SIGNATURES FOR NETWORK CODING

DENIS CHARLES, KAMAL JAIN, AND KRISTIN LAUTER

Abstract. This paper presents a practical digital signature scheme to be used in conjunction with network coding. This signature scheme seems to be the first example of a homomorphic signature scheme. Furthermore, our idea simultaneously provides authentication and detects malicious nodes that intentionally corrupt content on the network.

1. Introduction

Following the important work of Ahlswede *et al* and Li *et al* ([ACLY00, CLY03]), network coding ([CJW03, CJL05, GR05]) has been established as a viable alternative to the store and forward mechanisms used in peer-to-peer networks. However, network coding is inherently vulnerable to pollution attacks by malicious nodes in the network. The pollution of packets spreads quickly since the output of (even an) honest node is corrupted if at least one of the incoming packets is corrupted. The question of how to prevent pollution attacks in the network coding scheme remained open and was the subject of the paper by Krohn *et al* [KFM04] in the generalized setting of rateless erasure codes (see also [GR06]). They show that a construction based on homomorphic hashing works to detect the polluted packets. This scheme, however, assumes that there is a separate secure channel which is used to transmit the hash values of the packets to all the nodes.

In this paper we propose a different solution to the problem of detecting pollution attacks. We design a new *homomorphic* signature scheme for use with network coding. The homomorphic property of the signatures allows nodes to sign any linear comination of the incoming packets without contacting the signing authority. At first glance one might think that this is a weakness of the signature scheme. This is not so, in our scheme it is computationally infeasible for a node to sign a linear combination of the packets without disclosing what linear combination was used in the generation of the packet. Furthermore, we can prove that the signature scheme is secure under well known cryptographic assumptions of the hardness of the Discrete-Log problem and the computational co-Diffie-Hellman problem on elliptic curves. Our scheme has a three-fold advantage over the scheme based on homomorphic hashing: Firstly, we do not need to securely transmit hash values of the packets that the source transmits; secondly, since our scheme is based on elliptic curves smaller security parameters suffice and this translates to improved efficiency since the bit lengths involved are smaller; finally, our scheme provides authentication of the data in addition to detecting pollution of packets.

2. Background on elliptic curves

In this section we briefly review some facts about elliptic curves over finite fields, the reader should consult Chapters III and V of [Sil86] for proofs of the number theoretic claims.

Let $\mathbb{F}_q$ be a finite field where q is a power of a prime relatively prime to 2 and 3. An elliptic curve E over $\mathbb{F}_q$ (sometimes abbreviated as $E/\mathbb{F}_q$), is a projective curve in $\mathbb{P}^2(\mathbb{F}_q)$ given by an equation of the form

$$Y^2Z = X^3 + AXZ^2 + BZ^3$$

with $A, B \in \mathbb{F}_q$ and $4A^3 + 27B^2 \neq 0$. The curve has two affine pieces: the piece with $Z \neq 0$ has the affine form $y^2 = x^3 + Ax + B$ (obtained by setting $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$); and the piece with $Z = 0$ which has only one (projective) point namely $(0 : 1 : 0)$ which we denote $\mathcal{O}$. Let K be a field (not necessarily finite) that contains $\mathbb{F}_q$, the set

$$E(K) = \{(x, y) \in K \times K : y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}\}$$

can be given the structure of an abelian group with $\mathcal{O}$ as the identity of the group. Moreover, the group operations can be efficiently computed. In particular, if P and Q are points on E with coordinates in $\mathbb{F}_q$, then $P+Q$ and $-P$ can be computed in $O(\log^{1+\epsilon} q)$ bit operations for any $\epsilon > 0$. Hasse's theorem gives a tight estimate for the size of the group $E(\mathbb{F}_q)$:

$$q + 1 - 2\sqrt{q} \leq \#E(\mathbb{F}_q) \leq q + 1 + 2\sqrt{q}.$$

The Schoof-Elkies-Atkin algorithm ([BSS99] Chapter VII) is a deterministic polynomial time algorithm that computes $\#E(\mathbb{F}_q)$.

2.1. **The Weil pairing.** Let $E/\mathbb{F}_q$ be an elliptic curve and let $\overline{\mathbb{F}}_q$ be an algebraic closure of $\mathbb{F}_q$. If m is an integer such relatively prime to the characteristic of the field $\mathbb{F}_q$, then the group of m-torsion points, $E[m] = \{P \in E(\overline{\mathbb{F}}_q) \ : \ mP = \mathcal{O}\}$, have the following structure:

$$E[m] = \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}.$$

There is a map $\mathbf{e}_m : E[m] \times E[m] \to \overline{\mathbb{F}}_q^*$ with the following properties:
(1) The map $\mathbf{e}_m$ is bilinear:

$$\mathbf{e}_m(S_1 + S_2, T) = \mathbf{e}(S_1, T)\mathbf{e}(S_2, T)$$
$$\mathbf{e}_m(S, T_1 + T_2) = \mathbf{e}(S, T_1)\mathbf{e}(S, T_2).$$

(2) Alternating: $\mathbf{e}_m(T, T) = 1$ and so $\mathbf{e}_m(T, S) = \mathbf{e}_m(S, T)^{-1}$.
(3) Non-degenerate: If $\mathbf{e}_m(S, T) = 1$ for all $S \in E[m]$ then $T = \mathcal{O}$.
Let $E/\mathbb{F}_q$ be an elliptic curve such that the m-torsion points on E have coordinates in $\mathbb{F}_q$. Then there is a probabilistic algorithm that can evaluate $\mathbf{e}_m(S, T)$ in $O(\log^{2+\epsilon} q)$ bit operations for all S, T in $E[m]$. If it is clear from the context we may drop the subscript m when writing $\mathbf{e}_m$. The algorithm for computing $\mathbf{e}_m$ was proposed by Miller in [Mil86]. See the paper by Eisenträger *et al* ([ELM04]) for a description of Miller's algorithm and also a deterministic variant for computing the square of the Weil pairing.

3. The signature scheme

3.1. **Network Coding.** We briefly describe the standard network coding framework for content distribution ([CJW03, GR05, CJL05]). Let $G = (\tilde{V}, \tilde{E})$ be a directed graph. A source $s \in \tilde{V}$ wishes to transmit some data to a set $T \subseteq \tilde{V}$ of the vertices. One chooses a vector space $W/\mathbb{F}_p$ (say of dimension d), where p is a prime, and views the data to be transmitted as a bunch of vectors $\mathbf{w}_1, \cdots, \mathbf{w}_k \in W$. The source then creates the augmented vectors $\mathbf{v}_1, \cdots, \mathbf{v}_k$ by setting $\mathbf{v}_i = \langle \underbrace{0, \cdots, 0}_{i-1 \text{ zeros}}, 1, \cdots, 0, w_{i1}, \cdots, w_{id} \rangle$ where w_{ij} is the j-th coordinate of the vector $\mathbf{w}_i$. One can assume without loss of generality that the vectors $\mathbf{v}_i$ are linearly independent. We denote the subspace (of $\mathbb{F}_p^{k+d}$) spanned by these vectors by V. Each outgoing edge $e \in \tilde{E}$ computes a linear combination, $\mathbf{y}(e)$, of the vectors entering the vertex $v = \text{in}(e)$ where the edge originates, that is to say

$$\mathbf{y}(e) = \sum_{f:\ \text{out}(f)=v} m_e(f)\mathbf{y}(f)$$

where $m_e(f) \in \mathbb{F}_p$. We consider the source as having k input edges carrying the k vectors $\mathbf{w}_i$. By induction one has that the vector $\mathbf{y}(e)$ on any edge is a linear combination $\mathbf{y}(e) = \sum_{1 \le i \le k} g_i(e)\mathbf{v}_i$ and is a vector in V. The k-dimensional vector $\mathbf{g}(e) = \langle g_1(e), \cdots, g_k(e) \rangle$ is simply the first k-coordinates of the vector $\mathbf{y}(e)$. We call the matrix whose rows are the vectors $\mathbf{g}(e_1), \cdots, \mathbf{g}(e_k)$, where e_i are the incoming edges for a vertex $t \in T$, the global encoding matrix for t and denote it G_t. In practice the encoding vectors are chosen at random so the matrix G_t is invertible with high probability. Thus any receiver, on receiving $\mathbf{y}_1, \cdots, \mathbf{y}_k$ can find $\mathbf{w}_1, \cdots, \mathbf{w}_k$ by solving

$$\begin{bmatrix} \mathbf{y}'_1 \\ \mathbf{y}'_2 \\ \vdots \\ \mathbf{y}'_k \end{bmatrix} = G_t \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_k \end{bmatrix},$$

where the $\mathbf{y}'_i$ are the vectors formed by removing the first k coordinates of the vector $\mathbf{y}_i$.

3.2. **The homomorphic signature scheme.** Let p be a prime number and q a power of a different prime with $p \ll q$. Let $V/\mathbb{F}_p$ be a vector space of dimension $d + k$ and let $E/\mathbb{F}_q$ be an elliptic curve such that $R_1, \cdots, R_k, P_1, \cdots, P_d$ are (distinct) points of p-torsion on $E(\mathbb{F}_q)$. We can define a function $h_{R_1,\cdots,R_k,P_1,\cdots,P_d} : V \to E(\mathbb{F}_q)$ as follows: for $\mathbf{v} = \langle u_1, \cdots, u_k, v_1, \cdots, v_d \rangle \in V$

$$h_{R_1,\cdots,R_k,P_1,\cdots,P_d}(\mathbf{v}) = \sum_j u_j R_j + \sum_i v_i P_i.$$

The function $h_{R_1,\cdots,R_k,P_1,\cdots,P_d}$ is a homomorphism (of additive abelian groups) from the vector space V to the group $E[p]$ of p-torsion points on the curve.

Suppose the server wishes to distribute the augmented vectors $\mathbf{v}_1, \cdots, \mathbf{v}_k \in V$. The server chooses $s_1, \cdots, s_k$ and $r_1, \cdots, r_d$ which are secrets in $\mathbb{F}_p$, then signs the packet $\mathbf{v}_i$ by computing

$$\mathfrak{h}_i = h_{s_1 R_1,\cdots,s_k R_k, r_1 P_1,\cdots,r_d P_d}(\mathbf{v}_i).$$

The server also publishes $R_1, \cdots, R_k, P_1, \cdots, P_d$, Q, $s_j Q$ for $1 \le j \le k$ and $r_i Q$ for $1 \le i \le d$. Here Q is another point of p-torsion on the elliptic curve distinct from the others such that $\mathbf{e}_p(R_j, Q) \ne 1$ and $\mathbf{e}_p(P_i, Q) \ne 1$ for $1 \le j \le k$ and $1 \le i \le d$.

This signature $\mathfrak{h}_j$ is also appended to the data $\mathbf{v}_j$ and transmitted according to the distribution scheme. Now, at any edge e that computes

$$\mathbf{y}(e) = \sum_{f:\mathrm{out}(f)=\mathrm{in}(e)} m_e(f)\mathbf{y}(f)$$

we also compute

$$\mathfrak{h}(e) = \sum_{f:\mathrm{out}(f)=\mathrm{in}(e)} m_e(f)\mathfrak{h}(f)$$

and transmit $\mathfrak{h}(e)$ together with the data $\mathbf{y}(e)$. Since the computation of the signature $\mathfrak{h}(e)$ is a homomorphism, we have that if $\mathbf{y}(e) = \sum_i \alpha_i \mathbf{v}_i$ then

$$\mathfrak{h}(e) = \sum_i \alpha_i \mathfrak{h}_i.$$

Next we describe the verification process. Suppose $\mathbf{y}(e) = \langle u_1, \cdots, u_k, v_1, \cdots, v_d \rangle$ we check whether

$$\prod_{1 \le j \le k} \mathbf{e}(u_j R_j, s_j Q) \prod_{1 \le i \le d} \mathbf{e}(v_i P_i, r_i Q) = \mathbf{e}(\mathfrak{h}(e), Q).$$

This works because if $\mathfrak{h}(e)$ is the legitimate signature of $\mathbf{y}(e)$ then by definition

$$\mathfrak{h}(e) = \sum_{1 \le j \le k} u_j s_j R_j + \sum_{1 \le i \le d} v_i r_i P_i,$$

thus

$$\begin{aligned}
\mathbf{e}(\mathfrak{h}(e), Q) &= \mathbf{e}\big(\sum_{1 \le j \le k} u_j s_j R_j + \sum_{1 \le i \le d} v_i r_i P_i, Q \big) \\
&= \prod_{1 \le j \le k} \mathbf{e}(u_j s_j R_j, Q) \prod_{1 \le i \le d} \mathbf{e}(v_i r_i P_i, Q) \text{ (by bilinearity)} \\
&= \prod_{1 \le j \le k} \mathbf{e}(u_j R_j, s_j Q) \prod_{1 \le i \le d} \mathbf{e}(v_i P_i, r_i Q) \text{ (again by bilinearity).}
\end{aligned}$$

The verification cruicially uses the bilinearity of the Weil-pairing. Note that all the terms in the above verification can either be computed from the vector $\mathbf{y}(e)$ or from the public information.

The signature is a point on the elliptic curve with coordinates in $\mathbb{F}_q$. Thus the size of the signature is $2 \log q$ bits (which is some constant times $\log(p)$ bits, depending on the relative size of p and q), and this is the transmission overhead. The computation of the signature $\mathfrak{h}(e)$ at each vertex requires $O(d_{in} \log p \log^{1+\epsilon} q)$ bit operations, where d_{in} is the in-degree of the vertex $\mathrm{in}(e)$. The verification of a signature requires $O((d+k) \log^{2+\epsilon} q)$ bit operations.

4. Proof of security

We preserve the notation of the previous section here. To thwart the signature scheme an adversary can either produce a hash collision for the function $h_{s_1 R_1, \cdots, s_k R_k, r_1 P_1, \cdots, r_d P_d}$ or he can forge the signature such that the verification goes through. Note that in this situation the adversary has no knowledge of the points $s_1 R_1, \cdots, s_k R_k$ and $r_1 P, \cdots, r_d P_d$. We first show that *even* if the adversary knew these points, producing a collision is still as hard as computing discrete logs. We make the claim precise next:

Problem: Hash-Collision.
Fix an integer $r > 1$.
Input: Given $P_1, \cdots, P_r$, points on an elliptic curve $E/\mathbb{F}_q$ contained in a cyclic subgroup of prime order p.
Output: Tuples $\mathbf{a} = \langle a_1, \cdots, a_r \rangle, \mathbf{b} = \langle b_1, \cdots, b_r \rangle \in \mathbb{F}_p^r$ such that $\mathbf{a} \neq \mathbf{b}$ and

$$\sum_{1 \le i \le r} a_i P_i = \sum_{1 \le j \le r} b_j P_j.$$

Proposition 4.1. *There is a polynomial time reduction from Discrete Log on the cyclic group of order p on elliptic curves to* Hash-Collision.

Proof : First we treat the case when $r = 2$. Let P and Q be points of order p on $E(\mathbb{F}_q)$ that are not the identity. Assume that Q lies in the subgroup generated by P. Our aim is to find a such that $Q = aP$. To this end we apply the alleged algorithm that solves Hash-Collision to the points P and Q. The algorithm produces two distinct pairs $(x, y), (u, v) \in \mathbb{F}_p^2$ such that

$$xP + yQ = uP + vQ.$$

This gives us a relation $(x-u)P + (y-v)Q = \mathcal{O}$. We claim that $x \neq u$ and $y \neq v$. Suppose that $x = u$, then we would have $(y-v)Q = \mathcal{O}$, but Q is a point of order p (a prime) thus $y - u \equiv 0 \mod p$ in other words $y = v$ in $\mathbb{F}_p$. This contradicts the assumption that (x, y) and (u, v) are distinct pairs in $\mathbb{F}_p^2$. Thus we have that $Q = -(x-u)(y-v)^{-1}P$, where the inverse is taken modulo p.

If we have $r > 2$ then we can do one of two things. Either we can take $P_1 = P$ and $P_2 = Q$ as before and set $P_i = \mathcal{O}$ for $i > 2$ (in this case the proof reduces to the case when $r = 2$), or we can take $P_1 = r_1 P$ and $P_i = r_i Q$ where r_i are chosen at random from $\mathbb{F}_p$. We get one equation in one unknown (the discrete log of Q). It is quite possible that the equation we get does not involve the unknown. However, this happens with very small probability as we argue next. Suppose the algorithm for Hash-Collision gave us that

$$a r_1 P + \sum_{2 \leq i \leq r} b_i r_i Q = \mathcal{O}.$$

Then as long as $\sum_{2 \leq i \leq r} b_i r_i \not\equiv 0 \mod p$, we can solve for the discrete log of Q. But the r_i's are unknown to the oracle for Hash-Collision and so we can interchange the order in which this process occurs. In other words, given b_i, for $2 \leq i \leq r$, not all zero, what is the probability that the r_i's we chose satisfy $\sum_{2 \leq i \leq r} b_i r_i = 0$? It is clear that the latter probability is $\frac{1}{p}$. Thus with high probability we can solve for the discrete log of Q. $\square$

One can also conclude the above proposition from the proof presented in [BGG94] (see Appendix A of that paper). The proof in that paper deals with finite fields but the argument applies equally well to the case of elliptic curves.

We have shown that producing hash collisions in our scheme is difficult. The other method by which an adversary can foil our system is by forging a signature. Our scheme for the signature is essentially the Aggregate Signature version of the Boneh-Lynn-Shacham signature scheme [BLS04]. In that paper it is shown that forging a signature is at least as hard as solving the so-called computational co-Diffie-Hellman problem on the elliptic curve. The only known way to solve this problem on elliptic curves is via computing discrete-logs. Thus forging a signature is at least as hard as solving the computational co-Diffie-Hellman on elliptic curves and probably as hard as computing discrete-logs.

5. Setup of the scheme

We preserve the notation of section §3 here. To initialize the signature scheme we need to pick a prime p and an elliptic curve over a field such that all its p-torsion is defined over that field. We also need to produce the collection of p-torsion points needed to define the homomorphic signature. In this section we discuss all these matters and provide an example.

We describe the outline of the steps below and then describe the steps in detail:

(1) Pick a large prime and call it p.
(2) Pick a suitable prime ℓ (described in §5.1)and an elliptic curve E over $\mathbb{F}_\ell$ such that the number of points $\#E(\mathbb{F}_\ell)$ is a multiple of p.
(3) Find an extension $\mathbb{F}_q$ of the field $\mathbb{F}_\ell$ such that $E[p] \subseteq E(\mathbb{F}_q)$ (here $E[p]$ refers to the set of *all* p-torsion points).
(4) Since $\#E(\mathbb{F}_\ell) \equiv 0 \mod p$ it has p-torsion points. Let $\mathcal{O} \neq P \in E(\mathbb{F}_\ell)$ be a p-torsion point on the curve. Take $R_i = a_i P$ for $1 \leq i \leq k$ and $P_j = b_j P$ for $1 \leq j \leq d$ where a_i and b_i are picked at random from the set $1, \cdots, p-1$.

(5) One of the requirements of our scheme is that Q be a point such that $\mathbf{e}(R_i, Q) \neq 1$ and $\mathbf{e}(P_j, Q) \neq 1$. To ensure this, we claim that it suffices to pick a point of p-torsion that is defined over $\mathbb{F}_q$ but not over the smaller field $\mathbb{F}_\ell$. Indeed, let Q be such a point. Then if $\mathbf{e}(R_i, Q) = 1$, this would imply that $\mathbf{e}(A, B) = 1$ for any $A, B \in E[p]$ (since R_i and Q generate $E[p]$), which contradicts the non-degeneracy of the Weil-pairing.
(6) Finally, we pick the secret keys $s_1, \cdots, s_k$ and $r_1, \cdots, r_d$ at random from $\mathbb{F}_p^*$.

5.1. **Finding a suitable elliptic curve.** In general, if we have an elliptic curve E over a finite field K, then the p-torsion points are defined over an extension of degree $\Theta(p^2)$ of the field K (see [CL05] Lemma 2.2). It is crucial for our scheme to have the p-torsion points defined over a small degree extension field so that the operations can be carried out in polynomial time. In this section we discuss how one can pick a suitable field $\mathbb{F}_\ell$ and an elliptic curve over this field that has all its p-torsion defined over a small degree extension field.

In the following paragraph we describe a construction that allows one to find an elliptic curve defined over a finite field $\mathbb{F}_\ell$ such that the entire p-torsion is defined over $\mathbb{F}_{\ell^2}$. Such curves are said to have embedding degree 2 (the construction we give also generalizes nicely to produce other embedding degrees). We note that the MOV attack reduces the discrete-log problem on the p-torsion of such curves to the discrete-log problem in the multiplicative group of the *finite field* $\mathbb{F}_{\ell^2}^*$. Thus, for security considerations one needs to take the embedding degree k to be large enough so that the finite field produced by the MOV attack is of cryptographic size. For a detailed discussion of these issues we invite the reader to see [MOV93, MNT01, BLS02] and also the book [BSS99].

The theory of complex multiplication of elliptic curves can be used to generate elliptic curves over a finite field with a certain number of points on them. The algorithm to do this is described in many sources [LL90, ALV02, AtMor93, Sch85]. The details of the algorithm are not necessary for our purposes, but its running time is important, so we describe it next.

Suppose we wish to produce an elliptic curve $E/\mathbb{F}_\ell$ (where ℓ is a prime) that has exactly N points, where N lies in the interval $\ell + 1 - 2\sqrt{\ell} \leq N \leq \ell + 1 + 2\sqrt{\ell}$. Write N as $\ell + 1 - t$ and set $Dy^2 = t^2 - 4\ell$, where D or $D/4$ is squarefree (note that D is negative because of the Hasse bound). Then the algorithm to produce such a curve runs in time $|D|^{O(1)}$. In our case, we seek an elliptic curve with N equal to a small multiple of p. This tells us that the field $\mathbb{F}_\ell$ over which we should look for such a curve must have $\ell + 1 - 2\sqrt{\ell} \leq mp \leq \ell + 1 + 2\sqrt{\ell}$. The other requirement is that $t^2 - 4\ell$ should have a small squarefree part, since this determines the running time of the method to generate such a curve. We pick a prime ℓ such that $4\ell = 4p^2 - Dy^2$ for a small (negative) D. We also require[1] $\ell \equiv -1 \mod p$, and we set $t = 2p$. Thus $\ell + 1 - t = \ell + 1 - 2p \equiv 0 \mod p$, and so the number of points on the elliptic curve will be a multiple of p. The time to produce such a curve will also be reasonable since $|D|$ is small. To produce such a prime ℓ, we pick a (negative) D (with $|D|$ small) and check to see if $(p^2 - \frac{Dy^2}{4})$ is prime for $y = 0, 1, \cdots$. Since we are only interested in primes which are congruent to $-1 \mod p$, we perform the above check *only* for those values of y such that $-Dy^2 \equiv -4 \mod p$. A conjecture of Lang-Trotter ([LTr76]) tells us that there will be many values of y that yield a prime. This is also related to a conjecture of Hardy-Littlewood on the prime values of quadratic polynomials.

Now the complex multiplication method produces for us an elliptic curve E over $\mathbb{F}_\ell$ that has some p-torsion points. However, we need an elliptic curve such that $E[p]$ is defined over a small degree

[1] To get embedding degree k we instead look for primes ℓ such that $\ell \equiv a_k \mod p$ where a_k is an element of order k in $\mathbb{F}_p^*$.

extension of $\mathbb{F}_\ell$. This is where the additional constraint that $\ell \equiv -1 \mod p$ is used. Since $\ell \equiv -1 \mod p$ the order of ℓ in $\mathbb{F}_p^*$ is 2. Now a theorem of Koblitz-Balasubramanian (see [BK98], Theorem 1) shows that in this case the entire p-torsion is defined over a degree 2 extension of the base field, in other words $E[p] \subseteq E(\mathbb{F}_{\ell^2})$. Now we have an elliptic curve $E/\mathbb{F}_\ell$ and we know that it has all its p-torsion defined over $\mathbb{F}_{\ell^2}$, but how do we find these points? This is the subject of the next paragraph.

Remark 5.1. We remark that the theory of complex multiplication tells us that, for each D, there is a finite list of elliptic curves $E_1, \cdots, E_h$ over some number field K such that $E_i \mod \ell$ satisfies our requirements. This is illustrated in the example in §5.3.

5.2. **Finding the p-torsion points.** Let $E/\mathbb{F}_\ell$ be the elliptic curve found using the method given above. Then $\#E(\mathbb{F}_\ell) = \ell + 1 - 2p$. Let m be the largest divisor of $\#E(\mathbb{F}_\ell)$ that is relatively prime to p. Let P be a random point on the curve $E(\mathbb{F}_\ell)$. If $mP \neq \mathcal{O}$, then mP is a non-trivial point of p-power torsion (by Lagrange's theorem). Let $i \geq 1$ be the smallest integer such that $mp^iP = \mathcal{O}$ but $mp^{i-1}P \neq \mathcal{O}$. Then $mp^{i-1}P$ is a non-trivial p-torsion point. Of course, if $mP = \mathcal{O}$, we restart with another random point P. The probability that $mP = \mathcal{O}$ for a random point P is at most $\frac{1}{p}$, so we will find a non-trivial p-torsion point with very high probability.

This gives us the piece of the p-torsion defined over $\mathbb{F}_\ell$. To find the piece of the p-torsion defined over $\mathbb{F}_{\ell^2}$ we repeat the above process over $\mathbb{F}_{\ell^2}$. To carry out this process we need to know the number of points on $E(\mathbb{F}_{\ell^2})$. If E is defined over a finite field K, then the number of points on E over any extension of K is determined by $\#E(K)$ ([Sil86, p. 136]). Specifically, $\#E(\mathbb{F}_{\ell^2}) = \ell^2 + 1 - \alpha^2 - \overline{\alpha}^2$, where $\alpha, \overline{\alpha}$ are the two roots (in $\mathbb{C}$) of the equation

$$\phi^2 - 2p\phi + \ell = 0.$$

5.3. **An Example.** The example provided here was produced using the computer algebra package MAGMA [BC03]. For this example we take $D = -4$. For any prime p, a suitable prime ℓ is one that satisfies $4\ell = 4p^2 + 4y^2$ such that $\ell \equiv -1 \mod p$. The congruence implies that $y^2 = -1 \mod p$, in other words -1 should be a quadratic residue modulo p. This in turn implies that $p \equiv 1 \mod 4$, and that values of y that we need to search should be congruent to one of the square roots of $-1 \mod p$.

Let $p = 263300183685717422065746325660655084022315089991 53$. We search for prime values of $p^2 + y^2$ with

$$y \equiv \begin{cases} 20611019915125603610370027322246404729378417721286 \mod p \\ 57189984534461385962046052438191036728530912778 67 \mod p \end{cases} \quad \text{or}$$

corresponding to the two square roots of $-1 \mod p$. We find that

$$y = 18751503026220398352630035174344702002312902302177 30$$

yields a prime, so we take

$$\begin{aligned} \ell &= p^2 + (18751503026220398352630035174344702002312902302177 30)^2 \\ &= 35168819272908168996348622156834481670445567551962199157265479 28 \\ &\quad 60046102641340797974735424442696107030 9. \end{aligned}$$

The complex multiplication method tells us that the elliptic curve

$$E : y^2 = x^3 + x \text{ (in affine form)}$$

is a suitable elliptic curve. MAGMA tells us that $\#E(\mathbb{F}_\ell)$ is

$$351688192729081689963486221568344816704455675519621986306651119145697661326414284761633743996394307200 4,$$

which is indeed $\equiv 0 \mod p$. This computation took 0.063 seconds on an AMD Opteron 252 (2.6Ghz) processor. The number of points on $E(\mathbb{F}_{\ell^2})$ according to MAGMA is

$$123684584905047707258686141200578231465582664681874593612259486008465018014484601426538373930078429096341769913557802164349311875508547262692347038857763841422688694938944680813194533367728120360965744626464$$

and this is $\equiv 0 \mod p^2$, which is a necessary condition for $E[p]$ being a subgroup of $E(\mathbb{F}_{\ell^2})$. We show that $E[p]$ is indeed contained in $E(\mathbb{F}_{\ell^2})$ by finding two points that generate the p-torsion subgroup. Following the method outlined in §5.2 we find two p-torsion points, P and Q, that generate the whole p-torsion of $E(\mathbb{F}_{\ell^2})$

$$P = (276701049983509532234106338452082440292711762773463732533683876759414814860205833084376323976972215486 2, 736895619074862870441993260428363309212341952700619999020137331297834986221601940750818713297548511336)$$

$$Q = (170343693342782875614389009934880452275069084044323551866473740367532495756430307839699252460478525033 3u + 157128874698661854995016811716722095152507760097755673129863778174369969862913861485893531567999094343962932629794146247765964324029396184318939075174280958297655205533263210294725652408140056656867954141 90u + 282722913652845416300118493715740616379521916237377189328124466481421733687054166538367154312288563850 81).$$

Here u is a variable that gives the isomorphism $\mathbb{F}_{\ell^2} \cong \mathbb{F}_\ell[u]/(f(u))$ for a quadratic irreducible $f \in \mathbb{F}_\ell[u]$. The Weil pairing of P and Q is

$$\mathbf{e}_p(P,Q) = 188036180299835372546533903820354629932054094777699080104603766041577935958159317265607540618580827567 2u+312846556839611170253789382650488975505407147891209527580710819940254935617188961672586079797958196531 5.$$

References

[ALV02] Agashe, A.; Lauter, K.; Venkatesan, R.; *Constructing elliptic curves with a known number of points over a prime field*. In High Primes and Misdemeanours: lectures in honour of the 60th birthday of Hugh Cowie Williams, Fields Institute Communication Series, **42**, 1 17, 2002.

[ACLY00] Ahlswede, R.; Cai, N.; Li, S.-Y. R.; Yeung, W.; *Network Information Flow*, IEEE Trans. Information Theory, **46**(4), 1204 - 1216, 2000.

[AtMor93] Atkin, A., O., L.; Morain, F.; *Elliptic curves and primality proving*, Math. Comp., **61**, no. 203, 29-68, 1993.

[BLS02] Barreto, P. S. L. M.; Lynn B.; Scott M.; *Constructing Elliptic Curves with Prescribed Embedding Degrees*, Security in Communication Networks – SCN'2002, Lecture Notes in Computer Science 2576, Springer-Verlag, 263–273, 2002.

[BK98] Balasubramanian, R.; Koblitz, N.; *The Improbability That an Elliptic Curve Has Subexponential Discrete Log Problem under the Menezes-Okamoto-Vanstone Algorithm*, Journal of Cryptology, **11**, No. 2, 141-145, 1998.

[BGG94] Bellare, M.; Goldreich, O.; Goldwasser, S.; *Incremental cryptography: The case of hashing and signing*, in Advances in Cryptology CRYPTO'94, Santa Barbara, CA, 1994.

[BSS99] Blake, I.; Seroussi, G.; Smart, N.; *Elliptic Curves in Cryptography*, Lond. Math. Soc., Lecture Note Series, **265**, Cambridge University Press, 1999.

[BLS04] Boneh, D.; Lynn, B.; Shacham, H.; *Short signatures from the Weil pairing*, J. of Cryptology, **Vol. 17**, No. 4, pp. 297-319, 2004.

[BC03] Bosma, W.; Cannon, J.; *Handbook of MAGMA functions,* Sydney, 2003.

[CLY03] Cai, N.; Li, S.-Y. R.; Yeung, W.; *Linear Network Coding*, IEEE Trans. Information Theory, **49**(2), 371-381, 2003.

[CL05] Charles, D.; Lauter K.; *Computing modular polynomials*, Lond. Math. Soc. Journal of Computation and Mathematics, **8**, 195-204, 2005.

[CJW03] Chou, P. A.; Jain, K.; Wu, Y.; *Practical network coding*, in Allerton Conference on Communication, Control, and Computing, Monticello, IL, 2003.

[CJL05] Chou, P. A.; Lovász, L.; Jain, K.; *Building scalable and robust peer-to-peer overlay networks for broadcasting using network coding*, ACM Symposium on Principles of Distributed Computing, Las Vegas, NV, July 2005.

[ELM04] Eisenträger, K.; Lauter, K.; Montgomery, P.; *Improved Weil and Tate pairings for elliptic and hyperelliptic curves*, In: Algorithmic Number Theory - ANTS-VI, Buell (Ed.), Lecture Notes in Computer Science vol. 3076, 169–183, Springer-Verlag, 2004.

[GR05] Gkantsidis, C.; Rodriguez, P.; *Network coding for large scale content distribution*, in IEEE INFOCOM, Miami, 2005.

[GR06] Gkantsidis, C.; Rodriguez, P.; *Cooperative security for network coding file distribution*, To appear in IEEE INFOCOM, Barcelona, April 2006. (Also as Microsoft Research Technical Report, MSR-TR-2004-137)

[KFM04] Krohn, M. N.; Freedman, M. J.; Maziéres, D.; *On-the-Fly Verification of Rateless Erasure Codes for Efficient Content Distribution*, In the Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, 2004.

[LTr76] Lang, Serge; Trotter, Hale, F.; *Frobenius distributions in* GL_2*-extensions*, Lecture Notes in Math., **504**, Springer-Verlag, 1976.

[LL90] Lenstra, A. K.; Lenstra, H. W., Jr.; *Algorithms in number theory*, Handbook of theoretical computer science, **Vol. A**, Elsevier, Amsterdam, 673-715, 1990.

[Mil86] Miller, V.; *Short programs for functions on curves*, unpublished manuscript, 1986.

[MOV93] Menezes, A.; Okamoto, T.; Vanstone, S.; *Reducing elliptic curve logarithms to logarithms in a finite field* IEEE Trans. on Information Theory, **39**, 1639-1646, 1993.

[MNT01] Miyaji, A.; Nakabayashi, M.; Takano, S.; *New explicit conditions of elliptic curve traces for FR-reductions*, IEICE Trans., Fundamentals. vol. E84-A, No.5(2001), 1234-1243.

[Sch85] Schoof, R.; *Elliptic curves over finite fields and Computation of square roots* mod p, Math. Comp., Vol. **44**, no. 170, 483-494, 1985.

[Sil86] Silverman, J.; *The Arithmetic of Elliptic Curves*, Graduate Texts in Math. Vol. 106, Springer-Verlag, 1986.

Microsoft Research, Redmond, WA 98052
E-mail address: {cdx, kjain, klauter}@microsoft.com