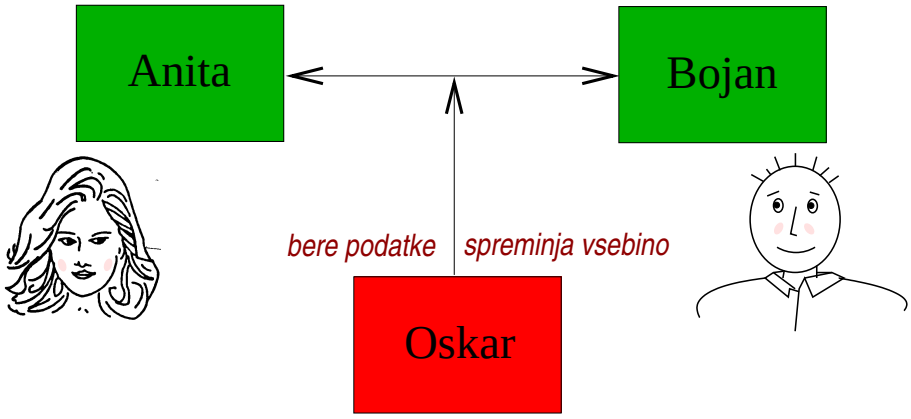
DIFFIE-HELLMANOV DOGOVOR O KLJUČU
2. del

Ali je možno, da se dve osebi, ki se želita pogovarjati po javnem kanalu (npr. telefonu ali Internetu), preko njega dogovorita o skrivnem ključu?

Le-tega naj bi poznali samo oni dve in ga potem uporabili za šifriranje njunega pogovora. Če sta se že prej srečali na štiri oči, potem bi se gotovo lahko spomnili nečesa, kar je skupno le njima in na osnovi tega sestavili skrivni ključ. Dovolj je na primer že, da se spomnita neke knjige, ki sta jo takrat skupaj občudovali, in že si lahko izbereta za relativno varno geslo npr. deveti odstavek v tretjem poglavju. Četudi bi prisluškovalec vedel, da gre za deveti odstavek v tretjem poglavju, nikakor ne bi mogel ugotoviti, za katero knjigo gre in mu tako pridobljena informacija brez knjige ne bi bila v kdo ve kakšno pomoč. Naj zveni še tako čudno, a ljudje so dolga leta uporabljali prav takšne kodne knjige za zaščito svojih zaupnih pogovorov. Danes, v času, ko so pri Googlu začeli skenirati skoraj vse knjige po vrsti in ko se da prav hitro preveriti več milijonov gesel, pa tak pristop ni nujno dovolj varen.

Kaj pa če se naši osebi nista še nikoli srečali na štiri oči ali pa če je bil poleg njiju (oziroma na javnem kanalu) vedno še nekdo, ki jima utegne kaliti srečo (glej sliko 1)? Ali se lahko potem preko javnega kanala dogovorita o skupnem ključu, ki je znan le njima?



Slika 1: Anita, Bojan ter napadalec Oskar.

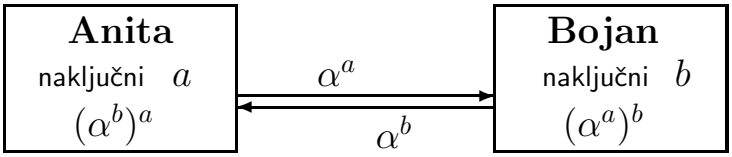
Rekli boste, da je to vendar nemogoče, saj s strani enega sogovornika ni mogoče ločiti drugega sogovornika od prisluškovalca, ki je slišal prav vse, kar je drugi sogovornik povedal oziroma slišal. Na našo srečo je tak dogovor o ključu iskal tudi **Whitfield Diffie**. Verjel je, da so ga v črnih kabinetih (ki so jih za potrebe lastne varnosti in vohunjenja ustanovile različne države) že našli in da si ne želijo, da bi o njem vedeli običajni ljudje. Ena izmed največjih takih organizacij je ameriški NSA (No Such Agency – morda že veste, kaj ta kratica pomeni v resnici). Tam dela največ matematikov na svetu, vendar pa so rezultati njihovega dela strogo zaupne narave in jih uporabljajo v vojaške oziroma vohunske namene. Diffija je bilo kar groza, da bi na ta način zadrževali razvoj vsega človeštva. Zato si je zadal za osrednjo nalogo, da se bo priboril do teh skrivnosti in jih dal na voljo vsem ljudem. Temu cilju je podredil svoje življenje. Kriptologi so ga imeli večinoma že za malce trčenega, saj niso verjeli, da je kaj takega sploh možno. Ko je Diffie nekega dne izvedel, da se z istim problemom ukvarja tudi **Martin Hellman**, ki je živel na čisto drugem koncu

Slika 2: Whitfield Diffie.

Slika 3: Martin Hellman.

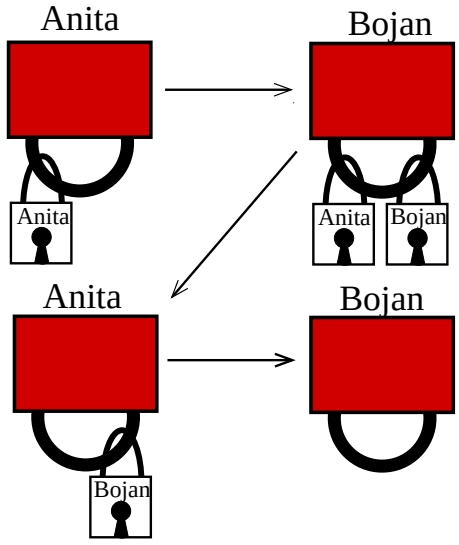
ZDA, se je nemudoma usedel v avto in se z vzhodne obale odpravil živet v njegovo bližino (San Francisco). Sledilo je obdobje skupnega iskanja rešitve in čez nekaj let se jima je zares posrečilo. Najprej sta rešila Diffijev problem, ki je osrednja tema tega sestavka, kmalu zatem pa so skupaj z **Ralphom Marklom** prišli do revolucionarnega koncepta **kriptosistemov z javnimi ključi**, ki ga bomo spoznali v tretjem delu. Čeprav so rešitev iskali tako dolgo, pa se je **Diffie-Hellmanov dogovor o ključu** (na kratko **DH-protokol**) izkazal za zelo enostavnega in elegantnega. Predstavimo ga!

Najprej se Anita in Bojan dogovorita, kako bosta računala, na primer kateri element α bosta uporabljala. Takemu številu bomo rekli **generator** in kmalu bo postalo jasno, zakaj je tako. Anita si naključno izbere svoje zasebno število a, Bojan pa svoje naključno zasebno število b. Nato izračunata vsak svojo potenco $A := \alpha^a$ ter $B := \alpha^b$ ter si ju izmenjata preko javnega kanala. Končno izračunata skupni tajni ključ $\alpha^{ab} = K = \alpha^{ba}$, Anita z $B^a = (\alpha^b)^a$, Bojan pa z $A^b = (\alpha^a)^b$. Glej sliko 4.



Slika 4: **Diffie-Hellmanov dogovor o ključu** – Anita in Bojan si izbereta vsak svoje naključno število a in b ter si na koncu delita skupni element grupe, ki ga izračunata iz izmenjanih potenc: $(\alpha^a)^b = (\alpha^b)^a = \alpha^{ab} = K$.

Če je bil namen dogovora o ključu med Anito in Bojanom pošiljanje zaupnega sporočila, potem po zgoraj opisanih dveh izmenjavah sledi še tretja, v kateri Anita s skupnim tajnim ključem zašifrira in pošlje zaupno sporočilo, ki ga na drugi strani Bojan brez težav odšifrira. Popolnoma brez matematike lahko opišemo podoben protokol, ki sta ga predlagala **Massey** in **Omura**. Tudi zanj potrebujemo tri izmenjave sporočil. Glej škatlo Protokol Massey-Omura. V digitalnem svetu, kjer si seveda ne želimo fizičnih ključavnic, si namesto njih lahko omislimo dve obrnljivi funkciji, ki med seboj komutirata, tj. velja pravilo o zamenjavi.



Protokol Massey-Omura. 1. korak: Anita zaklene sporočilo v škatlo s svojo ključavnico ter vse skupaj pošlje Bojanu. **2. korak:** Bojan prejme pošiljko, prepozna Anitino ključavnico, doda še svojo ključavnico in vrne pošiljko Aniti. **3. korak:** Anita prejme pošiljko, prepozna svojo in Bojanovo ključavnico ter odklene in odstrani svojo ključavnico. Nato še drugič pošlje pošiljko Bojanu. **4. korak:** Bojan prejme pošiljko, odklene in odstrani svojo ključavnico ter pride do sporočila.

UČINKOVITOST

Kaj je tako posebnega pri DH-protokolu? Najprej moramo znati učinkovito izvesti vse potrebne računske operacije. V našem primeru je to potenciranje elementa α.

Če želimo izračunati α^n, kjer je $n \in \mathbb{N}$, potem lahko to storimo z $n-1$ množenji:

$$((\cdots((\alpha * \alpha) * \alpha) * \cdots) * \alpha) * \alpha.$$

Žal s tem ne moremo biti zadovoljni, saj čas računanja merimo v odvisnosti od dolžine zapisa števil, s katerimi imamo opravka. Naj bo dolžina največjega zapisa enaka k. Produkt dveh števil v dvojiškem zapisu lahko izračunamo v grobem s k^2 operacijami (premisli zakaj!). Problem pa je v tem, da iz

$$2^k \approx n$$

sledi, da moramo opraviti eksponentno mnogo množenj. Eksponentni algoritmi so izredno počasni (glej škatlo Zgodba iz antične Indije), v praksi pa je pogosto $k > 100$, tako da takšno računanje potence v nobenem primeru ne pride v poštev.

Zgodba iz antične Indije. O njej smo v Preseku že pisali (A. Lipovec, Kaj imajo skupnega Gari Kasparov, Jamie Oliver in ozonske luknje? *Presek* **33** (2005), str. 4–6), zato le na kratko omenimo, da naj bi na prvo polje šahovnice položili eno zrno riža, na drugo polje dve, na tretje štiri ... (za vsako naslednje polje podvojimo število zrn). Šahovnica ima 64 polj, kar pomeni, da bi morali nanjo postaviti

$$1 + 2 + 2^2 + \cdots + 2^{63} = 2^{64} - 1 \text{ zrn riža.}$$

Za kriptografijo je zelo pomembno, da razumemo velike številke. Ali si lahko predstavljamo vsaj v grobem koliko riža bi to bilo? Recimo, da lahko poje povprečen človek pol kg riža na dan. To je kar precej, ko je riž enkrat kuhan. Koliko zrn napolni čajno žličko? Približno 330, za pol kg riža pa jih potrebujemo nekaj čez 80 oziroma $27.500 = 2,75 * 10^4$ zrn riža. Na svetu je danes 6,6 milijarde $= 6,6 * 10^9$ ljudi. Če bi vsi ljudje jedli samo riž, bi torej lahko pojedli $2,75 * 10^4 * 6,6 * 10^9 \approx 1,8 * 10^{14}$ zrn riža na dan.

Sedaj pa se vrnimo k številu $2^{64} - 1 \approx 1,8 * 10^{19}$ zrn in ga delimo s številom $1,8 \times 10^{14}$ zrn, ki jih poje človeštvo na dan. Dobimo 100.000 dni, kar pomeni, da bi lahko svet jedel riž s šahovnice toliko časa, dan za dnem, oziroma $100.000/365 \approx 274$ let!

Pa si poglejmo zapis števila n v dvojiškem sistemu:

$$n = n_k\, 2^k + \cdots + n_1\, 2^1 + n_0\, 2^0, \qquad \text{kjer je } n_i \in \{0, 1\}.$$

Potem je

$$\alpha^n = \alpha^{n_k\, 2^k} * \cdots * \alpha^{n_1\, 2^1} * \alpha^{n_0\, 2^0}. \tag{1}$$

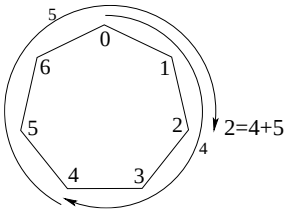
Potence $\alpha^2, \alpha^{2^2}, \ldots, \alpha^{2^k}$ izračunamo s k kvadriranji (v ta namen kvadriramo $\alpha^{2^{i-1}}$, da dobimo α^{2^i}). S tem smo dobili vse faktorje v (1). Potem pa za izračun produkta iz (1) potrebujemo še največ k množenj. Skupaj nanese $2k$ množenj oziroma v grobem k^3 procesorskih operacij, kar pomeni, da je naše računanje učinkovito in s tem tudi DH-protokol.

VARNOST

Kako pa je z možnimi napadi na DH-protokol? Mlajši bralci, ki se še niste srečali s funkcijami, lahko mirno preskočite ta razdelek. Na prvi pogled se zdi, da je dovolj, da znamo učinkovito izračunati bodisi $\log_\alpha A = a$ bodisi $\log_\alpha B = b$. Če bi bila α, a in b realna števila, potem to ne bi smel biti problem – vsaj od takrat naprej, ko so ljudje sestavili dobre logaritemske tablice, da o današnjih računalnikih niti ne govorimo. Vendar pa si kriptografi namesto obsega realnih števil v praksi raje izberemo diskretno grupo, v kateri ne znamo

učinkovito izračunati logaritma poljubnega njenega elementa. Glej škatlo Problem diskretnega logaritma (DLP). Spomnimo se, da smo pred leti v Preseku pisali o računalih nove dobe (1 del, Presek 30/1 (2002-03), 226-231 in 2. del, Presek 30/5 (2002-03), 291–296), kjer smo vpeljali nekaj zanimivih grup. Pa DLP ni edini razlog za uporabo diskretnih grup. Če računamo z običajnimi tipi števil, celimi, racionalnimi ali realnimi števili, se zna zgoditi, da bodo rezultati izračunov hitro zrastli preko vseh meja (glej škatlo Zgodba iz antične Indije) ali pa bomo zaradi omejene velikosti prisiljeni nenehno zaokrožati. V kriptografiji je učinkovitost osrednjega pomena, približki pa ne zadoščajo (predstavljajte si, da vam bankomat ali pa žepni telefon odgovorita, da danes nista povsem zadovoljna z vašim geslom, pa naj si bo to zato, ker še vedno računata ali pa se jima zdi geslo samo približno prav).

Tako si za računanje raje omislimo končne množice kot pri številčnici na uri. Tak zgled so kolobarji $\mathbb{Z}_n$, $n \in \mathbb{N}$, v katerih računamo po modulu števila n. Za elemente vzamemo $\{0, 1, \ldots, n-1\}$, računamo pa tako, da seštejemo ali zmnožimo dve števili tako, da pravi rezultat nadomestimo z njegovim ostankom pri deljenju z **modulom** n. Na primer za $n = 7$ velja $4 +_7 5 = 4 + 5 \mod 7 = 2$ in $5 *_7 4 = 5 \cdot 4 \mod 7 = 6$, saj ima vsota 9 ostanek 2 pri deljenju s 7, produkt 20 pa ostanek 6, glej sliko 3. Če želimo še deliti z vsakim neničelnim številom, potem si moramo za modul n izbrati neko praštevilo p, tako da v resnici delamo v praštevilskem obsegu $\mathbb{Z}_p$ oziroma za potrebe DH-protokola kar v njegovi podgrupi $(\mathbb{Z}_p \backslash \{0\}, *)$, ki ji pravimo **multiplikativna podgrupa**, oznaka $\mathbb{Z}_p^*$. Potem vsak element $\alpha \in \mathbb{Z}_p^*$, za katerega velja $D(\alpha, p-1) = 1$, potence $\alpha, \alpha^2, \ldots, \alpha^{p-1}$ pretečejo vse elemente grupe $\mathbb{Z}_p^*$ (zadnji element je seveda enota, saj Fermatov izrek za izbrana α in p pravi $\alpha^{p-1} \equiv 1 \pmod{p}$, glej članek D. Pagona, Kongruence in Eulerjev izrek, *Presek* **15** (1987-88), str. 194–196), tj. α **generira** grupo $\mathbb{Z}_p^*$. Multiplikativna grupa poljubnega končnega obsega je torej *ciklična*, tj. v grupi obstaja tak element, da so vsi elementi njegove potence, zato lahko njene elemente predstavimo na krogu, tako kot smo to storili na sliki 5.



Slika 5: Računanje po modulu 7.

Problem diskretnega logaritma (DLP).
Za dano grupo G in elementa $\alpha, \beta \in G$, kjer je red elementa α enak n, poišči učinkovito metodo, ki določi celo število x, $0 \le x \le n-1$ (če obstaja), tako da je $\alpha^x = \beta$. Število x imenujemo **diskretni logaritem** števila β pri osnovi α.
Požrešna metoda za reševanje DLP, ki računa zaporedne potence elementa α, tj. $\alpha, \alpha^2, \alpha^3, \ldots$ in jih primerja z β, je včasih znala odpovedati tam nekje za n nad 2^{40}, danes pa verjetno tam nekje pri 2^{80} (odvisno koliko opreme in časa imamo na voljo).
Metoda mali-veliki korak je v nekaterih grupah še vedno najboljša metoda (izboljšati se da le njeno prostorsko zahtevnost) in odpove pri dvakrat večjih eksponentih, ki smo jih omenili pri požrešni metodi. Pri tej metodi si najprej naredimo tabelo na enak način, kot smo začeli pri požrešni metodi, le da se ustavimo pri k-ti potenci, kjer je k celi del kvadratnega korena iz n (lahko si predstavljamo, da smo z malimi koraki prehodili in s tem pokrili lok dolžine k na zgoraj omenjenem krogu, ki ga sestavljajo potence elementa α). Nato pa zaporedoma pregledujemo katerega izmed elementov $\beta, \beta\alpha^k, \beta\alpha^{2k} \ldots, \beta\alpha^{k*k}$ najdemo v tej tabeli (enega bomo zagotovo, saj pravkar omenjenega loka ne moremo preskočiti s korakom dolžine k). Ko se to zgodi, velja

$$\beta(\alpha^k)^j = \alpha^i,$$

od koder dobimo $\log_\alpha \beta = i - k * j \mod n$.

Morda izgleda na prvi pogled, da bi lahko namesto $\mathbb{Z}_p^*$ uporabljali kar grupo $(\mathbb{Z}_n, +_n)$,

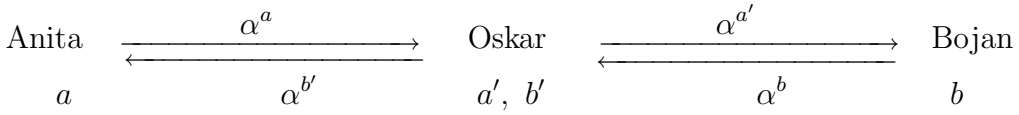
$n \in \mathbb{N}$. Potem bi namesto potence α^a v DH-protokolu zapisali večkratnik $V = a\alpha$, kjer je $\alpha \in \mathbb{Z}_n$, $D(\alpha, n) = 1$ in $a \in \mathbb{N}$. Vendar pa v tem primeru poznamo algoritem, ki nam iz večkratnika V izračuna a. V ta namen je namreč potrebno rešiti naslednjo diofantsko enačbo:

$$a\alpha + kn = 1,$$

za a (in k). To pa znamo učinkovito rešiti z razširjenim Evklidovim algoritmom, saj je $D(\alpha, n) = 1$ (glej članek M. Juvana, O Evklidovem algoritmu, *Presek* **21** (1993-94), str. 116–121). Izbira najboljše grupe G za DH-dogovor o ključu, v kateri bo računanje potenc učinkovito, reševanje DLP pa nedosegljivo, je izredno občutljiva naloga, ki jo morajo opraviti izkušeni kriptologi.

AKTIVNI SREDNJI NAPADALEC

Zgoraj predstavljen Diffie-Hellmanov protokol imenujemo tudi **neoverjen** Diffie-Hellmanov protokol. Pri njem udeleženci namreč ne morejo z gotovostjo ugotoviti izvora javnih vrednosti ostalih udeležencev (tj. potenc, ki si jih morajo izmenjati). To pa naredi neoverjen Diffie-Hellmanov protokol ranljiv na napad, ki ga imenujemo **napad srednjega napadalca** (angl. *middle man attack*), glej sliko 6.



Slika 6: Napad srednjega napadalca na DH-protokol. Pri tem napadu nasprotnik Oskar prestreže Anitino javno vrednost α^a in pošlje svojo javno vrednost $\alpha^{a'}$ Bojanu. Ko Bojan pošlje svojo javno vrednost α^b, jo Oskar prestreže in zamenja s svojo javno vrednostjo $\alpha^{b'}$ in jo pošlje naprej Aniti. Tako delita Anita in Oskar skupen skrivni sejni ključ $K_{ab'} = \alpha^{ab'}$ ter Bojan in Oskar drug skrivni sejni ključ $K_{a'b} = \alpha^{a'b}$. Po tej izmenjavi Oskar preprosto odšifrira sporočila poslana s strani Anite ali Bojana, jih prebere in po potrebi spremeni, preden jih zopet zašifrira in pošlje naprej.

Besedna zveza "*napad srednjega napadalca*" se uporablja za opis napada, kadar Oskar sedi nekje na sredini komunikacijskega kanala/poti med Anito in Bojanom ter oba prelisiči. Je pomemben napad, ki povzroči številna dopolnila in spremembe komunikacijskih protokolov. Vendar pa je, kot pravi B. Schneier, to tudi napad, ki ga lahko uporabimo v vsakdanjem življenju:

Premožna gospa želi najti pomoč pri pospravljanju hiše in v časopisu objavi oglas, da rabi pomoč. Ko se javi primerna oseba, jo gospa vpraša po priporočilih, da preveri njene izkušnje in zanesljivost. Ker pa gospa v resnici sploh ni lastnica hiše, ampak gre za staro znanko policije, sama uporabi pravkar pridobljena priporočila in si z njimi najde zaposlitev pomočnice v bogati hiši.

Službo je dobila z dobrimi priporočili – pa saj so vendar prava, čeprav niso njena – in nato odnese iz hiše vso zlatnino in ostale vrednosti.

Seveda lahko ta postopek lažna gospa-pomočnica ponovi tudi večkrat.

Kaj se je zgodilo v zgornjem primeru? Navidezna gospa-pomočnica se je postavila na sredino komunikacije med pravo pomočnico ter pravo gospo in se predstavila vsaki kot tista druga. Pomočnica pošlje priporočila nekomu, ki ni prava gospa. Prava gospa preveri priporočila, ne da bi preverila, da v resnici ne pripadajo lažni pomočnici.

Napad srednjega napadalca je uspešen zato, ker Diffie-Hellmanov protokol ne preveri identitete udeležencev v protokolu oziroma njihovih javnih vrednosti. Rešitev tega problema lahko dosežemo tako, da se udeleženci in javne vrednosti, ki nastopajo v protokolu, ustrezno overijo z digitalnimi certifikati. To pa je že nova zgodba, ki si jo lahko ogledate v tretjem delu hkrati z zgodbo o volku in sedmih kozličkih.

NALOGE

1. Funkcija $f(x) = \log_\alpha x$ nam pove, kateri eksponent moramo uporabiti na α, da dobimo x. Določi definicijsko območje realnih funkcij $f(x)$ in $g(y) = \alpha^y$. Prepričaj se, da je $\alpha^{\log_\alpha x} = x$ in $\log_\alpha(\alpha^x) = x$, kar pomeni, da sta funkciji $f(x)$ in $g(x)$ **inverzni**.

2. Preveri enakost
$$1 + 2 + 2^2 + \cdots + 2^{63} = 2^{64} - 1,$$
tako da levo stran pomnožiš z $(2 - 1)$ in uporabiš zakon o razčlenjevanju oziroma distributivnosti. Bi znal to enakost posplošiti?

3. Ali se lahko domisliš še kakšne zanimive zgodbe, ki bi nazorno ponazorila hitro rast eksponentne funkcije? Pomisli, npr. koliko prednikov bi imel v času, ko je bila zgrajena arena v Puli (pred približno 2000 leti – glej `http://hr.wikipedia.org/wiki/Pula`), če si med njimi nobena dva iz iste generacije ne bi bila v sorodu?

4. Oceni, koliko riža potrebujemo, da pokrijemo cel svet (vključno z oceani), če meri polmer zemlje 6400 km?

5. Oceni, koliko besed lahko "obdela" tvoj ali pa šolski PC v eni uri? Kolikšna je velikost največjega števila, ki bi ga še lahko zapisal v spomin svojega ali šolskega računalnika?

6. Kolikšna je dolžina produkta dveh binarnih števil, katerih zapisa sta dolga m in n?

7. Koliko operacij potrebuješ za izračun produkta dveh 100-bitnih binarnih števil s procesorjem/vodilom svojega ali šolskega računalnika?

8. Pri učinkovitem algoritmu računanja potence α^n, $n \in \mathbb{N}$, smo vnaprej izračunali potence $\alpha^2, \alpha^{2^2}, \alpha^{2^3}, \ldots$ Poišči učinkovit algoritem za potenciranje, ki potrebuje bistveno manj spomina. (Namig: en tak algoritem je znan pod imenom **kvadriraj in zmnoži**, njegov spomin pa je neodvisen od števila n.)

9. Poišči čim več parov funkcij, za katere velja pravilo o zamenjavi.
Ali znaš s takim parom sestaviti digitalno različico protokola Massey-Omura, ki bo varna?

10.* Ni se težko prepričati, da je ciklična grupa z n elementi izomorfna grupi $(\mathbb{Z}_n, +_n)$. Kaj lahko poveš o relaciji med zahtevnostjo problema diskretnega logaritma v poljubni ciklični grupi G z n elementi in problema iskanja izomorfizma med grupo G in aditivno grupo $(\mathbb{Z}_n, +)$?

Aleksandar Jurišić