Криптографија
- шести део -

доц. др Драган Ђокић

Математички факултет, Универзитет у Београду

dragan.djokic@matf.bg.ac.rs

5. април 2024.

Проблем дискретног логаритма

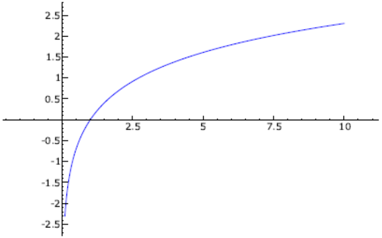
Дефиниција

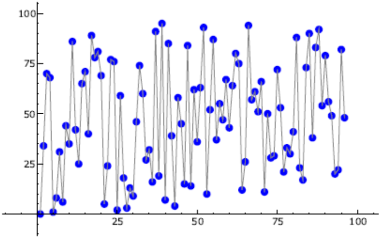
Нека је G група нпр. $\mathbb{F}_q^*$, и нека су $a, g \in G$. Најмањи природан број n (ако постоји) такав да је $a = g^n$ зовемо дискретни логаритам од a у основи g и означавамо са $\log_g a$.

Проблем дискретног логаритма

Дефиниција

Нека је G група нпр. $\mathbb{F}_q^*$, и нека су $a, g \in G$. Најмањи природан број n (ако постоји) такав да је $a = g^n$ зовемо дискретни логаритам од a у основи g и означавамо са $\log_g a$.





Класичан логаритам и дискретни log_2 у групи $\mathbb{Z}_{53}^*$ (други делује непредвидиво)

- Немамо формулу за израчунавање $n = \log_g a$ у $\mathbb{F}_q^*$

- Немамо формулу за израчунавање $n = \log_g a$ у $\mathbb{F}_q^*$
- Пример: $\log_2 6 =?$ у групи $\mathbb{Z}_{23}^*$
  (рачунамо 2, 2^2, 2^3, $\ldots$ и чекамо да се појави 6)

n	0	1	2	3	4	5	6	7	8	9	10	11
$2^n (mod\ 23)$	1	2	4	8	16	9	18	13	3	6	12	1

n	12	13	14	15	16	17	18	19	20	21	22	
$2^n (mod\ 23)$	2	4	8	16	9	18	13	3	6	12	1	

Добићемо да је $\log_2 6 = 20$ али траје предуго

- Немамо формулу за израчунавање $n = \log_g a$ у $\mathbb{F}_q^*$
- Пример: $\log_2 6 =?$ у групи $\mathbb{Z}_{23}^*$
(рачунамо 2, 2^2, 2^3, $\ldots$ и чекамо да се појави 6)

n	0	1	2	3	4	5	6	7	8	9	10	11
$2^n (mod\ 23)$	1	2	4	8	16	9	18	13	3	6	12	1

n	12	13	14	15	16	17	18	19	20	21	22	
$2^n (mod\ 23)$	2	4	8	16	9	18	13	3	6	12	1	

Добићемо да је $\log_2 6 = 20$ али траје предуго

- Најгори случај за $n = \log_g a$ у $\mathbb{F}_q^*$: када је g генератор n велико - практично прођемо целу горњу табелу

- Немамо формулу за израчунавање $n = \log_g a$ у $\mathbb{F}_q^*$
- Пример: $\log_2 6 =?$ у групи $\mathbb{Z}_{23}^*$
  (рачунамо 2, 2^2, 2^3, ... и чекамо да се појави 6)

n	0	1	2	3	4	5	6	7	8	9	10	11
$2^n (mod\ 23)$	1	2	4	8	16	9	18	13	3	6	12	1

n	12	13	14	15	16	17	18	19	20	21	22	
$2^n (mod\ 23)$	2	4	8	16	9	18	13	3	6	12	1	

Добићемо да је $\log_2 6 = 20$ али траје предуго

- Најгори случај за $n = \log_g a$ у $\mathbb{F}_q^*$: када је g генератор n велико - практично прођемо целу горњу табелу
  - То је $O(q)$ тестирања! (А временска сложеност степеновања је $O\left(\log^4 q\right)$)

Степен и дискретни логаритам у $\mathbb{F}_q$ и (њихово време израчунавања) помоћу Sage-а (који има на располагању све познате алгоритме)

Save Copy Run SageMath 10.3

```
K = GF(39916801^4,'x')
g = K.multiplicative_generator()
%time a=pow(g, 1234567891011121314151617181920, K.order())
%time b=discrete_log(a, g, K.order() - 1)
is_prime(39916801), g, a, b
```

```
CPU times: user 55 µs, sys: 0 ns, total: 55 µs
Wall time: 58.9 µs
CPU times: user 364 ms, sys: 3.76 ms, total: 367 ms
Wall time: 393 ms

(True,
 x + 3,
 12508294*x^3 + 18207805*x^2 + 38793833*x + 2187988,
 1234567891011121314151617181920)
```

$GF(q,'x')$ је коначно поље са $q = p^d$ елемененаta, други аргумент је симбол којим желимо да Sage означи променљиву x у $\mathbb{Z}_p[x]$

- Не постоји довољно брз алгоритам који решава проблем дискретног логаритма у $\mathbb{F}_q^*$, тј. алгоритам чија је брзина упоредива са степеновањем

- Не постоји довољно брз алгоритам који решава проблем дискретног логаритма у $\mathbb{F}_q^*$, тј. алгоритам чија је брзина упоредива са степеновањем
- Постоје алгоритми који могу да смање време претраживања $O(q)$ из претх. примера
  - Такав је алгоритам Гељфонд-Шенкса временске сложености $O\left(\sqrt{q}\log^2 q\right)$

- Не постоји довољно брз алгоритам који решава проблем дискретног логаритма у $\mathbb{F}_q^*$, тј. алгоритам чија је брзина упоредива са степеновањем
- Постоје алгоритми који могу да смање време претраживања $O(q)$ из претх. примера
  - Такав је алгоритам Гељфонд-Шенкса временске сложености $O\left(\sqrt{q}\log^2 q\right)$
- Постоје и алгоритми који раде довољно ефикасно за неке специфичне q-ове
  - Такав је Полиг-Хелманов алгоритам који предпоставља да су сви прости чиниоци броја $q-1$ „мали“
  - Да ли Дифи-Хелман може да се имплементира тако да не бира такве q-ове?

Алгоритам Гељфонд-Шенкса (baby-step-giant-step алгоритам)

- Алгоритам имитира идеју „сусрета на пола пута“

Алгоритам Гељфонд-Шенкса (baby-step-giant-step алгоритам)

- Алгоритам имитира идеју „сусрета на пола пута"
- Циљ: израчунати $n = \log_g a$ у $\mathbb{F}_q^*$, где су a и g познати

Алгоритам Гељфонд-Шенкса (baby-step-giant-step алгоритам)

- Алгоритам имитира идеју „сусрета на пола пута“
- Циљ: израчунати $n = \log_g a$ у $\mathbb{F}_q^*$, где су a и g познати
- Запишемо $n = mi + j$, где је $m = [\sqrt{q}]$ (цео део), $0 \leqslant i \leqslant m$, $0 \leqslant j \leqslant m - 1$

Алгоритам Гељфонд-Шенкса (baby-step-giant-step алгоритам)

- Алгоритам имитира идеју „сусрета на пола пута“
- Циљ: израчунати $n = \log_g a$ у $\mathbb{F}_q^*$, где су a и g познати
- Запишемо $n = mi + j$, где је $m = [\sqrt{q}]$ (цео део), $0 \leqslant i \leqslant m$, $0 \leqslant j \leqslant m - 1$
- Тада је $g^j = a\left(g^{-m}\right)^i$

Алгоритам Гељфонд-Шенкса (baby-step-giant-step алгоритам)

- Алгоритам имитира идеју „сусрета на пола пута“
- Циљ: израчунати $n = \log_g a$ у $\mathbb{F}_q^*$, где су a и g познати
- Запишемо $n = mi + j$, где је $m = [\sqrt{q}]$ (цео део), $0 \leqslant i \leqslant m$, $0 \leqslant j \leqslant m - 1$
- Тада је $g^j = a\left(g^{-m}\right)^i$
- Рачунамо парове $\left(j, g^j\right)$ и $\left(i, a\left(g^{-m}\right)^i\right)$, за све i, j и чувамо их сортиране по другој координати

Алгоритам Гељфонд-Шенкса (baby-step-giant-step алгоритам)

- Алгоритам имитира идеју ,,сусрета на пола пута"
- Циљ: израчунати $n = \log_g a$ у $\mathbb{F}_q^*$, где су a и g познати
- Запишемо $n = mi + j$, где је $m = [\sqrt{q}]$ (цео део), $0 \leqslant i \leqslant m$, $0 \leqslant j \leqslant m - 1$
- Тада је $g^j = a\left(g^{-m}\right)^i$
- Рачунамо парове $\left(j, g^j\right)$ и $\left(i, a\left(g^{-m}\right)^i\right)$, за све i, j и чувамо их сортиране по другој координати
- Када нађемо i и j за које се поклапа друга координата лако долазимо до n

Алгоритам Гељфонд-Шенкса (baby-step-giant-step алгоритам)

- Алгоритам имитира идеју ,,сусрета на пола пута"
- Циљ: израчунати $n = \log_g a$ у $\mathbb{F}_q^*$, где су a и g познати
- Запишемо $n = mi + j$, где је $m = [\sqrt{q}]$ (цео део), $0 \leqslant i \leqslant m$, $0 \leqslant j \leqslant m - 1$
- Тада је $g^j = a\left(g^{-m}\right)^i$
- Рачунамо парове $\left(j, g^j\right)$ и $\left(i, a\left(g^{-m}\right)^i\right)$, за све i, j и чувамо их сортиране по другој координати
- Када нађемо i и j за које се поклапа друга координата лако долазимо до n
- Временска (али и просторна) сложеност $O\left(\sqrt{q}\log^2 q\right)$

Пример: $\log_3 37$ по модулу 101

- тада je $\left[\sqrt{101}\right] = 10$ и

giant step i	0	1	2	3	4	5	6	7	8	9
$3^{-10i} \pmod{101}$	1	14	95	17	36	100	87	6	84	65
$37 \cdot 3^{-10i} \pmod{101}$	37	13	81	23	19	64	88	20	78	82

- 3, 3^2 и 3^3 се не појављују у табели, али $3^4 = 81 \equiv 37 \cdot 3^{-20} \pmod{101}$
- следи $\log_3 37 = 20 + 4$

Полиг-Хелманов алгоритам

Дефиниција

Нека су $N, B \in \mathbb{N}$. За број $N = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ кажемо да је B-гладак ако важи $p_i^{\alpha_i} \leqslant B$, за све $1 \leqslant i \leqslant k$

Полиг-Хелманов алгоритам

Дефиниција

Нека су $N, B \in \mathbb{N}$. За број $N = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ кажемо да је B-гладак ако важи $p_i^{\alpha_i} \leqslant B$, за све $1 \leqslant i \leqslant k$

- Пример: $70 = 2 \cdot 5 \cdot 7$ је 7-гладак, али $150 = 2 \cdot 3 \cdot 5^2$ није 7-гладак

Полиг-Хелманов алгоритам

Дефиниција

Нека су $N, B \in \mathbb{N}$. За број $N = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ кажемо да је B-гладак ако важи $p_i^{\alpha_i} \leqslant B$, за све $1 \leqslant i \leqslant k$

- Пример: $70 = 2 \cdot 5 \cdot 7$ је 7-гладак, али $150 = 2 \cdot 3 \cdot 5^2$ није 7-гладак
- Полиг-Хелманов метод омогућава (брзо) израчунавање дискретног логаритма у $\mathbb{F}_q^*$. Претпоставке су
  - $B \ll n$ нпр. $B \sim \log n$
  - $q - 1$ је B-гладак

Полиг-Хелманов алгоритам

Дефиниција

Нека су $N, B \in \mathbb{N}$. За број $N = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ кажемо да је B-гладак ако важи $p_i^{\alpha_i} \leqslant B$, за све $1 \leqslant i \leqslant k$

- Пример: $70 = 2 \cdot 5 \cdot 7$ је 7-гладак, али $150 = 2 \cdot 3 \cdot 5^2$ није 7-гладак
- Полиг-Хелманов метод омогућава (брзо) израчунавање дискретног логаритма у $\mathbb{F}_q^*$. Претпоставке су
  - $B \ll n$ нпр. $B \sim \log n$
  - $q - 1$ је B-гладак
- Циљ: израчунати n за које је $g^n = a$ у $\mathbb{F}_q^*$, где су a и g познати

Полиг-Хелманов алгоритам

Дефиниција

Нека су $N, B \in \mathbb{N}$. За број $N = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ кажемо да је B-гладак ако важи $p_i^{\alpha_i} \leqslant B$, за све $1 \leqslant i \leqslant k$

- Пример: $70 = 2 \cdot 5 \cdot 7$ је 7-гладак, али $150 = 2 \cdot 3 \cdot 5^2$ није 7-гладак
- Полиг-Хелманов метод омогућава (брзо) израчунавање дискретног логаритма у $\mathbb{F}_q^*$. Претпоставке су
  - $B \ll n$ нпр. $B \sim \log n$
  - $q - 1$ је B-гладак
- Циљ: израчунати n за које је $g^n = a$ у $\mathbb{F}_q^*$, где су a и g познати
  - МФТ $g^{q-1} = 1$, па n треба тражити као остатак по модулу $q - 1$

Полиг-Хелманов алгоритам

Дефиниција

Нека су $N, B \in \mathbb{N}$. За број $N = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ кажемо да је B-гладак ако важи $p_i^{\alpha_i} \leqslant B$, за све $1 \leqslant i \leqslant k$

- Пример: $70 = 2 \cdot 5 \cdot 7$ је 7-гладак, али $150 = 2 \cdot 3 \cdot 5^2$ није 7-гладак
- Полиг-Хелманов метод омогућава (брзо) израчунавање дискретног логаритма у $\mathbb{F}_q^*$. Претпоставке су
  - $B \ll n$ нпр. $B \sim \log n$
  - $q - 1$ је B-гладак
- Циљ: израчунати n за које је $g^n = a$ у $\mathbb{F}_q^*$, где су a и g познати
  - МФТ $g^{q-1} = 1$, па n треба тражити као остатак по модулу $q - 1$
  - ако је $q - 1 = p_1^{\alpha_1} \dots p_k^{\alpha_k}$, по Кинеској теореми о остацима довољно је одредити $n \equiv ? \pmod{p_i^{\alpha_i}}$ (и изоставићемо индекс i ради једноставности)

$n \equiv ? \pmod{p^\alpha}$

- Израчунамо $\zeta_p = g^{\frac{q-1}{p}}$, а затим и $1 = \zeta_p^0, \zeta_p, \zeta_p^2, \ldots, \zeta_p^{p-1}$ и чувамо (у некој табели) парове $\left(j, \zeta_p^j\right)$
  - све ζ_p^j зовемо p-ти корени из 1 (јер су решења једначине $x^p \equiv 1 \pmod{q-1}$)

$n \equiv ? \pmod{p^\alpha}$

- Израчунамо $\zeta_p = g^{\frac{q-1}{p}}$, а затим и $1 = \zeta_p^0, \zeta_p, \zeta_p^2, \ldots, \zeta_p^{p-1}$ и чувамо (у некој табели) парове $\left(j, \zeta_p^j\right)$
  - све ζ_p^j зовемо p-ти корени из 1 (јер су решења једначине $x^p \equiv 1 \pmod{q-1}$)
- Запишемо $n \equiv n_0 + n_1 p + n_2 p^2 + \cdots + n_{\alpha-1} p^{\alpha-1} \pmod{p^\alpha}$ (запис у основи p, евентуално су водеће цифре изгубљене или је допуњено нулама)
  - треба одредити n_i-ове

$n \equiv ? \pmod{p^\alpha}$

- Израчунамо $\zeta_p = g^{\frac{q-1}{p}}$, а затим и $1 = \zeta_p^0, \zeta_p, \zeta_p^2, \ldots, \zeta_p^{p-1}$ и чувамо (у некој табели) парове $\left(j, \zeta_p^j\right)$
  - све ζ_p^j зовемо p-ти корени из 1 (јер су решења једначине $x^p \equiv 1 \pmod{q-1}$)
- Запишемо $n \equiv n_0 + n_1 p + n_2 p^2 + \cdots + n_{\alpha-1} p^{\alpha-1} \pmod{p^\alpha}$ (запис у основи p, евентуално су водеће цифре изгубљене или је допуњено нулама)
  - треба одредити n_i-ове
- Можемо израчунати $a^{\frac{q-1}{p}}$, али $a^{\frac{q-1}{p}} = g^{\frac{n(q-1)}{p}} = \zeta_p^n = \zeta_p^{n_0}$. Како у табели имамо све вредности ζ_p^j (и све су различите) лако ћемо пронаћи вредност n_0

$n \equiv ? \pmod{p^\alpha}$

- Израчунамо $\zeta_p = g^{\frac{q-1}{p}}$, а затим и $1 = \zeta_p^0, \zeta_p, \zeta_p^2, \ldots, \zeta_p^{p-1}$ и чувамо (у некој табели) парове $\left(j, \zeta_p^j\right)$
  - све ζ_p^j зовемо p-ти корени из 1 (јер су решења једначине $x^p \equiv 1 \pmod{q-1}$)
- Запишемо $n \equiv n_0 + n_1 p + n_2 p^2 + \cdots + n_{\alpha-1} p^{\alpha-1} \pmod{p^\alpha}$ (запис у основи p, евентуално су водеће цифре изгубљене или је допуњено нулама)
  - треба одредити n_i-ове
- Можемо израчунати $a^{\frac{q-1}{p}}$, али $a^{\frac{q-1}{p}} = g^{\frac{n(q-1)}{p}} = \zeta_p^n = \zeta_p^{n_0}$. Како у табели имамо све вредности ζ_p^j (и све су различите) лако ћемо пронаћи вредност n_0
- Слично, можемо израчунати $\left(\frac{a}{g^{n_0}}\right)^{\frac{q-1}{p^2}}$, али $\left(\frac{a}{g^{n_0}}\right)^{\frac{q-1}{p^2}} = g^{\frac{(n-n_0)(q-1)}{p^2}} = \zeta_p^{\frac{n-n_0}{p}} = \zeta_p^{n_1}$ даје n_1

$n \equiv ? \pmod{p^\alpha}$

- Израчунамо $\zeta_p = g^{\frac{q-1}{p}}$, а затим и $1 = \zeta_p^0, \zeta_p, \zeta_p^2, \ldots, \zeta_p^{p-1}$ и чувамо (у некој табели) парове $\left(j, \zeta_p^j\right)$
  - све ζ_p^j зовемо p-ти корени из 1 (јер су решења једначине $x^p \equiv 1 \pmod{q-1}$)
- Запишемо $n \equiv n_0 + n_1 p + n_2 p^2 + \cdots + n_{\alpha-1} p^{\alpha-1} \pmod{p^\alpha}$ (запис у основи p, евентуално су водеће цифре изгубљене или је допуњено нулама)
  - треба одредити n_i-ове
- Можемо израчунати $a^{\frac{q-1}{p}}$, али $a^{\frac{q-1}{p}} = g^{\frac{n(q-1)}{p}} = \zeta_p^n = \zeta_p^{n_0}$. Како у табели имамо све вредности ζ_p^j (и све су различите) лако ћемо пронаћи вредност n_0
- Слично, можемо израчунати $\left(\frac{a}{g^{n_0}}\right)^{\frac{q-1}{p^2}}$, али $\left(\frac{a}{g^{n_0}}\right)^{\frac{q-1}{p^2}} = g^{\frac{(n-n_0)(q-1)}{p^2}} = \zeta_p^{\frac{n-n_0}{p}} = \zeta_p^{n_1}$ даје n_1
- $\left(\frac{a}{g^{n_0+n_1 p}}\right)^{\frac{q-1}{p^3}}$ даје n_2, $\left(\frac{a}{g^{n_0+n_1 p+n_2 p^2}}\right)^{\frac{q-1}{p^4}}$ даје n_3, ...

Пример: Израчунати $\log_3 304$ у $\mathbb{Z}_{401}^*$

$3^n \equiv 304 \pmod{401}$ (3 је генератор $\mathbb{Z}_{401}^*$)

$401-1 = 2^4 \cdot 5^2$: $n \equiv ?\ (400) \iff n \equiv ? \pmod{16}$ и $n \equiv ? \pmod{25}$

$n \equiv n_0 + 2n_1 + 4n_2 + 8n_3 \pmod{16}$ $n_i \in \{0, 1\}$

$\zeta_2 = 3^{\frac{401-1}{2}} \equiv 400 \pmod{401}$ (лако се види да је $-1 \equiv 400$ друго решење $x^2 \equiv 1 \pmod{401}$)

j	0	1
ζ_2^j	1	400

$304^{\frac{401-1}{2}} \equiv 400 \pmod{401} \Rightarrow n_0 = 1$

$\left(\frac{304}{3^1}\right)^{\frac{401-1}{4}} \equiv (304 \cdot 134)^{100} \equiv 400 \pmod{401} \Rightarrow n_1 = 1$

$\left(\frac{304}{3^{1+2\cdot 1}}\right)^{\frac{401-1}{8}} \equiv 338^{50} \equiv 1 \pmod{401} \Rightarrow n_2 = 0$

$\left(\frac{304}{3^{1+2\cdot 1+4\cdot 0}}\right)^{\frac{401-1}{16}} \equiv 338^{25} \equiv 400 \pmod{401} \Rightarrow n_3 = 1$

$n \equiv 1 + 2\cdot 1 + 4\cdot 0 + 8\cdot 1 \equiv 11 \pmod{16}$

$3^n \equiv 304 \pmod{401}$ (3 je генератор $\mathbb{Z}_{401}^*$)

$401-1 = 2^4 \cdot 5^2$: $n \equiv ?\ (400) \Leftrightarrow n \equiv ? \pmod{16}$ и $n \equiv ? \pmod{25}$

$n \equiv n_0 + 5n_1 \pmod{25}$

$\zeta_5 = 3^{\frac{401-1}{5}} \equiv 72 \pmod{401}$

j	0	1	2	3	4
ζ_5^j	1	72	372	318	39

$304^{\frac{401-1}{5}} \equiv 372 \pmod{401} \Rightarrow n_0 = 2$

$\left(\frac{304}{3^2}\right)^{\frac{401-1}{25}} \equiv 212^{16} \equiv 318 \pmod{401} \Rightarrow n_1 = 3$

$n \equiv 2 + 5 \cdot 3 \equiv 17 \pmod{25}$

$n = 25k + 17 \equiv 11 \pmod{16}$

$3k \equiv 10 \pmod{16}$

$k \equiv 14 \pmod{16}$

$n \equiv 25 \cdot 14 + 17 \equiv 267 \pmod{400}$

$\log_3 304 = 267$

- Приметимо да опет нема израчунавања дискретног логаритма, већ се траже вредности у табели! Још сад додатно чувамо табелу (трошимо меморију)
  - Али дужина табеле је мања од B

- Приметимо да опет нема израчунавања дискретног логаритма, већ се траже вредности у табели! Још сад додатно чувамо табелу (трошимо меморију)
  - Али дужина табеле је мања од B
- Затим за сваки p имамо α пута понављање рачуна, и на крају Кинеску теорему

- Приметимо да опет нема израчунавања дискретног логаритма, већ се траже вредности у табели! Још сад додатно чувамо табелу (трошимо меморију)
  - Али дужина табеле је мања од B
- Затим за сваки p имамо α пута понављање рачуна, и на крају Кинеску теорему
- Временска сложеност свега је полином од B, што је прихватљиво за мало B

- Приметимо да опет нема израчунавања дискретног логаритма, већ се траже вредности у табели! Још сад додатно чувамо табелу (трошимо меморију)
  - Али дужина табеле је мања од B
- Затим за сваки p имамо α пута понављање рачуна, и на крају Кинеску теорему
- Временска сложеност свега је полином од B, што је прихватљиво за мало B
- Опасност: Уколико Алиса и Бобан изаберу јавни кључ q за тд. $q-1$ је B-гладак, за довољно мало B, дискретни логаритам је решив - Цица може да декриптује поруку

- Приметимо да опет нема израчунавања дискретног логаритма, већ се траже вредности у табели! Још сад додатно чувамо табелу (трошимо меморију)
  - Али дужина табеле је мања од B
- Затим за сваки p имамо α пута понављање рачуна, и на крају Кинеску теорему
- Временска сложеност свега је полином од B, што је прихватљиво за мало B
- Опасност: Уколико Алиса и Бобан изаберу јавни кључ q за тд. $q-1$ је B-гладак, за довољно мало B, дискретни логаритам је решив - Цица може да декриптује поруку
- Алиса и Бобан могу да тестирају B-глаткост за неко мало B да би се провера обавила у реалном времену (а самим тим покривају и све $B' \leqslant B$)
  - Али: увек постоји опасност од $(B+1)$-глаткости што они неће приметити, а временска сложеност је практично иста као за B

Меси-Омура криптосистем

- Користи се за размену кључева или порука. Ако је дужа порука дели се на блокове

Меси-Омура криптосистем

- Користи се за размену кључева или порука. Ако је дужа порука дели се на блокове
- Алгоритам:
  - Фиксира се коначно поље $\mathbb{F}_q$ и то је свима познато (q је јавни кључ)

Меси-Омура криптосистем

- Користи се за размену кључева или порука. Ако је дужа порука дели се на блокове
- Алгоритам:
  - Фиксира се коначно поље $\mathbb{F}_q$ и то је свима познато (q је јавни кључ)
  - И Алиса и Бобан бирају своје тајне кључеве e_A и e_B тд. НЗД $(e_A, q-1) =$ НЗД $(e_B, q-1) = 1$

Меси-Омура криптосистем

- Користи се за размену кључева или порука. Ако је дужа порука дели се на блокове
- Алгоритам:
  - Фиксира се коначно поље $\mathbb{F}_q$ и то је свима познато (q је јавни кључ)
  - И Алиса и Бобан бирају своје тајне кључеве e_A и e_B тд. НЗД $(e_A, q-1) =$ НЗД $(e_B, q-1) = 1$
  - Свако од њих рачуна свој тајни кључ $d_i \equiv e_i^{-1} \pmod{q-1}$, $i \in \{A, B\}$

Меси-Омура криптосистем

- Користи се за размену кључева или порука. Ако је дужа порука дели се на блокове
- Алгоритам:
  - Фиксира се коначно поље $\mathbb{F}_q$ и то је свима познато (q је јавни кључ)
  - И Алиса и Бобан бирају своје тајне кључеве e_A и e_B тд. НЗД $(e_A, q-1) =$ НЗД $(e_B, q-1) = 1$
  - Свако од њих рачуна свој тајни кључ $d_i \equiv e_i^{-1} \pmod{q-1}$, $i \in \{A, B\}$
  - Ако је M блок поруке (кодиран елементом поља $\mathbb{F}_q$) коју треба послати Алиса рачуна M^{e_A} и шаље Бобану

Меси-Омура криптосистем

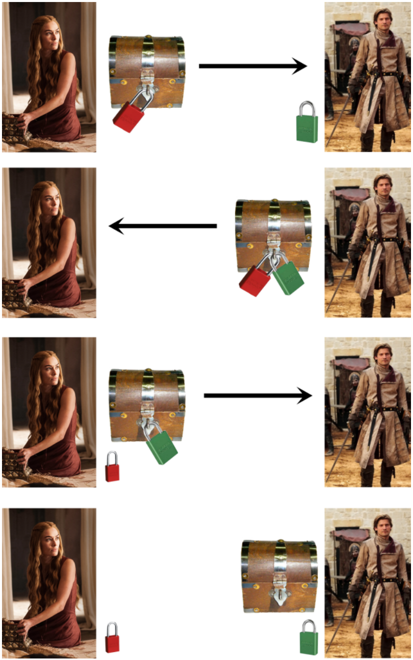
- Користи се за размену кључева или порука. Ако је дужа порука дели се на блокове
- Алгоритам:
  - Фиксира се коначно поље $\mathbb{F}_q$ и то је свима познато (q је јавни кључ)
  - И Алиса и Бобан бирају своје тајне кључеве e_A и e_B тд. НЗД $(e_A, q-1) =$ НЗД $(e_B, q-1) = 1$
  - Свако од њих рачуна свој тајни кључ $d_i \equiv e_i^{-1} \pmod{q-1}$, $i \in \{A, B\}$
  - Ако је M блок поруке (кодиран елементом поља $\mathbb{F}_q$) коју треба послати Алиса рачуна M^{e_A} и шаље Бобану
  - Бобан не може да прочита M^{e_A}, али може да израчуна $(M^{e_A})^{e_B} = M^{e_A e_B}$ и пошаље назад Алиси

Меси-Омура криптосистем

- Користи се за размену кључева или порука. Ако је дужа порука дели се на блокове
- Алгоритам:
  - Фиксира се коначно поље $\mathbb{F}_q$ и то је свима познато (q је јавни кључ)
  - И Алиса и Бобан бирају своје тајне кључеве e_A и e_B тд. НЗД$\,(e_A, q-1) =$ НЗД$\,(e_B, q-1) = 1$
  - Свако од њих рачуна свој тајни кључ $d_i \equiv e_i^{-1} \pmod{q-1}$, $i \in \{A, B\}$
  - Ако је M блок поруке (кодиран елементом поља $\mathbb{F}_q$) коју треба послати Алиса рачуна M^{e_A} и шаље Бобану
  - Бобан не може да прочита M^{e_A}, али може да израчуна $(M^{e_A})^{e_B} = M^{e_A e_B}$ и пошаље назад Алиси
  - Сада Алиса рачуна $(M^{e_A e_B})^{d_A} = M^{e_B}$ и шаље опет Бобану. Овде се користи $e_A d_A \equiv 1 \pmod{q-1}$ што повлачи $M^{e_A d_A} = M$ у $\mathbb{F}_q$

Меси-Омура криптосистем

- Користи се за размену кључева или порука. Ако је дужа порука дели се на блокове
- Алгоритам:
  - Фиксира се коначно поље $\mathbb{F}_q$ и то је свима познато (q је јавни кључ)
  - И Алиса и Бобан бирају своје тајне кључеве e_A и e_B тд. НЗД $(e_A, q-1) =$ НЗД $(e_B, q-1) = 1$
  - Свако од њих рачуна свој тајни кључ $d_i \equiv e_i^{-1} \pmod{q-1}$, $i \in \{A, B\}$
  - Ако је M блок поруке (кодиран елементом поља $\mathbb{F}_q$) коју треба послати Алиса рачуна M^{e_A} и шаље Бобану
  - Бобан не може да прочита M^{e_A}, али може да израчуна $(M^{e_A})^{e_B} = M^{e_A e_B}$ и пошаље назад Алиси
  - Сада Алиса рачуна $(M^{e_A e_B})^{d_A} = M^{e_B}$ и шаље опет Бобану. Овде се користи $e_A d_A \equiv 1 \pmod{q-1}$ што повлачи $M^{e_A d_A} = M$ у $\mathbb{F}_q$
  - Бобан рачуна $(M^{e_B})^{d_B} = M$ и долази до почетне поруке.



- Ако Цица пресретне комуникацију највише што може да зна је M^{e_A}, M^{e_B} и $M^{e_A e_B}$ и јавни кључ q.

- Ако Цица пресретне комуникацију највише што може да зна је M^{e_A}, M^{e_B} и $M^{e_A e_B}$ и јавни кључ q.
- Да би дошла до информације M мора да израчуна:
  - $e_A = \log_{M^{e_B}}(M^{e_A e_B})$ дискретни логаритам
  - $d_A \equiv e_A^{-1} \pmod{q-1}$
  - $M = (M^{e_A})^{d_A}$

- Ако Цица пресретне комуникацију највише што може да зна је M^{e_A}, M^{e_B} и $M^{e_A e_B}$ и јавни кључ q.
- Да би дошла до информације M мора да израчуна:
  - $e_A = \log_{M^{e_B}} \left(M^{e_A e_B}\right)$ дискретни логаритам
  - $d_A \equiv e_A^{-1} \pmod{q-1}$
  - $M = (M^{e_A})^{d_A}$
- Као додатна заштита: кључеви e_A, e_B, d_A и d_B се могу мењати код сваког блока.
  - Зато се понекад каже да Меси-Омура није ни симетричан ни асиметричан (јер се сматра да има само један јавни кључ q, а остало су параметри који се једнократно генеришу)

- Показаћемо како се рачуна мултипликативни инверз a^{-1} по модулу b
  - Еуклидов алгоритам (у $\mathbb{N}$) даје $ma + nb = 1 =$ НЗД(a, b) за неке $m, n \in \mathbb{Z}$
  - Следи $ma \equiv 1 \pmod{b}$, тј. $a^{-1} \equiv m \pmod{b}$

- Показаћемо како се рачуна мултипликативни инверз a^{-1} по модулу b
  - Еуклидов алгоритам (у $\mathbb{N}$) даје $ma + nb = 1 =$ НЗД(a, b) за неке $m, n \in \mathbb{Z}$
  - Следи $ma \equiv 1 \pmod{b}$, тј. $a^{-1} \equiv m \pmod{b}$
- Други начин (само за прост модул p): $a^{-1} \equiv a^{p-2} \pmod{p}$

- Показаћемо како се рачуна мултипликативни инверз a^{-1} по модулу b
  - Еуклидов алгоритам (у $\mathbb{N}$) даје $ma + nb = 1 =$ НЗД(a, b) за неке $m, n \in \mathbb{Z}$
  - Следи $ma \equiv 1 \pmod{b}$, тј. $a^{-1} \equiv m \pmod{b}$
- Други начин (само за прост модул p): $a^{-1} \equiv a^{p-2} \pmod{p}$
- Оба алгоритма раде и у $\mathbb{F}_q$ и имају исту временску сложеност $O\left(\log^3 q\right)$

Пример: $421^{-1} \equiv 281 \pmod{676}$

$$
\begin{aligned}
676 &= 1 \cdot 421 + 255 \\
421 &= 1 \cdot 255 + 166 \\
255 &= 1 \cdot 166 + 89 \\
166 &= 1 \cdot 89 + 77 \\
89 &= 1 \cdot 77 + 12 \\
77 &= 6 \cdot 12 + 5 \\
12 &= 2 \cdot 5 + 2 \\
5 &= 2 \cdot 2 + 1
\end{aligned}
$$

$$
\begin{aligned}
1 &= 5 - 2 \cdot (12 - 2 \cdot 5) \\
&= -2 \cdot 12 + 5 \cdot (77 - 6 \cdot 12) \\
&= 5 \cdot 77 - 32 \cdot (89 - 77) \\
&= -32 \cdot 89 + 37 \cdot (166 - 89) \\
&= 37 \cdot 166 - 69 \cdot (255 - 166) \\
&= -69 \cdot 255 + 106 \cdot (421 - 255) \\
&= 106 \cdot 421 - 175 \cdot (676 - 421) \\
&= -175 \cdot 676 + \underline{281} \cdot 421
\end{aligned}
$$

Пример: $421^{-1} \equiv 281 \pmod{676}$

$$\begin{aligned}
676 &= 1 \cdot 421 + 255 \\
421 &= 1 \cdot 255 + 166 \\
255 &= 1 \cdot 166 + 89 \\
166 &= 1 \cdot 89 + 77 \\
89 &= 1 \cdot 77 + 12 \\
77 &= 6 \cdot 12 + 5 \\
12 &= 2 \cdot 5 + 2 \\
5 &= 2 \cdot 2 + 1
\end{aligned}$$

$$\begin{aligned}
1 &= 5 - 2 \cdot (12 - 2 \cdot 5) \\
&= -2 \cdot 12 + 5 \cdot (77 - 6 \cdot 12) \\
&= 5 \cdot 77 - 32 \cdot (89 - 77) \\
&= -32 \cdot 89 + 37 \cdot (166 - 89) \\
&= 37 \cdot 166 - 69 \cdot (255 - 166) \\
&= -69 \cdot 255 + 106 \cdot (421 - 255) \\
&= 106 \cdot 421 - 175 \cdot (676 - 421) \\
&= -175 \cdot 676 + \underline{281} \cdot 421
\end{aligned}$$

Пример: $255^{-1} \equiv -281 \equiv 395 \pmod{676}$

$$\begin{aligned}
676 &= 2 \cdot 255 + 166 \\
255 &= 1 \cdot 166 + 89 \\
166 &= 1 \cdot 89 + 77 \\
89 &= 1 \cdot 77 + 12 \\
77 &= 6 \cdot 12 + 5 \\
12 &= 2 \cdot 5 + 2 \\
5 &= 2 \cdot 2 + 1
\end{aligned}$$

$$\begin{aligned}
1 &= 5 - 2 \cdot (12 - 2 \cdot 5) \\
&= -2 \cdot 12 + 5 \cdot (77 - 6 \cdot 12) \\
&= 5 \cdot 77 - 32 \cdot (89 - 77) \\
&= -32 \cdot 89 + 37 \cdot (166 - 89) \\
&= 37 \cdot 166 - 69 \cdot (255 - 166) \\
&= -69 \cdot 255 + 106 \cdot (676 - 2 \cdot 255) \\
&= 106 \cdot 676 \underline{-281} \cdot 255
\end{aligned}$$

Пример: Помоћу Меси-Омура криптосистема са јавним кључем $q = 677$ (прост) треба послати $M = 470$

Алиса бира тајни кључ $e_A = 255$ и $d_A = 395$. Бобан бира тајни кључ $e_B = 421$ и $d_B = 281$ (претх. примери)

$$M^{e_A} = 470^{255}(mod\ 677) = 470^{128} \cdot 470^{64} \cdot 470^{32} \cdot 470^{16} \cdot 470^{8} \cdot 470^{4} \cdot 470^{2} \cdot 470^{1}(mod\ 677)$$

$$470^2 \equiv 198$$

$$470^4 \equiv 198^2 \equiv 615 \equiv -62$$

$$470^8 \equiv (-62)^2 \equiv 459$$

$$470^{16} \equiv 459^2 \equiv 134$$

$$470^{32} \equiv 134^2 \equiv 354$$

$$470^{64} \equiv 354^2 \equiv 71$$

$$470^{128} \equiv 71^2 \equiv 302$$

$$M^{e_A} = 302 \cdot 71 \cdot 354 \cdot 134 \cdot 459 \cdot 615 \cdot 198 \cdot 470(mod\ 677) = 292(mod\ 677)$$

Алиса шаље Бобану 292

$$
\begin{aligned}
M^{e_A e_B} &= 292^{421}(mod\ 677) = 292^{256}\cdot 292^{128}\cdot 292^{32}\cdot 292^{4}\cdot 292^{1}(mod\ 677)\\
& 292^4 \equiv (292^2)^2 \equiv 639^2 \equiv (-38)^2 \equiv 90\\
& 292^{32} \equiv (292^4)^8 \equiv (90^2)^4 \equiv 653^4 \equiv ((-24)^2)^2 \equiv 576^2 \equiv 46\\
& 292^{128} \equiv (292^{32})^4 \equiv (46^2)^2 \equiv 85^2 \equiv 455\\
& 292^{256} \equiv (292^{128})^2 \equiv 455^2 \equiv 540\\
M^{e_A e_B} &= 540\cdot 455\cdot 46\cdot 90\cdot 292(mod\ 677) = 156
\end{aligned}
$$

Бобан шаље Алиси 156

$$
\begin{aligned}
M^{e_A e_B d_A} &= 156^{395}(mod\ 677) = 156^{256}\cdot 156^{128}\cdot 156^{8}\cdot 156^{2}\cdot 156^{1}(mod\ 677)\\
& 156^2 \equiv 641 \equiv -36\\
& 156^8 \equiv (156^2)^4 \equiv ((-36)^2)^2 = 619^2 \equiv 656 \equiv -21\\
& 156^{128} \equiv (156^8)^{16} \equiv (441^2)^4 \equiv (182^2)^2 \equiv 628^2 \equiv 370\\
& 156^{256} \equiv (156^{128})^2 \equiv 370^2 \equiv 146\\
M^{e_A e_B d_A} &= 146\cdot 370\cdot (-21)\cdot (-36)\cdot 156(mod\ 677) = 313(mod\ 677)
\end{aligned}
$$

Алиса шаље Бобану 313

$$
\begin{aligned}
M^{e_A e_B d_A d_B} &= 313^{281}(mod\ 677) = 313^{256} \cdot 313^{16} \cdot 313^{8} \cdot 313^{1}(mod\ 677) \\
& 313^{8} \equiv (313^{2})^{4} \equiv (481^{2})^{2} \equiv 504^{2} \equiv 141 \\
& 313^{16} \equiv (313^{8})^{2} \equiv 141^{2} \equiv 248 \\
& 313^{256} \equiv (313^{16})^{16} \equiv (248^{2})^{8} \equiv (574^{2})^{4} \equiv (454^{2})^{2} \equiv 308^{2} \equiv 84 \\
M^{e_A e_B d_A d_B} &= 84 \cdot 248 \cdot 141 \cdot 313(mod\ 677) = 470(mod\ 677)
\end{aligned}
$$

Бобан коначно добија поруку 470

ЕлГамалов криптосистем

- Јавни кључ q (степен простог броја) и $g \in \mathbb{F}_q^*$ генератор

ЕлГамалов криптосистем

- Јавни кључ q (степен простог броја) и $g \in \mathbb{F}_q^*$ генератор
- Бобан бира свој тајни кључ e_B и помоћу њега прави јавни кључ g^{e_B} који шаље Алиси тј. објављује (као код Дифи-Хелмана)

ЕлГамалов криптосистем

- Јавни кључ q (степен простог броја) и $g \in \mathbb{F}_q^*$ генератор
- Бобан бира свој тајни кључ e_B и помоћу њега прави јавни кључ g^{e_B} који шаље Алиси тј. објављује (као код Дифи-Хелмана)
- $M \in \mathbb{F}_q$ кодирани блок (део поруке) коју Алиса жели да пошаље, она генерише случајан природан број $k < q$ који ће користити само једном за блок M (за наредно M бира ново k)

ЕлГамалов криптосистем

- Јавни кључ q (степен простог броја) и $g \in \mathbb{F}_q^*$ генератор
- Бобан бира свој тајни кључ e_B и помоћу њега прави јавни кључ g^{e_B} који шаље Алиси тј. објављује (као код Дифи-Хелмана)
- $M \in \mathbb{F}_q$ кодирани блок (део поруке) коју Алиса жели да пошаље, она генерише случајан природан број $k < q$ који ће користити само једном за блок M (за наредно M бира ново k)
- Алиса шаље Бобану пар информација g^k и $Mg^{e_B k} = M\left(g^{e_B}\right)^k$ (ово лако рачуна јер зна g, g^{e_B}, k и M)

ЕлГамалов криптосистем

- Јавни кључ q (степен простог броја) и $g \in \mathbb{F}_q^*$ генератор
- Бобан бира свој тајни кључ e_B и помоћу њега прави јавни кључ g^{e_B} који шаље Алиси тј. објављује (као код Дифи-Хелмана)
- $M \in \mathbb{F}_q$ кодирани блок (део поруке) коју Алиса жели да пошаље, она генерише случајан природан број $k < q$ који ће користити само једном за блок M (за наредно M бира ново k)
- Алиса шаље Бобану пар информација g^k и $Mg^{e_B k} = M\left(g^{e_B}\right)^k$ (ово лако рачуна јер зна g, g^{e_B}, k и M)
- Бобан рачуна $g^{e_B k} = \left(g^k\right)^{e_B}$, затим његов инверз и добија $M = Mg^{e_B k}\left(g^{e_B k}\right)^{-1}$

ЕлГамалов криптосистем

- Јавни кључ q (степен простог броја) и $g \in \mathbb{F}_q^*$ генератор
- Бобан бира свој тајни кључ e_B и помоћу њега прави јавни кључ g^{e_B} који шаље Алиси тј. објављује (као код Дифи-Хелмана)
- $M \in \mathbb{F}_q$ кодирани блок (део поруке) коју Алиса жели да пошаље, она генерише случајан природан број $k < q$ који ће користити само једном за блок M (за наредно M бира ново k)
- Алиса шаље Бобану пар информација g^k и $Mg^{e_B k} = M\left(g^{e_B}\right)^k$ (ово лако рачуна јер зна g, g^{e_B}, k и M)
- Бобан рачуна $g^{e_B k} = \left(g^k\right)^{e_B}$, затим његов инверз и добија $M = Mg^{e_B k}\left(g^{e_B k}\right)^{-1}$
- Цица мора да реши проблем дискретног логаритма да би урадила претходни корак

ЕлГамалов криптосистем

- Јавни кључ q (степен простог броја) и $g \in \mathbb{F}_q^*$ генератор
- Бобан бира свој тајни кључ e_B и помоћу њега прави јавни кључ g^{e_B} који шаље Алиси тј. објављује (као код Дифи-Хелмана)
- $M \in \mathbb{F}_q$ кодирани блок (део поруке) коју Алиса жели да пошаље, она генерише случајан природан број $k < q$ који ће користити само једном за блок M (за наредно M бира ново k)
- Алиса шаље Бобану пар информација g^k и $M g^{e_B k} = M \left(g^{e_B}\right)^k$ (ово лако рачуна јер зна g, g^{e_B}, k и M)
- Бобан рачуна $g^{e_B k} = \left(g^k\right)^{e_B}$, затим његов инверз и добија $M = M g^{e_B k} \left(g^{e_B k}\right)^{-1}$
- Цица мора да реши проблем дискретног логаритма да би урадила претходни корак

Напомена: Бобан само једном шаље e_B Алиси (на почетку), код сваког блока имамо једну размену (Алиса шаље пар $\left(g^k, M g^{e_B k}\right)$ Бобану)

Игре на срећу засноване на проблему дискретног логаритма

- Илустроваћемо на примеру бацања новчића, јасно је да се лако може уопштити на било коју игру са малим бројем исхода

Игре на срећу засноване на проблему дискретног логаритма

- Илустроваћемо на примеру бацања новчића, јасно је да се лако може уопштити на било коју игру са малим бројем исхода
- Бобан баца новчић. Алиса жели да се клади на исход писмо/глава, није присутна (не види бацање) и треба да буде сигурна да је Бобан неће преварити

Игре на срећу засноване на проблему дискретног логаритма

- Илустроваћемо на примеру бацања новчића, јасно је да се лако може уопштити на било коју игру са малим бројем исхода
- Бобан баца новчић. Алиса жели да се клади на исход писмо/глава, није присутна (не види бацање) и треба да буде сигурна да је Бобан неће преварити
  - Јавни кључ је q степен простог
  - Алиса бира два различита генератора g_1 и g_2 групе $\mathbb{F}_q^*$ и шаље их Бобану (g_1 =писмо, g_2 =глава)

Игре на срећу засноване на проблему дискретног логаритма

- Илустроваћемо на примеру бацања новчића, јасно је да се лако може уопштити на било коју игру са малим бројем исхода
- Бобан баца новчић. Алиса жели да се клади на исход писмо/глава, није присутна (не види бацање) и треба да буде сигурна да је Бобан неће преварити
  - Јавни кључ је q степен простог
  - Алиса бира два различита генератора g_1 и g_2 групе $\mathbb{F}_q^*$ и шаље их Бобану ($g_1 =$писмо, $g_2 =$глава)
  - Бобан бира случајан број $1 \leqslant n \leqslant q-1$

Игре на срећу засноване на проблему дискретног логаритма

- Илустроваћемо на примеру бацања новчића, јасно је да се лако може уопштити на било коју игру са малим бројем исхода
- Бобан баца новчић. Алиса жели да се клади на исход писмо/глава, није присутна (не види бацање) и треба да буде сигурна да је Бобан неће преварити
  - Јавни кључ је q степен простог
  - Алиса бира два различита генератора g_1 и g_2 групе $\mathbb{F}_q^*$ и шаље их Бобану ($g_1 =$писмо, $g_2 =$глава)
  - Бобан бира случајан број $1 \leqslant n \leqslant q-1$
  - Бобан баца новчић и види која страна g_i је пала. Затим он рачуна $a = g_i^n$ и шаље Алиси

Игре на срећу засноване на проблему дискретног логаритма

- Илустроваћемо на примеру бацања новчића, јасно је да се лако може уопштити на било коју игру са малим бројем исхода
- Бобан баца новчић. Алиса жели да се клади на исход писмо/глава, није присутна (не види бацање) и треба да буде сигурна да је Бобан неће преварити
  - Јавни кључ је q степен простог
  - Алиса бира два различита генератора g_1 и g_2 групе $\mathbb{F}_q^*$ и шаље их Бобану ($g_1 =$писмо, $g_2 =$глава)
  - Бобан бира случајан број $1 \leqslant n \leqslant q - 1$
  - Бобан баца новчић и види која страна g_i је пала. Затим он рачуна $a = g_i^n$ и шаље Алиси
  - Када добије a Алиса бира на које g_i се клади

Игре на срећу засноване на проблему дискретног логаритма

- Илустроваћемо на примеру бацања новчића, јасно је да се лако може уопштити на било коју игру са малим бројем исхода
- Бобан баца новчић. Алиса жели да се клади на исход писмо/глава, није присутна (не види бацање) и треба да буде сигурна да је Бобан неће преварити
  - Јавни кључ је q степен простог
  - Алиса бира два различита генератора g_1 и g_2 групе $\mathbb{F}_q^*$ и шаље их Бобану (g_1 =писмо, g_2 =глава)
  - Бобан бира случајан број $1 \leqslant n \leqslant q-1$
  - Бобан баца новчић и види која страна g_i је пала. Затим он рачуна $a = g_i^n$ и шаље Алиси
  - Када добије a Алиса бира на које g_i се клади
  - Бобан тада саопштава Алиси ко је победио

Игре на срећу засноване на проблему дискретног логаритма

- Илустроваћемо на примеру бацања новчића, јасно је да се лако може уопштити на било коју игру са малим бројем исхода
- Бобан баца новчић. Алиса жели да се клади на исход писмо/глава, није присутна (не види бацање) и треба да буде сигурна да је Бобан неће преварити
  - Јавни кључ је q степен простог
  - Алиса бира два различита генератора g_1 и g_2 групе $\mathbb{F}_q^*$ и шаље их Бобану (g_1 =писмо, g_2 =глава)
  - Бобан бира случајан број $1 \leqslant n \leqslant q-1$
  - Бобан баца новчић и види која страна g_i је пала. Затим он рачуна $a = g_i^n$ и шаље Алиси
  - Када добије a Алиса бира на које g_i се клади
  - Бобан тада саопштава Алиси ко је победио
  - Да би игра била регуларна, Бобан на крају мора да сопшти n како би Алиса проверила да је a (које има) заиста једнако g_i^n (за оно g_i које је Бобан саопштио у претх. кораку)

Да ли је игра регуларна?

- Алиса добија a као гаранцију да неће бити преварена, али она тада не зна n и не може израчунати g_i
  - g_1 и g_2 су генератори, па је a свакако степен и једног и другог

Да ли је игра регуларна?

- Алиса добија a као гаранцију да неће бити преварена, али она тада не зна n и не може израчунати g_i
  - g_1 и g_2 су генератори, па је a свакако степен и једног и другог
- Ако би Бобан желео да промени g_i (након Алисиног избора) он мора да реши проблем дискретног логаритма.

Да ли је игра регуларна?

- Алиса добија a као гаранцију да неће бити преварена, али она тада не зна n и не може израчунати g_i
  - g_1 и g_2 су генератори, па је a свакако степен и једног и другог
- Ако би Бобан желео да промени g_i (након Алисиног избора) он мора да реши проблем дискретног логаритма.
  - Зато је битно да генераторе g_1 и g_2 бира Алиса

Да ли је игра регуларна?

- Алиса добија a као гаранцију да неће бити преварена, али она тада не зна n и не може израчунати g_i
  - g_1 и g_2 су генератори, па је a свакако степен и једног и другог
- Ако би Бобан желео да промени g_i (након Алисиног избора) он мора да реши проблем дискретног логаритма.
  - Зато је битно да генераторе g_1 и g_2 бира Алиса
- Ако Бобан покуша (на почетку) да нађе једно поклапање $g_1^{n_1} = g_2^{n_2}$ и тај број подметне Алиси
  - чак ни ово не може да уради брзо, видели смо да је овакво претраживање најдужи корак у Гељфонд-Шенксовом алгоритму

Да ли је игра регуларна?

- Алиса добија a као гаранцију да неће бити преварена, али она тада не зна n и не може израчунати g_i
  - g_1 и g_2 су генератори, па је a свакако степен и једног и другог
- Ако би Бобан желео да промени g_i (након Алисиног избора) он мора да реши проблем дискретног логаритма.
  - Зато је битно да генераторе g_1 и g_2 бира Алиса
- Ако Бобан покуша (на почетку) да нађе једно поклапање $g_1^{n_1} = g_2^{n_2}$ и тај број подметне Алиси
  - чак ни ово не може да уради брзо, видели смо да је овакво претраживање најдужи корак у Гељфонд-Шенксовом алгоритму
  - како се инверз (нпр. од n_2) по модулу $q-1$ брзо рачуна: налажење n_1, n_2 тд. $g_1^{n_1} = g_2^{n_2}$ је еквивалентно рачунању $\log_{g_1} g_2$

Да ли је игра регуларна?

- Алиса добија a као гаранцију да неће бити преварена, али она тада не зна n и не може израчунати g_i
  - g_1 и g_2 су генератори, па је a свакако степен и једног и другог
- Ако би Бобан желео да промени g_i (након Алисиног избора) он мора да реши проблем дискретног логаритма.
  - Зато је битно да генераторе g_1 и g_2 бира Алиса
- Ако Бобан покуша (на почетку) да нађе једно поклапање $g_1^{n_1} = g_2^{n_2}$ и тај број подметне Алиси
  - чак ни ово не може да уради брзо, видели смо да је овакво претраживање најдужи корак у Гељфонд-Шенксовом алгоритму
  - како се инверз (нпр. од n_2) по модулу $q-1$ брзо рачуна: налажење n_1, n_2 тд. $g_1^{n_1} = g_2^{n_2}$ је еквивалентно рачунању $\log_{g_1} g_2$
- Исто, Бобан не може да намести ни да је изгубио.