each such embedding extends to an optimal embedding of a unique order of $K_j := \mathbb{Q}(\sqrt{-p\ell^j})$ into $\mathrm{End}(E)$. To fix ideas, assume $p \equiv 1 \mod 4$. Then each such order is of the form $\mathcal{O}_s$ with $s \leq j$ and $s \equiv j \mod 2$. It is well known that the number of such embeddings is the class number of $\mathcal{O}_s$ and this, in turn, is $\ell^{s/2} O(\sqrt{p})$. Thus, we get the estimate that $\alpha(i)$ is $(\sum_{r=0}^{i/2} \ell^{(i-2r)/2}) O(\sqrt{p}) = \ell^{i/2} O(\sqrt{p})$. $\square$

The lemma implies that to have that the distance between two randomly chosen supersingular elliptic curves is less than i, with probability greater than some constant independent of p and ℓ, one must take i close to the limit posed in the lemma, i.e. $\log_\ell(p/4)$, and this is essentially the diameter of G. This shows that the generic attack using the Frobenius automorphism fails.

LPS graphs. The LPS graphs, defined in section 7 are Cayley graphs. Let $C(G,S)$ be the Cayley graph of a group G relative to a symmetric set of generators S of G, such that $1_G \notin S$. Recall that the vertices of the graph are the elements of G and that we connect g to gs if $s \in S$. The graph $C(G,S)$ is a simple regular connected graph. The group G acts as automorphisms of $C(G,S)$. Given $x \in G$ we have an automorphism $[x]$ of $C(G,S)$ such that $[x](g) = xg$. Note that if g is connected to gs then $[x](g)$ is connected to $[x](gs)$. The Cayley graph could have other automorphisms. Indeed, any automorphism ϕ of G such that $\phi(S) = S$ induces an automorphism of $C(G,S)$. Those, however, will not be studied here.

Let $x \neq 1_G$. Then $[x]$ has no fixed points. Suppose that for some $g \in G$, $\mathrm{dist}(g,[x]g) = n$, where the distance is the minimal length of a walk in $C(G,S)$ starting at g and ending in xg. Thus, there are elements $s_1, s_2, \ldots, s_n$ of S such that $xg = gs_1s_2\cdots s_n$. Then $x = gs_1s_2\ldots s_ng^{-1}$. Assume that also $x = hs_1s_2\ldots s_nh^{-1}$ then $h \in g\mathrm{Cent}_G(s_1\cdots s_n)$, and vice versa. Note that this condition on h depends only on the product $s_1s_2\cdots s_n$ and not on the particular choice of elements $s_1, s_2, \ldots, s_n$. We conclude that following:

$$\sharp\{g \in G : \mathrm{dist}(g,[x]g) \leq n\} = \sum_{\{y \in G : 1 \leq \mathrm{dist}(1_G, y) \leq n, x \sim y\}} \sharp\mathrm{Cent}(y),$$

where we used the notation $x \sim y$ to indicate that x is conjugate to y. Let x^G denote the conjugacy class of x in G. Since conjugacy is an equivalence relation, we conclude that

$$\sharp\{g \in G : \mathrm{dist}(g,[x]g) \leq n\} = \sum_{\{y \in x^G : 1 \leq \mathrm{dist}(1_G, y) \leq n\}} \sharp\mathrm{Cent}(x).$$

Remark that $\sharp x^G \cdot \sharp\mathrm{Cent}(x) = \sharp G$ and so the essential point is how are the lengths of the elements in x^G (relative to the Cayley graph) are distributed. This is an interesting question in general. Here we just note that if G is k regular then there are at most $k \cdot (k-1)^{n-1}$ elements whose distance from 1_G is not larger than n. In fact, since our interest is in good expanders, we are justified in assuming a worst case scenario.

We now specialize our considerations to the group $PSL_2(\mathbb{F}_p)$. The centralizer of a non-central element in $SL_2(\mathbb{F}_p)$ is roughly of size p and is at most of size $p+1$ (that element generate in $M_2(\mathbb{F}_p)$ a quadratic algebra over $\mathbb{F}_p$ isomorphic to $\mathbb{F}_{p^2}, \mathbb{F}_p \oplus \mathbb{F}_p$ or $\mathbb{F}_p[\epsilon]/(\epsilon^2)$). Up to a factor of 2, this is also the size of the centralizer in $PSL_2(\mathbb{F}_p)$. Thus, for $1 \neq x \in PSL_2(\mathbb{F}_p)$ the number of vertices g such that the distance in the LPS graph (relative to ℓ and p) between g and $[x]g$ is less than n is at most $(p+1)(\ell+1)\ell^{n-1} \sim p\ell^n$, while the number of vertices is $(p^3-p)/2$. We see that in order to have that the probability of picking an element g such that $\mathrm{dist}(g,[x]g) \leq n$ exceed some constant, we must choose n to be about $2\log_\ell(p)$, which is essentially the lower bound one has on the girth of the LPS graph. Again, we find that the generic attack method fails.

9 Related work

A proposal for using the hardness of lattice reduction problems can be found in the trapdoor one-way function defined by Goldreich, Goldwasser, and Halevi. In [7], the authors propose a public-key cryptosystem based on the hardness of finding the closest lattice vector to a given vector in a vector space. The system had the disadvantage that for security parameter k-bits, the key size needed was $O(k^2)$ bits while the running time was $O(k^3)$. Ajtai and Dwork (in [1]) proposed a public-key cryptosystem based on the hardness of finding the shortest vector in a lattice. This system had an even worse relation between the security parameter and the key-size. In particular, for security parameter of k-bits, the key size and running time were both $O(k^4)$. However, this was the first system that was based on a hard problem known to have the Worst-case to Average-case connection. In other words, if there was an efficient algorithm to solve the shortest vector problem on average, then the worst case problem also admitted an efficient algorithm. Our proposal (using the Pizer graphs) differs from these constructions in the sense that the lattices are implicitly present, and the translation to the lattice formulation itself seems to be hard.

The work of Zémor and Tillich is more closely related to our second construction of the hash function. They propose using the standard generators for the group $\mathrm{SL}(2, \mathbb{F}_{2^n})$ and doing a walk on the resulting Cayley graph to define a hash function. In spirit, this is very similar to our approach; however, there are a few key differences. The first is that we work with the group $\mathrm{PSL}(2, \mathbb{F}_p)$ and the second and more crucial difference is that we use a set of expanding generators for defining the Cayley graph. Consequently, the distribution properties of the final vertex in the walk can be analyzed using the rapid mixing properties of random walks on expanders. A related proposal was also made by Goldreich [6], where he suggested using expander graphs such as the LPS graph to construct one-way functions. An interesting application of our scheme is given in a paper of Quisquater and Joye ([14]). The authors point out that the scheme of Zémor and Tillich has a nice property which they term the concatenation property: the hash scheme satisfies the following $\mathrm{Hash}(x|y) = \mathrm{Hash}(x) \times \mathrm{Hash}(y)$, where $x|y$ refers to the concatenation of the messages x and y and the product is computed on the group $PSL(2, F_p)$. To satisfy the concatenation property in our scheme, the hash function would have to be designed to always start at the identity matrix and use the generators as determined by the input string. This property is used for authenticating sequences, and there is some application to signing video images. We remark that the concatenation property suggests a possible attack. Indeed, if one can find an element y such that $\mathrm{Hash}(y) = 1$, then for every input x we have $\mathrm{Hash}(x) = \mathrm{Hash}(x|y) = \mathrm{Hash}(y|x)$ and the inputs $x|y$, $y|x$, have the same length. To find such an input y could be easy when the girth of the graph is small. In the LPS graphs, where the hash function has the concatenation property, the girth is essentially as large as possible and a brute force approach to finding such y, i.e. to finding a short cycle, is infeasible when the size of the graph is large enough.

References

1. Ajtai, M.; Dwork, C.; *A Public-Key Cryptosystem with Worst-Case/Average case Equivalence*, In 29th ACM Symposium on Theory of Computing, 284-293, 1997.
2. Alon, N.; *Eigevalues and Expanders*, Combinatorica 6(1986), 83-96.
3. Cerviño, J.M.; *On the correspondence between supersingular elliptic curves and maximal quaternionic orders*, http://arxiv.org/abs/math/0404538.
4. Contini, S.;. Lenstra, A.K.; Steinfeld, R.; *VSH, an Efficient and Provable Collision Resistant Hash Function.* http://www.eprint.iacr.org/2005/193.
5. Goldreich, O.; *Randomized methods in Computation*, Lecture Notes. http://www.wisdom.weizmann.ac.il/∼oded/rnd-sum.html
6. Goldreich, O.; *Candidate One-Way Functions Based on Expander Graphs*, 2000.
7. Goldreich, O.; Goldwasser, S.; Halevi, S.; *Public-Key Cryptosystems from Lattice Reduction Problems*, Advances in Cryptology - CRYPTO '97. Lecture Notes in Computer Science, vol. 1294, Pages 112-131, Springer-Verlag, 1997.
8. Galbraith, S.; *Constructing isogenies between elliptic curves over finite fields*, London Math. Soc., Journal of Computational Mathematics, Vol. 2, pp. 118-138 (1999)
9. Hafner, J. L.; McCurley, K. S.; *A rigorous subexponential algorithm for computation of class groups.* Journal of the American Mathematical Society **2** (1989), 837–850.
10. Hamdy, S.; Möller, B.; *Security of Cryptosystems Based on Class Groups of Imaginary Quadratic Orders* T. Okamoto (Ed.): Advances in Cryptology ASIACRYPT 2000, Springer-Verlag LNCS 1976, pp. 234-247.
11. Lubotzky, A.; Phillips, R.; Sarnak, P.; Ramanujan graphs. Combinatorica 8 (1988), no. 3, 261–277.
12. Pizer, A.K.; *An algorithm for computing modular forms on* $\Gamma_0(N)$. J. Algebra 64 (1980), no. 2, 340–390.
13. Pizer, A.K.; *Ramanujan Graphs and Hecke Operators*, Bulletin of the AMS, Volume 23, Number 1, July 1990.
14. Quisquater, J.-J.; Joye, M.; *Authentication of sequences with the* SL_2 *hash function: Application to video sequences*, Journal of Computer Security, 5(3), pp. 213-223, 1997.
15. Sarnak, P.; *Some Applications of Modular Forms*, Series: Cambridge Tracts in Mathematics **99**, Cambridge University Press, 1990.
16. Silverman, Joseph, H.; *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics, **106**, Springer-Verlag, 1986.
17. Vélu, Jacques; *Isogénies entre courbes elliptiques*, C. R. Acad. Sc. Paris, **273**, 238-241, 1971.

18. Zémor, G.; *Hash functions and Cayley Graphs*, Designs, Codes and Cryptography, 4, 381-394, 1994.
19. Zémor, G.; Tillich, J.-P.; *Hashing with* SL_2, Advances in Cryptology, Crypto'94, Lecture Notes in Computer Science, Vol. 839, 1994.