Abelian Variety Cryptosystems

March 28, 2015

Contents

In recent years there has been increased interest in public key cryptography. The use of elliptic curves in cryptography was suggested independently by Neal Koblitz and Victor S. Miller in 1985 and have entered wide use in 2004 to 2005. In this report we give explicit descriptions for algorithms need to implement public key cryptosystems using abelian varieties.

0.1 The Discrete Logarithm Problem

Given an arbitrary finite cyclic group $\mathfrak{G}$ with group operation $\cdot$ and generator $g \in \mathfrak{G}$, discrete exponentiation by a in $\mathfrak{G}$ is defined by

$$g^a = \overbrace{g \cdot g \cdots g}^{a \text{ times}}$$

If $y = g^a$ is known, computing a is called finding the discrete logarithm of y. With the method of fast exponentiation, y can be computed quickly, only $O(\log a)$ group operations. On the other hand, computing a can be much harder. In fact, in [2] it was shown that in an arbitrary group for which only the group operation and discrete exponentiation can be applied to group elements, computing discrete logarithms will take at least $O(\sqrt{|\mathfrak{G}|})$ operations. In most cases though, more structure is kno about the group in use.

0.1.1 The Diffie-Hellman Key Exchange

This one-way property of discrete exponention has proven to be very useful for cryptographic purposes. The most notable of these is in the Diffie-Hellman Key Exchange Protocal in which two parties A and B wish to share a secret key k.

1. A and B share a publicly known group $\mathfrak{G}$ and generator $\mathfrak{g}$.
2. A chooses a random private exponent a and computes g^a.
3. B chooses a random private exponent a and computes g^b.
4. A sends g^a to B and B sends g^b to A.
5. A raises g^b to their own private exponent a to obtain $k = (g^b)^a = g^{ab}$.
6. B raises g^a to their own private exponent b to obtain $k = (g^a)^b = g^{ba}$.

The two parties may now use k to communicate with a cryptographically secure communication protocol. The described protocol relies on the hardness of computing g^{ab} given g^a and g^b, which is conjectured in [3] to be equivalent to computing discrete logarithms.

0.1.2 The Digital Signature Algorithm

Another nice property about this protocol is it's applicable to any finite cyclic group. The simplist example of this is in the multiplicative units of the integers modulo a prime, denoted $\mathbb{Z}_p^*$. Let g be a primite root mod p, then every element of $y \in \mathbb{Z}_p^*$ can be written in the form $y = g^a$ for some $a < p - 1$. Thus $\mathbb{Z}_p^*$ is a finite cyclic group and one can compute discrete logarithms in $\mathbb{Z}_p^*$.

0.2 Algebraic Varieties

When working with polynomials in more than one variable, it is sometimes simpler to use the formalism of classic algebraic geometry as found in [1]. Let K be an arbitray field (for the purposes of this report we will only consider $K = \mathbb{Z}_p$, where $p > 3$).
Let $A = K[X_1, ..., X_n]$ represent the polynomial ring in n variables over K. Given a subset $T \subseteq A$, we may define

$$Z(T) \stackrel{\text{def}}{=} \{x \in K^n \mid f(x) = 0 \text{ for all } f \in T\}$$

to be the set of all common zeros of polynomials in T. We call $Y \subseteq K^n$ an *algebraic subset* if $Y = Z(T)$ for some subset $T \subseteq A$. Similary, given an subset $Y \subseteq K^n$, may define the set of A

$$I(Y) \stackrel{\text{def}}{=} \{f \in A \mid f(x) = 0 \text{ for all } x \in Y\}$$

In fact, $I(Y)$ is an ideal in A often called *the ideal of* Y. So we may consider the quotient ring

$$A(Y) \stackrel{\text{def}}{=} A/I(Y)$$

We refer to this ring as the *the coordinate ring of* Y.

0.2.1 Dimension

0.2.2 Genus

0.3 Elliptic Curves

Let p be an odd prime and let $E = Z(y^2 - x^3 - ax - b)$. We call E and *elliptic curve* defined over $K = \mathbb{Z}_p$ if $4a^3 + 27b^2 \not\equiv 0 \bmod p$. It was first realized by Abel and Jacobi in the 1700's that remarkably, the $\mathbb{Z}_p$-rational points of E can be transformed into a group using a very specific group operation. In this section we describe this group operation along with other algorithms needed for cryptographic purposes.

0.3.1 The Group Operation

Given two $\mathbb{Z}_p$-rational points $P_1 = (x_1, y_1), P_2 = (x_2, x_2)$ which we want to add together, we first define the value

$$s = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2} \mod p & \text{if } P_1 \neq P_2 \\ \frac{3x_1^2 - a}{2y_1} \mod p & \text{if } P_1 = P_2 \end{cases}$$

Then we define the coordinates of the point $P_3 = P_1 + P_2$ to be

$$x_3 = s^2 - x_1 - x_2$$
$$y_3 = y_1 + s(x_3 - x_1)$$

It's not immediately obvious that $P_3 = (x_3, y_3)$ is even a point on E and less obvious that this operation satisfies the axioms of a group. SEE IF YOU CAN DO THE CALCULATIONS

0.3.2 Scalar Multiplication of a Point

For many discrete logarithm protocols (such as Diffie-Hellman of DSA), we require to add point P to itself many times in order to perform discrete exponentiation. That is, given an integer m we need to calculate

$$mP = \overbrace{P + P + \cdots + P} m \text{ times}$$

fast in order to be cryptographically reasonable. The following algorithm does this.

Algorithm 1. FastScalarMultiplication

INPUT: An integer m, a point P on an elliptic curve $E : y^2 = x^3 + ax + b$
OUTPUT: mP

---

1. **if** $m == 0$ **then return** $\emptyset$
2. **if** $m == 1$ **then return** P
3. **if** $m == 0 \mod 2$
4. **return FastScalarMultiplication**($m/2$,$P + P$,E)
5. **else**
6. **return** $P+$ **FastScalarMultiplication**($m - 1$,P,E)

0.3.3 Finding Points

Now that we have developed arithemtic on and elliptic curve E, then next step is figure out how to find points on E. If $y^2 = x^3 + ax + b$ for $a, b \in \mathbb{Z}_p$, then finding $\mathbb{Z}_p$-rational points on E is equivalent to determining if $x^3 + ax + b$ is a square mod p. This is a classical problem in number theory which can be reformulated to determing the value of the *Legendre Symbol*, which is defined as

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \text{ is a square mod } p \\ -1 & \text{if } a \text{ is not a square mod } p \\ 0 & \text{if } p \text{ divides } a \end{cases}$$

where (in our case) $a = x^3+ax+b$. The following five properties let us determine whether a is a square mod p in polynomial time. Let $a, b \in \mathbb{Z}$ and p, q be odds primes.

(i) If $a \equiv b \bmod p$, then $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$

(ii) $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{p}\right)$

(iii) $\left(\frac{-1}{p}\right) = 1$ if $p \equiv 1 \bmod 4$, and $\left(\frac{-1}{p}\right) = -1$ if $p \equiv 3 \bmod 4$

(iv) $\left(\frac{2}{p}\right) = 1$ if $p \equiv \pm 1 \bmod 8$, and $\left(\frac{2}{p}\right) = -1$ if $p \equiv \pm 3 \bmod 8$

(v) If p, q are distinct, then

$$\left(\frac{p}{q}\right) = \left(\frac{q}{p}\right) \quad \text{if } p \text{ or } q \equiv 1 \bmod 4$$

$$\left(\frac{p}{q}\right) = -\left(\frac{q}{p}\right) \quad \text{if } p \equiv q \equiv 3 \bmod 4$$

For example, to determine if 105 is a square mod 227, we simply compute

$$\begin{aligned} \left(\frac{105}{227}\right) &\overset{\text{(ii)}}{=} \left(\frac{3}{227}\right)\left(\frac{5}{227}\right)\left(\frac{7}{227}\right) \\ &\overset{\text{(v)}}{=} (-1)\left(\frac{227}{3}\right)\left(\frac{227}{5}\right)(-1)\left(\frac{227}{7}\right) \\ &\overset{\text{(i)}}{=} \left(\frac{2}{3}\right)\left(\frac{2}{5}\right)\left(\frac{3}{7}\right) \\ &\overset{\text{(iv)}}{=} (-1)(-1)(-1)\left(\frac{7}{3}\right) \\ &\overset{\text{(i)}}{=} (-1)\left(\frac{1}{3}\right) \\ &= -1 \end{aligned}$$

So 105 is a not a square mod 227. This simple process requires $O(\log^3 p)$ reference bit operations but doesn't actually tell us the squareroot of a, if a is indeed a square mod p. If $\left(\frac{a}{p}\right) = 1$, the following method finds x such that $x^2 = a$ mod p. We break the calcultions into two cases. If $p \equiv 3$ mod 4, then $x = a^{(p+1)/4}$ satisfies $x^2 \equiv a$ mod p. The second case where $p \equiv 1$ mod 4 is more involved.

1. Pick random r such that $\left(\frac{r^2-4a}{p}\right) = -1$ and write $d = r^2 - 4a$.
2. let $\alpha = \frac{r+\sqrt{d}}{2}$ and $\alpha^k = \frac{V_k+U_k\sqrt{d}}{2}$ where V_k, U_k are the coefficients of $1, \sqrt{d}$ in the k-th power of α.
3. $x = 2^{-1}V_{(p+1)/2}$ satisfies $x^2 \equiv a$ mod p.

For this case, the proof that x does satisfy $x^2 \equiv a$ mod p is quite involved but can be found in GarrWALSH NOTES. Putting all this together we obtain the following algorthm which finds random points on an elliptic curve E.

Algorithm 2. Find Point on Elliptic Curve

INPUT: An elliptic curve E and an odd prime p
OUTPUT: A point $P = (x, y)$ on E

---

1. Set $x \overset{r}{\in} \mathbb{Z}_p^*$, $c \leftarrow x^3 + ax + b$, $s \leftarrow$ False

2.

0.3.4 Counting Points

When using an elliptic curve E for discrete logarithm based cryptosystems, it's of fundamental importance to know the number of $\mathbb{Z}_p$-ratiomal points on E. This is because in 1978 Stephan Pohlig and Martin Hellman came up with an attack, described in [5], which uses the order of the group to solve discrete logs. Let G be a finite cyclic group with order N. We may factor $N = p_1^{e_1}p_2^{e_2}\cdots p_s^{e_s}$ where $p_1, p_2, ..., p_s$ are primes and $e_1, e_2, ..., e_s \in \mathbb{Z}^+$. All the subgroups $G_1, G_2, ..., G_s$ of G will have order $p_1^{e_1}, p_2^{e_2}, ..., p_s^{e_s}$ respectively. Given an element $h = g^a \in G$, the Pohlig-Hellman attack solves for a in the subgroups $G_1, G_2, ..., G_s$ and uses the chinese remainder theorem to piece these solutions back together to solve for a in the bigger group G. What they noticed is that using this method the complexity of solving for a in G (which can be done in $O(\sqrt{N})$) gets reduced to $O(p_1^{e_1/2}) + O(p_2^{e_2/2}) + O(p_s^{e_s/2})$. This means a group's bits of security is only as high as the bits of security of its largest subgroup.

0.3.5 Generating Provably Random Elliptic Curves

0.4 Hyperelliptic Curves

Unlike elliptic curves, when the genus $\mathfrak{g}$ of a curve $\mathfrak{C}$ is greater than 1, the set of points on $\mathfrak{C}$ will not always form a group.

Example 0.4.1.

0.4.2 The Jacobian of a Hyperelliptic Curve

Luckily, there is another way to form an abelian group with hyperelliptic curves. Indeed, let $\mathfrak{D}$ be the set of all formal finite sums

$$\sum_i m_i P_i$$

where $m_i \in \mathbb{Z}$ and P_i are points on the curve $\mathfrak{C}$. We call elements of $\mathfrak{D}$ divisors of $\mathfrak{C}$. Given a rational function f in $\mathbb{Z}_p[\mathfrak{C}]$, we can define the corresponding divisor to f as

$$(f) = \sum_i m_i P_i$$

where P_i are the zeros and poles of f with multiplicities m_i.

Example 0.4.3.

Divisors of this form are called principal divisors and we let $\mathfrak{P}$ denote the subset of all of them in $\mathfrak{D}$. If we define the operation on $\mathfrak{D}$ by

$$\sum_i m_i P_i + \sum_i m'_i P_i = \sum_i (m_i + m') P_i$$

then $\mathfrak{D}$ becomes and abelian group. Unfortunetly, this group is far too large and unstructured for cryptographic purposes. So we consider the subgroup $\mathfrak{D}^0$ of all divisors of $\mathfrak{D}$ whose coefficients sum to 0. That is, divisors $\sum_i m_i P_i$ such that $\sum_i m_i = 0$.

This subgroup is still infinite, but that can be remedied by defining two divisors D_1, D_2 of $\mathfrak{D}^0$ to be equal if $D_1 - D_2$ is equal to the divisor of a rational function on $\mathfrak{C}$. That is, $D_1 - D_2 = (f)$ for $f \in \mathbb{Z}_p[\mathfrak{C}]$. This new quotient group, denoted

$$\mathfrak{J} = \mathfrak{D}^0 / \mathfrak{P}$$

is called the jacobian of the curve $\mathfrak{C}$ and is a finite cyclic group. This will be the group used to build hyperelliptic cryptosystems.

0.4.4 Representation of Divisors

Athough the Jacobian $\mathfrak{J}$ of an hyperelliptic curve $\mathfrak{C}$ is a finite abelian group, elements of $\mathfrak{J}$ are very hard to represent.

Example 0.4.5.

To make the group operation in $\mathfrak{J}$ tractable, we ustilize the mumford representation of a divisor which is described as follows. Let D be a semi-reduced with points $P_i = (x_i, y_i)$. We associate to D polynomials $a, b \in \mathbb{Z}_p[x]$ such that

$$a(x) = \prod_i^r (x - x_i)$$

$$b(x_i) = y_i \ 1 \leq i \leq r$$

where $\deg b < \deg a$ and $(x - x_i)^{k_i} \mid b - y_i$, if k_i is the multiplicity of P_i. Denote this representation $D \stackrel{\text{def}}{=} \text{div}(a, b)$.

0.4.6 The Group Operation

The group operation can be divided into two parts - *Composition* and *reduction* as described in [4].

Given two divisors represented as $D_1 = \text{div}(a_1, b_1), D_2 = \text{div}(a_2, b_2)$

1. compute $d_0 = \gcd(a_1, a_2)$ and find the unique $c_1, e_1 \in \mathbb{Z}_p[x]$ such that $d_0 = c_1a_1 + e_1a_2$
2. compute $d = \gcd(d_1, b_1 + b_2)$ and find the unique $c_2, e_2 \in \mathbb{Z}_p[x]$ such that $d = c_2d_1 + e_2(b_1 + b_2)$
3. compute $a_3 = \frac{a_1, a_1}{d^2}$
4. compute $b_3 = \frac{c_2c_1a_1 + c_2e_1a_2 + e_2(b_1b_2 + f)}{d} \bmod \frac{a_1, a_1}{d^2}$
5. compute $a_3' = \frac{f - b_3^2}{a_3}$ and $b_3' = -b_3 \bmod a_3'$
6. while $\deg(a_3') > g$, reassign $a_3 = a_3', b_3 = b_3'$ and repeat step 5
7. divide a_3' by its leading coefficient so that a_3' becomes monic
8. the output $div(a_3', b_3') = D_1 + D_2$

Why Does this work?

Example 0.4.7.

0.4.8 Finding Points

0.4.9 Counting Points

0.5 Abelian Varieties

0.6 Applications

0.7 References

1 Robin Hartshorne, *Algebraic Geometry, Graduate Texts in Mathematics, vol. 52*, Springer-Verlag, New York, 1977, ISBN 0-387-90244-9.

2 Victor Shoup, *Lower bounds for discrete logarithms and related problems*, Theory and Application of Cryptographic Techniques, 1997, pp. 256 - 266.

3 Whitfield Diffie and Martin E. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory **IT-22** (1976), no. 644-654.

4 Tanja Lange, *Formulae for Arithmetic on Genus 2 Hyperelliptic Curves*, ... not complete.

5 Stephan Pohlig and Martin E. Hellman, *An Improved Algorithm for Computing Logarithms over $GF(p)$ and its Cryptographic Significance*, IEEE Transactions on Information Theory, volL. m24, NO. 1, January 1978, pg 106.