1 Plan for lectures 27

- Last lecture, we discussed the Diffie-Hellman protocol at a high level. Recall, Alice picks a prime p, a number $g \in \mathbb{Z}_p^*$, and a number $a \in \mathbb{Z}_p$. She sends $(p, g, v = g^a \mod p)$. Next, Bob picks $b \in \mathbb{Z}_p$ and sends $w = g^b \mod p$. Alice computes $w^a \mod p$, Bob computes $v^b \mod p$, and these two are equal to $g^{a \cdot b} \mod p$. The adversary sees $(p, g, g^a \mod p, g^b \mod p)$ and tries to learn something about $g^{a \cdot b} \mod p$. What are good/bad choices for g, p, a, b?
- We saw a few useful theorems/facts from number theory.
  - For any $a \in \mathbb{Z}_p^*$, $a^{p-1} \mod p = 1$.
  - For any $a \in \mathbb{Z}_p^*$, let $\langle a \rangle_p = \{1, a, a^2, \ldots\} \subseteq \mathbb{Z}_p^*$. The size of this set is a divisor of $p - 1$.
  - There exists an $a \in \mathbb{Z}_p^*$ such that $|\langle a \rangle_p| = p - 1$. Such an element is called a generator of $\mathbb{Z}_p^*$.

2 Density of generators in $\mathbb{Z}_p^*$

Last time, we saw that if p is a prime, then there exists at least one element $a \in \mathbb{Z}_p^*$ such that $\langle a \rangle_p = \mathbb{Z}_p^*$. Today, we will first show that there are many generators of $\mathbb{Z}_p^*$.

First, let us see a simple property of generators of $\mathbb{Z}_p^*$. If g is a generator of $\mathbb{Z}_p^*$, then we know that $g^{p-1} \mod p = 1$, and there exists no number smaller d than $p - 1$ such that $g^d \mod p = 1$. As a simple corollary, we get that for any integer z, if $g^z \mod p = 1$, then $(p - 1)$ divides z.

Claim 27.01. Let p be a prime and g a generator of $\mathbb{Z}_p^*$. Then, if $g^z \mod p = 1$ for some number $z > 0$, then $(p - 1)|z$.

Easy hain

Proof. Suppose $(p-1) \nmid z$, then $z = q \cdot (p-1) + r$ for some integer q and remainder r such that $0 < r < (p-1)$. Since $1 = g^z \mod p = (g^{q(p-1)} \mod p) \cdot (g^r \mod p) \mod p = g^r \mod p$, we get a contradiction. □

Next, we will use the above claim for proving the following theorem.

Theorem 27.01. Take any generator g of $\mathbb{Z}_p^*$. Then, for any number $\gamma < p - 1$, g^γ is a generator of $\mathbb{Z}_p^*$ if and only if $\gcd(\gamma, p - 1) = 1$.

Yeh bhi aasan hain

Before we prove this theorem, let us see an example. Consider $\mathbb{Z}_{13}^*$. We saw that 2 is a generator of $\mathbb{Z}_{13}^*$. Note that $5, 7, 11$ are the other numbers less than 12 that are co-prime to 12. Check that $2^5 \mod 13 = 6$, $2^7 \mod 13 = 11$ and $2^11 \mod 13 = 7$ are all generators of $\mathbb{Z}_{13}^*$.

Proof. **'If' part:** Take any $\gamma < (p-1)$ such that $\gcd(p-1, \gamma) = 1$, and let $d = |\langle g^\gamma \rangle_p|$. We know that $d|(p-1)$, and therefore $g^{\gamma \cdot d} \mod p = 1$. Using Claim 27.01, it follows that $(p - 1) \mid (\gamma \cdot d)$. Since $\gcd(p - 1, \gamma) = 1$, this implies that $(p - 1) \mid d$. This implies that $d = (p - 1)$, and hence g^γ is a generator of $\mathbb{Z}_p^*$.

'Only if' part: Let $\gamma < p - 1$ be such that $\gcd(\gamma, (p - 1)) = d > 1$. Then $|\langle g^\gamma \rangle_p| \leq (p - 1)/d$ (and therefore g^γ is **not** a generator of $\mathbb{Z}_p^*$). Suppose $\gamma = k \cdot d$ for some integer k. Note that $(g^\gamma)^{(p-1)/d} \mod p = (g^k)^{(p-1)} \mod p = (g^{p-1})^k \mod p = 1$. Hence $|\langle g^\gamma \rangle_p| \leq (p - 1)/d$ and g^γ is not a generator of $\mathbb{Z}_p^*$.

□

The above theorem gives us a clean characterization of all generators of $\mathbb{Z}_p^*$. The number of generators of $\mathbb{Z}_p^*$ is precisely equal to the number of integers less than $p - 1$ that are co-prime to $p - 1$. If $p = 13$, then we saw that there are four generators of $\mathbb{Z}_{13}^*$.

yeh badhiya chiz hain

2.1 Number of integers less than $p-1$ that are co-prime to $p-1$

Let us look at a special case where $p = 2q+1$, and q is prime. Such primes are widely used in practice, and are known as 'safe primes'. In this case, $p-1 = 2q$, and the set of integers less than $p-1$ is $S = \{1, 2, \ldots, 2q-1\}$. If a number is even, then clearly it is not co-prime to $p-1$. That leaves us with the set of odd numbers in S. Out of these, q is not co-prime to $p-1$. Removing q from this set, we are left with $S' = \{1, 3, \ldots, q-2, q+2, \ldots, 2q-1\}$. Check that all these numbers are co-prime to $p-1$, and there are $q-1$ such numbers.

Suppose Alice and Bob choose a safe prime p for the key exchange. There are still a few questions unaddressed:

- how does Alice pick g?
- how do Alice and Bob pick a and b respectively?

Let us go with the obvious choice for g: pick a random generator of $\mathbb{Z}_p^*$. How to sample a random generator of $\mathbb{Z}_p^*$?

2.1.1 Sampling a random generator of $\mathbb{Z}_p^*$ for safe prime p

U can sample and check whether a number is a generator or not

Out of the $2q$ elements in $\mathbb{Z}_p^*$, $q-1$ of them are generators for $\mathbb{Z}_p^*$. We sample a uniformly random element in $\mathbb{Z}_p^*$, and check if it is a generator of $\mathbb{Z}_p^*$. Suppose we sample g, then g is a generator if $|\langle g \rangle_p| = \mathbb{Z}_p^*$. The obvious way of checking this (by generating $\langle g \rangle_p$ explicitly) is inefficient (requires $O(p)$ operations), but we can use Theorem 26.02 from last lecture. Note that $|\langle g \rangle_p|$ divides $p-1$, and hence there are only 4 options for $|\langle g \rangle_p|$: the size is either $1, 2, q$ or $2q$. As a result, it suffices to just check that $g^1 \neq 1$, $g^2 \neq 1$, $g^q \neq 1$ (all operations modulo p). Hence, we can efficiently check if g is a generator of $\mathbb{Z}_p^*$, and therefore we can sample a random generator of $\mathbb{Z}_p^*$.

3 The discrete log problem

In the last section, we discussed how to sample a generator for $\mathbb{Z}_p^*$ when p is a safe prime. This brings us to the first number-theoretic computational problem that is believed to be hard: the discrete log problem over $\mathbb{Z}_p^*$. Let $\mathsf{Gen\text{-}Safe\text{-}Prime}(1^n)$ be a p.p.t. algorithm that takes as input 1^n, and generates an n-bit prime p such that $p = 2q+1$ and q is also prime.[1]

Assumption 27.01 (Discrete Log assumption over $\mathbb{Z}_p^*$). For any p.p.t. adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that for all n,

$$\Pr\left[\mathcal{A}(p, g, g^a) = a \;:\; \begin{array}{c} p \leftarrow \mathsf{Gen\text{-}Safe\text{-}Prime}(1^n) \\ g \leftarrow \mathbb{Z}_p^* \text{ s.t. } |\langle g \rangle_p| = p-1 \\ a \leftarrow \mathbb{Z}_{p-1} \end{array}\right] \leq \mu(n)$$

We don't need 'safe primes' to define the discrete log problem; the computational problem can be defined for any prime p, as long as we have a generator for $\mathbb{Z}_p^*$. We used safe primes here since we discussed how to sample the generator efficiently. In general, if we know the factorization of $p-1$, then we can find a generator for $\mathbb{Z}_p^*$ efficiently, and there are methods for sampling a random prime together with the factorization of $p-1$.

The discrete log problem is a well studied problem, and despite significant efforts, there are no polynomial time algorithms that can compute the discrete log. At a very high level, the hardness arises due to the mod p operation, which disturbs the monotonicity of the log function.

[1] Strictly speaking, it is not known if there are infinitely many 'safe primes', and therefore we don't know such an algorithm for large n. We know safe primes for sufficiently large n, which suffices for practical purposes. Also, there are other 'groups' that can be used instead.

4 Back to Diffie-Hellman Key Exchange

In the last section, we saw the Discrete Log assumption over $\mathbb{Z}_p^*$. Given this assumption, we can now complete our description of the Diffie-Hellman protocol: Alice chooses a safe prime p and a generator for $\mathbb{Z}_p^*$. Next, she chooses $a \leftarrow \mathbb{Z}_p$ and sends (p, g, g^a). Bob chooses $b \leftarrow \mathbb{Z}_p$ and sends g^b. Both can now compute $K = g^{a \cdot b}$.[2]

The adversary sees p, g, g^a, g^b, and wants to learn some information about g^{ab}. Assuming that the Discrete Log problem is hard, the adversary cannot obtain a or b from the protocol transcript. But does $g^{a \cdot b}$ look like a truly random element in $\mathbb{Z}_p^*$, given (g, g^a, g^b)? This brings us to the second computational problem of this lecture: the Decisional Diffie-Hellman (DDH) problem.

Assumption 27.02 (Decisional Diffie-Hellman assumption over $\mathbb{Z}_p^*$). For any p.p.t. adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that for all n, the following is bounded by $\mu(n)$:

$$\left| \Pr\left[\mathcal{A}\left(\begin{matrix} p, g \\ g^a, g^b, g^{a \cdot b} \end{matrix} \right) = 1 : \begin{matrix} p \leftarrow \textsf{Gen-Safe-Prime}(1^n) \\ g \leftarrow \mathbb{Z}_p^*, |\langle g \rangle_p| = p - 1 \\ a, b \leftarrow \mathbb{Z}_{p-1} \end{matrix} \right] - \Pr\left[\mathcal{A}\left(\begin{matrix} p, g \\ g^a, g^b, g^c \end{matrix} \right) = 1 : \begin{matrix} p \leftarrow \textsf{Gen-Safe-Prime}(1^n) \\ g \leftarrow \mathbb{Z}_p^*, |\langle g \rangle_p| = p - 1 \\ a, b, c \leftarrow \mathbb{Z}_{p-1} \end{matrix} \right] \right|$$

4.1 The Decisional Diffie-Hellman assumption is false over $\mathbb{Z}_p^*$

Odd even waala chacker hain. Odd Power and even power mein distinguish kar sakte hain. Usko (p-1)/2 power mein raise kar do.

Unfortunately, Assumption 27.04 (Decisional Diffie-Hellman assumption over prime-order subgroup of $\mathbb{Z}_p^*$) does not hold, and there is a very simple attack. The attack follows from the following observation: given $T = g^z$ for any z, there exists an efficient algorithm to determine if z is even or odd. Consider the number $T^{(p-1)/2}$. If $z = 2k$ for some k, then $T^{(p-1)/2} = g^{k \cdot (p-1)} = 1$. If $z = 2k+1$, then $T^{(p-1)/2} = g^{(2k+1) \cdot ((p-1)/2)} = g^{k \cdot (p-1) + (p-1)/2} = g^{(p-1)/2}$. Since g is a generator of $\mathbb{Z}_p^*$, $g^{(p-1)/2} \neq 1$. Therefore, an attacker can tell if the exponent is odd or even.

Now, if we check the DDH problem, $a \cdot b$ is even with probability $3/4$, while c is even with probability $1/2$. Therefore, the DDH adversary works as follows: given (g, g^a, g^b, T) where T is either $g^{a \cdot b}$ or g^c, the adversary checks if $T^{(p-1)/2} = 1$. If so, it guesses that $T = g^{a \cdot b}$, else it guesses that $T = g^c$.

4.2 How to fix the Diffie-Hellman protocol

There are a few natural alterations to prevent the above attack. First, we can force one of the parties to pick an odd exponent. That is, suppose Bob must pick an odd $b \in \mathbb{Z}_p$. Then $a \cdot b$ is even with probability $1/2$, same as the probability of c being even. Does the discrete log assumption hold in this case? In Question 27.02, you will show that the discrete log assumption holds here.

However, this attempt is also not secure. Suppose the adversary receives $(g, A = g^a, B = g^b, T)$ where a is odd, and T is either $g^{a \cdot b}$ or g^c. The adversary checks if the exponents of B and T have the same parity (that is, it checks if $B^{(p-1)/2} = T^{(p-1)/2}$). If so, it concludes that this is a DDH tuple. Else, it concludes that it is a uniformly random tuple.

One can force both a and b to be odd, and restrict the key space to be $\{g^c : c \text{ is a random odd number in } \mathbb{Z}_p\}$. The Diffie-Hellman key exchange protocol is probably secure for this setting. However, a far more useful alteration is to force the exponent to be even. That is, Alice picks a random generator $\widetilde{g}$ of $\mathbb{Z}_p^*$, but instead of using $\widetilde{g}$, both Alice and Bob use $g = \widetilde{g}^2$. First, note that $|\langle g \rangle_p| = q$. This subset is very interesting for us, primarily because (a) its size is a prime number itself, (b) the set is closed under multiplication modulo p, (c) the set contains 1, and for every element z in this set, there exists z' in the set s.t. $z \times_p z' = 1$. Subsets of this type are called 'prime order subgroups of $\mathbb{Z}_p^*$', and are widely used in cryptography.

We can define the discrete log problem, and the decisional Diffie-Hellman problem for this set.

[2] Everything is modulo p, I will be skipping $\mod p$ when it is clear from the context.

This is also not secure because if a is always odd. Then, parity of c will only depend on b. and we can distinguish between the two distribution very easily. Toh secure banane ke liye, ab hum g^2 ka use karenge.

Assumption 27.03 (Discrete Log assumption over prime-order subgroup of $\mathbb{Z}_p^*$). For any p.p.t. adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that for all n,

$$\Pr\left[\mathcal{A}(p,g,g^a)=a \ : \ \begin{array}{c} p \leftarrow \textsf{Gen-Safe-Prime}(1^n), q=(p-1)/2 \\ g \leftarrow \mathbb{Z}_p^* \text{ s.t. } |\langle g\rangle_p| = q \\ a \leftarrow \mathbb{Z}_q \end{array}\right] \leq \mu(n)$$

Assumption 27.04 (Decisional Diffie-Hellman assumption over prime-order subgroup of $\mathbb{Z}_p^*$). For any p.p.t. adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that for all n, the following is bounded by $\mu(n)$:

$$\left|\Pr\left[\mathcal{A}\left(\begin{array}{c} p,g \\ g^a, g^b, g^{a\cdot b}\end{array}\right)=1 : \begin{array}{c} p \leftarrow \textsf{Gen-Safe-Prime}(1^n) \\ q=(p-1)/2 \\ g \leftarrow \mathbb{Z}_p^*, |\langle g\rangle_p| = q \\ a,b \leftarrow \mathbb{Z}_q \end{array}\right] - \Pr\left[\mathcal{A}\left(\begin{array}{c} p,g \\ g^a, g^b, g^c\end{array}\right)=1 : \begin{array}{c} p \leftarrow \textsf{Gen-Safe-Prime}(1^n) \\ q=(p-1)/2 \\ g \leftarrow \mathbb{Z}_p^*, |\langle g\rangle_p| = q \\ a,b,c \leftarrow \mathbb{Z}_q \end{array}\right]\right|$$

5 Lecture summary, plan for next lecture, additional resources

Summary: We first showed that $\mathbb{Z}_p^*$ has many generators. By choosing a safe prime p (that is, $p=2q+1$ for some prime q), we get that nearly half the elements of $\mathbb{Z}_p^*$ are generators of $\mathbb{Z}_p^*$. Moreover, given an element in $\mathbb{Z}_p^*$, we can efficiently check if it is a generator of $\mathbb{Z}_p^*$. This allows us to completely specify the Diffie-Hellman key exchange protocol. Unfortunately, the protocol is not secure if we use $\mathbb{Z}_p^*$. Instead, we should be using a prime order subgroup of $\mathbb{Z}_p^*$. This can be generated by first choosing a generator $\widetilde{g}$ for $\mathbb{Z}_p^*$, then generating the subgroup using $\widetilde{g}^2$.

Yahan bas ye likha hua hain ki how to check if generator hain ya nhi. And we should not use Z_p* instead we should use Z_q*. Now, g = g'^2.

6 Questions

Question 27.01. In Theorem 27.01, show that if $\gcd(\gamma,(p-1))=d$, then $|\langle g^\gamma\rangle_p| = (p-1)/d$.

g^(2a+1) mod p.

Question 27.02. Suppose Assumption 27.01 (Discrete Log assumption over $\mathbb{Z}_p^*$) holds. Show that for any p.p.t. adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that for all n,

$$\Pr\left[\mathcal{A}(p,g,g^a)=a \ : \ \begin{array}{c} p \leftarrow \textsf{Gen-Safe-Prime}(1^n) \\ g \leftarrow \mathbb{Z}_p^* \text{ s.t. } |\langle g\rangle_p| = p-1 \\ a \leftarrow \mathbb{Z}_{p-1} \cap \{1,3,5,\ldots,p-2\} \end{array}\right] \leq \mu(n)$$

Question 27.03. Let p be a safe prime, and g a generator of $\mathbb{Z}_p^*$. Consider the following two distributions.

$$\mathcal{D}_0 = \{(g^a, g^b, g^{a\cdot b}) : a,b \leftarrow \mathbb{Z}_p\}$$
$$\mathcal{D}_1 = \{(g^a, g^b, g^c) : a,b,c \leftarrow \mathbb{Z}_p\}$$

Compute the statistical distance of the two distributions (as a function of p).

sirf (p-1)^2 possibility hi honge D_0 mein, Then (p-1)^3 - (p-1)^2 tuples nhi honge usme. Then calculate easily the statistical distance. For reference see this image attached.