Pohlig-Hellman Applied in Elliptic Curve Cryptography

Martin Lysoe Sommerseth and Haakon Hoeiland
martin.sommerseth@hotmail.no, haakon.hoeiland@gmail.com

December 7, 2015

1 Abstract

The Pohlig-Hellman algorithm is an algorithm that solves the discrete logarithm problem. The algorithm simplifies the problem by solving the elliptic curve discrete logarithm problem (ECDLP) in the prime subgroups of the point $\langle \mathrm{P} \rangle$. The difficulty of solving the ECDLP in its prime order subgroups is no harder than solving the ECDLP in $\langle \mathrm{P} \rangle$ [1, 3].

In our paper we will present the Pohlig-Hellman algorithm and its applications. We will discuss its complexity and how to construct the elliptic curves in order to defend against the Pohlig-Hellman attack. At the very end we will briefly discuss how to choose parameters such that the effectiveness of other attacks is also minimized.

2 Introduction

2.1 Groups and fields

It is important to know what groups and fields are when we discuss elliptic curve cryptography. A group consists of a set S and a binary operation that usually is addition or multiplication denoted by $\oplus$. The set S and the binary operation have to satisfy the rule of closure, associativity and commutativity. It also has to satisfy that every element $a \in S$ has an inverse $\mathrm{inv}(a) \in S$, and a neutral element $e \in S$ such that $a \oplus e = e \oplus a = a$ exist [4].

A field F consists of the two operations, addition and multiplication, and a set S. The additive group is denoted by $G_a = (S,\oplus)$ and the multiplicative group is denoted by $G_m = (S^*,\oplus)$ [5].

2.2 Elliptic Curve

An elliptic curve (EC) is defined as the solution set of a nonsingular cubic polynomial equation with two variables over a field $\mathcal{F}$:

$$\mathcal{E} = \{(x, y) \in \mathcal{F} \times \mathcal{F} \mid f(x, y) = 0\} \tag{1}$$

The general equation in a cubic with two different variables is given by:

$$\begin{aligned} &ax^3 + by^3 + cx^2y + dxy^2 + ex^2 \\ &+fy^2 + gxy + hx + iy + j = 0 \end{aligned} \tag{2}$$

The Weierstrass form of the elliptic curve is only valid when $char(\mathcal{F}) \neq \{2, 3\}$. This curve is the most known, and is given by: [6]

$$y^2 = x^3 + ax + b \tag{3}$$

2.3 Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is an approach to public-key cryptography. The concept is based on the algebraic structure of EC over finite fields. The advantage of ECC compared to non-ECC is that it requires smaller keys to provide equivalent security. The key exchange in ECC is described in *figure 1.* Here user A and B agree on an elliptic curve E and a point P on the curve with order n. User A chooses a random number a, where $a \in [0, n-1]$, and then computes the private key which is $[a]P$. User B does the same and chooses a random number b, where $b \in [0, n-1]$ and computes B's private key $[b]P$. The public key, S, will then be $S = [a][b]P$ [7].

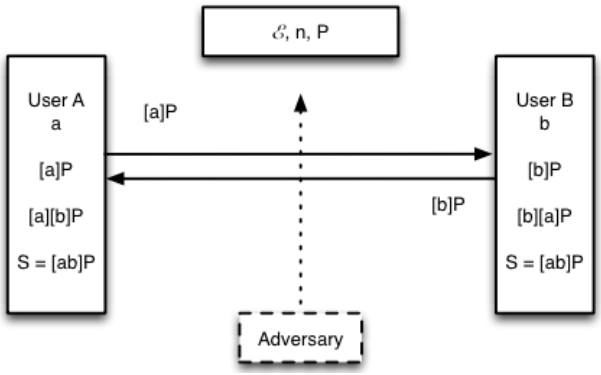

Figure 1: Key exhange in ECC [6].

2.4 Elliptic curve discrete logarithm problem

Take the elliptic curve E, defined over the finite field $\mathbb{F}_q$. The two points P and Q are given, where P has order n, and $P \in E(\mathbb{F}_q)$. Q is a point on the curve, that is $Q \in \langle P \rangle$. Then write $Q = lP$, where the integer $l \in [0, n-1]$. The elliptic curve discrete logarithm problem (ECDLP) is then defined as finding the integer l that solves the equation $Q = lP$. l is called the discrete logarithm of Q to the base P, meaning $l = \log_p Q$ [1].

Solving ECDLP in reasonable time is believed to be beyond what is computable with today's technology, as long as the parameters are chosen to avoid the known attacks. The hardness of the problem creates the fundament of the security in all elliptic curve cryptography. The best general known attack on elliptic curve cryptography is a combination of the Pohlig-Hellman and Pollard's rho algorithms. Given the largest primal divisor of n, denoted p; such an attack solves ECDLP in $O(\sqrt{p})$ time. An algorithm that solves the problem in polynomial time is likely to exist, because if there is no such algorithm, it would imply P$\neq$ NP [1].

3 Pohlig-Hellman Attack

The Pohlig-Hellman algorithm was presented by Stephan C. Pohlig and Martin E. Hellman in 1978. In the original paper it is presented as an improved algorithm used to compute discrete logarithms over the cyclic field $G = GF(p)$, and how their findings impact elliptic curve cryptography [2].

Given the ECDLP $Q = lP$, the Pohlig-Hellman algorithm is a recursive algorithm that reduces the problem by computing discrete logarithms in the prime order subgroups of $\langle$P$\rangle$. Each of these smaller subproblems can then be solved using methods, such the Pollard's rho alogrithm.

The Pohlig-Hellman algorithm works as follows:

- Write n as $n = p_{11}^{e} p_2^{e_2} \dots p_r^{e_r}$
- Compute $l_i = l \bmod p_i^{e_i}$ for all $1 \leq i \leq r$.

The Chinese Remainder Theorem is then used to obtain a unique solution to the following system of congruences:

$$\begin{aligned} l &\equiv l_1 \pmod{p_1^{e_1}} \\ l &\equiv l_2 \pmod{p_2^{e_2}} \\ &\dots \\ l &\equiv l_r \pmod{p_r^{e_r}} \end{aligned} \tag{4}$$

Here $p_1, p_2, ..., p_r$ is a mutually coprime set of positive integers, i.e. $\gcd(p_i, p_j) = 1$ for all $i = j$. $l_1, l_2, ..., l_r$ are all positive integers such that $0 \leq l_i < p_i$. The unique positive integer l can be computed efficiently by using the Extended Euclidean Algorithm to compute the linear congruences above [11].

Each l_i can then be expressed in base p the following way:

$$l_i = z_0 + z_1 p + z_2 p^2 + ... + z_{e-1} p^{e-1} \tag{5}$$

where $z_i \in [0, p-1]$. Let

$$P_0 = \frac{n}{p_i} \tag{6}$$

and

$$Q_0 = \frac{n}{p_i} Q \tag{7}$$

Rewriting the equations, and using the fact that the order of P_0 is p, we get

$$Q_0 = lP_0 = z_0 P_0 \tag{8}$$

Then z_0 is found by computing the ECDLP solution in $\langle P_0 \rangle$. Every $z_0, ..., z_{e-1}$ is then computed by solving

$$Q_i = z_i P_0 \tag{9}$$

in $\langle P_0 \rangle$ [1, 3].

3.1 Complexity

The Pohlig-Hellman algorithm's worst-case time complexity is $O(\sqrt{n})$ for a group of order n. What is interesting is that it is a lot more efficient when the order is B-smooth. A B-smooth order means that the order has prime factors, which are less than B. Then, if the prime factorisation of order n is $\prod_i p_i^{e_i}$, the complexity will be $O\left(\sum_i e_i(\log n + \sqrt{p_i})\right)$ [8].

It turns out that the best known general-purpose attack on ECDLP is a combination of the Pohlig-Hellman algorithm and the Pollard's rho algorithm.

Pollard's rho algorithm is an algorithm based on integer factorization, and is really effective for a composite number with a small prime factor. The combination of these algorithms has a fully-exponential running time of $O(\sqrt{p})$ where p is the highest prime factor of the order n.

Therefore it is really important to choose elliptic curve parameters so that n is divisible by a large prime number p, which makes it extremely hard and infeasible to finish the computation in a reasonable time. This will be discussed in subsection 3.2 [9, 10].

3.2 Defending Against Pohlig-Hellman Attacks

Pohlig-Hellman is designed to reduce the problem so that the ECDLP is solved in the prime order subgroups of n, where n is the order of the given point P. It is possible to choose the elliptic curve paramaters in such a way that it reduces the effectiveness of the Pohlig-Hellman attack. The way this is done is by making n divisible by a large prime, p [1].

If the largest prime that divides n is sufficiently large, it follows that solving this ECDLP over this prime is very hard. P is sufficiently large when $p > 2^{160}$ [1]. The reason why p has to be bigger than 2^{160} is because this is the limit where solving the Chinese Remainder Theorem is taking so much time and solving the problem is infeasible. If the parameters of elliptic curves are also carefully chosen to defeat all other known attacks, then by today's computer technology it is infeasible to solve the ECDLP problem, which implies that using ECC in today's cryptography is still secure [10].

It is important to keep in mind that there is no mathematical proof that the ECDLP is impossible to solve within reasonable time. There exists no proof that there does not exist an efficient algorithm that solves the ECDLP. A proof of this kind would revamp the whole computer science industry, due to the non-existence of a polynomial-time algorithm for the ECDLP implies that P $\neq$ NP [10].

To show the difference in time of solving the ECDLP with Pohlig-Hellman we will show two examples, choosing two different points in a large Elliptic group. One with order that has a relatively small highest prime factor and the other with a order that has a large highest prime factor. We will look on the difference in time spent for solving the ECDLP and refers to the article "Weak Curves In Elliptic Curve Cryptography" written by Novotney. He looks at the elliptic curve $\mathcal{E}$: $y^2 = x^3 + 7x + 1$ and Galios Field over prime $p = 4516284508517$:

- When the order of the point P is 4516285972627 it has the prime factors $11 \cdot 13 \cdot 31582419389$. Solving the ECDLP with Pohlig-Hellman takes 49.14 seconds.
- When the order of the point P is 9254332285624 it has the prime factors $2^3 \cdot 19 \cdot 23 \cdot 67 \cdot 2089 \cdot 18913$. Solving the ECDLP with the same Pohlig-Hellman code takes 0.16 seconds.

This example confirms that using a order n with a larger prime factor takes much more time. Just imagine how long it would have taken if the largest prime factor was bigger than 2^{160} [3].

4 Defending againt other attacks

In the previous section we explained how the Pohlig-Hellman algorithm can be used to exploit the security of certain elliptic curves. In this section we will describe other ways of attacking systems that uses ECC and how to choose the parameters to avoid the ECDLP being easily solved.

4.1 Exhaustive search

This attack is a brute-force way of breaking the ECC. The search is done by computing lP for every l until $lP = Q$. The expected running time of the algorithm is $n/2$. Choosing $n \geq 2^{80}$ is sufficient to make the method unfeasible, given the current technology [1].

4.2 Pollard's rho

There are several versions of the Pollard's rho algorithm, including the ρ and the λ methods, as well as parallelized versions. The parallelized version using M processors, has time complexity $O(\sqrt{\frac{\pi n}{2}}/M)$ [12]. Hankerson, Menezes and Vanstone suggests using curves where the largest prime factor of n is $p > 2^{160}$, to combat this attack [1].

4.3 Anomalous curves

An elliptic curve is anomalous if $E(\mathbb{F}_p) = p$. It is important to avoid using such curves in ECC, because it allows ECDLP to be solved in $O(\log p)$ time [12].

4.4 Pairing attacks

Several different pairing attacks are presented by Musson in "Attacking the Elliptic Curve Discrete Logarithm Problem", 2006. Two of the most known algorithms of this kind are the MOV attack and the Frey-Rück attack, which uses Weil and Tate pairing [12]. These attacks exploit isomorphic properties to reduce ECDLP to the DLP. Because DLP can be solved in subexponential-time, one must avoid curves that are vulnerable to these methods.

Let $q^k \equiv 1 \mod n$. The pairing attacks are inefficient if n does not divide

$$q^k - 1 \tag{10}$$

for small k. When $n > 2^{160}$, it is then sufficient to check (10) for $k \in [1, 20]$ [1].

4.5 GHS attack

The GHS attack uses the Weil Descent technique to reduce the ECDLP to Hyperelliptic Curve DLP. The algorithms has different extensions, such as the use of isogenies. There is a simple way to avoid this attack, which is done by defining the curve over $\mathbb{F}_{2^p}$ where p is prime and $p \in [160, 600]$ [12].

4.6 Index calculus

Index calculus methods for abelian varieties have been developed, but the running time is proven to be $O(q^{2-2/n})$, over the field $\mathbb{F}_q$. This makes the attack unfeasible for big n [12].

The Xedni Calculus is an index calculus type of attack on ECDLP that has shown some interesting results. Much research has been made, trying to make it effective. However, this attack is proven to be inefficient for large primes. This is because its running time is $O(p)$ and it must also be repeated at least $O(p)$ times to solve the ECDLP [12].

5 Summary

The Pohlig-Hellman algorithm solves the ECDLP problem in a very efficient way if the order is B-smooth. The Pohlig-Hellman is a recursive algorithm that reduces the problem to computing the DLP in prime order subgroups of $\langle \mathrm{P} \rangle$. To defend against these attacks it is essential to choose an order which has biggest prime factor $p > 2^{160}$. If you always follow this guideline, using Pohlig-Hellman to solve the ECDLP in elliptic curve cryptography is infeasible with today's computer technology.

In order to defend against the other known attacks covered in this paper there are additional things to consider. One is to avoid anomalous curves, because they only take $O(\log p)$ to solve. Pairing attacks are inefficient if n does not divide $q^k - 1$ for small k, which also restricts the parameters. The GHS attacks are eliminated by making sure p is prime and $p \in [160, 600]$ in the field $\mathbb{F}_{2^p}$. The index calculus attacks are less effective than other attacks for large numbers, and one is protected as long as the constraints above are applied.

References

[1] Hankerson, Menezes, Vanstone: "Guide to Elliptic Curve Cryptography", Chapter 4.1.1, 2004, Springer-Verlag New York, Inc.

[2] Pohlig, Hellman: "An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significance", 1978, http://www-ee.stanford.edu/ hellman/publications/28.pdf

[3] Novotney: "Weak Curves In Elliptic Curve Cryptography", 2010, http://wstein.org/edu/2010/414/projects/ novotney.pdf

[4] Çetin Kaya Koç: "Elliptic Curve Cryptography Fundementals", lect02 groups in cryptography, 2015

[5] Çetin Kaya Koç: "Elliptic Curve Cryptography Fundementals", lect03 fields in cryptography, 2015

[6] Çetin Kaya Koç: "Elliptic Curve Cryptography Fundementals", lect09 ecc fundamentals, 2015

[7] https://en.wikipedia.org/wiki/Elliptic_curve_cryptography, December 7, 2015

[8] https://en.wikipedia.org/wiki/Pohlig-Hellman_algorithm, December 7, 2015

[9] https://en.wikipedia.org/wiki/Pollard%27s_rho_algorithm, December 7, 2015

[10] Hankerson, Menezes, Vanstone: "Guide to Elliptic Curve Cryptography", Chapter 4.1, 2004, Springer-Verlag New York, Inc.

[11] https://en.wikipedia.org/wiki/Chinese_remainder_theorem, December 7, 2015

[12] Matthew Musson: "Attacking the Elliptic Curve Discrete Logarithm Problem", 2006 http://security.cs.pub.ro/hexcellents/wiki/_media/10.1.1.132.6034.pdf