CS 7810 Foundations of Cryptography October 23, 2017

Lecture 13: Crypto in Cyclic groups, DL, CDH, DDH assumptions

Lecturer: Daniel Wichs *Scribe: Akshar Varma*

1 Topic Covered

- Primes
- Crypto in cyclic groups
- Discrete Logarithm and Diffie-Hellman assumptions
- Diffie-Hellman key exchange

2 Primes

Before we can start using the number theory and algebra we have seen for cryptography, we need some properties about prime numbers.

Fact 1 *The distribution of primes is "dense enough for effective sampling".*

We need this fact so that we can find prime numbers large enough for the groups we use in crypto.

Let $\pi(x)$ be the number of primes $\leq x$, then

$$\pi(x) \geq \frac{x}{3\log_2(x)} \approx \frac{x}{\log(x)} \tag{1}$$

Using 1, we get that

$$\mathsf{Pr}[x \text{ is prime} : x \in \{1, 2, \ldots, 2^n - 1\}] \geq \frac{1}{3n}$$

Fact 2 *Primality testing is possible in polynomial time. [Miller-Rabin'80, AKS'02]*

Given a number x, we can determine whether x is prime or not efficiently. Miller-Rabin is probabilistic in nature but is the more efficient algorithm, while AKS is deterministic but less efficient.

The above two facts allow for our cryptographic algorithms to efficiently sample random n bit prime numbers. This is done by simply sampling a random n bit number and then testing whether it is prime or not.

3 Crypto in Cyclic groups

Recall from earlier that all cyclic groups of order q are isomorphic to the group $\mathbb{Z}_q$.

$$\mathbb{G} = \{g^0, g^1, \ldots, g^{q-1}\} =< g >$$
$$g^i = g^{i \mod q} \Rightarrow \mathbb{G} \cong \mathbb{Z}_q$$

Further, given an element i of $\mathbb{Z}_q$ and a generator g of the group, it is easy to find g^i but given g^i, finding the corresponding i is not easy. For example, for prime p, $\mathbb{Z}_p^*$ is cyclic; setting $p = 7, g = 3$ we get the following group. Given $6 = g^i \in \mathbb{Z}_7^*$, it is difficult to determine the corresponding element $i \in \mathbb{Z}_6$.

$$\mathbb{Z}_p^* = \{1, \ldots, 6\} =< 3 >= \{g^0 = 1, g^1 = 3, , g^2 = 2, g^3 = 6, g^4 = 4, g^5 = 5\} \cong (\mathbb{Z}_6, +)$$

This is formalized by the hardness assumptions that we make when using cyclic groups for crypto. In all these scenarios, it is assumed that everyone (algorithms and the adveraries) have access to the cyclic group in which the crypto is being done, the generator being used and the order of the group. We let GroupGen be a function that on input 1^n (n being the security parameter), outputs a group along with a generator and the order of the group. The tuple $(\mathbb{G}, g, q)$ is always assumed to be public and available to everyone.

$$(\mathbb{G}, g, q) \leftarrow \mathsf{GroupGen}(1^n)$$

4 Discrete Logarithm and Diffie-Hellman assumptions

The following three problems are assumed to be hard for cyclic groups in general. The assumptions are ordered from weakest to strongest, that is the later assumptions imply the preceding ones (DDH$\Rightarrow$ CDH$\Rightarrow$ DL).

Discrete Logarithm

This assumption captures the intuitive idea that going from an element of a cyclic group $\mathbb{G}$ to the corresponding element of $\mathbb{Z}_q$ is difficult. For all PPT adversaries $\mathcal{A}$

$$\mathsf{Pr}[A(\mathbb{G}, g, q, g^x) = x : (\mathbb{G}, g, q) \leftarrow \mathsf{GroupGen}(1^n), x \leftarrow \mathbb{Z}_q] = \mathsf{negl}(n) \tag{2}$$

As mentioned earlier, this is often written without the group generator being made explicit as the group related information is assumed to be public.

$$\mathsf{Pr}[A(g^x) = x : x \leftarrow \mathbb{Z}_q] = \mathsf{negl}(n)$$

Note that this gives us a concrete construction for a one-way function. Assuming that q is super-polynomial[1], we can make a OWF f as follows, where r is the randomness that the GroupGen algorithm uses to come up with a group:

$$f(x,r) = (\mathbb{G}, g, q, g^x) \quad ; \quad (\mathbb{G}, g, q) \leftarrow \mathsf{GroupGen}(1^n, r)$$

Computational Diffie-Hellman

For all PPT adversaries $\mathcal{A}$

$$\Pr[A(g^x, g^y) = g^{xy} : x, y \leftarrow \mathbb{Z}_q] = \mathsf{negl}(n) \tag{3}$$

This assumption essentially states that multiplying the exponents is difficult if we have only been given the exponentiated terms. CDH implies DL as otherwise one can compute x, y given g^x, g^y and then compute g^{xy}.

Decisional Diffie-Hellman

Where CDH says that computing g^{xy} is difficult, DDH is a stronger assumption because it says that no adversary can differentiate it from a uniformly random element of $\mathbb{G}$.

$$\begin{aligned}(g^x, g^y, g^{xy}) &\approx (g^x, g^y, g^z) \quad ; \quad x, y, z \leftarrow \mathbb{Z}_q \\ &\approx (g^x, g^y, h) \quad ; \quad x, y \leftarrow \mathbb{Z}_q, h \leftarrow \mathbb{G}\end{aligned}$$

Limits of these assumptions

These assumptions are known to hold in the Generic Group Model. In this model, the adversary only knows how to perform the group operations but doesn't know about any additional structure the group may or may not have. However, if additional information is known, then these assumptions are not necessarily valid.

DDH doesn't hold in $\mathbb{Z}_p^*$

We show that the DDH assumption doesn't hold in $\mathbb{Z}_p^*$ by using a subset of $\mathbb{Z}_p^*$, called the quadratic residue which is defined as follows.

$$\begin{aligned}QR_p = \{f : \exists h \in \mathbb{Z}_p^* \quad s.t. \quad f = h^2\} &= \{g^i : i \text{ is even}\} \\ f = h^2 = g^{2j \mod p-1} &= g^i, i \text{ is even}\end{aligned}$$

Claim: $f \in QR_p \iff f^{(p-1)/2} = 1$. This is easy to verify by looking at $f = g^i \Rightarrow f^{(p-1)/2} = g^{i(p-1)/2}$. If i is even, then this $f^{(p-1)/2} = f^{(p-1)} = 1$ and it is not 1 if i is odd.

[1]If q is not super-polynomial then the OWF can be broken by a brute force attack.

This can be used to determine whether $g^{xy} \in QR_p$ which happens with probability 3/4 (only one of x, y need to be from QR_p). On the other hand $g^z \in QR_p$ with probability 1/2. Another distinguisher would be to look at $parity(x)parity(y) = parity(xy)$, this happens with probability 1 if g^{xy} is the input and happens with probability 1/2 when its g^z.

Removing the QR problem for DDH

We can overcome the problem created due to the quadratic residue by using Sophie-Germain primes, which are primes of the form $p = 2q+1$, where p, q are both primes. This will make $|\mathbb{Z}_p^*| = 2q$ and $|QR_p| = q$ at which point we can use QR_p as our group.

One problem in this approach is that it is not known whether there are an infinite number of Sophie-Germain primes. This means that the $(\mathbb{G}, g, q) \leftarrow \mathsf{GroupGen}(1^n)$ algorithm may not be able to come up with groups of order q for arbitrarily large q.

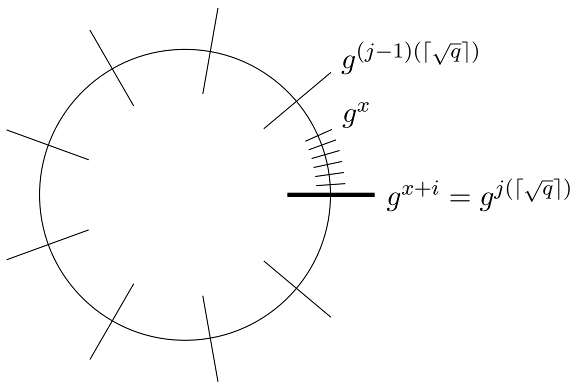
Solving DL in $\sqrt{q}$ time (Baby step-Giant step algorithm)

This algorithm improves on the naive manner of finding the discrete logarithm which takes q time. The algorithm takes giant steps starting at g and then it takes baby steps starting at g^x. It then finds where these two sequences collide and uses that to find x by using $x = j\lceil\sqrt{q}\rceil - i$.

Giant steps: $g, g^{\lceil\sqrt{q}\rceil}, g^{\lceil 2\sqrt{q}\rceil}, \ldots, g^{\lceil(\sqrt{q}-1)\sqrt{q}\rceil}$

Baby steps: $g^x, g^{x+1}, \ldots, g^{x+i} = g^{j(\lceil\sqrt{q}\rceil)}$

Since $\mathbb{G}$ is a cyclic group, its elements can be written on a circle and the giant steps can be thought of as big marks on the circle and the baby steps would be smaller marks on the circle between some two large marks. In the figure below, the bold line is the one where the collision occurs.


$g^{(j-1)(\lceil\sqrt{q}\rceil)}$
g^x
$g^{x+i} = g^{j(\lceil\sqrt{q}\rceil)}$

5 Diffie-Hellman key exchange

We now look at a construction that uses the DDH assumption to tackle the problem of exchanging a common secret key in symmetric key cryptography. Once again $(\mathbb{G}, g, q) \leftarrow \mathsf{GroupGen}(1^n)$ is assumed to be publicly available to everyone. In this protocol Alice and Bob communicate over a public channel on which Eve can see everything.

- Alice sends g^x over to Bob after selecting a random $x \leftarrow \mathbb{Z}_q$.
- Bob sends g^y over to Alice after selecting a random $y \leftarrow \mathbb{Z}_q$.
- Both Alice and Bob get the key by computing $k := g^{xy}$ (Alice does $(g^y)^x$ and Bob does $(g^x)^y$).

The aim is that the key Alice and Bob share seems random (uniformly over the key space) to Eve based on her view of the channel. That is, $(View_E, k) \approx (View_E, U_{\mathcal{K}})$ which is the same as $(g^x, g^y, g^{xy}) \approx (g^x, g^y, g^z)$ which is exactly the DDH assumption.

Diffie-Hellman key exchange without DDH assumption

If we do not want to assume that DDH holds, but can assume that CDH holds, then we can still perform a modified version of the Diffie-Hellman key exchange. We simply set $k := RO(g^{xy})$, which in practice would use a CRHF.