More Cryptographic Hardness Assumptions
Cyclic Groups and Generators

Foundations of Cryptography
Computer Science Department
Wellesley College

Fall 2016

Table of contents

Cyclic Groups

Definition. Let $\mathbb{G}$ be a finite group of order m. For $g \in \mathbb{G}$, define the set

$$\langle g \rangle \stackrel{\text{def}}{=} \{g^0, g^1, \ldots\}$$

Remark. We know that $g^m = 1$. Let $i \leq m$ be the smallest positive integer such that $g^i = 1$. The above sequence repeats after i terms

$$\langle g \rangle \stackrel{\text{def}}{=} \{g^0, g^1, \ldots, g^{i-1}\}$$

and $\langle g \rangle$ contains at most i elements. In fact, it contains exactly i elements* and forms a subgroup of $\mathbb{G}$ called the *subgroup generated by g*.

*Why?

The order of a group element

Definition 8.51. Let $\mathbb{G}$ be a finite group and $g \in \mathbb{G}$. The *order of g* is the smallest positive integer i with $g^i = 1$. Equivalently, the order of g equals $|\langle g \rangle|$.

Proposition 8.53. Let $\mathbb{G}$ be a finite group and $g \in \mathbb{G}$ an element of order i. Then $g^x = g^y$ if and only if $x = y \mod i$.

Proof.

Cyclic groups and generators

Definition. If there exists $g \in \mathbb{G}$ whose order equals that of the group, then $\langle g \rangle = \mathbb{G}$. In this case, we call $\mathbb{G}$ a *cyclic group* and say that g is a *generator* of $\mathbb{G}$.

Different elements of $\mathbb{G}$ may have different orders. However, ...

Proposition 8.54. Let $\mathbb{G}$ be a finite group of order m, and say $g \in \mathbb{G}$ has order i. Then $i \mid m$.

Proof.

Corollary 8.55. Let $\mathbb{G}$ be a finite group of prime order p, then $\mathbb{G}$ is cyclic. Furthermore, all elements of $\mathbb{G}$ except the identity are generators of $\mathbb{G}$.

Proof.

Examples of cyclic groups

Example. The additive group $\mathbb{Z}_{12}$ is cyclic. What elements are generators?

Example. Is multiplicative group $\mathbb{Z}_{12}^*$ cyclic?

Example. How about the multiplicative group $\mathbb{Z}_7^*$?

In fact,

Theorem 8.56. If p is prime, then $\mathbb{Z}_p^*$ is a cyclic group of order $p - 1$.

*Cyclic groups of the same order are all "the same"**

Example 8.61. Let $\mathbb{G}$ be a cyclic group of order n, and let $g \in \mathbb{G}$ be a generator of $\mathbb{G}$. Then the mapping $f : \mathbb{Z}_n \to \mathbb{G}$ given by $f(a) = g^a$ is an isomorphism between $\mathbb{Z}$ and $\mathbb{G}$.

Indeed, for $a, a' \in \mathbb{Z}_n$, we have

$$f(a + a') = g^{[a+a' \mod n]} = g^{a+a'} = g^a \cdot g^{a'} = f(a) \cdot f(a').$$

*This is not true in the *computational sense*. In particular, $f^{-1} : \mathbb{G} \to \mathbb{Z}_n$ need not be efficiently computable.

The discrete logarithm

- If $\mathbb{G}$ is a cyclic group of order q, then there exists a generator $g \in \mathbb{G}$ such that $\{g^0, g^1, \ldots, g^{q-1}\} = \mathbb{G}$..
- Put another way, for every $h \in \mathbb{G}$, there is a *unique* $x \in \mathbb{Z}_q$ such that $g^x = h$
- We call this x the *discrete logarithm of h with respect to g* and write $x = \log_g h$.

The discrete logarithm problem

The discrete logarithm experiment $\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n)$:

1. Run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$, where $\mathbb{G}$ is a cyclic group of order q (with $\|q\| = n$), and g is a generator of $\mathbb{G}$.
2. Choose $h \leftarrow \mathbb{G}$. (This can be done by choosing $x' \leftarrow \mathbb{Z}_q$ and setting $h := g^{x'}$).
3. $\mathcal{A}$ is given $\mathbb{G}, q, g, h$, and outputs $x \in \mathbb{Z}_q$.
4. The output of the experiment is defined to be 1, if $g^x = h$, and 0 otherwise.

Definition 8.62 We say that *the discrete logarithm problem is hard relative to* $\mathcal{G}$ if for all probabilistic polynomial-time algorithms $\mathcal{A}$ there exists a negligible function negl such that

$$\Pr[\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n) = 1] \leq \mathsf{negl}(n).$$

Computational Diffie-Hellman (CDH) problem

Definition. Fix a cyclic group $\mathbb{G}$ and a generator $g \in \mathbb{G}$. Given two group elements h_1 and h_2, define

$$\mathsf{DH}_g(h_1, h_2) \stackrel{\text{def}}{=} g^{\log_g(h_1)\cdot\log_g(h_2)}$$

That is, given $h_1 = g^x$ and $h_2 = g^y$, then

$$\mathsf{DH}_g(h_1, h_2) = g^{x\cdot y} = h_1^y = h_2^x.$$

The *Diffie-Hellman (CDH) problem* is to compute $\mathsf{DH}_g(h_1, h_2)$ given randomly-chosen h_1 and h_2.

Remark. If the discrete logarithm problem is easy relative to some $\mathcal{G}$ given randomly chosen h_1 and h_2, then so is CDH. The converse is not so clear.

*Decisional Diffie-Hellman (DDH) problem**

The *decisional Diffie-Hellman (DDH) problem* is to distinguish $\mathrm{DH}_g(h_1, h_2)$ from a random group element for randomly chosen h_1, h_2.

Definition 8.63. We say that the *DDH problem is hard relative to* $\mathcal{G}$ if for all probabilistic polynomial-time algorithms $\mathcal{A}$ there exists a negligible function negl such that

$$|\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1]| \leq \mathrm{negl}(n),$$

where in each case the probabilities are taken over the experiment in which $\mathcal{G}(1^n)$ outputs $(\mathbb{G}, q, g)$, and the random $x, y, z \in \mathbb{Z}_q$ are chosen.

*If the CDH problem is easy relative to some $\mathcal{G}$, then so is the DDH problem. The converse does not appear to be true.

For discrete log and Diffie-Hellman, groups of prime order are preferred

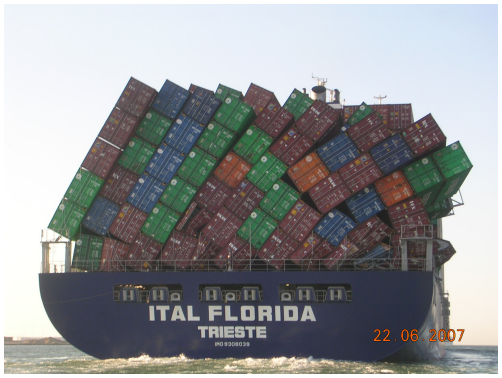
Here are some of the reasons why:

1. The discrete logarithm problem seem to be "hardest" in groups of prime order.*
2. Finding a generator is really, really easy.
3. Every non-zero exponent will be invertible modulo q (required in some security proofs).
4. DDH boils down to distinguishing between $(h_1, h_2, \mathrm{DH}_g(h_1, h_2))$ and (h_1, h_2, z) for random h_1, h_2, y, so a necessary condition for DDH to be hard is that $\mathrm{DH}_g(h_1, h_2)$ be indistinguishable from a random group element. When q is prime, this is very nearly true.

*An instance of the discrete logarithm problem in a group of order $q = q_1 \cdot q_2$ can be reduced to two instances of the problem for groups of orders q_1 and q_2.

Houston, we have a problem

- Groups of the form $\mathbb{Z}_p^*$, for p prime, give one class of cyclic groups in which the discrete logarithm problem is believed hard.
- However, for $p > 3, \mathbb{Z}_p^*$ does not have prime order.
- Worse still, the decision Diffie-Hellman problem is *not hard* in such groups.



Thankfully there is a solution

Definition. An element $y \in \mathbb{Z}_p^*$ is a *rth residue modulo p* if there exists an $h \in \mathbb{Z}_p^*$ such that $y = h^r \mod p$.

Theorem 8.64. Let $p = rq + 1$ with p, q prime. Then

$$\mathbb{G} \stackrel{\text{def}}{=} \{[h^r \mod p] \mid h \in \mathbb{Z}_p^*\}$$

is a subgroup of $\mathbb{Z}_p^*$ of order q.

Working in $\mathbb{G}$

Remark. To generate a uniform element of $\mathbb{G}$ choose a uniform element $h \in \mathbb{Z}_p^*$ and compute $[h^r \mod p]$.*

Remark. Testing whether an $h \in \mathbb{Z}_p^*$ is also in $\mathbb{G}$ can be done by checking whether $h^q \stackrel{?}{=} 1 \mod p$. To see why let g be a generator of $\mathbb{Z}_p^*$ and $h = g^i$. Then

$$\begin{aligned} h^q = 1 \mod p &\iff g^{iq} = 1 \mod p \\ &\iff iq = 0 \mod (p-1) \\ &\iff rq \mid iq \iff r \mid i. \end{aligned}$$

So

$$h = g^i = g^{cr} = (g^c)^r$$

for some c.

*Since $\mathbb{G}$ has prime order, every element of $\mathbb{G}$ is a generator.

Generating a uniform element of $\mathbb{G}$

Algorithm 8.65.
A group generation algorithm $\mathcal{G}$

Input: Security parameter $\ell = \ell(n)$
Output: Cyclic group $\mathbb{G}$, its prime order q, and a generator g.

generate a uniform n-bit prime q.
generate a uniform ℓ-bit prime p such that $q \mid (p-1)$
    // details omitted
choose a uniform $h \in \mathbb{Z}_p^*$ with $h \neq 1$
set $g := [h^{(p-1)/q} \mod p]$
return p, q, g // $\mathbb{G}$ is the order-q subgroup of $\mathbb{Z}_p^*$