Let G be a finite cyclic group and $g \in G$ one of its generators. Let $|G| = m$.

Let $h \in G$. Then there exists a unique $x \in \{0, \ldots, m-1\}$, such that $g^x = h$.

This x is the (discrete) logarithm of h to the base g. Denote $x = \log_g h$.

If g is clear from the context then we do not mention it: $\log h$ is the discrete logarithm of h.

We can also define $\log_g$ if g is not a generator of G, but then $\log_g$ is a partial function.

A particular instance: $G = \mathbb{Z}_p^*$ for $p \in \mathbb{P}$.

- Operation — multiplication.

Supposedly discrete logarithm is hard for this instance.

- if p is a randomly generated prime of sufficient length.

In the following, if we speak about a group G, we assume that multiplication, taking inverses and finding the unit element are simple operations.

Example: $\mathbb{Z}_{13}^*$. A generator of it is 2.

i	0	1	2	3	4	5	6	7	8	9	10	11
$2^i \bmod 13$	1	2	4	8	3	6	12	11	9	5	10	7

Inverting this table gives us

h	1	2	3	4	5	6	7	8	9	10	11	12	$\in \mathbb{Z}_{13}^*$
$\log_2 h$	0	1	4	2	9	5	11	3	8	10	7	6	$\in \mathbb{Z}$

On the other hand, 3 is not a generator of $\mathbb{Z}_{13}^*$.

i	0	1	2	3	4	5	6	7	8	9	10	11
$3^i \bmod 13$	1	3	9	1	3	9	1	3	9	1	3	9

Hence $\log_3 1 = 0$, $\log_3 3 = 1$ and $\log_3 9 = 2$. The function $\log_3$ is undefined for other values.

Exercise. Give an example of a (family of) cyclic group(s) where finding the discrete logarithms is an easy problem.

Hybrid usage of asymmetric and symmetric cryptosystems to encrypt a plaintext x:

Let a symmetric cryptosystem be fixed. It may be a block cipher with a fixed mode of operation.

1. Generate a new key k_s of the symmetric cryptosystem.
2. Let $y = E_{k_s}^{\mathrm{symm}}(x)$.
3. Let $k' = E_{k_p}^{\mathrm{asymm}}(k_s)$.
4. The cryptotext is (k', y).

In a bit more general terms:

If A wants to send a message x to B then

- A and B somehow agree on the key k_s for the symmetric cryptosystem.
  - The eavesdropper must not learn k_s.
- A sends to B the message $E^{\mathrm{symm}}_{k_s}(x)$.

Using an asymmetric cryptosystem, the agreement on k_s is achieved with the following steps:

- B generates a new asymmetric keypair $(k_{\text{pub}}, k_{\text{sec}})$ and sends k_{pub} to A.
- A generates k_s and sends $k' = E^{\text{asymm}}_{k_{\text{pub}}}(k_s)$ to B.
- B decrypts $k_s = D^{\text{asymm}}_{k_{\text{sec}}}(k')$.

Diffie-Hellman key agreement protocol:

- Let a cyclic group G and its generator g be fixed. Let $|G| = m$.
  - They may be fixed globally, or be chosen at each run of the protocol.
- A randomly chooses $a \in \{0, \ldots, m-1\}$. B randomly chooses $b \in \{0, \ldots, m-1\}$.
- A sends g^a to B. B sends g^b to A.
- Both A and B compute $k_0 = g^{ab}$.
  - A computes $(g^b)^a$. B computes $(g^a)^b$.
- A hash of k_0 is taken as the key k_s.
  - k_0 is distributed differently than the keys for typical symmetric cryptosystems.

The adversary sees (the description of) G, g, g^a and g^b.

The adversary wants to compute g^{ab}.

This problem is the Diffie-Hellman problem.

It is no harder than discrete logarithm.

It is also presumed to be hard for $\mathbb{Z}_p^*$.

Example: let $G = \mathbb{Z}_{13}^*$. Let $g = 2$.

Let A generate $a = 7$. Let B generate $b = 4$.

Then A sends to B $g^a = 2^7 \equiv 11 \pmod{13}$. And B sends to A $g^b = 2^4 \equiv 3 \pmod{13}$.

A computes $3^7 = 729 \equiv 3 \pmod{13}$. And B computes $11^4 = 14641 \equiv 3 \pmod{13}$.

The adversary only sees 11 and 4 and has to solve the Diffie-Hellman problem.

ElGamal public key cryptosystem:

Let a cyclic group G, $|G| = m$ and its generator g be fixed.

- Key generation: randomly choose $a \in \{0, \ldots, m-1\}$. Let $h = g^a$.
  - Public key: h. Secret key: a.
    - If G and g are not global, then they are part of the public (and secret) key.
- Set of possible plaintexts: G.
- Encryption of $x \in G$: randomly generate $r \in \{0, \ldots, m-1\}$.

$$E_h(x, r) = (g^r, x \cdot h^r)$$

- Decryption:

$$D_a(c_1, c_2) = c_2 \cdot c_1^{-a}$$

Decryption works:

We had $E_h(x, r) = (g^r, x \cdot h^r)$ and $g^a = h$.

$$D_a(g^r, x \cdot h^r) = x \cdot h^r \cdot (g^r)^{-a} = x \cdot h^r \cdot (g^a)^{-r} = x \cdot h^0 = x$$

Example. Let $G = \mathbb{Z}_{19}^*$ and $g = 2$.

Let the secret key be 13. The public key is then 3.

Let the message be 8. To encrypt, we generate $r \in \{0, \ldots, 17\}$. Let r be 10.

The cryptotext is $(g^r, xh^r) = (2^{10}, 8 \cdot 3^{10}) \equiv (17, 14)$.

To decrypt we compute $c_1^a = 17^{13} \equiv 16$. We invert it and obtain $c_1^{-a} = 6$. The plaintext is $c_2 \cdot c_1^{-a} = 14 \cdot 6 \equiv 8$.

If we can solve the Diffie-Hellman problem then we can break ElGamal cryptosystem.

Let cyclic G, $m = |G|$ and generator g be fixed. Let $h \in G$ be an ElGamal public key.

We are given a ciphertext $(c_1, c_2) = (g^r, x \cdot h^r)$ where r and x are unknown. We want to find x.

We solve the DH problem instance (G, g, c_1, h). Here $c_1 = g^r$ and $h = g^a$. We obtain $y = g^{ar} = h^r$.

We find $x = xh^r \cdot h^{-r} = c_2 \cdot y^{-1}$.

If we can break ElGamal cryptosystem then we can solve the Diffie-Hellman problem.

Let the problem instance (G, g, g', g'') need solving, where $g' = g^a$ and $g'' = g^b$ but a and b are unknown to us.

Let ElGamal cryptosystem use the same G and g.

Let the public key be $(g'')^{-1}$ and the message be $(g', 1)$. We break the system and find the plaintext x satisfying

$$(g', 1) = (g^a, x \cdot (g^{-b})^a) = (g^a, x \cdot g^{-ab})$$

hence $x = g^{ab}$ is the solution to the Diffie-Hellman problem.

Assume that ElGamal cryptosystem is used to create several different ciphertexts using the same key.

What do we have to keep in mind when choosing r?

Can we reuse a random r?

Given $(g^r, x_1 h^r)$ and $(g^r, x_2 h^r)$ we can find x_1/x_2. Hence a r should not be reused.

Property	ElGamal	RSA
Encryption complexity	two modular exponentiations	one modular exponentiation (with small modulus)
Decryption complexity	one modular exponentiation	one modular exponentiation
Randomized?	yes	no
Message expansion	twice	none (i.e. once)
Genericity	applicable to any cyclic group	usable in a single structure

Given a cyclic G with $m = |G|$, how do we verify that $g \in G$ is a generator?

Assume that we can factor m: $m = p_1^{e_1} \cdots p_k^{e_k}$.

- If we cannot, pick some other G.
- To generate $p \in \mathbb{P}$, such that we can factor $|\mathbb{Z}_p^*| = p-1$, we can let p be a strong prime.

The order of g must divide m.

If the order of g is not m then it must divide one of the numbers m/p_i, where $i \in \{1, \ldots, k\}$.

We verify whether $g^{m/p_i} = 1$ for some $i \in \{1, \ldots, k\}$. If not, then g is a generator.

Given a cyclic G with $m = |G|$ and a generator $g \in G$, how do we compute $\log_g h$ for some $h \in G$?

Simplest method — enumeration. Compute $g^0, g^1, g^2, \ldots$ until $g^n = h$ for some n. Then $\log_g h = n$.

Time complexity: $O(m)$. Space complexity: $O(1)$.

Shanks' baby-step giant-step algorithm ("meet-in-the-middle"):

Let $l = \lceil \sqrt{m}\, \rceil$. Then $\log_g h = ql + r$ for some $q \in \{0, \dots, l-1\}$ and $r \in \{0, \dots, l-1\}$. Let

$$S = \{(hg^{-r}, r) \,|\, 0 \leqslant r < l\}$$

be organized as a hash table with hg^{-r} as the key.

If $(1, r) \in S$ then $\log_g h = r$.

Otherwise compute $g^l, g^{2l}, g^{3l}, \dots$ until $(g^{ql}, r) \in S$ for some q and r. Then $\log_g h = ql + r$.

Time complexity: $O(\sqrt{m})$. Space complexity: $O(\sqrt{m})$.

Still infeasible if $|G| \geqslant 2^{160}$.

Birthday paradox: let there be 23 random people in the same room. The probability that two of them have the same birthday is more than 50%.

In general, let X be a set, $|X| = n$. Let $x_1, \ldots, x_k$ be mutually independent uniformly distributed random variables over X. The probability that $x_1, \ldots, x_k$ are all different is

$$\prod_{i=1}^{k} \frac{n+1-i}{n} = \prod_{i=1}^{k-1} \frac{n-i}{n} = \prod_{i=1}^{k-1} \Big(1 - \frac{i}{n}\Big) \leqslant^{\forall x \in \mathbb{R}\,:\,1+x \leqslant e^x}$$

$$\prod_{i=1}^{k-1} e^{-i/n} = e^{-\sum\limits_{i=1}^{k-1} i/n} = e^{-k(k-1)/(2n)}$$

If $k \geqslant \frac{1}{2}(1 + \sqrt{1 + 8n \ln 2})$ then this probability is at most $1/2$.

Pollards ρ-algorithm: partition the group G into three parts G_1, G_2, G_3, such that membership tests for all parts are easy. Let $1 \notin G_2$.

Define $f : G \to G$ by

$$f(x) = \begin{cases} gx, & x \in G_1 \\ x^2, & x \in G_2 \\ hx, & x \in G_3 \end{cases}$$

Define $f^0(x) = x$ and $f^i(x) = f(f^{i-1}(x))$.

Let $z \in \{0, \dots, m-1\}$ be randomly chosen. Let $x_i = f^i(g^z)$. There exist α_i and β_i, such that $x_i = g^{\alpha_i} h^{\beta_i}$, where $\alpha_0 = z$, $\beta_0 = 0$ and

$$\alpha_{i+1} = \begin{cases} \alpha_i + 1, & x_i \in G_1 \\ 2\alpha_i, & x_i \in G_2 \\ \alpha_i, & x_i \in G_3 \end{cases} \qquad \beta_{i+1} = \begin{cases} \beta_i, & x_i \in G_1 \\ 2\beta_i, & x_i \in G_2 \\ \beta_i + 1, & x_i \in G_3 \end{cases}$$

(all computations are modulo m).

Suppose that we have found such i and j, where $i \neq j$ but $x_i = x_j$. Then

$$g^{\alpha_i} h^{\beta_i} = g^{\alpha_j} h^{\beta_j}$$

meaning that

$$h^{\beta_j - \beta_i} = g^{\alpha_i - \alpha_j}$$

Hence

$$\log_g h = \frac{\alpha_i - \alpha_j}{\beta_j - \beta_i} \pmod{m} .$$

If $(\beta_j - \beta_i)^{-1} \pmod{m}$ does not exist then we try again with a different z.

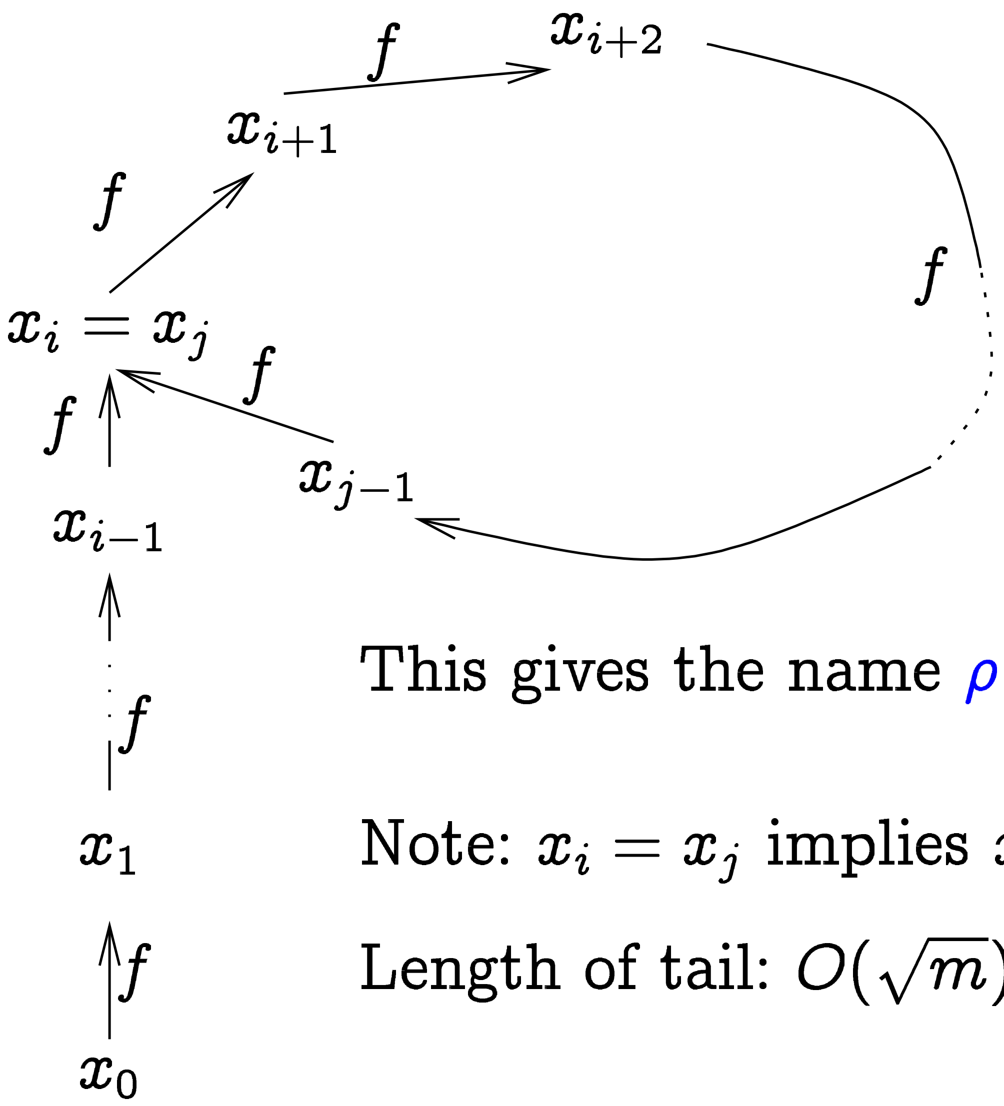
Or... there definitely exists such k that $k(\beta_j - \beta_i) = \alpha_i - \alpha_j$ (take $k = \log_g h$). If there are not too many such k-s then we can try them all out.

Consider the values $\{x_i\}_{i\in\mathbb{N}}$. If the values x_i were mutually independent uniformly distributed random variables then two equal values exist among $O(\sqrt{m})$ first ones with high probability.

They are not independent, but for the purpose of our analysis, we do not care.

To find $\log_g h$: compute $x_0, x_1, \ldots, \alpha_0, \alpha_1, \ldots$ and $\beta_0, \beta_1, \ldots$ until $x_i = x_j$ for $i \neq j$. Then proceed as in the previous slide.

Time complexity: $O(\sqrt{m})$. Space complexity: $O(\sqrt{m})$. (both expected)



This gives the name ρ

Note: $x_i = x_j$ implies $x_{i+1} = x_{j+1}$

Length of tail: $O(\sqrt{m})$. Length of cycle: $O(\sqrt{m})$.

Floyd's cycle-finding algorithm: compute the sextuples

$$(x_i, \alpha_i, \beta_i, x_{2i}, \alpha_{2i}, \beta_{2i})$$

(here $i = 0, 1, 2, \ldots$) until $x_i = x_{2i}$.

Here $(x_{i+1}, \alpha_{i+1}, \beta_{i+1}, x_{2(i+1)}, \alpha_{2(i+1)}, \beta_{2(i+1)})$ can be computed from $(x_i, \alpha_i, \beta_i, x_{2i}, \alpha_{2i}, \beta_{2i})$, which can then be discarded.

$x_i = x_{2i}$ is reached while x_i is making the first round on the cycle. Hence $i = O(\sqrt{m})$ at that moment.

Discrete logarithm's algorithm's time complexity: $O(\sqrt{m})$ (expected). Space complexity: $O(1)$.

Example: let $G = \mathbb{Z}_{197}^*$. Let $g = 2$. Then g is a generator.

Indeed, $m = |G| = 196 = 2^2 \cdot 7^2$. We have

$$2^{\frac{196}{2}} \equiv -1 \text{ and } 2^{\frac{196}{7}} \equiv 104 \pmod{197} \ .$$

Let us find $\log_2 133$ in $\mathbb{Z}_{197}^*$.

Partition: $G_1 = \{1, \ldots, 65\}$, $G_2 = \{66, \ldots, 131\}$, $G_3 = \{132, \ldots, 196\}$.

Randomly pick $z = 20$. Then $x_0 = 66$, $\alpha_0 = 20$, $\beta_0 = 0$.

i	x_i	α_i	β_i	x_{2i}	α_{2i}	β_{2i}
0	142	20	0	142	20	0
1	171	20	1	88	20	2
2	88	20	2	122	41	4
3	61	40	4	61	164	16

Hence

$$\log_2 133 = \frac{40 - 164}{16 - 4} \pmod{196}$$

$12^{-1} \pmod{196}$ does not exist. We have to consider all k-s satisfying the following congruence as possible values for $\log_2 133$:

$$12k \equiv -124 \pmod{196} .$$

Dividing everything by $\gcd(12, 196) = 4$ gives us

$$3k \equiv -31 \equiv 18 \pmod{49}$$

I.e. $k \equiv 6 \pmod{49}$. The possible values for k modulo 196 are 6, 55, 104 and 153. We try all of them:

$$\begin{aligned} 2^6 &\equiv 64 \pmod{197} & 2^{104} &\equiv 133 \pmod{197} \\ 2^{55} &\equiv 89 \pmod{197} & 2^{153} &\equiv 108 \pmod{197} \ . \end{aligned}$$

Hence $\log_2 133 = 104$ in $\mathbb{Z}_{197}^*$.

Suppose that we know the factorization of $|G| = m$: let $m = p_1^{e_1} \cdots p_k^{e_k}$. Pohlig-Hellman algorithm lets us to reduce the computation of discrete logarithms in G to the computation of discrete logarithms in groups of order p_i.

Let g be a generator of G and let us look for $\log_g h$.

For each $i \in \{1, \ldots, k\}$ define

$$m_i = \frac{m}{p_i^{e_i}} \qquad g_i = g^{m_i} \qquad h_i = h^{m_i} \quad .$$

g_i generates of subgroup of G of order $p_i^{e_i}$ and h_i belongs to that subgroup.

Let $x_i = \log_{g_i} h_i$. Then $x = \log_g h$ satisfies the system of congruences

$$\{x \equiv x_i \pmod{p_i^{e_i}}\}_{1 \leqslant i \leqslant k}$$

which has a unique solution modulo m (use chinese remainder theorem to find it).

Indeed, for all $i \in \{1, \ldots, k\}$,

$$(g^{-x}h)^{m_i} = (g^{m_i})^{-x}h^{m_i} = g_i^{-x}h_i = g_i^{-(lp_i^{e_i}+x_i)}h_i = g_i^{-x_i}h_i = 1$$

for some $l \in \mathbb{Z}$.

Hence the order of $g^{-x}h$ divides m_i for all i. Then it also divides $\gcd(m_1, \ldots, m_k) = 1$. Hence the order of $g^{-x}h$ is 1, i.e. $g^{-x}h = 1$ and $g^x = h$.

We have reduced the finding of discrete logarithms in G to the finding of discrete logarithms in the subgroups of G whose orders are prime powers.

Assume now that $|G| = p^e$ for some $p \in \mathbb{P}$. We want to find $\log_g h$ in G where g is a generator of G.

Denote $x = \log_g h$. Then $x = x_0 + x_1 p + x_2 p^2 + \cdots + x_{e-1} p^{e-1}$ for some $x_0, \ldots, x_{e-1} \in \{0, \ldots, p-1\}$. Our task is to find these x_i.

We are going to have $g^x = h$. Then also $g^{p^{e-1}x} = h^{p^{e-1}}$. But

$$p^{e-1}x = p^{e-1}x_0 + p^e(x_1 + px_2 + \ldots + p^{e-2}x_{e-1}) \equiv p^{e-1}x_0 \pmod{p^e}$$

As $g^{p^e} = 1$, the value x_0 must satisfy $g^{p^{e-1}x_0} = h^{p^{e-1}}$. Hence x_0 can be found be solving a discrete logarithm in the subgroup generated by $g^{p^{e-1}}$. Its order is p.

Assume that we have already found $x_0, \ldots, x_{j-1}$. To find x_j we note that we must have

$$g^{x_j p^j + \cdots + x_{e-1} p^{e-1}} = h g^{-x_0 - x_1 p - \cdots - x_{j-1} p^{j-1}}$$

Denote the right hand side by h_j. Then we must also have

$$g^{x_j p^{e-1} + x_{j+1} p^e + \ldots + x_{e-1} p^{2e-j-2}} = h_j^{p^{e-j-1}}$$

Here the left hand side equals $g^{x_j p^{e-1}}$. We find x_j from the equation $(g^{p^{e-1}})^{x_j} = h_j^{p^{e-j-1}}$.

Example: let $G = \mathbb{Z}^*_{64153}$. Then $|G| = 64152 = 2^3 \cdot 3^6 \cdot 11$.
Let $g = 5$. Then g is a generator of G. Indeed,

$$5^{\frac{64152}{2}} \equiv 64152 \pmod{64153}$$

$$5^{\frac{64152}{3}} \equiv 58563 \pmod{64153}$$

$$5^{\frac{64152}{11}} \equiv 57412 \pmod{64153}$$

Let us find $\log_5 43210$ in G.

Reduce finding that discrete logarithm to finding discrete logarithms modulo prime powers:

$$m_1 = \frac{64152}{2^3} = 8019 \quad m_2 = \frac{64152}{3^6} = 88 \quad m_3 = \frac{64152}{11} = 5832$$

$$g_1 = 5^{8019} = 6899 \quad g_2 = 5^{88} = 45332 \quad g_3 = 5^{5832} = 57412$$

$$h_1 = 43210^{8019} = 5325 \qquad h_2 = 43210^{88} = 60946$$

$$h_3 = 43210^{5832} = 37326$$

(all powers modulo 64153).

We must find $x_1 = \log_{g_1} h_1 = \log_{6899} 5325$ in G. We know that this logarithm must belong to $\{0, \ldots, 7\}$. By trying all possibilities we find that $x_1 = 6$.

We must find $x_3 = \log_{g_3} h_3 = \log_{57412} 37326$. We know that this logarithm must belong to $\{0, \ldots, 10\}$. By trying all possibilities we find that $x_3 = 9$.

We must find $x_2 = \log_{g_2} h_2 = \log_{45332} 60946$. We know that this logarithm must belong to $\{0, \ldots, 3^6 - 1\}$. We reduce finding this logarithm to finding logarithms in the group of three elements.

We have

$$x_2 = y_0 + 3y_1 + 9y_2 + 27y_3 + 81y_4 + 243y_5,$$

where $y_i \in \{0, 1, 2\}$.

We find y_0 from $g_2^{243y_0} = h_2^{243}$. I.e.

$$58563^{y_0} = (45332^{243})^{y_0} = (g_2^{243})^{y_0} = h_2^{243} = 60946^{243} = 5589$$

By trying all three possibilities we find $y_0 = 2$.

In the following we need $g_2^{-1} = 45332^{-1} = 29774 \pmod{64153}$.

As next, we have $g_2^{243y_1} = (h_2 g_2^{-2})^{81}$. I.e.

$$58563^{y_1} = 45332^{243y_1} = (60946 \cdot 45332^{-2})^{81} = 5589$$

and $y_1 = 2$.

Then we have $g_2^{243y_2} = (h_2 g_2^{-(2+3 \cdot 2)})^{27}$. I.e.

$$58563^{y_2} = (60946 \cdot 45332^{-8})^{27} = 58563$$

and $y_2 = 1$.

Then we have $g_2^{243y_3} = (h_2 g_2^{-2+3 \cdot 2+9})^{9}$. I.e.

$$58563^{y_3} = (60946 \cdot 45332^{-17})^{9} = 5589$$

and $y_3 = 2$.

Then we have $g_2^{243y_4} = (h_2 g_2^{-(2+3\cdot 2+9+27\cdot 2)})^3$. I.e.

$$58563^{y_4} = (60946 \cdot 45332^{-71})^3 = 58563$$

and $y_4 = 1$.

Finally, $g_2^{243y_5} = h_2 g_2^{-(2+3\cdot 2+9+27\cdot 2+81)}$. I.e.

$$58563^{y_5} = 60946 \cdot 45332^{-152} = 5589$$

and $y_5 = 2$.

Thus $x_2 = \sum_{i=0}^{5} y_i 3^i = 638$.

We have the system of congruences

$$\begin{cases} x \equiv x_1 \pmod{p_1^{e_1}} \\ x \equiv x_2 \pmod{p_2^{e_2}} \\ x \equiv x_3 \pmod{p_3^{e_3}} \end{cases} \quad \text{or} \quad \begin{cases} x \equiv 6 \pmod{2^3} \\ x \equiv 638 \pmod{3^6} \\ x \equiv 9 \pmod{11} \end{cases}$$

Using the chinese remainder theorem we find $x = 58958$. This is the discrete logarithm of 43210 to the base 5 in $\mathbb{Z}^*_{64153}$.