

CS559 Curve Based Cryptography - Prof. Ming-Deh Huang
Scribe: Iftikhar A Burhanuddin
burhanud@usc.edu
Class #5 - February 5, 2002

Administrivia: Scribe notes for class #4 are online at the course webpage: <http://www-rcf.usc.edu/~mdhuang/cs599>

Today's class:

1. Elliptic Curve Digital Signature Algorithm
2. Pollard Rho Algorithm
3. Silver-Pohlig-Hellman Algorithm

Elliptic Curve Digital Signature Algorithm - ECDSA

Alice signs $h(m)$ the hash of message m with her private key and sends the encrypted message and her signature to Bob. Bob extracts the signature and verifies whether the sender of the message was Alice using Alice's public key. Eve is an intruder who eavesdrops on the conversation.

Setup: Steps taken by Alice and Bob in accordance

1. Choose E and \mathbf{F}_p and use $G = E(\mathbf{F}_p)$, where p is a large prime. (The case where the number of elements in the field is a power of a prime will be discussed later.)
2. Pick a point $T \in E(\mathbf{F}_p)$ with the *nice* property that $\text{ord}(T)$ contains a large prime. For convenience we shall assume that $\#E(\mathbf{F}_p) = n$ is a prime. (The size of the group will determine the length of the signature.)

Protocol:

1. Alice generates a private key d such that $\text{gcd}(d, n) = 1$ and computes the key $Q = d.T$ and makes it public

Protocol - Alice - Signature Part:

1. Alice generates random secret number k

2. Computes $k.T = (x_1, y_1) \neq O$, $x_1, y_1 \in \mathbf{F}_p = \{0, 1, \dots, p-1\}$
3. Computes r where $r \equiv x_1 \pmod n$
4. Computes s where $s \equiv k^{-1}(h(m) + d.r) \pmod n$
5. $(r, s), 0 < r, s < n$ is Alice's signature which she sends to Bob along with the encrypted message

Protocol - Bob - Verification Part:

1. Bob recovers $r, s, h(m)$
2. Computes $s^{-1}.h(m).T \oplus s^{-1}.r.Q$
 $= s^{-1}.h(m).T \oplus s^{-1}.r.d.T$
 $= s^{-1}(h(m) \oplus r.d).T = k.T$, where \oplus denotes addition on the curve
3. Say $s^{-1}.h(m).T \oplus s^{-1}.r.Q = (\tilde{x}_1, \tilde{y}_1)$. Bob computes $\tilde{x}_1 \pmod n$ and checks if it matches r . If it does Bob is convinced that Alice must have sent the message

Even if Eve recovers $r, s, h(m)$ forgery is going to be tough without d , which is presumably secure due to the hardness of ECDLP.

How is ECDSA better than the classical DSA?

Not only is the signature short but computation is 160-bit arithmetic

	DSA	ECDSA
Signature size (bits)	320	320
Computation - Bit Arithmetic	1024	160

The DSA security claim is that solving DLP on the subgroup depends on solving DLP on the bigger group. On the other hand the ECDSA rests on the no-known-existence of a sub-exp algorithm for the ECDLP.

Remarks:

1. ECDSA can be extended to \mathbf{F}_{p^n} fields with a mapping from \mathbf{F}_{p^n} to the integers so that step 3 of the signature protocol ($r \equiv \langle x_1 \rangle \pmod n$) can be redefined. One possible mapping is $\mathbf{F}_{p^d} \rightarrow \mathbf{F}_p^d$ that is mapping a field element to a d -dimensional vector and viewing it as an integer with an appropriate basis.

For example $(a_0, a_1, \dots, a_{d-1}) \rightarrow a_0 + a_1p + a_2p^2 + \dots + a_{d-1}p^{d-1}$

2. Modulo p , inversion takes around $\log p$ divisions/multiplications and taking inverse seems to be a bottleneck in finite field arithmetic. How else can you more efficiently compute the inverse?
3. If $2^n \sim p$, is working with \mathbf{F}_{2^n} *better* than \mathbf{F}_p since you can look at elements as 0 – 1 vectors? Are there any security trade-offs?
4. Cryptography relies on a whole lot more than the $P \neq NP$ question
5. Does existence of one-way functions imply the existence of a one-way trapdoor function

Birthday Paradox: How many people must be there in a room so that atleast 2 were born on the same day of the year with probability $\geq \frac{1}{2}$?

Answer: 28 and not 366!

Proof: Let's define X_{ij} to be a random (indicator) variable and say that a year has n days and there are k persons in the room

$$X_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ have the same birthday} \\ 0, & \text{otherwise} \end{cases}$$

$$\begin{aligned} E(X_{ij}) &= 1 \cdot Pr[X_{ij} = 1] + 0 \cdot Pr[X_{ij} = 0] \\ &= \sum_{m=1}^n Pr[i \& j \text{ have the same } m\text{th birthday}] \\ &= \sum_{m=1}^n \frac{1}{n^2} = \frac{n}{n^2} = \frac{1}{n} \end{aligned}$$

Let's define another random variable $X = \sum_{i \neq j} X_{ij}$

$$\begin{aligned} E(X) &= E(\sum_{i \neq j} X_{ij}) = \sum_{i \neq j} E(X_{ij}) \text{ (linearity of expectation)} \\ &= \binom{k}{2} \frac{1}{n} \end{aligned}$$

We would like $E(X) = 1$ and for this it is sufficient that $k \geq \frac{\sqrt{8n+1}}{2}$. The minimum k when we plug 365 for n turns out to be 28.

So if you have a *random enough* function f on a set of cardinality n , then the size of the subset with atleast 2 in the subset having the same image under f with *good* probability is \sqrt{n}

I. Pollard Rho Method:

This method is a general purpose technique which can be used to compute the discrete log in a cyclic group G in exponential time $\sim O(\sqrt{|G|})$ group

operations (which is better than the $O(|G|)$ naive brute-force algorithm). The crux of the algorithm is the Birthday Paradox. As elliptic curves have a natural additive group law defined on them, lets regard G as an additive group.

Discrete Logarithm Problem: Given the generator of the group G , $|G| = n$ and $R \in G$, to compute $x \in \mathbf{Z}$ such that $R = xQ$

Consider the map f ,

$$\begin{aligned} f : G &\rightarrow G \\ P &\mapsto aP + bQ \end{aligned}$$

where a, b are random integers and $(a, n) = 1$. f turns out to be a bijection.

Also consider the following iterative way to generate new elements in G starting with the element R :

$$\begin{aligned} R_1 &= f(R) = aR + bQ = axQ + bQ = (ax + b)Q \\ &= (a_1x + b_1)Q, \text{ where } a_1 = a, b_1 = b \\ R_2 &= f(R_1) = aR_1 + bQ = a(ax + b)Q + bQ \\ &= (a^2x + ab + b)Q = (a_2x + b_2)Q, \text{ where } a_2 = a^2, b_2 = ab + b \\ \dots & \\ R_i &= f(R_{i-1}) = (a_i x + b_i)Q, \text{ where } a_i = a^i, b_i = \sum_{j=0}^{i-1} a^j b \\ \dots & \end{aligned}$$

Since f is a random function we can think of R_1, R_2, \dots as a random sequence of elements in G . By the *Birthday problem* paradox we'll encounter $R_i = R_j$, with $i \neq j$ in about $O(\sqrt{n})$ iterations with probability ≥ 0.5 .

$$\begin{aligned} R_i &= R_j, i \neq j \\ \Rightarrow (a_i x + b_i)Q &= (a_j x + b_j)Q \\ \Rightarrow a_i x + b_i &\equiv a_j x + b_j \pmod{n} \\ \Rightarrow x &= (b_j - b_i)(a_i - a_j)^{-1} \pmod{n} \end{aligned}$$

Therefore we start with $R = R_1$ and generate R_i 's until we compute an element we have already generated R_j and as we know a_i, b_i, a_j, b_j we can recover x . Hence instead of enumerating all possibilities - the naive method we essentially do a random walk (which looks the symbol rho: ρ , hence the

name for this algorithm). We can also parallelize this scheme to get better results. Pollard-rho is one kind of random walk and any other similar scheme will also work. The survey paper [1] talks about refinements to the above algorithm.

Fact: Say $|G| = n = \prod p_i^{e_i}$, p_i small and $\max_i p_i = b$, that is n is b -smooth. Then DL/G can be solved in time ($\#$ group ops) polynomial in b and $\log n$. In particular if $b \leq \text{poly}(\log n)$ then the running time is $\text{poly}(\log n)$ and this is the algorithm [2] we discuss next.

II. Silver-Pohlig-Hellman:

We need to compute x such that $R = xQ$. Once we find $x \bmod p_i^{e_i}$, $\forall i$, we can use the Chinese Remainder Theorem to obtain x .

Say $p = p_i$, $e = e_i$ and let $x \equiv x_0 + x_1p + x_2p^2 + \dots + x_{e-1}p^{e-1} \bmod p^e$. We want to determine $x \bmod p^e$ i.e., $x_0, x_1, \dots, x_{e-1} \in \mathbf{F}_p = \{0, 1, \dots, p-1\}$

1. Compute $\frac{n}{p}R$. Determine x_0 by plugging $x_0 = \mathbf{F}_p$ in $\frac{n}{p}x_0Q$ until the result matches $\frac{n}{p}R$

$$\frac{n}{p}x \equiv \frac{n}{p}x_0 \bmod n$$

$$\text{Hence } \frac{n}{p}R = \frac{n}{p}xQ = \frac{n}{p}x_0Q$$

There are p possible values for x_0 and each iteration takes $O(\log n)$ group operations. Hence this step takes time $O(p \log n)$ group ops

2. Compute $\frac{n}{p^2}R$. Determine x_1 by plugging $x_1 = \mathbf{F}_p$ in $\frac{n}{p}x_1(xQ - x_0Q)$ until the result matches $\frac{n}{p^2}R$

$$x - x_0 \equiv x_1p + x_2p^2 + \dots + x_{e-1}p^{e-1} \bmod p^e$$

$$\Rightarrow \frac{n}{p^2}(x - x_0) \equiv \frac{n}{p}x_1 \bmod n$$

$$\text{Hence } \frac{n}{p^2}(R - x_0Q) = \frac{n}{p^2}(xQ - x_0Q) = \frac{n}{p^2}(x - x_0)Q = \frac{n}{p}x_1Q$$

There are p possible values for x_1 and each iteration takes $O(\log n)$ group operations. Hence this step takes time $O(p \log n)$ group ops

3. Proceed inductively to compute the other x'_i s

Hence when the group order is smooth we can solve DL in $\text{poly}(\log n)$ group operations

III. Easy cases

By Hasse's theorem we know that $\#E(\mathbf{F}_p) = p + 1 - t = n$, where $|t| \leq 2\sqrt{p}$ and t is called the trace of Frobenius

1. $t = 1 \Rightarrow \#E(\mathbf{F}_p) = p$. In this case $DL/E(\mathbf{F}_p) \leq_p DL/\text{additive group } \mathbf{F}_p$. Given $x, y \in \mathbf{F}_p$, to compute d so that $y = d.x$ in \mathbf{F}_p . This is polynomial time solvable and we see that solving $DL/E(\mathbf{F}_p)$ is even simpler than solving DL/\mathbf{F}_p^*

2. $t = 0, 2 \Rightarrow \#E(\mathbf{F}_p) = p \pm 1$.

Generally $DL/E(\mathbf{F}_p) \leq_p DL/\mathbf{F}_{p^k}^*$ for some $k, n|p^k - 1$. In particular when $n = p + 1, p - 1, k = 1, 2$ respectively. The paper [1] talks about $t = 0, 2$ as two separate cases but we follow the later work by Frey and Rück. This gives us sub exp algorithms for $n = p \pm 1$.

IV. Xedni Calculus method

This technique is *similar* to the Index Calculus method for DLP over finite fields. We'll see that Xedni fails when applied to ECDLP and this seems to be the strongest evidence on the hardness of ECDLP.

Bibliography:

1. Neal Koblitz, Elliptic curve cryptosystems, Mathematics of Computation, Vol. 48, No. 177, 1987, 203-209.
2. S. C. Pohlig and M. E. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance, IEEE-Transactions on Information Theory 24, 1978, 106-110