Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Využití prvočísel při šifrování dat

2023

Matěj Ošťádal

Vedoucí práce:
doc. RNDr. Miroslav Kolařík, Ph.D.

Studijní program: Informatika,
Specializace: Obecná informatika

Bibliografické údaje

Autor:	Matěj Ošťádal
Název práce:	Využití prvočísel při šifrování dat
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2023
Studijní program:	Informatika, Specializace: Obecná informatika
Vedoucí práce:	doc. RNDr. Miroslav Kolařík, Ph.D.
Počet stran:	24
Přílohy:	žádné
Jazyk práce:	český

Bibliographic info

Author:	Matěj Ošťádal
Title:	Use of prime numbers in data encryption
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2023
Study program:	Computer Science, Specialization: General Computer Science
Supervisor:	doc. RNDr. Miroslav Kolařík, Ph.D.
Page count:	24
Supplements:	none
Thesis language:	Czech

Anotace

TODO ANOTACE

Synopsis

TODO ANOTACE ANGLICKY

Klíčová slova: šifrování; prvočísla; bezpečnost; modulární aritmetika;

Keywords: encryption; prime numbers; security; modular arithmetic

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce podpis autora

Obsah

Seznam tabulek

Část I
Úvod

V celé práci budeme hledat různá řešení následujícího problému:

Mějme dva uživatele, Alici a Boba, kteří si chtějí po síti poslat tajnou zprávu m. V síti je také protivník, Eva, který komunikaci odposlouchává (může zprávu zachytit). Potřebujeme zařídit, aby Eva nemohla zjistit obsah zprávy m.[1]

Velmi zjednodušeně můžeme popsat poslání zprávy Alice Bobovi takto: Alice upraví zprávu m (upravenou zprávu označíme c)[2] a pošle ji sítí Bobovi. Bob přijme c, upraví ji do původní podoby m a poté si ji přečte. Alice s Bobem využívají toho, že Eva neví jakým způsobem byla m upravena na c, a tudíž nemá jak získat m.

Tomu, co myslíme tím, že je zpráva upravována, se budeme věnovat dále.

Pro zjednodušení budeme nyní předpokládat následující:

- Eva není schopna modifikovat zachycenou zprávu. Bob tedy nebude muset kontrolovat, zda zprávu opravdu poslala Alice, a naopak (tohoto předpokladu se zbavíme v kapitole III).
- Alice i Bob mají možnost si zprávu přečíst v bezpečném prostředí.

Základní způsob úpravy zprávy budeme nazývat *šifrování*. Proces úpravy zprávy m na c budeme nazývat *zašifrování* a proces úpravy c zpět na m *dešifrování*.

[1] Jména Alice a Bob byla poprve použita v článku *A method for obtaining digital signatures and public-key cryptosystems* z roku 1978. Jméno Eva (z angl. *eavesdropper*) bylo jedno z dalších, které se v kryptografii objevilo. Jména nám pomáhají udržet přehlednější a jednodušší výklad.

[2] Použití písmena m a c plyne z angl. slov *message* a *cipher*.

Část II
Symetrické šifrování

Symetrické šifrování je způsob šifrování, ve kterém je v procesu zašifrování zprávy použit stejný klíč k[3], jako v procesu dešifrování. [4]
[5]

Symetrická šifra pro nás bude uspořádaná dvojice $\mathcal{E} = (E, D)$, kde:

- E je funkce zašifrování (E z ang. *encryption*). E přijímá na vstupu klíč k a zprávu m. Jako výstup vrací zašifrovaný text c.

  $$E : \mathcal{K} \times \mathcal{M} \to \mathcal{C},$$

  kde $\mathcal{K}$ je množina klíčů, $\mathcal{M}$ je množina zpráv a $\mathcal{C}$ je množina šifrovaných zpráv.

- D je funkce dešifrování (D z ang. *decryption*). D přijímá na vstupu klíč k a zašifrovaný text c. Jako výstup vrací dešifrovanou zprávu m.

  $$D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$$

Každá $\mathcal{E}$ je tedy definována nad $(\mathcal{K}, \mathcal{M}, \mathcal{C})$.

Teď už můžeme konkrétněji formulovat postup poslání zprávy mezi Alicí a Bobem:
Alice zašifruje zprávu m (tedy sestrojí $c = E(k, m)$) a pošle c sítí Bobovi. Bob přijme c, rozšifruje ho (tedy získá $m = D(k, c)$), a zprávu si přečte.

Všimněme si teď několika věcí:

1. Přirozeně požadujeme, aby $D(k, E(k, m)) = m$. (Bob získá stejnou zprávu, jako poslala Alice.) Této podmínce budeme říkat *podmínka korektnosti* a nadále se budeme zabývat pouze takovými šiframi, které ji splňují.

2. Alice a Bob musí být předem domluveni na používaném klíči k.

3. To, že si Eva přečte m nám nevadí (z c nejde snadno získat m)[6].

4. Eva nesmí znát k, jinak by totiž z c mohla získat původní m.

[3] Písmeno k opět používáno kvůli anglickému *key*.
[4] Místo názvu *symetrické šifrování* se často používá název *šifrování se soukromým (tajným) klíčem* (anglicky *private-key cryptography*).
[5] Písmeno k opět používáno kvůli anglickému *key*.
[6] Slovem *snadno* myslíme, že získání m z c je značně výpočetně náročné. Tomuto se ještě budeme konkrétněji věnovat později.

Zkusme se zamyslet nad tím, jak bychom mohli zařídit bod 2, tedy jak by se Alice mohla s Bobem domluvit na klíči k, a přitom zajistit bod 4.

Určitě nás napadne, že by si Alice s Bobem mohli k poslat zprávou. Pokud ale Eva čte všechny zprávy v síti, určitě by si mohla k zapamatovat.

Mohli bychom tedy klíč k zašifrovat pomocí dalšího tajného klíče k_2. Alice by tedy sestrojila $c = E(k_2, k)$ a c by poslala Bobovi. Bob by pomocí $D(k_2, c)$ získal k, který by Eva neznala. Tím bychom sice vyřešili náš původní problém, ale vytvořili bychom další: Jak se Alice s Bobem domluví na k_2? (Jistě nám dojde, že při použití stejného postupu by vznikal stále dokola ten samý problém.)

Potřebujeme, aby se Alice s Bobem na k domluvili v nějaké bezpečné síti, kterou Eva nemůže odposlouchávat. (Například by se mohli sejít v parku a k si tajně sdělit.)[7]

Kdyby ale existovala bezpečná síť, kterou by Eva nemohla odposlouchávat, jistě by mohli Alice s Bobem vést veškerou komunikaci rovnou přes ni. Tím pádem by se vůbec nepotřebovali domluvit na k, jelikož by nebylo potřeba zprávy šifrovat. Nebylo by tedy ani potřeba řešit problém, který byl představen v úvodu.

K tomu, jak se Alice s Bobem mohou domluvit na tajném klíči k i přes kanál, který Eva odposlouchává, se dostaneme později (konkrétně v kapitole 5). Budeme k tomu potřebovat širší aparát znalostí.

Nyní uvedeme některé základní příklady symetrických šifer.

1 Caesarova šifra

Caesarova šifra $\mathcal{E}$ spadá do kategorie substitučních šifer.[8] $\mathcal{E}$ je definovaná nad $(\mathbb{N}_0, \Sigma^L, \Sigma^L)$, kde Σ je konečná množina symbolů a L je libovolně zvolená délka.

Pokud bychom symboly v abecedě oindexovali (tedy $\Sigma = \{a_0, a_1, \ldots, a_n\}$), funkce zašifrování E by každý symbol zaměnila za symbol, který je v abecedě o k míst dále. Symboly na konci abecedy bychom posunovali ve smyslu mod (např.: $E(2, y) = a$, $E(2, z) = b$ pro klasickou anglickou abecedu). Analogicky by funkce dešifrování D každý symbol zaměnila za symbol, který mu v abecedě o k míst předchází. Vidíme, že takto zvolená šifra splňuje *podmínku korektnosti*.

Formálně můžeme zapsat:

$$E(k, a_i) = a_l, \text{ kde } l = (i + k) \mod |\Sigma|$$
$$D(k, a_j) = a_m, \text{ kde } m = (j - k) \mod |\Sigma|$$

Je jasné, že kdybychom chtěli zašifrovat celou zprávu m, tak podle klíče k zašifrujeme všechny symboly jednotlivě na nové a jejich spojením vznikne zašifrovaná zpráva c.

[7] To bude zřejmě problém, pokud se Alice s Bobem nachází na opačných koncích světa.

[8] Substituční šifra je druh šifry, při kterém dochází k záměně množiny symbolů za jinou množinu symbolů podle daného klíče.

Na okraj ještě uveďme, že se u klíčů můžeme omezit na podmnožinu nezáporných celých čísel a pracovat pouze s $\mathcal{K} = \{n \in \mathbb{N}_0 \mid n < |\Sigma|\}$ bez újmy na obecnosti. Je zřejmé, že například pro množinu symbolů velikosti 2 by každý lichý klíč prohodil každý symbol na opačný a každý sudý klíč by nechal m beze změny. Mohli bychom se tedy omezit pouze na $k \in \{0, 1\}$ aniž bychom jakkoliv změnili počet možností zašifrování zprávy m. Pro abecedu Σ tedy obecně existuje $|\Sigma|$ klíčů, které zprávu m zašifrují unikátním způsobem.[9]

1.1 Bezpečnost Caesarovy šifry

Představme si nyní, že Alice a Bob spolu komunikují přes síť a využívají přitom Caesarovy šifry (pro zjednodušení uvažujme, že už jsou dohodnuti na společném klíči k). Je komunikace bezpečná?

Pokud Eva zašifrovanou zprávu c může číst, zřejmě její obsah nebude ihned zřetelný. Mohla by ale vyzkoušet všechny možnosti pro klíč k. Již jsme provedli úvahu o tom, že se s klíči můžeme omezit na $\{n \in \mathbb{N} \mid 0 \leq n < |\Sigma|\}$. Eva tedy může postupně vyzkoušet všechny tyto možnosti. Jedna z nich jistě bude správně dešifrovat c a Eva si m přečte.

Pokud by Alice například chtěla Bobovi poslat zprávu v anglickém jazyce, stačilo by Evě otestovat 26 možností, jelikož anglická abeceda má pouze 26 znaků. Kdyby Alice chtěla Bobovi poslat zprávu skládající se z libovolných znaků ASCII, stačilo by Evě otestovat 128 možností. Kdyby Alice například posílala tajný číselný kód (přirozené číslo), stačilo by Evě vyzkoušet 10 možností.

Obecně tedy k prolomení[10] Caesarovy šifry stačí čas $O(|\Sigma|)$.[11]

K prolomení Caesarovy šifry lze také použít tzv. frekvenční analýzu, která umožňuje některé symboly odhadnout na základě jejich statistického výskytu v přirozeném jazyce.

Je vidět, že Caesarova šifra je pro malou množinu znaků velmi snadno prolomitelná a tím pádem pro praktické problémy nevyužitelná.

2 Vernamova šifra

Vernamova šifra (anglicky *one-time pad*) spočívá v posunu každého znaku zprávy o náhodně zvolený počet míst v abecedě. Oproti Caesarově šifře tedy nemusí být shodné symboly posunuty vždy o stejný počet míst.

Vernamova šifra $\mathcal{E}$ je definována nad $(\{0, 1, \ldots, |\Sigma|-1\}^L, \Sigma^L, \Sigma^L)$ pro zvolenou délku L. Klíč je tedy L-tice čísel, kde člen na pozici i určuje počet míst v abecedě, o které posuneme znak zprávy na pozici i.

[9] Krajní případ, kdy $m = c$ uznáme jako platný, ikdyž by zřejmě nebyl prakticky využitelný.

[10] Intuitivně chápeme jako proces, díky kterému bude Eva schopna získat dešifrované zprávy.

[11] Tímto myslíme časovou složitost v nejhorším případě. Eva samozřejmě může v (pro ni) nejlepším případě klíč uhádnout hned na první pokus. Tomu samozřejmě nezabráníme žádnou šifrou. Můžeme včak vysokým počtem klíčů výrazně snížit pravděpodobnost, že se to stane.

Operace zašifrování a dešifrování jsou tedy definovány následovně: (předpokládejme, že $m_i = a_r$ a $c_j = a_s$)

$$E(k_i, m_i) = a_l, \text{ kde } l = (r + k_i) \mod |\Sigma|$$
$$D(k_j, c_j) = a_m, \text{ kde } m = (s - k_j) \mod |\Sigma|.$$

Obdobně jako u Caesarovy šifry bude zašifrování celé zprávy probíhat tak, že podle klíče k zašifrujeme všechny znaky m jednotlivě na nové a jejich spojením vznikne zašifrovaná zpráva c.

Existuje i binární varianta Vernamovy šifry, ve které jsou odesílané zprávy pouze sekvence bitů. Binární varianta je definována nad $(\{0,1\}^L, \{0,1\}^L, \{0,1\}^L)$. Fakt, že zprávy jsou sekvence bitů nás nijak neomezuje. Víme, že v praxi jsou všechny zprávy na nějaké úrovni reprezentovány sekvencí jedniček a nul.

Operace šifry jsou pak definovány takto:

$$E(k, m) = k \oplus m$$
$$D(k, c) = k \oplus c$$

(symbol $\oplus$ značí operaci XOR aplikovanou po bitech)

Obě verze šifry zřejmě splňují *podmínku korektnosti* (u binární varianty si stačí uvědomit, že $x \oplus x = 0^L$ pro každé $x \in \{0,1\}^L$).

2.1 Bezpečnost Vernamovy šifry

Pokud chceme zjistit, jak je Vernamova šifra bezpečná, zamysleme se nad tím, jak těžké ji bude prolomit. Eva by k získání původní zprávy m z c potřebovala zjistit klíč. Počet možných klíčů je počet binárních kódů délky L (těch je $|\Sigma|^L$).

K prolomení Vernamovy šifry je tedy potřeba čas $O(|\Sigma|^L)$. (Při použití binární varianty pro zprávu o velikosti 1 MB existuje $2^{8\times10^6}$ možných klíčů.)

Platí také, že Vernamova šifra je odolná vůči frekvenční analýze, pokud pro zašifrování každé další zprávy vybereme náhodně nový klíč. Za předpokladu, že klíč k je vybrán dokonale náhodně, a že klíč není použit opakovaně, je Vernamova šifra tzv. *dokonale bezpečná*.

3 Bezpečnost v teorii a praxi

Je jasné, že pro zajištění bezpečnosti šifry je nutné, aby byl k vybrán z dostatečně velké množiny $\mathcal{K}$. Potom bude totiž pro Evu těžší zjistit použitý klíč k.

Klíč k musí být z množiny $\mathcal{K}$ vybrán dokonale náhodně (pravděpodobnost výběru každého z klíčů musí být stejná). Pokud by tomu tak nebylo, Eva by přirozeně nejprve vyzkoušela nejvíce pravděpodobné klíče.

Je také nutné, aby c byla nezávislá na m a nijak s ní nesouvisela. Případná souvislost by Evě mohla zjednodušit získání m.

Poznámka 1

Riziko přináší také opakované použití stejného klíče. Pokud by napříč celou komunikací byl použit stejný klíč, útočník by mohl (například po náhodném získání klíče) zpětně dešifrovat každou zprávu, která komunikací prošla.

Dokonalé zabezpečení

Jako zlatý standard, nebo ideál bezpečnosti se uvádí takzvané *dokonalé zabezpečení.*[12]

Definice 2

Dokonalé zabezpečení

Nechť $\mathcal{E} = (E, D)$ je šifra definovaná nad trojicí $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Uvažujme pravděpodobnostní experiment, ve kterém je náhodná proměnná $\mathbf{k}$ rovnoměrně rozdělena na $\mathcal{K}$. Pokud pro každé $m_0, m_1 \in \mathcal{M}$, a každé $c \in \mathcal{C}$ platí:

$$\Pr[E(\mathbf{k}, m_0) = c] = \Pr[E(\mathbf{k}, m_1) = c]$$

nazýváme $\mathcal{E}$ dokonale bezpečnou šifrou.

Za předpokladu, že $\mathcal{E}$ je dokonale bezpečná a že každý klíč k má stejnou pravděpodobnost výběru z $\mathcal{K}$ lze ukázat, že zpráva $c = E(k, m)$ bude nezávislá na m, což jak víme, je žádoucí.

Věta 3

Vernamova šifra šifra je dokonale bezpečná.

Když tedy máme šifru, která je dokonale bezpečná, k čemu potřebujeme šifry ostatní? Důvodem je praxe.

Prvním problémem je dokonale náhodný výběr klíče. Současné generátory nejsou dokonale náhodné, ale pouze pseudonáhodné. To nám ale pro potřeby bezpečnosti nestačí. (Eva by mohla využít pseudonáhodnosti k snazšímu uhádnutí klíče.)

Tím druhým je paměťová náročnost. Pokud by si Alice s Bobem chtěli například poslat zprávu m o velikosti 1 GB, museli by být předem domluveni na klíči k stejné velikosti. Museli by tedy mít k uložený někde v paměti. Vzhledem k tomu, že by pro každou zprávu Alice s Bobem potřebovali nový klíč, není nutnost takové velikosti vhodná.

Následující věta nám ukazuje, že pokud chceme dosáhnout dokonalé bezpečnosti, musíme volit klíče alespoň stejné velikosti, jako jsou jimi šifrované zprávy. Tedy nedokážeme najít „stejně bezpečnou“ šifru, která by využívala klíče efektivněji než Vernamova šifra.

[12] V anglicky psané literatuře se nejčastěji používá pojem *perfect security.*

Věta 4 (Shannonova věta)
Nechť $\mathcal{E} = (E, D)$ je šifra definovaná nad $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Je-li $\mathcal{E}$ dokonale bezpečná, potom $|\mathcal{K}| \geq |\mathcal{M}|$.

Díky předchozí větě můžeme snadno tvrdit, že Caesarova šifra není dokonale bezpečná.

Zejména kvůli těmto dvěma problémům se v praxi vzdáváme jisté míry bezpečnosti za cenu toho, že jsme schopni zprávy šifrovat efektivněji.

4 Modulární aritmetika

[Obsah doplním následně po DH a RSA dle použitého obsahu, kterému tuto kapitolu přizpůsobím. Případně věty rozdělím přímo ke kapitolám, u kterých jsou využity.]

V této kapitole uvedeme několik definic a vět, které nám pomůžou v dalších pasážích tohoto textu. Na některé z nich budeme přímo odkazovat.

U čtenáře předpokládáme základní znalosti z teorie grup, kongruence modulo n a dělitelnosti.

Definice 5
Grupa $\mathbb{G}$ se nazývá cyklická, jestliže existuje prvek $g \in \mathbb{G}$ takový, že $\mathbb{G} = \{g^k \mid k \in \mathbb{Z}\}$. Prvek g se nazývá generátor cyklické grupy.

Definice 6
Řád prvku a v grupě $\mathbb{G}$ je nejmenší přirozené číslo b takové, že $a^b = e$, kde e je neutrálním prvkem grupy $\mathbb{G}$.

Věta 7 (Lagrangova věta)
Nechť $\mathbb{H}$ je podgrupa grupy $\mathbb{G}$. Pak $|\mathbb{H}|$ dělí $|\mathbb{G}|$.

Důsledek 8
Řád každého prvku $a \in \mathbb{G}$ dělí řád grupy $\mathbb{G}$.

Věta 9 (Cauchyho věta)
Nechť $\mathbb{G}$ je konečná grupa řádu n a p je prvočíslo. Pokud p dělí n, pak v G existuje prvek (a tedy i cyklická podgrupa) řádu p.

Důsledek 10
Každá grupa prvočíselného řádu je cyklická.

Věta 11
Nechť $\mathbb{G}$ je cyklická grupa řádu n, g její generátor a $b = g^a \in \mathbb{G}$. Pak b generuje cyklickou podgrupu $\mathbb{H}$ řádu n/d, kde d je $NSD(n, s)$.

Definice 12
Multiplikativní grupu nenulových zbytkových tříd modulo p budeme značit $\mathbb{Z}_p^*$.

Věta 13
Pro libovolné prvočíslo p je $\mathbb{Z}_p^*$ cyklická.

Definice 14
Neutrální prvek multiplikativní grupy budeme nazývat *jednička*.

Definice 15
Eulerova funkce $\varphi : \mathbb{N} \to \mathbb{N}$ přiřazuje každému přirozenému číslu n počet přirozených čísel menších nebo rovno n, která jsou s n nesoudělná, tedy

$$\varphi(n) = |K|, \text{ kde } K = \{k \in \mathbb{N} \mid k \leq n; NSD(k, n) = 1\}.$$

Věta 16
Každá konečná cyklická grupa řádu n má právě $\varphi(n)$ různých generátorů, kde φ je Eulerova funkce.

Důsledek 17
Každá konečná cyklická grupa prvočíselného řádu p má $p - 1$ různých generátorů.

5 Diffieho-Hellmanova výměna klíčů

V úvodu k symetrickému šifrování jsme zmínili, že výměna (respektive domluva) tajného klíče k mezi Alicí a Bobem tak, aby jej nezískala Eva, není jednoduchý úkol.

S pomocí některých znalostí, které jsme uvedli v sekci 4, zmíníme protokol, pomocí kterého to bude možné.

Jeho idea stojí na myšlence *jednosměrných funkcí*. Jednosměrná funkce F je taková funkce, kde pro každý vstup x lze snadno spočítat $F(x)$, ale z $F(x)$ nelze snadno zjistit původní x.[13] Jinými slovy, není snadné k funkci F najít inverzní funkci F^{-1}.

Celý protokol pak bude probíhat obecně takto:

- Alice náhodně vygeneruje svůj tajný klíč α a spočítá $H(\alpha)$. To stejné provede Bob pro svůj náhodně vygenerovaný tajný klíč β.
- Alice a Bob si přes síť vymění $H(\alpha)$ a $H(\beta)$.

[13] Jako praktický příklad se často uvádí smíchání dvou barev. Dvě různé barvy lze snadno smíchat a následně zjistit barvu, která vznikne. Z výsledné barvy samotné ale zřejmě není jednoduché zjistit barvy, jejichž smícháním vznikla.

- Alice s pomocí svého tajného klíče α a obdrženého $H(\beta)$ vypočítá $C(\alpha, \beta)$. Stejně učiní Bob se svým tajným klíčem β a obdrženého klíče $H(\alpha)$.
- Alice a Bob v komunikaci použijí $k = C(\alpha, \beta)$ jako jejich společný tajný klíč.

Aby protokol fungoval korektně a efektivně, požadujeme následující:

1. Pro každý vstup x lze $H(x)$ snadno spočítat.
2. Z α a $H(\beta)$ lze snadno spočítat $C(\alpha, \beta)$.
3. Z β a $H(\alpha)$ lze snadno spočítat $C(\alpha, \beta)$.
4. Z $H(\alpha)$ a $H(\beta)$ nelze snadno spočítat $C(\alpha, \beta)$.

Tyto podmínky implikují to, že H musí být jednosměrná funkce. Za splnění těchto podmínek platí, že Alice i Bob s pomocí protokolu efektivně získají stejný klíč k (to plyne přímo z podmínky 2 a 3).

Teď jen stačí nalézt vhodné funkce H a C tak, aby splňovaly podmínky.

Pro vhodně vybraný generátor g zvolme:

$$H(x) = g^x$$
$$C(x, y) = (g^x)^y.$$

Tyto funkce už zřejmě splňují podmínky 1–3. Abychom předešli tomu, že vygenerovaný klíč bude příliš velký (což jak víme není vhodné), budeme pracovat s adekvátní konečnou algebraickou doménou.

Ke splnění podmínky 4 tedy musí platit, že funkce H^{-1} je výpočetně náročná. Tou bude v našem případě funkce *diskrétního logaritmu*.

5.1 Protokol D-H

Alice a Bob se nejprve musí domluvit na velkém prvočísle p (to bude určovat potřebnou konečnou doménu). Prvočíslo p by mělo být alespoň 2048 bitů dlouhé. Zároveň by mělo platit, že existuje prvočíslo q (alespoň 256 bitů dlouhé), které dělí $p - 1$. K tomu, proč je existence q důležitá se budeme věnovat v podkapitole **??**.

Pro funkčnost protokolu potřebujeme, aby Alice i Bob znali generátor g grupy $\mathbb{Z}_p^*$ (viz 12). Nalezení generátoru grupy není obecně jednoduchý úkol. Budeme ale předpokládat, že g je parametr sdílený všemi uživateli v síti (i Evou).

Poznámka 18

Generátorem grupy $\mathbb{Z}_p^*$ **není** její libovolný prvek různý od jedničky. Grupa $\mathbb{Z}_p^*$ je řádu $p - 1$, neplatí pro ni tedy Důsledek 17.

1. Alice pošle Bobovi velké prvočíslo p. Domluvit se můžou i nezabezpečenou komunikací přes síť. Nevadí nám, že Eva p zachytí.

2. Alice náhodně vybere (velké) číslo $\alpha \in \mathbb{N}$, vypočítá $A = g^\alpha \mod p$ a A pošle po síti Bobovi.

3. Bob náhodně vybere (velké) číslo $\beta \in \mathbb{N}$, vypočítá $B = g^\beta \mod p$ a B pošle po síti Alici.

4. Alice vypočítá $k = B^\alpha \mod p$.

5. Bob vypočítá $k = A^\beta \mod p$.

Všimněme si, že $B^\alpha \equiv g^{\beta\alpha} \pmod p$ a $A^\beta \equiv g^{\alpha\beta} \pmod p$. Z komutativity násobení plyne, že $g^{\beta\alpha} = g^{\alpha\beta}$. Z tohoto vyplývá, že Alice a Bob nezávisle na sobě získají stejný klíč k.

5.2 Bezpečnost D-H výměny klíčů

Celou komunikaci na síti poslouchá Eva. Z návrhu našeho protokolu víme, že Eva zachytila p, A (tedy g^α) a B (tedy g^β). Navíc zná generátor g. Jestliže chce Eva šifrované zprávy dešifrovat, musí získat k. Musí tedy zjistit $g^{\alpha\beta}$ z g, g^α, g^β (aniž by znala α nebo β). Tomuto problému budeme říkat *Diffieho-Hellmanův problém* (zkráceně DHP).

Platí, že Eva je schopna vyřešit DHP, pokud umí vyřešit tzv. *problém diskrétního logaritmu*. Ikdyž opačná implikace zatím nebyla dokázána, panuje shoda, že oba zmíněné problémy jsou ekvivalentní.

Bezpečnost D-H výměny klíčů se tedy výrazně opírá o složitost řešení problému diskrétního logaritmu. Tomu, za jakých podmínek jej považujeme za obtížně řešitelný (a tedy D-H výměnu jako bezpečnou), se budeme věnovat v následující kapitole.

6 Problém diskrétního logaritmu

V podkapitole 5.2 jsme představili problém DHP. Úkolem bylo zjistit $g^{\alpha\beta}$ z g, g^α, g^β bez znalosti α a β.

Pokud bychom byli schopni z g^α efektivně získat α (případně z g^β získat β), uměli bychom také efektivně řešit DHP.

Definice 19 (Problém diskrétního logaritmu (DLP))
Nechť $\mathbb{G}$ je cyklická grupa řádu n, g její generátor a x libovolný prvek z $\mathbb{G}$. Číslo $e \in \mathbb{Z}_n$ takové, že

$$g^e \equiv x \pmod n. \tag{1}$$

nazveme diskrétním logaritmem o základu g z x.

Nalezení takového čísla e budeme nazývat **problém diskrétního logaritmu**.

Existuje řada grup $\mathbb{G}$, u nichž předpokládáme, že je v nich problém diskrétního logaritmu obtížně řešitelný. Na tomto předpokladu stojí bezpečnost řady šifrovacích protokolů, včetně Diffieho-Hellmanovy výměny klíčů (jak jsme uvedli v podkapitole 5.2).

Definice 20 (Předpoklad diskrétního logaritmu)
Předpoklad diskrétního logaritmu platí pro cyklickou grupu $\mathbb{G}$, právě tehdy když je pravděpodobnost správného určení diskrétního logaritmu zanedbatelná.

V následujících podkapitolách se zaměříme na volbu grupy $\mathbb{G}$ a vliv jejích vlastností na obtížnost DLP. Předem jen uveďme, že obtížnost DLP je úzce provázána s prvočísly.

6.1 Obtížnost DLP

V současnosti se za vhodnou volbu $\mathbb{G}$ považuje grupa řádu $n = p$, kde p je velké prvočíslo (alespoň 2048 bitů dlouhé). Pro p také musí platit, že číslo $p - 1$ má alespoň jednoho velkého prvočíselného dělitele q (alespoň 256 bitů).

Poznámka 21
Prvočíselný řád grupy $\mathbb{G}$ nám automaticky zaručuje cykličnost (viz Důsledek 10).

Složitost nalezení diskrétního logaritmu x v grupě řádu p, kde p má k číslic, je prakticky stejná jako faktorizace k-ciferného celého čísla. Díky tomu můžeme říct, že výpočet diskrétních logaritmů je zhruba stejně obtížný jako faktorizace. Pro faktorizaci (stejně jako pro DLP) není znám žádný polynomiální algoritmus, který by ji řešil.

Je třeba zdůraznit, že jsme zatím ani nedokázali, že takový algoritmus neexistuje.[14]

6.2 Výpočet diskrétního logaritmu

6.2.1 Hrubá síla

Nejjednodušším způsobem k nalezení diskrétního logaritmu je hrubá síla. Algoritmus pracující tímto přístupem jednoduše vyzkouší všechny možné exponenty, dokud nenajde exponent e vyhovující (1).

Takto navržený algoritmus je korektní. V nejhorším případě potřebuje provést n násobení v grupě $\mathbb{G}$.

[14]Tento fakt souvisí s problémem P versus NP. Ten patří k nejznámějším otevřeným problémům teoretické informatiky.

```
function brute_force_dlog(base, result, modulus):

    exponent = 0

    while result != current_result:

        current_result = pow(base, exponent, modulus)

        exponent += 1
```

Zdrojový kód 1: Nalezení diskrétního logaritmu hrubou silou

6.2.2 Rekurzivní algoritmus

Nechť $\mathbb{G}$ je grupa řádu q^y s generátorem g, kde $q > 1, y \geq 1$. Algoritmus, který v této podkapitole uvedeme, nám umožní redukovat řešení DLP v této grupě na řešení DLP v grupě řádu q.

Nalezení diskrétního logaritmu o základu g z x probíhá rekurzivně.

Ukončovací podmínkou bude případ, kdy $y = 1$. V tomto případě k výpočtu použijeme algoritmus z podkapitoly 6.2.1.

V případě $y > 1$ zvolíme $z = \lfloor y/2 \rfloor$. Diskrétní logaritmus e lze vyjádřit jako $e = q^z v + u$, pro $u, v \in \mathbb{N}$, kde $0 \leq u < q^z$ a $0 \leq v < q^{y-z}$.

Potom platí

$$x = g^e = g^{(q^z)v+u} = g^{(q^z)v} \cdot g^u. \tag{2}$$

Umocněním obou stran na q^{y-z} získáme

$$x^{q^{y-z}} = g^{q^{y-z}u}.^{15} \tag{3}$$

Dle Věty 11 platí, že $g^{q^{y-z}}$ je řádu q^z. Můžeme tedy rekurzivně spočítat diskrétní logaritmus o základu $g^{q^{y-z}}$ z $x^{q^{y-z}}$. Tímto způsobem můžeme rekurzivně pokračovat, dokud nenarazíme na ukončovací podmínku a nezískáme u.

Vydělením obou stran rovnice (2) hodnotou g^u získáme

$$x/g^u = g^{q^z v}. \tag{4}$$

Opět z Věty 11 plyne, že g^{q^z} je řádu q^{y-z}. Rekurzivně tedy můžeme spočítat diskrétní logaritmus o základu g^{q^z} z x/g^u, dokud nezískáme v.

Po získání u a v snadno vypočítáme $x = q^z v + u$.

6.2.3 Silver-Pohlig-Hellmanův algoritmus

Tento algoritmus nám kromě dalšího způsobu řešení problému DLP také ukazuje zajímavý vztah mezi faktorizací řádu grupy $\mathbb{G}$ a obtížností DLP v ní.

[15] Protože $g^{(q^z)v}$ je řádu q^{y-z}. Umocněním tedy dle definice získáme jednotkový prvek, který se vzhledem k násobení chová neutrálně.

```
function recursive_dlog(base, result, q, y, modulus):

   if y = 1:
      return brute_force_dlog(base, result, modulus)

   z = y / 2

   pow(base, exponent, modulus)

   u_base = pow(base, q ** (y-z), modulus)
   u_result = pow(result, q ** (y-z), modulus)
   u = recursive_dlog(u_base, u_result, q, z, modulus
       )


   v_base = pow(base, q ** z, modulus)
   v_result = result / pow(base, u, modulus)
   v = recursive_dlog(v_base, v_result, q, y-z,
       modulus)

   return (q ** z) * v + u
```

Zdrojový kód 2: Nalezení diskrétního logaritmu rekurzivním algoritmem

Věta 22 (Základní věta aritmetiky)
Každé přirozené číslo větší než jedna lze vyjádřit jednoznačně až na pořadí činitelů jako součin prvočísel.

Nechť n je řád grupy $\mathbb{G}$. Pak dle Věty 22 existuje rozklad

$$n = q_1^{e_1} \cdots q_r^{e_r}. \tag{5}$$

Algoritmus využívá algoritmu RDL z podkapitoly 6.2.2 a *Čínské věty o zbytcích*. Jeho konkrétní implementace je uvedena v příloze této práce.

Pokud je faktorizace (5) dána, časová složitost Silver-Pohlig-Hellmanova algoritmu je

$$O(\sum_{i=1}^{r} a_j(\ln n + \sqrt{q_{max}})) = O(\sqrt{q_{max}} \cdot \ln n),^{16} \tag{6}$$

kde hodnota q_{max} značí největší prvočíslo, které dělí n.

Ze složitosti (6) získáváme několik důležitých faktů.

[16] Předpokládáme, že operaci násobení v grupě provádíme v konstantním čase. Přesná časová složitost samozřejmě vždy závisí na konkrétní implementaci.

Věta 23
Složitost výpočtu diskrétního logaritmu v cyklické grupě řádu n je dána velikostí největšího prvočísla, které dělí n.

Tato věta jasně určuje, v jakých grupách platí předpoklad diskrétního logaritmu (viz 20).

Důsledek 24
Aby platil předpoklad diskrétního logaritmu v grupě $\mathbb{G}$, její řád musí mít alespoň jednoho velkého prvočíselného dělitele.

Pokud bychom například chtěli spočítat diskrétní logaritmus v grupě $\mathbb{G}$ s řádem $n = 2^l$, bude nám stačit provést $O(\ln n)$ násobení v grupě.

Kdyby se tedy například Alice s Bobem u D-H výměny klíčů domluvili na prvočísle p ve tvaru $p = 2^l + 1$, Evě by pro prolomení celé komunikace stačilo spočítat diskrétní logaritmus v grupě $\mathbb{Z}_p^*$. Grupa $\mathbb{Z}_p^*$ by v tomto případě byla řádu 2^l. Výpočet diskrétního logaritmu (a tedy i získání tajného klíče k) by tedy pro Evu byl snadný úkol.

Poznámka 25 (Zefektivnění protokolu D-H)
Složitost Silver-Pohlig-Hellmanova algoritmu také dává návod pro zefektivnění D-H protokolu. Namísto spuštění D-H protokolu v celé grupě $\mathbb{Z}_p^*$ stačí protokol spustit v podgrupě $\mathbb{G}_{q_{max}}$ řádu q_{max}.
Dle Věty 23 takovéto omezení bude mít pouze zanedbatelný vliv na bezpečnost protokolu.
Alici i Bobovi tímto omezením zmenšíme množinu exponentů, ze kterých budou vybírat své tajné klíče.

6.3 Kvantové útoky na DLP

[TODO]

7 Napadení protokolu D-H trochu jinak

V části I jsme uvedli, že Eva umí komunikaci proudící po síti pouze číst. Eva tedy nemůže zprávy posílat, mazat, ani modifikovat. Pokud se v síti nachází protivník, který takové schopnosti má, stává se námi uvedený protokol snadno napadnutelným. Protivníka, který bude mít schopnost posílat, mazat a modifikovat zprávy v síti, nazveme Mallory.[17]

Ukažme si, jak by Mallory mohl komunikaci mezi Alicí a Bobem jednoduše

[17] Jméno Mallory (z angl. *malicious attacker*) se nejčastěji používá jako označení útočníka, který je (oproti Evě) aktivní.

napadnout. Alice s Bobem se chtějí domluvit na tajném klíči k (kterým budou šifrovat zprávy) pomocí protokolu D-H.

- Alice pošle Bobovi velké prvočíslo p. Mallory p zachytí a zapamatuje si ho.
- Alice náhodně vybere (velké) číslo $\alpha \in \mathbb{N}$, vypočítá $A = g^\alpha \mod p$ a A pošle po síti Bobovi.
- Mallory zprávu A zachytí a nepošle ji dále Bobovi. Místo toho vybere velké číslo $\gamma \in \mathbb{N}$, vypočítá $C = g^\gamma \mod p$ a C pošle po síti Bobovi.
- Bob náhodně vybere (velké) číslo $\beta \in \mathbb{N}$, vypočítá $B = g^\beta \mod p$ a B pošle po síti Alici.
- Mallory zprávu B zachytí a nepošle ji Alici. Místo toho Alici pošle C.
- Alice vypočítá $k_1 = C^\alpha \mod p$. Bob vypočítá $k_2 = C^\beta \mod p$.
- Mallory vypočítá $k_1 = A^\gamma \mod p$ a $k_2 = B^\gamma \mod p$.

Alice a Bob si nyní můžou myslet, že se dohodli na společném klíči, pomocí kterého povedou zabezpečenou komunikaci. Ve skutečnosti bude ale jejich veškerou komunikaci číst (případně měnit) Mallory, aniž by o tom Alice s Bobem věděli. Ukažme si to na příkladu komunikace, která probíhá po výše zmíněné výměně klíče.

Alice pomocí k_1 zašifruje zprávu m a pošle c po síti Bobovi. Mallory c zachytí a pomocí k_1 ji dešifruje. Zprávu m si nyní může přečíst a kompletně změnit její obsah. Upravenou zprávu m' zašifruje pomocí k_2 a c' pošle Bobovi.[18] Bob zprávu c' dešifruje pomocí k_2.

Problém je jistě v autentifikaci zpráv. Alice ani Bob nemají možnost zjistit, že byla zpráva někým upravena. Bob nemá nikdy jistotu, že obdrženou zprávu poslala opravdu Alice (a naopak).

Šifrovací metody, které autentifikaci zajišťují (a budou tedy komunikaci chránit i před Mallorym), představíme v části III.

[18] Pokud Mallory zprávy pouze čte, potom zřejmě $m = m'$.

Část III
Asymetrické šifrování

Asymetrické šifrování je způsob šifrování, ve kterém je v procesu zašifrování zprávy použit odlišný klíč, než při procesu dešifrování.

Podobně jako v části II budeme hledat řešení základního problému uvedeného v části I. Později navíc představíme systémy, které budou odolné také proti silnějšímu protivníkovi Mallorymu.

Nejprve uveďme definici asymetrické šifry.

- E je pravděpodobnostní algoritmus zašifrování. E přijímá na vstupu klíč k a zprávu m. Jako výstup vrací zašifrovaný text c.

  $$E : \mathcal{K} \times \mathcal{M} \to \mathcal{C},$$

  kde $\mathcal{K}$ je množina klíčů, $\mathcal{M}$ je množina zpráv a $\mathcal{C}$ je množina šifrovaných zpráv.

- D je deterministický algoritmus dešifrování. D přijímá na vstupu klíč k a zašifrovaný text c. Jako výstup vrací dešifrovanou zprávu m.

  $$D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$$

Každá $\mathcal{E}$ je tedy definována nad $(\mathcal{K}, \mathcal{M}, \mathcal{C})$.

Podobně jako u symetrické šifry označme množinu zpráv $\mathcal{M}$, a množinu šifrovaných zpráv $\mathcal{C}$. Každá $\mathcal{E}$ je definována nad $(\mathcal{M}, \mathcal{C})$.

Stejně jako u symetrického šifrování budeme požadovat, aby byl Bob dešifrováním c schopen získat stejnou zprávu m, kterou mu Alice poslala. Bude tedy muset platit *podmínka korektnosti*.

$$D(k_s, E(k_p, m)) = m$$

Poznámka 26

Všimněme si rozdílu oproti podmínce korektnosti u symetrického šifrování. Rozdíl přímo vyplývá z definice asymetrického šifrování.

7.1 Volba algoritmů zašifrování a dešifrování

8 Klíčový pár

Dvojici klíčů k_s (soukromý klíč) a k_p (veřejný klíč), které zaručují splnění podmínky korektnosti budeme říkat *klíčový pár*. [19]

[19] Indexy s a p jsou z angl. *secret* a *public*.

Už pojmenování klíčů nám říká, že k_s bude chtít uživatel udržet v tajnosti a k_p bude moct veřejně sdělit.

Průběh poslání zprávy by tedy mohl vypadat následovně:

- Bob vygeneruje svůj klíčový pár k_s a k_p. Soukromý klíč si uchová v tajnosti a veřejný klíč zveřejní. Například jej nahraje na internet, případně jej Alici pošle po síti.
- Alice pomocí k_p zašifruje zprávu m na $c = E(k_p, m)$ a c pošle po síti Bobovi.
- Bob s použitím svého soukromého klíče spočítá $D(k_s, c)$ a získá původní zprávu m.

Jelikož je Bobův veřejný klíč k_p zpřístupněn všem uživatelům v síti, každý uživatel může pomocí něj zašifrovat zprávu a poslat ji Bobovi. Bezpečnost celého systému se potom opírá o fakt, že šifrovaná zpráva lze dešifrovat pouze pomocí soukromého klíče k_s. A jelikož si Bob svůj soukromý klíč uchoval v tajnosti, bude šifrované zprávy moct dešifrovat pouze on.

Poznámka 27
Opět si všimněme rozdílu oproti symetrickému šifrování. U něj měl každý, kdo mohl zprávy zašifrovat, možnost zprávy také dešifrovat (stejným klíčem).

To, že dešifrování zprávy může provést vlastník soukromého klíče k_s nám již poskytuje základní úroveň bezpečnosti. Oproti symetrickému šifrování, nemusela proběhnout žádná domluva mezi Alicí a Bobem. Soukromý klíč (ani žádná informace o něm) přes síť nebyl poslán. Tím přirozeně snižujeme riziko, že jej někdo odhalí.

Zároveň, pokud bude libovolný jiný uživatel v síti, Carlos, chtít poslat zprávu Bobovi, nebude se s Bobem muset na ničem domlouvat. Pouze použije volně přístupný veřejný klíč k_p k zašifrování zprávy, kterou pošle Bobovi. Carlos bude nevíc mít jistotu, že zprávu zpětně dešifruje pouze vlastník soukromého klíče k_s, tedy Bob. [20]

[20] Místo názvu *asymetrické šifrování* se často používá název *šifrování s veřejným klíčem* (anglicky *public-key cryptography*).