Cryptographic hash functions from expander graphs

Denis X. Charles[1], Eyal Z. Goren[2], and Kristin E. Lauter[1]

[1] Microsoft Research, One Microsoft Way, Redmond, WA 98052
cdx@microsoft.com, klauter@microsoft.com

[2] Department of Mathematics and Statistics, McGill University, 805 Sherbrooke St. W., Montreal H3A 2K6, QC, Canada. goren@math.mcgill.ca

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 [5]. We propose constructing provable cryptographic hash functions from expander graphs. The input to the hash 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 and thus have excellent mixing properties.

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 ([14]), computing 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 square-root 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 relatively inefficient compared to other provable hash functions such as [5], 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 [19], [20]. 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.

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.

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.

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 ([7, 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: ([3])

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

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 breaking the input to the hash function into chunks of size e, so that $2^e = k-1$. 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 e bits of the input. We do not allow backtracking in the walk, so only $k-1$ choices for the next edge are allowed at each step. Variations could allow input to be written in base b, so that other choices for k would be allowable. Alternatively, e could be chosen such that $2^e < k-1$.

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 ([7, Lecture 10, Cor 6]).

4 Pizer's Ramanujan graphs

The graphs. We first define the family of graphs ([14]). 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 $\mathbb{F}_{p^2}$. 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 [17, Chapter 5, Thm 4.1]. 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 E is as follows: there is an edge between the vertices E_1 and E_2 if there is an isogeny of degree ℓ between them. An isogeny is a morphism of elliptic curves which takes the identity element to the identity (for background on isogenies see [17, Chapter 3, Section 4]). For separable isogenies, to have degree ℓ means to have kernel of size ℓ. Actually, we can identify isogenies with their kernels, and the isogeny itself can be computed using Vélu's formulas [18] from the knowledge of the subgroup of size ℓ. To take a step in a walk on the graph, we compute isogenies as in [6, Algorithm 1] by explicitly writing down generators for the (rank 2) ℓ-torsion and listing the $\ell+1$ subgroups of order ℓ. It follows from this that the graph $G(p,\ell)$ is $(\ell+1)$-regular. More details follow 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).

Walking around the graph. Let C be a subgroup of E, Vélu in [18] 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. 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.
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

Explicitly finding a collision in this hash function is equivalent to finding two isogenies of the same ℓ-power degree between a pair of supersingular elliptic curves. If the graph $G(p,\ell)$ does not have small cycles then this problem is very hard, since constructing isogenies of large degree between curves is a well-known computationally hard problem, as we explain below. 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.

To find to distinct paths from E to E', each of length n, is to give two isogenies of degree ℓ^n, $f, g : E \to E'$. In this case $\ell^n g^{-1} f$ will be an endomorphism of degree ℓ^{2n} of E. The degree map is a rank 4 quadratic form, which can also be described as the Norm map on a maximal order in a quaternion algebra. The endomorphism ring (over $\mathbb{F}_p$) of a supersingular elliptic curve is isomorphic to an order in the quaternion algebra $B = B_{p,\infty}$ over $\mathbb{Q}$ ramified only at p and ∞ ([17, Chapter 5, Theorem 3.1]). However, the best known algorithms for determining the endomorphism ring of a supersingular elliptic curve as a maximal order in B are exponential in p ([4]).

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 (i.e. not as the norm of ℓ^n). 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.

Producing two isogenies of the same ℓ-power degree between two elliptic curves is a special case of finding an isogeny of a given ℓ-power degree between two given elliptic curves. Starting at any elliptic curve, one can take a walk on the graph (corresponding to an isogeny) to arrive at another elliptic curve. Then the task is to produce another isogeny of the same degree between those two elliptic curves. Thus collision resistance for the graph of supersingular elliptic curves can be viewed in several ways.

1. Heuristically in analogy with the ordinary case, which we know is hard. In [10], 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 class groups of imaginary quadratic number fields, which has been well-studied. Although subexponential index calculus methods apply ([11]), taking quadratic orders with large discriminants makes the problem as hard as factoring integers of that size ([12]). Note the difference between the ECDLP situation and here: problems on supersingular elliptic curves are not necessarily easier than the corresponding problems on ordinary elliptic curves. In fact, for our problem, there is no class group to work in in the supersingular case, and the degree map is a rank 4 quadratic form instead of rank 2 (see the next point).
2. Finding vectors of a given norm in a lattice. Problem: Given two supersingular elliptic curves over $\mathbb{F}_{p^2}$, to find two ℓ-power isogenies between them of the same degree ℓ^n. As explained above, this can also be described as finding a vector of a given norm in a lattice, namely $\mathrm{End}(E)$ with the degree map defining a positive definite rank 4 quadratic form. This problem seems to be hard in general. For example, the related problem of finding a vector of L_2-norm bounded by a given quantity in a lattice is NP-hard under randomized reductions (see [1]).
3. Random walks on optimal expander graphs. The Pollard Rho attack will succeed in expected time $O(\sqrt{p})$. Thus taking $p \approx 2^{256}$ would give roughly 128 bits of security against this attack.

7 LPS Ramanujan graphs

An alternative to using the graph $G(p, \ell)$ is to use the Lubotzky-Phillips-Sarnak expander graph ([13]). 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.

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 $y = gx$.

Collision resistance.

The problem of collision resistance is essentially the problem of explicitly calculating the product of generators giving the shortest cycle on the graph. In Sarnak, ([16, §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 $t \geq 2 \log_\ell p$. 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 (as discussed in Section 6). 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 ([20] §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. A walk of length 1000, takes 0.188 seconds for p a 1024-bit prime with $\ell = 5$ on a 64-bit HP Workstation xw9300, Opteron 252/2.6GHz using Visual C++ and the NTL library without optimizations. The input is divided into chunks of size $\log(\ell)$. 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 this is too much. For a 128-bit prime p with $\ell = 5$, a walk of length 1 million requires only 14.312 seconds. One step of 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 7α.

8 Related work

Another 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 [9], 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 [2]) 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 [8], 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 ([15]). The authors point out that the scheme of Zémor and Tillich has a nice property which they term the concatenation property: the signature scheme satisfies the following Sign($x|y$) = Sign(x)*Sign(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, we would 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.

References

1. Ajtai, M.; *The Shortest Vector Problem in L_2 is* NP*-hard for Randomized Reductions (Extended Abstract)*, ACM Symposium on Theory of Computing (STOC), 1998, 10-19.
2. 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.
3. Alon, N.; *Eigevalues and Expanders*, Combinatorica 6(1986), 83-96.
4. Cerviño, J.M.; *On the correspondence between supersingular elliptic curves and maximal quaternionic orders*, http://arxiv.org/abs/math/0404538.
5. Contini, S.;. Lenstra, A.K.; Steinfeld, R.; *VSH, an Efficient and Provable Collision Resistant Hash Function.* http://www.eprint.iacr.org/2005/193.
6. Charles, D.; Lauter, K.; *Computing Modular Polynomials*, London Math. Soc., Journal of Computational Mathematics, Vol. 8, pp. 195-204 (2005).
7. Goldreich, O.; *Randomized methods in Computation*, Lecture Notes. http://www.wisdom.weizmann.ac.il/ oded/rnd-sum.html
8. Goldreich, O.; *Candidate One-Way Functions Based on Expander Graphs*, 2000.
9. 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.
10. Galbraith, S.; *Constructing isogenies between elliptic curves over finite fields*, London Math. Soc., Journal of Computational Mathematics, Vol. 2, pp. 118-138 (1999)
11. 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.
12. 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.
13. Lubotzky, A.; Phillips, R.; Sarnak, P.; Ramanujan graphs. Combinatorica 8 (1988), no. 3, 261–277.

14. Pizer, A.K.; *Ramanujan Graphs and Hecke Operators*, Bulletin of the AMS, Volume 23, Number 1, July 1990.
15. Quisquater, J.-J.; Joye, M.; *Authentication of sequences with the SL2 hash function: Application to video sequences*, Journal of Computer Security, 5(3), pp. 213-223, 1997.
16. Sarnak, P.; *Some Applications of Modular Forms*, Series: Cambridge Tracts in Mathematics **99**, Cambridge University Press, 1990.
17. Silverman, Joseph, H.; *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics, **106**, Springer-Verlag, 1986.
18. Vélu, Jacques; *Isogénies entre courbes elliptiques*, C. R. Acad. Sc. Paris, **273**, 238-241, 1971.
19. Zémor, G.; *Hash functions and Cayley Graphs*, Designs, Codes and Cryptography, 4, 381-394, 1994.
20. Zémor, G.; Tillich, J.-P.; *Hashing with* SL_2, Advances in Cryptology, Crypto'94, Lecture Notes in Computer Science, Vol. 839, 1994.