A bit more fun with $\mathbb{Z}_n$ and $\mathbb{F}_p^\times$.

Let $a \in \mathbb{F}_p^\times$ be an element. How can we find a^{-1}?

Answer 1. Use extended Euclid's algorithm (see previous notes).

Answer 2. Use FLT: $a^{p-1} \equiv 1 \pmod p$ or

$$a \cdot a^{p-2} \equiv 1, \text{ so } a^{-1} \equiv a^{p-2}.$$

Remark (Fermat's primality test). This way we can also check if a number n is prime. Namely, pick a random $a \in \mathbb{Z}_n$ and check if $a^{n-1} \equiv 1$. In case $a^{n-1} \not\equiv 1$, then n is composite and a is called Fermat's witness of compositeness. If n is composite, but for chosen a, we have $a^{n-1} \equiv 1$, a is called Fermat's liar.

Flaw: there are infinitely many composite numbers n, s.t. for any a with $\gcd(a, n) = 1$, a is a Fermat's liar. These numbers are known as Carmichael numbers.

Rmk: if n is composite, but not Carmichael number, then at least half of $\{a \in \mathbb{Z}_n \mid \gcd(a,n) = 1\}$ are witnesses of compositeness.

'Proof': indeed, let a be such that $\gcd(a, n) = 1$ and $a^{n-1} \not\equiv 1$ (it exists, since n is not a Carmichael number). Then for any Fermat's liar b, we have $(ab)^{n-1} \equiv a^{n-1} b^{n-1} \equiv a^{n-1} \not\equiv 1$, so ab is a witness as well. The assertion follows $\boxtimes$.

We clearly see that ~~w~~ need to compute $a^k \pmod p$ arises time and again. How can we do that effectively?

The Fast Powering Algorithm.

__Step 1.__ Write the binary expression of K, i.e.

$$K = k_0 + k_1 \cdot 2 + k_2 \cdot 2^2 + \dots + k_r \cdot 2^r, \quad k_0, k_1, \dots, k_r \in \{0, 1\}$$

(bits.

$k_r = 1$.

__Step 2.__ Compute the powers $a^{2^i} \pmod p$ via

$$a_0 \equiv a$$

$$a_1 \equiv a_0^2 \equiv a^2$$

$$a_2 \equiv a_1^2 \equiv a^4$$

$$\vdots$$

$$a_r \equiv a_{r-1}^2 \equiv a^{2^r}$$

Then $a^k \equiv a^{k_0} \cdot a^{k_1 \cdot 2} \dots a^{k_r \cdot 2^r} \equiv a_0^{k_0} \cdot a_1^{k_1} \dots a_r^{k_r} \equiv \prod\limits_{\substack{i \in \{1, \dots, r\} \\ k_i \equiv 1 \pmod 2}} a_i^{k_i}$.

Rmk (running time). We will need at most $2r$ multiplications (r to compute a_i's and $\leq r$ to find $\prod_{\substack{i\in\{1,\dots,r\}\\ k_i\equiv 1}} a_i$) to find a^k. Notice that $2^r \leq k \leq 2^{r+1}$, so $r \approx \log_2 k$ and $2r \approx 2\log_2 k$.

Compare to the 'stupid multiplication': $a^k = \underbrace{a \cdot a \cdot \ldots \cdot a}_{k}$.

Say, if $k = 2^{100}$, then $2\log_2 k = 200$, while $2^{100} > 10^{30}$.

$10^{30} >>>> 400$!

Example. Let's find $3^{13} \pmod{17}$.

$13 = 2^3 + 2^2 + 2^0$ $(r=3)$.

✓ $a_0 \equiv 3$

$a_1 \equiv 3^2 \equiv 9$

✓ $a_2 \equiv 9^2 \equiv 81 \equiv 13$

✓ $a_3 \equiv 13^2 \equiv 169 \equiv -1 \equiv 16$.

$3^{13} \equiv 3 \cdot 13 \cdot 16 \equiv 39 \cdot 16 \equiv 624 \equiv 12$, so $3^{13} \equiv 12$.

Solving quadratic equations.

$x^2 \equiv 1 \pmod p$

$(x^2 - 1) \equiv 0$ or $(x-1)(x+1) \equiv 0$.

As p is prime, either $x-1 \equiv 0$ or $x+1 \equiv 0$, in other words,

$$\left[\begin{array}{l} x \equiv 1 \\ x \equiv -1 \end{array}\right.$$

Now $x^2 \equiv 1 \pmod{n}$ and $n = pq$ with p & q prime.
Again, $(x-1)(x+1) \equiv 0$, but now there are 4 possibilities:

(1) $x \equiv 1 \pmod{n}$
(2) $x \equiv -1 \pmod{n}$
(3) $x \equiv 1 \pmod{p}$ and $x \equiv -1 \pmod{q}$
(4) $x \equiv -1 \pmod{p}$ and $x \equiv 1 \pmod{q}$

<u>Example.</u> $x^2 \equiv 1 \pmod{15}$

(1) $x \equiv 1 \pmod{15}$
(2) $x \equiv -1 \equiv 14 \pmod{15}$
(3) $x \equiv 1 \pmod{3}$ and $x \equiv -1 \equiv 4 \pmod{5}$
$\Downarrow$
$x = 1 + 3k,\ k \in \mathbb{Z}$
$1 + 3k \equiv 4 \pmod{5}$
$3k \equiv 3$ or $k \equiv 1$

Conclusion: $x \equiv 1 + 3 \cdot 1 \equiv 4 \pmod{15}$. <u>Check</u>: $4^2 = 16 \equiv 1 \pmod{15}$

(4) $x \equiv -1 \equiv 2 \pmod{3}$ and $x \equiv 1 \pmod{5}$.
$x = 2 + 3k$
$2 + 3k \equiv 1 \pmod{5}$
$3k \equiv -1 \equiv 4$ or $2 \cdot 3k \equiv 2 \cdot 4$ or $k \equiv 3$

Conclusion: $x \equiv 2 + 3 \cdot 3 \equiv 11 \pmod{15}$ <u>Check</u>: $11^2 = 121 \equiv 1 \pmod{15}$.

Babystep - Giantstep Algorithm.

Let G be a (finite) group and $g \in G$ an element of order $N \geq 2$. We would like to solve the DLP problem, i.e. find k, s.t. $g^k = h$ (for a given h). The following algorithm is due to Shanks.

Let $n = 1 + \lfloor \sqrt{N} \rfloor$ (here $\lfloor \sqrt{N} \rfloor$ is the ~~lar~~gest integer smaller than $\sqrt{N}$). Notice that $n > \sqrt{N}$.

Step 1. Create two lists (n elements in each):

$$L_1 = \{e, g, g^2, \dots, g^n\}$$

$$L_2 = \{h, hg^{-n}, hg^{-2n}, \dots, hg^{-n^2}\}$$

Step 2. Find an element in both lists ($x \in L_1 \cap L_2$).

We will show below that such an x always exists.

Then $x = g^i = hg^{-jn}$,

$$g^i = hg^{-jn} \Leftrightarrow g^{i+jn} = h, \text{ hence } k \equiv i + jn \pmod{N}$$

Proof of existence of x:

as $g^k = h$ (for some unknown k), we consider (write)

$k = an + r$ for some a and $0 \leq r < n$.

Then $g^k = g^{an+r}$, giving $x = g^r = h \cdot g^{-an}$ $\boxtimes$

<u>Example</u>. Recall the example from previous lecture:

$G = \mathbb{F}_{17}^{\times}$, $g = 3$, $N = 16$, $3^{13} \equiv 12$. Let us pretend we don't know that $3^{13} \equiv 12$ and find k, s.t. $3^k \equiv 12$ using the algorithm above.

$n = 1 + \lfloor \sqrt{N} \rfloor = 5$.

<u>Step 1.</u> $L_1 = \{1, 3, 9, 10, 13, 5\}$

$L_2 = \{12, 12 \cdot 3^{-5}, 12 \cdot 3^{-10}, 12 \cdot 3^{-15}, 12 \cdot 3^{-20}, 12 \cdot 3^{-25}\}$.

Let's simplify the elements in L_2.

$3^{-1} \equiv 6$ (since $3 \cdot 6 \equiv 18 \equiv 1$)

$3^{-5} \equiv 6^5 \equiv 6^2 \cdot 6^2 \cdot 6 \equiv 2 \cdot 2 \cdot 6 \equiv 7$, so

$L_2 = \{12, 12 \cdot 7, 12 \cdot 7^2, 12 \cdot 7^3, 12 \cdot 7^4, 12 \cdot 7^5\} = \{12, 16, 10, 2, 14, 13\}$.

$(12 \cdot 7 \equiv -5 \cdot 7 \equiv -35 \equiv -1)$

We find that $L_1 \cap L_2 = \{10, 13\}$.

<u>Step 2.</u> $10 \overset{L_1}{\equiv} 3^3 \overset{L_2}{\equiv} 3^{-2 \cdot 5} \cdot 12$, so $12 \equiv 3^{3+2 \cdot 5} \equiv 3^{13}$, hence $k = 13$.

$13 \overset{L_1}{\equiv} 3^4 \overset{L_2}{\equiv} 3^{-5 \cdot 5} \cdot 12$, so $12 \equiv 3^{4+5 \cdot 5} \equiv 3^{29} \equiv 3^{13}$ (since $3^{29} \equiv 3^{13} \cdot 3^{16}$ and $3^{16} \equiv 1$ by FLT).

<u>Remark</u>. The intersection $L_1 \cap L_2$ may consist of more than one element. It doesn't matter which one to use.

The Chinese remainder theorem.

The following problem appeared in Master Tzu Suang Ching Math. Manual, ca. 300 AD.

We have a number of things, but we do not know exactly how many. If we count them by threes, we have two left over. If we count them by fives, we have three left over. If we count them by sevens, we have two left over. How many things do we have?

Let's find out: For this we need to find a number n, s.t. $\begin{cases} n \equiv 2 \pmod 3 \\ n \equiv 3 \pmod 5 \\ n \equiv 2 \pmod 7 \end{cases}$.

Any $n = 2+3k_1$, $k_1 \in \mathbb{Z}$, satisfies the first equation.

Then $2+3k_1 \equiv 3 \pmod 5 \iff 3k_1 \equiv 1 \iff k_1 \equiv 2 \pmod 5$, i.e.

$k_1 = 2+5k_2$, $k_2 \in \mathbb{Z}$ and $n = 2+3(2+5k_2) = 8+15k_2$.

Finally, we need $8+15k_2 \equiv 2 \pmod 7 \iff$

$\iff k_2 \equiv -6 \equiv 1$, hence $n = 8+15(1+7l) = 23+105l$, $l \in \mathbb{Z}$

<u>Check</u>: $23 \equiv 2 \pmod 3$

$23 \equiv 3 \pmod 5$

$23 \equiv 2 \pmod 7$.

Thm (CRT). Let $n = a_1 \cdot a_2 \cdot \ldots \cdot a_k$ be an integer with a_i's pairwise coprime ($\gcd(a_i, a_j) = 1$). Then the system of equivalences

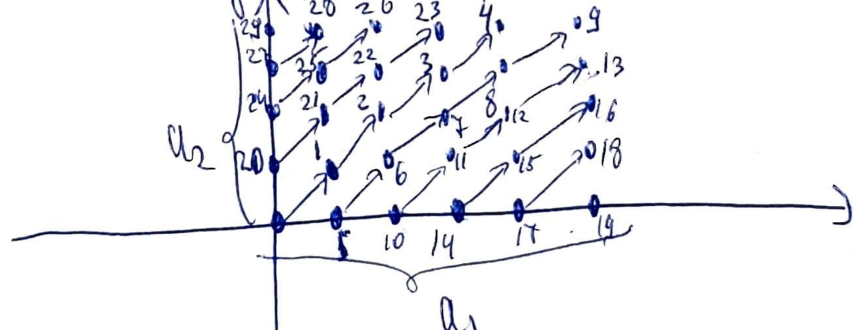
$$\begin{cases} x \equiv s_1 \pmod{a_1} \\ x \equiv s_2 \pmod{a_2} \\ \vdots \\ x \equiv s_k \pmod{a_k} \end{cases} \quad (*)$$

has a solution. Each such solution has the form $x = s + k \cdot n$, $k \in \mathbb{Z}$, where s is the unique sol-n modulo n.

Alternative formulation. There is a group isomorphism $\varphi : \mathbb{Z}_n \xrightarrow{\sim} \mathbb{Z}_{a_1} \times \ldots \times \mathbb{Z}_{a_k}$ given by

$$\varphi(1) = (1, \ldots, 1).$$

Geometric visualization (2d)



Adding vector $(1, \ldots, 1)$ to itself, will 'visit' every point prior to landing at the origin.

Proof: induction on the number of factors.

Base: $n = a_1$. Need to find $x \equiv s_1 \pmod{a_1}$ ✓.

Step: suppose we have found a sol-n of $\leftarrow b_i$

$$\begin{cases} x \equiv s_1 \pmod{a_1} \\ \vdots \\ x \equiv s_i \pmod{a_i} \end{cases}$$

and want to find a sol-n of

$$\begin{cases} x \equiv s_1 \pmod{a_1} \\ \vdots \\ x \equiv s_i \pmod{a_i} \\ x \equiv s_{i+1} \pmod{a_{i+1}}. \end{cases} \quad (**)$$

Well $\underset{k \in \mathbb{Z}}{b_i + k \cdot a_1 \cdot \ldots \cdot a_i}$ satisfies all congruences, probably, except $x \equiv s_{i+1} \pmod{a_{i+1}}$. As $\gcd(a_1 \ldots a_i, a_{i+1}) = 1$ we can find $\alpha, \beta \in \mathbb{Z}: \alpha a_1 \ldots a_i + \beta a_{i+1} = 1$ (Euclid's algorithm) or $\alpha \cdot s_{i+1} a_1 \ldots \cdot a_i + \beta s_{i+1} a_{i+1} = s_{i+1}$, thus,

$$b_{i+1} = b_i + \alpha \cdot s_{i+1} a_1 \ldots a_i \equiv b_i \pmod{a_j},\ j \in \{1, \ldots, i\}.$$

and $b_{i+1} \equiv s_{i+1} \pmod{a_{i+1}}$ is a sol-n of $(**)$.

The Pohlig-Hellman algorithm.

Recall that to solve the discrete logarithm problem (DLP), we needed to find s:

$$g^s \equiv h \text{ for } g \in G \text{ of order } N.$$

As g has order N, s is defined modulo N, i.e. any $k \equiv s \pmod{N}$ is a sol-n ($g^k = h$).

Now $N = p_1^{a_1} \cdots p_k^{a_k}$ (decomposition into product of coprime factors).

Let $g_i = g^{N/p_i^{a_i}}$ and $h_i = h^{N/p_i^{a_i}}$.

Suppose we found s satisfying

$$s \equiv s_i \pmod{p_i^{a_i}} \text{ with } g_i^{s_i} \equiv h_i.$$

Notice that the cyclic subgroup generated by g inside G is $\mathbb{Z}_N$, i.e. $\langle g \rangle \simeq \mathbb{Z}_N \underset{CRT}{\cong} \mathbb{Z}_{p_1^{a_1}} \cdots \mathbb{Z}_{p_k^{a_k}}$ and $\varphi(g) = 1$, $\varphi(h) = (s_1, \dots, s_k)$ (since $h \sim s \in \mathbb{Z}_N$). We conclude that $g_1^{s_1} \cdots g_k^{s_k} = g^s = h$. In other words to solve the DLP:

1. Solve each 'sub DLP' for $p_i^{a_i}$, i.e. find $s_i : g_i^{s_i} \equiv h_i$.
2. Use CRT to find s from $(s_1, \dots, s_k)$.

Example. $G = \mathbb{F}_{19}^{\times}$, $g = 2$, $N = \mathrm{ord}(2) = 18 = 2 \cdot 3^2$, $h = 15$.

$$g_1 = 2^{18/2} = 2^9 \equiv 18, \quad h_1 = 15^9 \equiv (-4)^9 \equiv -4^9 \equiv -1 \equiv 18$$

$$g_2 = 2^{18/9} = 4, \quad h_2 = 15^2 \equiv (-4)^2 \equiv 16.$$

$$\begin{cases} 18^{s_1} \equiv 18 \\ 4^{s_2} \equiv 16 \end{cases}$$, hence, $s_1 \equiv 1 \pmod 2$ and $s_2 \equiv 2 \pmod 9$

We need to find s:

$$\begin{cases} s \equiv 1 \pmod 2 \\ s \equiv 2 \pmod 9 \end{cases}$$

$$s = 1 + 2k,\ k \in \mathbb{Z} \rightsquigarrow 1 + 2k \equiv 2 \pmod 9$$

$$2k \equiv 1 \iff 5 \cdot 2k \equiv 5 \text{ or } k \equiv 5, \text{ so,}$$

$$s = 1 + 2 \cdot 5 \equiv 11 \pmod{18}.$$

Check: $2^{11} = 2^8 \cdot 2^2 \cdot 2 = 16^2 \cdot 4 \cdot 2 \equiv 9 \cdot 8 \equiv 72 \equiv 15$. ✓