One-way functions

for cryptgrafy we need functions that are easy to calculate but their inverse is not (almost impossible)

$f: X \xrightarrow{\text{easy}} X$

$f: X \xleftarrow{\text{hard}} X$

The discrete logarithm

We know that $\mathbb{F}_p^*$ is a cyclic group with $p-1$ elements.

We also know that $\exists k \in \mathbb{F}_p^* \mid \forall x \in \mathbb{F}_p^*, x = k^n, n \in \mathbb{Z}$

the function $f: x \longrightarrow g^x \pmod p$ is a good candidate for a one-way function

p is not necessarily a prime number, but later it will be

> given a base g, an exponent v and n $(g, v, n \in \mathbb{Z})$ n not necessarily prime
> determine $g^v \pmod n$

powers modulo p can easily be calculated by looking at v in its binary decomposition and calculating first the powers of g where the exponent is a power of 2

e.g. $17^{246} \pmod{135}$

lets write a list of powers of 17 modulo 135

$17 \equiv 17 \pmod{135}$

$17^2 \equiv 289 \equiv 19 \pmod{135}$

$17^4 \equiv (17^2)^2 \equiv 19^2 \equiv 361 \equiv 91 \pmod{135}$

$17^8 \equiv (17^4)^2 \equiv 91^2 \equiv 8281 \equiv 46 \pmod{135}$

$17^{16} \equiv (17^8)^2 \equiv 46^2 \equiv 2116 \equiv 91 \pmod{135}$

$17^{32} \equiv (17^{16})^2 \equiv 91^2 \equiv 8281 \equiv 46 \pmod{135}$

$17^{64} \equiv (17^{32})^2 \equiv 46^2 \equiv 2116 \equiv 91 \pmod{135}$

$17^{128} \equiv (17^{64})^2 \equiv 91^2 \equiv 8281 \equiv 46 \pmod{135}$

now we write v as a sum of powers of 2

$246 = 128 + 64 + 32 + 16 + 4 + 2$

now we can easily make the calculation

$17^{246} = 17^{128+64+32+16+4+2} = 17^{128} \cdot 17^{64} \cdot 17^{32} \cdot 17^{16} \cdot 17^4 \cdot 17^2 = 46 \cdot 91 \cdot 46 \cdot 91 \cdot 91 \cdot 19 \equiv 3029656848 \equiv 109 \pmod{135}$

this method can be fine-tuned into powermod and is $O(\log_2 n)$

powermod algorithm: exponential bs repeated squaring

calculate $y = g^v \pmod n$, write its binary decomposition:

$v = (v_{m-1} v_{m-2} \dots v_2 v_1)_2$ so $v = v_{m-1} 2^{m-1} + \dots v_1 \cdot 2 + v_0$

algo:

```
y = 1
for(i = m-1 to 0){
(1) y = y^2 (mod n)
    if(v_i = 1){
(2)     y = g·y (mod n)
    }
}
```

example

g^{45} $\quad 45 = (101101)_2$

i	5	4	3	2	1	0
y (after step 1)	1	g^2	g^4	g^{10}	g^{22}	g^{44}
v_i	1	0	1	1	0	1
y (after step 2)	g	g^2	g^5	g^{11}	g^{22}	g^{45}

here I have the number I wanted

the other was around

given base g, y and n determine v such that $y = g^v \pmod n$ that is called discrete logarithm problem

In practice always n prime noted as p

from Fermat's little theorem we know $g^{p-1} \equiv 1 \pmod p$

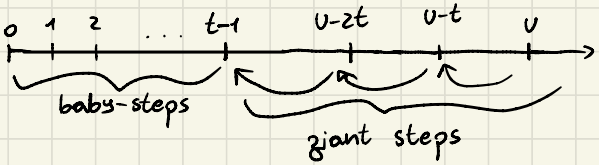
Solving DLP

You can try brute force but it's impossible for large numbers

Baby-step Giant-Step

we have to divide the $\{1, \dots p-1\}$ interval into t different ones, all of them with approssimate lenght t ($t=p$ also works)

the baby steps take place within the intervals, the giant-steps go to another sub-interval



let's make an example

$7^U \equiv 78 \pmod{97}$

7 it's a generator of $\mathbb{F}_{97}^*$

now we consider $t = \lceil \sqrt{97} \rceil = 10$ so now we are going to do 10 baby steps of 7^i with $i = 0, 1, \dots 9$

i	0	1	2	3	4	5	6	7	8	9
$7^i \pmod{97}$	1	7	49	52	73	26	85	13	91	55

let's sort by power

7^i	1	7	13	26	49	52	55	73	85	91
i	0	1	7	5	2	3	9	4	6	8

now the giant steps with size $g^t = 7^{10}$

we start with $y = 78$ and multiply it by g^{-t} until we find a number from the baby list

in this case $g^{-t} = 7^{-10} = 7^{86}$

$7^{-10} \equiv 32 \pmod{97}$ $\quad 7^{86} \equiv 32 \pmod{97}$

since $7^{-1} \equiv 14 \pmod{97}$ $\quad 14^{10} \equiv 32 \pmod{97}$

	37	7	
R_1	1	0	37
R_2	0	1	7
$R_3 = R_1 - 13R_2$	1	-13	$6 = 37 - 13 \cdot 7$
$R_4 = R_2 - R_1 + 13R_2$ $-R_1 + 14R_2$	-1	(14)	$1 = 7 - 1 \cdot 6$

$7^{-1} \pmod{37} = 16$

i	0	1	2	3	4	5	6
$78 \cdot (7^{-10})^i \pmod{37}$	78	71	44	51	80	38	52

we found one of the baby steps

we can conclude:

$7^3 = 52 \pmod{37}$

$78 \cdot (7^{-10})^6 = 52 \pmod{37}$

$\Rightarrow 78 = 7^{3+6 \cdot 10} \pmod{37}$

so $v = 63$

$78 \cdot (7^{-10})^6 = 7^3 \pmod{37}$

$\frac{78}{7^{60}} = 7^3 \quad 78 = 7^3 \cdot 7^{60} \Rightarrow 78 = 7^{63}$

worst case $\sqrt{p}$ giant steps and $\lceil \sqrt{p} \rceil$ baby-steps.

In short

$g^v = y \pmod{p}$ find v

1) $t = \lceil \sqrt{p} \rceil$, do t baby steps of $g^i \pmod{p}$ $i \in \{0, \dots, t-1\}$
2) sort the results
3) do giant steps of g^t
   start with y and multiply it by g^{-t} until we find a number of the baby step list
4) solve the equation and find v

Pollard algorithm $g^v = y \pmod p$

This method consists of arranging a series of numbers $(x_i)_{i\in\mathbb{N}} \mid x_i = x_{2i}$ for some i then we can determine v

take $x_0 = 1$ $\quad x_{i+1} = \begin{cases} yx_i & \text{if } x_i \equiv 1 \pmod 3 \\ gx_i & \text{if } x_i \equiv 2 \pmod 3 \\ x_i^2 & \text{if } x_i \equiv 0 \pmod 3 \end{cases}$

we can now write x_i as a product of powers of y and g:

$x_i = g^{a_i} y^{b_i}$ where $(a^0, b^0) = (0,0)$ then the recursion becomes:

$$(a_{i+1}, b_{i+1}) = \begin{cases} (a_i, b_i+1) & \text{if } x_i \equiv 1 \pmod 3 \\ (a_i+1, b_i) & \text{if } x_i \equiv 2 \pmod 3 \\ (2a_i, 2b_i) & \text{if } x_i \equiv 0 \pmod 3 \end{cases}$$

we carry on until we encounter the first time that $x_i = x_{2i}$, then we have:

$$x_i \equiv x_{2i} \pmod p$$

$$g^{a_i} y^{b_i} \equiv g^{a_{2i}} y^{b_{2i}} \pmod p$$

because $y = g^v$

$$g^{a_i} g^{v\cdot b_i} \equiv g^{a_{2i}} g^{v\cdot b_{2i}} \pmod p$$

$$g^{a_i + v b_i} \equiv g^{a_{2i} + v b_{2i}} \pmod p$$

$$a_i + v b_i \equiv a_{2i} + v b_{2i} \pmod{p-1}$$

$$v \equiv (a_i - a_{2i})(b_{2i} - b_i)^{-1} \pmod{p-1}$$

we can only perform the last step if $b_{2i} - b_i$ is reversible $(\gcd(b_{2i} - b_i, p-1) = 1)$

if we cannot do the last step then we have to start over with a different $x^0 = g^{a_0} y^{b_0}$

The probabilty of not being able to finish the algorithm is circa $1 - \frac{\varphi(p-1)}{p-1}$ (euler phi function)

example $g=2 \quad y=228$

$2^u = 228 \pmod{383}$ find u

since $g=2$ is a generator of the group $\mathbb{F}_{383}^*$ of order 191 we can use it instead of mod 382

we'll take x_i as mod 383 and a_i, b_i mod 191

i	x_i	a_i	b_i	x_{2i}	a_{2i}	b_{2i}
1	228	0	1	279	0	2
2	279	0	2	184	1	4
3	92	0	4	14	1	6
4	184	1	4	256	2	7
5	205	1	5	304	3	8
6	14	1	6	121	6	18
7	28	2	6	144	12	38
8	256	2	7	235	48	152
9	152	2	8	72	48	154
10	304	3	8	14	96	118
11	372	3	9	256	97	119
12	121	6	18	304	98	120
13	12	6	19	121	5	51
14	**144**	**12**	**38**	**144**	**10**	**104**

$2^{110} \equiv 228 \pmod{383}$

$X_{14} = X_{28}$

$u = (a_{28} - a_{14})(b_{14} - b_{28})^{-1}$

$\equiv (-2)(-66)^{-1} \pmod{191}$

have to check if reversible

$\equiv (189)(125)^{-1} \pmod{191}$

$\equiv (189)(136) \pmod{191} = 110 \pmod{191}$

	191	125	
R_1	1	0	191
R_2	0	1	125
$R_3 = R_1 - R_2$	1	-1	$191 = 66 + 1 \cdot 125$
$R_4 = R_2 - R_3 = R_2 - (R_1 - R_2)$	-1	2	$125 = 59 + 1 \cdot 66$
$R_5 = R_3 - R_4 = 2R_1 - 3R_2$	2	-3	$66 = 7 + 1 \cdot 59$
$R_6 = R_4 - 8(R_5) = -17R_1 + 26R_2$	-17	+26	$59 = 3 + 8 \cdot 7$
$R_7 = R_5 - 2(R_6) =$	36	-55	$7 = 1 + 2 \cdot 3$
$= 2R_1 - 3R_2 - 2(-17R_1 + 26R_2) =$			$3 = 0 + 1 \cdot 3$
$= 2R_1 - 3R_2 + 34R_1 - 52R_2 =$			
$= 36R_1 - 55R_2$			

$R_3 = 66 = 191 - 125$

$R_4 = 59 = 125 - 1 \cdot 66$

$R_5 = 7 = 66 - 1 \cdot 59$

$R_6 = 3 = 59 - 8 \cdot 7$

$R_7 = 1 = 7 - 2 \cdot 3$

$125^{-1} \equiv -55 \pmod{191}$

125 is reversible! $\Rightarrow \equiv 136 \pmod{191}$

Index calculus algorithm

the currently fastest known DLP solving algorithm.

Like Pollard's it may take some tries

Suppose we want to find $v = \log_g y \pmod{p-1}$ where g is a producer of $\mathbb{Z}_p^*$.

With this algorithm we first take a mixture s of small prime numbers

Then we look for all the prime factors of $g^{k_i} \pmod p$ that also appear in s

if we have found enough k_i's we can go through a system to determine the discrete logarithm from s.

Now we look for a $k \mid y \cdot g^k \pmod p$ can be decomposed into prime factors from s

We can then use the logarithms of the elements of s to determine the v we are looking for

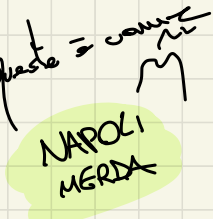
example

$p = 229 \quad g = 6 \quad y = 13 \quad 6$ is a generator of $\mathbb{Z}_{229}^*$

$6^v = 13 \pmod{229}$

- we take $S = \{2, 3, 5, 7, 11\}$.
- we take powers of g in such way that the results $(\text{mod } 229)$ can be factored into prime factors $2, 3, 5, 7, 11$

$$6^{100} \pmod{229} = 180 = 2^2 \cdot 3^2 \cdot 5$$
$$6^{18} \pmod{229} = 176 = 2^4 \cdot 11$$
$$6^{12} \pmod{229} = 165 = 3 \cdot 5 \cdot 11$$
$$6^{62} \pmod{229} = 154 = 2 \cdot 7 \cdot 11$$
$$6^{143} \pmod{229} = 198 = 2 \cdot 3^2 \cdot 11$$
$$6^{206} \pmod{229} = 210 = 2 \cdot 3 \cdot 5 \cdot 7$$



if we now convert the equations to equalities in the exponents (i.e we take $\log_6$ of both members), then we have the following equations:

$$100 \equiv 2\log_6 2 + 2\log_6 3 + \log_6 5 \pmod{228}$$
$$18 \equiv 4\log_6 2 + \log_6 11 \pmod{228}$$
$$12 \equiv \log_6 3 + \log_6 5 + \log_6 11 \pmod{228}$$
$$62 \equiv \log_6 2 + \log_6 7 + \log_6 11 \pmod{228}$$
$$143 \equiv \log_6 2 + 2\log_6 3 + \log_6 11 \pmod{228}$$
$$206 \equiv \log_6 2 + \log_6 3 + \log_6 5 + \log_6 7 \pmod{228}$$

we are working with $(\text{mod } 228)$ since we have a system in the powers

if we solve the equations we get:

$\log_6 2 = 21 \quad \log_6 3 = 208 \quad \log_6 5 = 38 \quad \log_6 7 = 107 \quad \log_6 11 = 162$

now we have to find a value K so that $y \cdot g^K \pmod{229}$ only has factors in S

with trial and error we find that for example $K=77$ we get:

$$y \cdot g^K \equiv 13 \cdot 6^{77} \equiv 147 \pmod{229}$$

since $147 = 3 \cdot 7^2$ we get:

$$13 \cdot 6^{77} \equiv 3 \cdot 7^2 \pmod{229}$$

$$\log_6 13 + \log_6 6^{77} \equiv \log_6 3 + 2\log_6 7 \pmod{228}$$

$$\log_6 13 = \log_6 3 + 2\log_6 7 - \log_6 6^{77} \pmod{228}$$

$$\log_6 13 \equiv 208 + 2 \cdot 107 - 77 = 345 \equiv 117 \pmod{228}$$

Proving the complexity of this algorithm is really hard, it's in polynomial time.

Pohlig and Hellman algorithm

if $p-1$ is the product of small prime numbers the DLP problem is easier to crack by adapting the index calculus algorithm. Here we use the chinese remainder theorem, wich provides solutions to systems of linear congruence equations.

Chinese remainder theorem

Let $m_1, m_2, \dots m_n$ and $b_1, b_2 \dots b_n$ given with $\gcd(m_i, m_j) = 1 \quad \forall i, j \in \{1, \dots n\}, i \neq j$

and let $M = \prod_{i=1}^{n} m_i$

there's a unique solution $x \pmod M$ of the system:

$$\begin{cases} x \equiv b_1 \pmod{m_1} \\ x \equiv b_2 \pmod{m_2} \\ \vdots \\ x \equiv b_n \pmod{m_n} \end{cases}$$

let $M_i = M/m_i$ and $y_i = M_i^{-1} \pmod{m_i}$ $\forall i \in \{1, \dots, n\}$ the solution then is:

$$x \equiv \sum_{i=1}^{n} b_i M_i y_i \pmod M$$

in Pohlig-Hellman the overall process looks like this:

$y = g^u \pmod p$

$\varphi(p) = p - 1 = p_1^{\alpha_1} \cdots p_K^{\alpha_K}$ — small prime num (simpler)

Pohlig-Hellmann

$u \pmod{p_1^{\alpha_1}}$ $\quad$ $u \pmod{p_K^{\alpha_K}}$

CRT

$u \pmod{\varphi(p)}$

suppose that $p-1$ can be written as $p_1^{\alpha_1} \cdots p_K^{\alpha_K}$ product of small prime factors.
First we look for the solution $u \pmod{p_i^{\alpha_i}}$ $\forall i$ and then we use them to determine the solutions u via the Chinese remainder theorem.

$y = g^u \pmod p$

$\downarrow$

$\varphi(p) = p - 1 = p_1^{\alpha_1} \cdot \dots \cdot p_K^{\alpha_K}$

$\downarrow$

find the solutions u for $\pmod{p_i^{\alpha_i}}$ $\forall i$ $i = 1, \dots K$

$\downarrow$

find $u \pmod{\varphi(p)}$ with CRT

example:

suppose we want to search for $v \mid 19^v \equiv 181 \pmod{1801}$

The solution v is given in modulo 1800 because $1800 = 2^3 \cdot 3^2 \cdot 5^2$

we first determine what $v \pmod{2^3}$, then $v \pmod{3^2}$ finally $v \pmod{5^2}$

$p_1^{\alpha_1} = 2^3$ we determine the following powers for both $a = 181$ and $a = 19$

$$a^{1800/2} \equiv a^{900}, \quad a^{1800/2^2} \equiv a^{450}, \quad a^{1800/2^3} \equiv a^{225} \pmod{1801}$$

we find:

$19^{900} \equiv -1$, $19^{450} \equiv 824$, $19^{225} \equiv -524 \pmod{1801}$

$181^{900} \equiv 1$, $181^{450} \equiv -1$, $181^{225} \equiv 824 \pmod{1801}$

we find from the 900th power that

$181 = 19^4 \pmod{1801}$

$$1 \equiv 181^{900} \equiv (19^v)^{900} \equiv (19^{900})^v \equiv (-1)^v \pmod{1801}$$

consequently we find that $v \equiv 0 \pmod 2$, we write that $v = 2x$

↳ we must obtain a positive number

$$-1 \equiv 181^{450} \equiv (19^v)^{450} \equiv (19^{2x})^{450} \equiv (19^{900})^x \equiv (-1)^x \pmod{1801}$$

and therefore $x \equiv 1 \pmod 2$, so $x = 2y+1$ so $v = 8y+2$ then:

$v \equiv 2 \pmod 8$ ↳ we must obtain a negative number

$p_2^{\alpha_2} = 3^2$ $a = 181$ and $a = 19$

$$a = a^{1800/3} = a^{600} \qquad a^{1800/3^2} = a^{200} \pmod{1801}$$

we find:

$$19^{600} \equiv 73 \qquad 19^{200} \equiv -12 \pmod{1801}$$

$$181^{600} \equiv 1 \qquad 181^{200} \equiv -74 \pmod{1801}$$

we proceed as usual:

$1 \equiv 181^{600} \equiv (19^v)^{600} \equiv 73^v \pmod{1801}$

since $73^v \equiv 1$, $73^1 \equiv 73$, $73^2 \equiv -74$, $73^3 \equiv 1$ we find from the previous equality that $v \equiv 0 \pmod 3$, so $v = 3x$, let's look at the exponent 200

$$-74 \equiv 181^{200} \equiv (19^v)^{200} \equiv (19^{600})^x \equiv (73)^x \pmod{1801}$$

so $x \equiv 2 \pmod 3$, so $x = 3y+2 \to v = 9y+6 \to v \equiv 6 \pmod 9$

$p_3^{\alpha_3} = 5^2$ in a similar way we find $v \equiv 1 \pmod{25}$

now we have the system

$$\begin{cases} v \equiv 2 \pmod 8 \\ v \equiv 6 \pmod 9 \\ v \equiv 1 \pmod{25} \end{cases}$$

the conditions of the CRT are satisfied and we can apply it, we get:

$M_1 = 9 \cdot 25 \equiv 225$ $\quad y_1 = 225^{-1} \equiv 1 \pmod 8$

$M_2 = 8 \cdot 25 = 200$ $\quad y_2 = 200^{-1} \equiv 5 \pmod 9$

$M_3 = 8 \cdot 9 = 72$ $\quad y_3 = 72^{-1} \equiv 8 \pmod{25}$

then we calculate v by:

$v = 2 \cdot 225 \cdot 1 + 6 \cdot 200 \cdot 5 + 1 \cdot 72 \cdot 8 = 7026 \equiv 1626 \pmod{1800}$

we can conclude that $19^{1626} \equiv 81 \pmod{1801}$

El Gamal

Asymmetric encryption. 2 keys are generated large random number $\xrightarrow{\text{program}}$

Private

Public

take

- p large prime number (so that DLP "uncrackable" in $\mathbb{F}_p^*$)
- g generator of the subgroup $\mathbb{F}_p^*$
- x_a private key
- $y_a = g^{x_a} \pmod p$ the public key
- m the message we want to send ($m<p$)

signing

we now want to add the digital signature. It consists of 2 numbers r and s

- choose k such that $\gcd(k, p-1)=1$ (to determine the inverse of k modulo $p-1$)
- calculate $\begin{cases} r = g^k \pmod p \\ s = (m - x_a r)k^{-1} \pmod{p-1} \end{cases}$
- sign m with r and s

checking signature

- must have public key y_a (p and g are also public)
- calculate $e_1 = y_a^r r^s \pmod p$ and $e_2 = g^m \pmod p$
- if $e_1 = e_2$ accept otherwise no.

In practice this will not be very usefull bcz r and s are in the same order of magnitude as p and m The signature is therefore twice the size of the message. We also have to work with a very large p if the message is large.

To fix this we use an hash function to convert m to a much smaller number.
$f: m \to H(m)$. r and s can be determined based on $H(m)$ instead of m, so that the signature is twice as long as $H(m)$ and not m.

Diffie Hellmann

Agree a secrect key togheter without sending it.

Alice and Bob want to agree on a secret key

General idea:

They both generate public and private key separately and exchange the public ones.
Alice combines her private key with Bob's public key, Bob does the same.
Even though they did different steps they will end up with the same key.

implementation

1) Alice and Bob agree on a common base g and a common p prime $(gcd(g, p-1)=1)$
2) Alice determines secrect key a and Bob b
3) Alice calculates $A=g^a \pmod p$ and sends it to Bob
4) Bob calculates $B=g^b \pmod p$ and sends it to Alice
5) Alice calculates $c_1=B^a \pmod p$, Bob calculates $c_2=A^b \pmod p$
   ↑ public other ∧ private mine
6) since

$$c_1 \equiv B^a \equiv (g^b)^a \equiv g^{ba} \pmod p$$

and

$$c_2 \equiv A^b \equiv (g^a)^b \equiv g^{ab} \pmod p$$

$c_1 = c_2$, they agreed on a key

Alice		Bob
generates p, g and a		
calculates $A=g^a \pmod p$	$\xrightarrow{p, g, a}$	generates b
	$\xleftarrow{B}$	calculates $B=g^b \pmod p$
$c=B^a \pmod p$		$c=A^b \pmod p$