# Discrete logs and Fourier Coefficients

by
David Mireles

Mathematics Department
Royal Holloway, University of London
Egham, UK

## 1 Introduction

We describe an algorithm that gives a reduction from the discrete log problem in the multiplicative group $\mathbf{F}_p^{\times}$, to the problem of calculating the Fourier coefficients of a Hecke eigenform of level $p$.

## 2 Some Theory

The algorithm uses standard results concerning the Fourier coefficients of a modular form $f$ of level $N$, weight $k$ and attached Dirichlet character $\chi$ when acted upon by the $m$-th Hecke operator $T_m$ (remember that $\chi$ is a homomorphism $\chi : (\mathbf{Z}/N\mathbf{Z})^* \longrightarrow \mathbf{C}^*$). The following proposition can be found in [Dia-Sh]:

**Proposition 2.1.** *Let $f$ be a modular form as described above having Fourier expansion $\sum a_n(f)q^n$. Suppose further that $f$ is a normalized eigenform for the Hecke algebra, and that*

$$T_m f = \lambda(m) f,$$

*then we have:*

1. *For every $n$, $\lambda(n) = a_n(f)$.*
2. *If $l$ is a prime and $r \geq 2$, then*

$$a_{l^{r+1}}(f) = a_l(f)a_{(l^r)}(f) - \chi(l)l^{k-1}a_{l^{r-1}}(f).$$

3. $a_n(f)a_m(f) = a_{nm}(f)$ *whenever* $(n, m) = 1$.

Using the previous proposition, it is clear that if we can compute the Fourier coefficients $a_l(f)$ and $a_{l^2}(f)$ of the eigenform $f$, then we can recover the value of $\chi(l)$ for $l$ prime. This is all we need to describe our algorithm.

## 3 The Algorithm

The input to the algorithm are elements $g, g^k \in (\mathbf{Z}/p\mathbf{Z})^*$ where $p$ is a prime, and we are asked to find the value of $k$. The algorithm runs as follows:

DLP-REDUCTION

1 Specify a modular form $f$ of level $p$, with associated Dirichlet character $\chi$.
2 Calculate random powers of $g$: $\{g^i\}$ until the unique representative $l$ of $g^{i_0}$ in $[-(p-1)/2, (p-1)/2]$ is a prime number.
3 Using an oracle to the Fourier coefficients of a modular form $f$ of level $p$, calculate $\chi(g^{i_0}) = \chi(l)$.
4 Using $\chi(g^{i_0})$, compute $\chi(g)$.
5 Repeat to compute $\chi(g^k)$.
6 Since $\chi(g^k) = \chi(g)^k$, and the DLP is trivial in $\mathbf{C}^*$, we can compute $k$.

A detailed account of this algorithm can be found in the author's personal web page:
`http://personal.rhul.ac.uk/pqai/107/`

## 4 Comments

- The Dirichlet character $\chi$ associated to $f$ shouldn't be trivial. Even if $\chi$ is not trivial, some information might be lost if the order of $\chi(g)$ is not $p-1$. One might try to solve this problem iterating the reduction using different $f$'s every time.
- The complexity depends on the representation of the Fourier coefficients $a_n(f)$. The fact that the value of $\chi(p)$ is an algebraic combination of the $a(p)$'s says that if $\chi$ is such that $\chi(g)$ has order sufficiently large to give us useful information, then the degree of the $a_p(f)$'s as algebraic integers will be very large, therefore, representing the coefficients as algebraic integers will lead to exponential complexity, so we propose to think of the coefficients as complex numbers calculated with enough precision to identify the root of unity corresponding to $\chi(g)$ and $\chi(g^a)$.
- The weight of the modular form doesn't play a role in the algorithm, so it is worth mentioning that there are techniques available to calculate the coefficients of Hecke eigenforms of weight 2; Cremona's tables of modular elliptic curves are based on the possibility of calculating the Fourier coefficients of forms of weight 2 and being able to distinguish when they are all rational.
- Bas Edixhoven has outlined an algorithm to calculate the $p$-th Fourier coefficient of a given modular form in polynomial time (polynomial in $p$ once the modular form is fixed); in [ECdJ] the authors restrict the algorithm to Ramanujan's $\tau$-function, but it seems this method should work for general modular forms. Another result related with the reduction presented in this note is in Denis Charles' thesis [Char], where he proves that being able to compute the values of Ramanujan's $\tau$-function is not more difficult that being able to factor RSA moduli, a difference between his approach and ours is that he considers the problem of calculating the $n$-th Fourier coefficient for arbitrary $n$, whereas we restrict ourselves to computing Fourier coefficients for $n$ a prime power. Charles' result supports a claim by Edixhoven, saying that in order to compute the $n$-th Fourier coefficient of an eigenform, one must be able to factor $n$.

## References

[Char] Denis X. Charles, *Computational Aspects of Modular Forms and Elliptic Curves*, PhD thesis, University of Winsconsin-Madison, 2005.

[Dia-Sh] F. Diamond and J. Shurman, *A First Course in Modular Forms*, Springer-Verlag 2005.

[ECdJ] B. Edixhoven, J. M. Couveignes, R. de Jong et. al. *On the computation of coefficients of a modular form*, ArXiv preprint *math. NT/0605244* .

[G-M] S. Galbraith and D. Mireles, *Discrete logarithms and modular forms*, `http://personal.rhul.ac.uk/pqai/107/`.