

CS556 Introduction to Cryptography - Prof. Ming-Deh Huang

Scribe: Iftikhar A Burhanuddin

burhanud@usc.edu

Lecture #19, #20 - November 5,7 2002

Announcements:

1. Lecture notes #17-18 are online.
2. Please read Discrete Logarithm [K, IV.3], Key Exchange [S,22.1], Encryption [S,19.6] and Digital Signature [S, 20.1-20.4]

Topics for today:

1. Quadratic Reciprocity
2. Solvay-Strassen Test
3. Discrete Logarithm Problem
4. Diffie Hellman key exchange

1. Quadratic Reciprocity

Gauss was always interested in finding short-cuts, for instance eight-year-old Gauss astonished his teacher by instantly solving the problem of finding the sum of the first 100 integers. He also rediscovered the Law of Quadratic reciprocity which we'll talk about today.

We know that \mathbf{F}_p^* is a cyclic group and every non-trivial element of \mathbf{F}_p^* is a generator. Say p is an odd prime and $\mathbf{F}_p^* = \langle \mathbf{g} \rangle$. Consider the set of squares or quadratic residues in this group.

$$\{x \in \mathbf{F}_p^* \mid x = y^2, y \in \mathbf{F}_p^*\}$$

Clearly this set is a subgroup of \mathbf{F}_p^* . Infact it contains exactly half the elements of \mathbf{F}_p^* as we shall prove. The other elements are called the quadratic non-residues.

Suppose $a \in \mathbf{F}_p^*$ is a square, that is, $a = b^2$ for some $b \in \mathbf{F}_p^*$. As \mathbf{F}_p^* is cyclic $b = g^m$ for some $m \in \{0, 1, \dots, p-1\}$. So $a = g^{2m} = (g^m)^2$. Raising to the $\frac{p-1}{2}$ power, we get $a^{\frac{p-1}{2}} = (g^{2m})^{\frac{p-1}{2}} = (g^{p-1})^m = 1$.

Conversely suppose $a = g^k$ for some k . Raising to the $\frac{p-1}{2}$ power, we have $a^{\frac{p-1}{2}} = g^{k\frac{p-1}{2}} = 1 \Rightarrow p-1 \mid k\frac{p-1}{2} \Rightarrow 2 \mid k$. And so we've shown that a is a square $\Leftrightarrow a \in \langle g^2 \rangle \Leftrightarrow a^{\frac{p-1}{2}} = 1$. Therefore the number of squares is $|\langle g^2 \rangle| = \frac{p-1}{2}$. More generally when we consider finite fields with the number of elements being the power of an odd prime we can show that the quadratic residues and non-residues partition \mathbf{F}_p^* into two equal halves.

When $p > 2$, $\forall a \in \mathbf{F}_p^* (a^{\frac{p-1}{2}})^2 = a^{p-1} = 1 \Rightarrow a^{\frac{p-1}{2}} = \pm 1$ as $x^2 = 1 \Rightarrow x = \pm 1$ in \mathbf{F}_p .

So for $a \in \mathbf{F}_p^*$

$$a^{\frac{p-1}{2}} = \begin{cases} 1 \Leftrightarrow a \text{ is a square} \\ -1 \Leftrightarrow a \text{ is not a square} \end{cases}$$

And Gauss came up with this test to determine whether a is square or not. Note that $a^{\frac{p-1}{2}}$ can be computed efficiently using repeated squaring.

Legendre symbol. When $p > 2$ and is a prime and $a \in \mathbf{Z}$

$$\left(\frac{a}{p}\right) = \begin{cases} 1 \Leftrightarrow a^{\frac{p-1}{2}} \equiv 1 \pmod{p} \\ -1 \Leftrightarrow a^{\frac{p-1}{2}} \equiv -1 \pmod{p} \\ 0 \Leftrightarrow p \mid a \end{cases}$$

Hence we can succinctly write $\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p}$.

Some properties. Clearly $\left(\frac{1}{p}\right) = 1$ and $\left(\frac{a^2}{p}\right) = 1$. Also

$$\begin{aligned} \left(\frac{ab}{p}\right) &\equiv (ab)^{\frac{p-1}{2}} \pmod{p} \\ &\equiv (a)^{\frac{p-1}{2}} (b)^{\frac{p-1}{2}} \pmod{p} \\ &\equiv \left(\frac{a}{p}\right) \left(\frac{b}{p}\right) \pmod{p} \end{aligned}$$

So the Legendre symbol is multiplicative $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right) \left(\frac{b}{p}\right)$

Proposition II.2.4. p is an odd prime

$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = \begin{cases} 1 \text{ if } p \equiv \pm 1 \pmod{8} \\ -1 \text{ if } p \equiv \pm 3 \pmod{8} \end{cases}$$

Proposition II.2.5. When p, q are odd primes

$$\left(\frac{q}{p}\right) = (-1)^{\frac{(p-1)(q-1)}{4}} \left(\frac{p}{q}\right) = \begin{cases} \left(\frac{p}{q}\right) & \text{if } p \equiv q \equiv 3 \pmod{4} \\ -\left(\frac{p}{q}\right) & \text{otherwise} \end{cases}$$

This is called the law of quadratic reciprocity as it determines quadratic residuosity and the term reciprocity is to indicate that upto sign you can flip p and q . This law is a non-trivial observation as it says q is a square mod p iff p is a square mod q .

So according to this law $x^2 \equiv 3 \pmod{131}$ has solutions $\Leftrightarrow x^2 \equiv 131 \pmod{3}$ has no solutions $\Leftrightarrow x^2 \equiv 2 \pmod{3}$ has no solution and we know that the latter is true. Notice that we reduce the size of the problem each time we flip.

Now that we know whether an element in \mathbf{F}_p^* is a square or not, a natural question to ask is how to compute *square roots* and this turns out to be difficult to answer efficiently.

How efficiently can we compute the Legendre symbol? We encounter composite numbers and the need to factor (which is presumably a hard problem) arises. Can we compute the Legendre symbol without factoring? Yes and for this we introduce the Jacobi symbol which is a natural extension of the Legendre symbol and has the same symbolic notation as the former and allows us to work with denominators that are not prime.

Jacobi Symbol. Say $n = \prod_i p_i^{e_i}$ and $\left(\frac{a}{n}\right) := \prod_i \left(\frac{a}{p_i}\right)^{e_i}$

This definition by itself doesn't do any good but with the help of the following theorems (which we state without proof) becomes quite useful.

Fact 1. When n is an odd integer $\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}}$

Fact 2. When n, m are odd integers $\left(\frac{m}{n}\right) = (-1)^{\frac{(m-1)(n-1)}{4}} \left(\frac{n}{m}\right)$

Hence once we've established these facts there is no need to factor as first we can factor out the 2's and use fact 1. Suppose the odd factors are m, n and wlog say $n < m$ then using fact 2 calculating $\left(\frac{n}{m}\right)$ reduces to computing $\left(\frac{m}{n}\right)$ which is equal to $\left(\frac{m \bmod n}{n}\right)$. As the problem size reduces each time we flip the Legendre symbol can be computed using the reciprocity laws on the Jacobi symbol in atmost $\log m$ flips.

Example. $\left(\frac{11}{37}\right) = (-1)^{\frac{10 \cdot 36}{4}} \left(\frac{37}{11}\right) = \left(\frac{4}{11}\right) = \left(\frac{2^2}{11}\right) = 1$

$$\left(\frac{13}{37}\right) = (-1)^{\frac{12 \cdot 36}{4}} \left(\frac{37}{13}\right) = \left(\frac{11}{13}\right) = (-1)^{\frac{10 \cdot 12}{4}} \left(\frac{13}{11}\right) = \left(\frac{2}{11}\right) = -1$$

2. Solvay-Strassen Test

A primality test works as follows. Given an odd integer n it decides whether n is a prime or not. On the other hand the Solvay-Strassen Test is a test for compositeness.

We know that if n is prime \Rightarrow for all a such that $\gcd(a, n) = 1$, $a^{\frac{n-1}{2}} \equiv \pm 1 \pmod n$ that is $a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) \pmod n$. On the other hand, when n is not a prime there is no guarantee that the congruence will hold. The reason for this is that when n is prime $n-1$ is the order of the group $\mathbf{Z}/n\mathbf{Z}^*$ and when n is composite the group order is $\leq n-1$.

Solvay-Strassen Test. Input: odd n . Choose random $a < n$. Check if $a^{\frac{n-1}{2}} = \left(\frac{a}{n}\right)$. If not n is composite.

Proposition. Composite numbers pass the test with probability $\leq \frac{1}{2}$

Proof. Let $H = \{a \in \mathbf{Z}/n\mathbf{Z}^* \mid a^{\frac{n-1}{2}} = \left(\frac{a}{n}\right)\}$. Clearly H is a subgroup of $\mathbf{Z}/n\mathbf{Z}^*$. By Lagrange's theorem it is enough to argue that $H \neq \mathbf{Z}/n\mathbf{Z}^*$, that is enough to show that there exists a such that $a^{\frac{n-1}{2}} \not\equiv \left(\frac{a}{n}\right) \pmod n$

case 1. n is not square free, that is, $p^2 \mid n$ for some prime p . Let $a = 1 + \frac{n}{p}$

For all $q \mid n$, $\left(\frac{a}{q}\right) = \left(\frac{1}{q}\right) = 1 \Rightarrow a \equiv 1 \pmod q \Rightarrow \left(\frac{a}{n}\right) = 1$

$$\begin{aligned} a^j &= \left(1 + \frac{n}{p}\right)^j \\ &= 1 + j\frac{n}{p} + C_2^j \frac{n^2}{p^2} + \dots + \frac{n^j}{p^j} \\ &\equiv 1 + j\frac{n}{p} \pmod n \\ &\equiv 1 \pmod n \text{ only if } p \mid j \end{aligned}$$

Consider $j = \frac{n-1}{2}$. $\frac{n-1}{2} = 2^{-1}(-1) \pmod p \not\equiv 0 \pmod p$. $p \nmid \frac{n-1}{2}$ so $a^{\frac{n-1}{2}} \not\equiv 1 \pmod n$. So RHS = 1 and LHS $\neq 1$. Hence $a^{\frac{n-1}{2}} \not\equiv \left(\frac{a}{n}\right)$.

Once we know there is one which fails the test, we know that at least half fail the test due to Lagrange's theorem.

Case 2. n is square free

Take a prime $p \mid n$. Consider an integer a such that, $\left(\frac{a}{p}\right) = -1$ and $a \equiv 1 \pmod{\left(\frac{n}{p}\right)}$.

To construct such an a , pick a $b < p$, $\left(\frac{b}{p}\right) = -1$. CRT says there exists a such that $a \equiv b \pmod{p}$, $a \equiv 1 \pmod{\left(\frac{n}{p}\right)}$.

So now we have a with $\left(\frac{a}{p}\right) = -1$ and $a \equiv 1 \pmod{\left(\frac{n}{p}\right)}$. Now $\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right) \left(\frac{a}{n/p}\right) = -1$ and hence RHS = -1 .

On the other hand $a^{\frac{n-1}{2}} = 1 \pmod{\frac{n}{p}} \Rightarrow a^{\frac{n-1}{2}} \not\equiv -1 \pmod{n}$. And so LHS $\not\equiv -1 \pmod{n}$. And so the congruence doesn't hold in this case either. \diamond

The fact that implicitly we are working with a group comes in handy. We are doing the group membership test. A composite n fails with probability $\geq 1/2$. By repeating the test, we can reduce the probability of error to arbitrarily low values. Observe that this is not a genuine primality test but a test for composites.

3. Discrete Logarithm Problem

Today we'll look at discrete logarithm based cryptosystems in particular the ones which are based on discrete log over finite fields.

Modular exponentiation with modulus a prime p . Given p, x, g to compute $g^x \pmod{p}$

$$\begin{array}{ccc} \mathbf{F}_p & \rightarrow & \mathbf{F}_p^* \\ x & \mapsto & g^x \pmod{p} \end{array}$$

This can be efficiently done using the squaring trick in $\leq 2 \log p$ modular multiplications. In particular if g is a generator, we get $\langle g \rangle = \mathbf{F}_p^*$ and that the map is bijective. On the other hand, the inverse modular exponentiation problem is defined as the discrete logarithm problem over \mathbf{F}_p .

Discrete logarithm over \mathbf{F}_p . Given g and a prime p such that $\langle g \rangle = \mathbf{F}_p^*$ and $y \in \mathbf{F}_p^*$ to compute x such that $y = g^x \pmod{p}$

More generally we can define the discrete logarithm problem over a finite cyclic group as follows.

Discrete logarithm (DL) over $G = \langle g \rangle$. Given $y \in G$ to compute e such that $y = g^e$.

Exponentiation in this general setting can be done efficiently in $\leq 2 \log e$ group operations with the implicit assumption being that each group operation in turn can be performed efficiently.

Example. When $G = \mathbf{F}_p^*$ or $G = \mathbf{F}_q^*$ we get the discrete logarithm over finite fields which is presumed to be a hard problem as all known algorithms take subexponential time.

Exponentiation is a basic arithmetic operation which is easy to compute but the inverse is presumably hard for a lot of interesting groups. This one-way-ness allows people to construct asymmetric cryptosystems.

When $G = \mathbf{F}_p = \mathbf{Z}/p\mathbf{Z}$ the problem is trivial as finding e amounts to solving a linear congruence. $y \equiv ge \pmod n$

There is a natural “addition” / group law which can be defined on an elliptic curve over any field. When we take G to be the elliptic curve group $E(\mathbf{F}_p)$ we get the Elliptic Curve Discrete Logarithm problem (ECDLP). Similarly we can work with more generalized curves called abelian varieties and their associated Jacobians. The last 5-10 years has seen a lot of activity in this area mainly because the known algorithms are of exponential time complexity (in the elliptic curve case).

Claim. Say G is a cyclic group and $|G|$ is k -smooth. Then DL problem over G can be solved in $\text{poly}(k, \log |G|)$ group operations (that is in time polynomial in k and $\log |G|$ group ops).

So if we are working with $G = \mathbf{F}_p^*$ and $|G| = p - 1$ is $\log p$ smooth, that is all primes occurring in the factorization of $p - 1$ are $\leq \log p$ then the DL problem over \mathbf{F}_p^* can be solved in time $\text{poly}(\log p)$. So we presume that DL over \mathbf{F}_p^* is hard except in the smooth case and construct DL based cryptosystems.

4. Diffie Hellman key exchange

Historically the Diffie-Hellman key exchange was the first public key crypto application. Suppose Alice and Bob want to generate a secret key for a session of symmetric key operation and they don't trust each other and so they want to cogenenerate the secret key.

Protocol.

- Alice and Bob generate a random element of \mathbf{F}_p^* (the “key”) together
- p, g are public such that $\mathbf{F}_p^* = \langle g \rangle$
- Alice and Bob randomly generate their private keys $x, y \in \mathbf{F}_p^*$
- Alice sends g^x to Bob and Bob sends g^y to Alice

- $K = g^{xy}$ is the key which Alice and Bob can compute efficiently as follows: Alice computes by $(g^y)^x$ and Bob computes $(g^x)^y$

Due to our assumption that DL is hard a malicious third party intercepting g^x or g^y will not be able to compute x . This scheme can be extended to more than 2 players. The 3 player case is as follows.

Protocol 3-party.

- Alice, Bob and Charlie randomly generate their private keys $x, y, z \in \mathbf{F}_p^*$
- Alice sends g^x to Bob and Bob sends g^y to Charlie and Charlie sends g^z to Alice, that is g^x, g^y, g^z are made public.
- Now Alice sends $(g^z)^x$ to Bob and Bob sends $(g^x)^y$ to Charlie and Charlie sends $(g^y)^z$ to Alice.
- $K = g^{xyz}$ is the key which Alice, Bob and Charlie can compute quickly. Alice computes by $(g^{yz})^x$, Bob computes $(g^{zx})^y$ and Charlie computes $(g^{xy})^z$

An attractive feature of this scheme is that a message can be sent to the players with the common key $K = g^{xyz}$ and presumably nobody else can compute K even with the knowledge of the public keys (g^x, g^y, g^z) of the individual players.

In RSA it is critical that everyone has a different modulus and different primes are used to generate the different moduli. But in DH everyone shares the same finite field and this is an advantage over RSA.

But DH is susceptible to the *Man-in-the-Middle* attack. Consider the two player case of Alice and Bob. Suppose Malory intercepts g^x which Alice sends to Bob and g^y which Bob sends to Alice and replaces them with g^z where z is an element which she generates. So Bob thinks he is talking only to Alice when he sends messages using the key g^{yz} and Alice thinks he is talking only to Bob with the key g^{xz} .

So there is a need for a certification authority to establish a correspondence between Alice and g^x , Bob and g^y , etc. An alternative is message authentication.

Let $K = g^{xy}$ where x, y are Alice and Bob's secret keys respectively. Alice sends Bob $a = g^x$. Bob doesn't simply send $b = g^y$ to Alice but computes $K = g^{xy} = a^y$ and instead sends $b, E_K(S_{Bob}(a, b))$.

Alice then computes $K = b^x$ and decrypts $D_K(E_K(S_{Bob}(a, b)))$ to get $S_B(a, b)$ and then recovers a, b . Finally he checks if the two b 's the one which was sent by Bob and the one hidden in $E_K(S_{Bob}(a, b))$ match. If they do he is convinced it was by Bob.

E_K could be DES. S_{Bob} could be any signature scheme even RSA. With this refinement Mallory cannot follow Alice as she will not be able to generate a message where the two b 's match.