INFORMATIKA

Moderní šifry II

EDUARD BARTL

Přírodovědecká fakulta UP, Olomouc

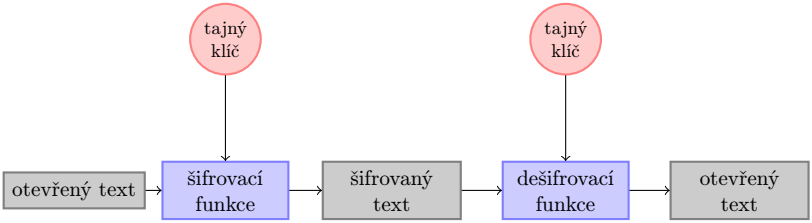
Předchozí díl série článků o šifrování byl zakončen výkladem jednosměrných šifrovacích funkcích se zadními vrátky. Tyto funkce jsou důležité proto, že pomocí nich jsme schopni realizovat asymetrické šifrování. V závěru jsme si řekli, že pro další výklad bude užitečné seznámit se s takzvanou modulární aritmetikou. Tato aritmetika se od běžné aritmetiky v ledasčem liší, v tomto pokračování se ji proto pokusíme podrobně vysvětlit. Dále se vrátíme k asymetrickému šifrování, vysvětlíme si, jak funguje šifra RSA a jak se používá v praktických aplikacích (například v elektronickém podpisu). Článek je doplněn dvěma řešenými úkoly, na kterých si může čtenář ověřit pochopení textu.

Na úvod krátké opakování

V předešlém dílu jsme si ukázali princip asymetrického šifrování. Než se podíváme na slíbenou modulární aritmetiku, kterou budeme potřebovat pro podrobný výklad asymetrické šifry RSA, zopakujeme si, jak asymetrické šifrování funguje a jaký je jeho vztah k symetrickému šifrování.

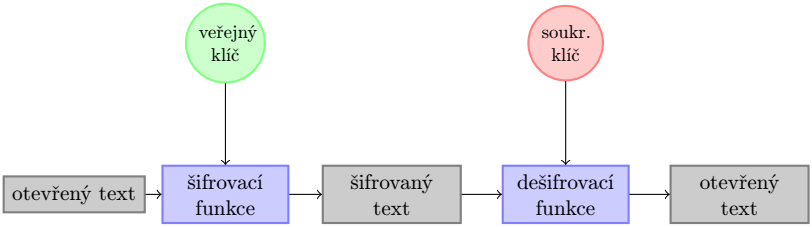
Princip symetrického šifrování můžeme znázornit pomocí diagramu na obr. 1. Podstatné na symetrickém šifrování je skutečnost, že odesilatel (Alice) i příjemce (Bob) disponují stejným klíčem, pomocí kterého Alice zašifruje otevřený text a Bob pomocí něj zašifrovaný text opět dešifruje. Pokud nějaká třetí osoba (Eva) získá tento klíč, může dešifrovat libovolnou zachycenou zprávu. Z tohoto důvodu je nutné tento klíč udržet v tajnosti. Proto je tento klíč nazýván *tajným*.

Problematické na symetrickém šifrování je zejména to, že se Alice i Bob musí na tajném klíči dohodnout. Tento problém velmi elegantně řeší



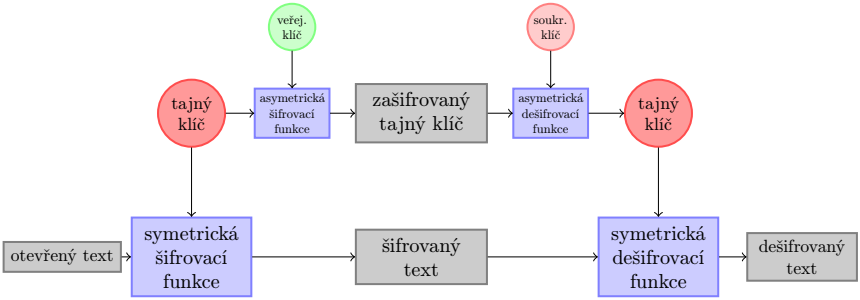
Obr. 1 Schéma symetrického šifrování

asymetrické šifrování, jehož princip je schematicky ukázán na obr. 2. Tento způsob šifrování již nepoužívá jediný klíč, ale používá klíče dva – jeden k šifrování, druhý k dešifrování. Klíč určený k šifrování může být zveřejněn (proto se mu také říká *veřejný klíč*), naopak klíč určený k dešifrování musí být udržen v soukromí (na což poukazuje jeho název – *soukromý klíč*). Přestože jsou tyto dva klíče různé, existuje mezi nimi určitý vztah. Veřejný klíč musí být snadno a rychle odvoditelný ze soukromého klíče; odvodit soukromý klíč z veřejného však musí být velmi obtížné, časově neschůdné. Právě tomuto vztahu bude v tomto dílu věnována pozornost.



Obr. 2 Schéma symetrického šifrování

Připomeňme také krátce, že v praktických aplikacích se oba přístupy – symetrické šifrování a asymetrické šifrování – vhodným způsobem kombinuje. Šifrování dat se provádí symetricky, asymetrické šifrování se používá pouze pro výměnu tajného klíče tak, jak ukazuje schema na obr. 3. Tento přístup se volí proto, že symetrické šifrovací algoritmy jsou řádově rychlejší než asymetrické. Tím, že se objemná data šifrují symetricky a krátký tajný klíč asymetricky, je tak celý šifrovací proces významným způsobem urychlen.

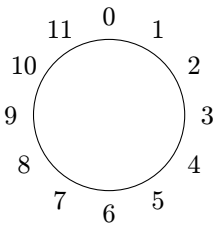


Obr. 3 Použití asymetrického šifrování při výměně tajného klíče

Počítání v kruhu

Pokračujme nyní ve slíbeném výkladu modulární aritmetiky. Této aritmetice se někdy říká *hodinová aritmetika*. Příměr s hodinami je trefný a umožní nám rychle pochopit podstatu věci.

Představme si klasický hodinový ciferník, jenž ukazuje čas v dvanáctihodinovém cyklu. Tento ciferník však mírně upravíme: místo, aby ukazoval hodiny od 1 do 12, bude ukazovat hodiny od 0 do 11. Čísla od 1 do 11 tedy zůstanou na svých místech, pouze číslo 12 nahradíme číslem 0, jak můžeme vidět na obr. 4. Tato změna nikterak neovlivní způsob počítání, na který jsme zvyklý; cyklus je stále dvanáctihodinový, pouze budeme *dvanácté* hodině říkat *nultá* hodina.



Obr. 4 Dvanáctihodinový ciferník

Počítání na takovém ciferníku je nám dobře známé. Pokud například dvouhodinové divadelní představení začíná v 11 hodin, pak jeho konec (počítáno v dvanáctihodinovém cyklu) není v 13 hodin, ale v 1 hodinu,

můžeme tedy napsat $11 \oplus 2 = 1$; pro operaci sčítání v hodinové aritmetice jsme zvolili symbol $\oplus$, abychom ji odlišili od operace sčítání $+$, kterou běžně používáme.[1)]

Jak by se dala popsat operace sčítání $\oplus$ v hodinové aritmetice pomocí běžného sčítání $+$? Podle předchozího příkladu se začátkem a koncem divadelního představení by se mohlo zdát, že jakmile dojde běžným součtem k „přetečení“, tak jednoduše od výsledku odečteme číslo 12, čímž se vrátíme zpět na číslo zobrazitelné na ciferníku. Mohli bychom tedy napsat:

$$11 \oplus 2 = 11 + 2 - 12 = 1.$$

Takto pojatá definice operace $\oplus$ však nebude fungovat vždy. K „přetečení“ totiž může dojít několikanásobně, odečtení čísla 12 pak k návratu na ciferník nepomůže. Například u součtu $11 \oplus 16$ musíme číslo 12 odečíst dvakrát:

$$11 \oplus 16 = 11 + 16 - 2 \cdot 12 = 3.$$

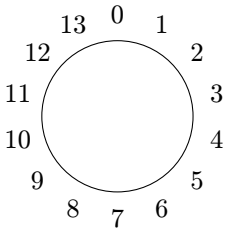
Čtenář již možná tuší, že správnou definici operace $\oplus$ pomocí operace $+$ je možné vyslovit s použitím *zbytku po celočíselném dělení*: součet $a \oplus b$ je roven zbytku po vydělení běžného součtu $a + b$ číslem 12. Skutečně, $11 + 16 = 27$; tento součet vydělíme číslem 12, podílem je pak 2, zbytek je 3. Platí tedy $11 \oplus 16 = 3$, jak jsme ostatně viděli výše. Operace odečítání se definuje obdobně: rozdíl $a \ominus b$ je roven zbytku po vydělení běžného rozdílu $a - b$ číslem 12.

Je celkem zřejmé, že operace $\oplus$ a $\ominus$ se dají jednoduše zobecnit i pro *ciferníky s jiným počtem cifer než je 12*. Jediný rozdíl bude v tom, že budeme uvažovat zbytek po dělení číslem, které udává počet cifer na ciferníku. Tomuto počtu se v hodinové aritmetice říká *modul* (proto této aritmetice říkáme modulární).

Pokud tedy bude mít ciferník například 14 cifer, jak můžeme vidět na obr. 5, pak dopadnou součty $11 \oplus 2$ a $11 \oplus 16$ zcela jinak. Konkrétně dostaneme, že $11 \oplus 2 = 13$, protože $11 + 2 = 13$ děleno 14 je 0, zbytek je 13. Dále pak můžeme psát, že $11 \oplus 16$ je opět 13, poněvadž $11 + 16 = 27$ děleno 14 je 1, zbytek je 13.

Protože hodnota modulu určuje, jakým číslem v definici operace $\oplus$ a $\ominus$ dělíme, bude jistě praktické dát tuto hodnotu jasně najevo. Uděláme to tak, že ji napíšeme do indexu symbolů $\oplus$ a $\ominus$. Budeme tedy například

[1)] Kolečko v symbolu $\oplus$ nám bude připomínat, že se jedná o sčítání v modulární aritmetice – tedy o „sčítání v kruhu“.



Obr. 5 Čtrnáctihodinový ciferník

psát:

$$11 \oplus_{12} 2 = 1,$$
$$11 \oplus_{12} 16 = 3,$$
$$11 \oplus_{14} 2 = 13,$$
$$11 \oplus_{14} 16 = 13.$$

Tím se vracíme zpět ke vztahům z předchozího dílu, které definují šifrovací a dešifrovací funkci posouvací šifry. Při výkladu této šifry jsme se domluvili na kódování dané tabulkou 1. Pracujeme tedy s čísly x, y a k,

znak	`a`	`b`	`c`	`d`	...	`y`	`z`	␣
kódové číslo	0	1	2	3	...	24	25	26

Tabulka 1 Jednoduché kódování písmen abecedy a mezery ␣

která mohou nabývat celkem 27 hodnot, konkrétně hodnot 0 až 26. Zmíněnou šifrovací funkci e a dešifrovací funkci d posouvací šifry můžeme – teď již zcela správně – napsat následujícím způsobem:

$$e(x, k) = x \oplus_{27} k, \tag{1}$$
$$d(y, k) = y \ominus_{27} k. \tag{2}$$

Pro úplnost dodejme, že operace výpočtu zbytku po celočíselném dělení je natolik užitečná, že se vyskytuje snad ve všech vyšších programovacích jazycích. Například v jazyce C je možné tento zbytek vypočítat pomocí operátoru `%`. Šifrovací funkci (1) posouvací šifry bychom tedy mohli definovat tak, jak ukazuje následující programový kód:

```
/* sifrovaci funkce posouvaci sifry */
short encipher(short x, short k)
{
   return (x + k) % 27;
}
```

Pro další výklad bude potřeba zamyslet se nad tím, jak v modulární aritmetice fungují i jiné operace, konkrétně nám půjde o násobení, umocnění a dělení. Pro tyto tři operace budeme používat po řadě symboly $\odot$, $\oslash$ a $\oslash$ (opět s dolním indexem určujícím modul, jak uvidíme za chvíli).

U prvních dvou zmíněných operací není celkem co řešit, situace je totiž analogická se sčítáním a odečítáním. Násobek $a \odot_n b$ je tak roven zbytku po vydělení čísla $a \cdot b$ číslem n. Proto například platí $3 \odot_{10} 5 = 5$. Podobně jednoduché je i umocnění: $a \oslash_n b$ je zbytek po vydělení čísla a^b číslem n; lehce tak například zjistíme, že $2 \oslash_{10} 6 = 4$.

Vypočítat podíl v modulární aritmetice ovšem tak přímočaré není. V běžné aritmetice definujeme dělení pomocí násobení, platí totiž například, že 20 děleno 5 je totéž jako 20 krát $\frac{1}{5}$. Obecně můžeme napsat, že b děleno a je rovno součinu b a „převrácené“ hodnoty k číslu a (samozřejmě za předpokladu, že a nebude nula). K tomu, abychom převedli tuto myšlenku do modulární aritmetiky, musíme dobře porozumět tomu, co se myslí zmíněnou převrácenou hodnotou k danému číslu.

Převrácená hodnota k číslu a je takové číslo, které vynásobeno (ať už zleva nebo zprava) číslem a dá jedničku.[2)] V běžné aritmetice je převrácenou hodnotou k číslu 5 skutečně zlomek $\frac{1}{5}$, protože

$$5 \cdot \frac{1}{5} = \frac{1}{5} \cdot 5 = 1.$$

Podobně v modulární aritmetice definujeme podíl $b \oslash_n a$ jako součin $b \odot_n A$, kde A značí převrácenou hodnotu k a. Otázkou ovšem je, jestli k číslu a převrácená hodnota A existuje, a pokud ano, jak ji vypočítat.[3)]

Zkusme si situaci představit na jednoduchém příkladu. Budeme uvažovat modulární aritmetiku s modulem $n = 12$, to znamená, obyčejné

[2)] Jedničku proto, že právě jednička se při násobení chová *neutrálně*, to znamená, že nemění výsledek součinu. Přesněji řečeno, pro jedničku a libovolné nenulové číslo a platí, že $a \cdot 1 = 1 \cdot a = a$.

[3)] Nezapomeňme, že v modulární aritmetice nemáme žádné zlomky. Převráceným číslem k číslu 5 jistě nebude $\frac{1}{5}$ už proto, že žádné takové číslo v modulární aritmetice (na ciferníku) nemáme.

hodiny s dvanáctihodinovým cyklem, které vidíme na obr. 4. Ptáme se, jestli k číslu 3 existuje převrácená hodnota. Formulováno v řeči hodin se tedy ptáme, jestli se poskládáním několika čtvrthodinovek dostaneme na pět minut po celé (číslo 1). Na první pohled je jasné, že se nám to nepodaří. Pomocí čtvrthodiny (čísla 3) se dostaneme pouze na půlhodinu (číslo 6), na třičtvrtěhodinu (číslo 9) nebo na celou (číslo 0). K číslu 3 tedy převrácená hodnota neexistuje. Podobně dopadneme i s čísly 2, 4 a 6 – ani k nim neexistuje převrácená hodnota. Můžeme tedy snadno vypozorovat, že pokud nějaké číslo *dělí modul beze zbytku*, pak k tomuto číslu neexistuje převrácená hodnota.

Právě navržené kritérium však není úplné, protože neříká, jak to dopadne v případě, kdy dané číslo *nedělí modul beze zbytku*. Tak například k číslu 9, které nedělí modul 12 beze zbytku, také neexistuje převrácená hodnota. Důvodem je, že třičtvrtěhodina (číslo 9) je poskládána ze tří čtvrthodin (čísel 3), ke kterým, jak už víme, převrácená hodnota neexistuje. Celkem tedy můžeme vyslovit následující tvrzení:

Tvrzení 1.
K danému číslu existuje převrácená hodnota, právě když je největším společným dělitelem tohoto čísla a modulu jednička (to znamená, toto číslo je s modulem nesoudělné*).*

Pokud převrácená hodnota k danému číslu existuje, pak se dá vypočítat takzvaným rozšířeným Euklidovým algoritmem. Tento algoritmus zde nebudeme popisovat, pro účely tohoto výkladu plně postačí, když vyzkoušíme všechny možné kombinace. V předchozím příkladu s dvanáctihodinovým ciferníkem tak dojdeme například ke zjištění, že převrácená hodnota k číslu 5 je opět číslo 5, protože $5 \odot_{12} 5 = 1$ (neboli $5 \cdot 5$ děleno 12 dává zbytek 1), viz také tabulka 2.

a	0	1	2	3	4	5	6	7	8	9	10	11
A	–	1	–	–	–	5	–	7	–	–	–	11

Tabulka 2 Převrácená hodnota A k číslu a v modulární aritmetice s modulem 12

Abychom počítání v modulární aritmetice dostatečně zažili, vyzkoušíme si na závěr této sekce několik jednoduchých příkladů. Vypočtěme následující příklady (výsledky jsou uvedeny na konci tohoto článku):

$$15 \ominus_{17} (10 \oplus_{17} 14 \oplus_{17} 0) = ?$$
$$14 \odot_{25} 24 = ?$$
$$3 \oslash_9 3 = ?$$
$$20 \oslash_6 5 = ?$$
$$10 \oslash_{10} 5 = ?$$

Nalezli jsme konečně jednosměrnou funkci?

Na konci předchozího dílu jsme prozradili, že kandidátem na jednosměrnou funkci se zadními vrátky je umocnění v modulární aritmetice. Nyní, když modulární aritmetiku docela dobře ovládáme, jsme schopni tuto funkci přesně popsat. Je-li tedy x číslo, které šifrujeme, k_e veřejný klíč a n modul, pak můžeme psát:

$$e(x, k_e) = x \oslash_n k_e.$$

V dalším výkladu budeme uvažovat nějaké konkrétní nastavení – veřejným klíčem k_e bude číslo 2 a pro modul bude platit $n = 10$ (počítat tedy budeme na ciferníku s deseti ciframi). Šifrovací funkce pak vypadá takto:

$$e(x, 2) = x \oslash_{10} 2. \tag{3}$$

Dále předpokládejme, že zašifrované číslo, které se podaří Evě získat, je číslo 9. Důvod, proč šifrovací funkce založená na běžném umocnění, tedy $e(x, 2) = x^2$, není jednosměrnou funkcí, jsme objasnili už v předchozím dílu. Zopakujme si, že číslo x, pro které platí, že $9 = x^2$ můžeme rychle získat metodou půlení intervalu: Nejprve zkusíme x uhádnout; řekněme, že prvním pokusem bude $x = 4$. Jelikož je ale 4^2 větší než požadované zašifrované číslo 9, odhad čísla x musíme zmenšit. Druhým pokusem tedy bude $x = 2$. Pokud tuto hodnotu dosadíme do šifrovací funkce, získáme 2^2, což je naopak menší číslo než 9. Dále zkusíme $x = 3$, čímž se dostáváme ke správnému výsledku.

Pokud bychom chtěli stejnou metodu využít i ve světě modulární aritmetiky, narazíme na značné obtíže. Začneme-li stejným počátečním odhadem $x = 4$ a dosadíme-li tento odhad do šifrovací funkce (3), obdržíme číslo $4 \oslash_{10} 2 = 6$ (zbytek po vydělení čísla 4^2 číslem 10 je 6). Dostali jsme výsledek, který je menší než požadované zašifrované číslo 9. Metoda půlení intervalu v tomto případě říká, že je potřeba odhad zvýšit. My ovšem

víme, že správným výsledkem je číslo menší, konkrétně $x = 3$, protože $3 \circledcirc_{10} 2 = 9$.

Můžeme tedy vypozorovat, že to, co dělá z šifrovací funkce $e(x, k_e) = x \circledcirc_n k_e$ dobrého kandidáta na jednosměrnou funkci je schopnost modulární aritmetiky (tím, že se počítání děje v kruhu) zamíchat výsledky umocnění $x \circledcirc_n k_e$ pro různé hodnoty x a zároveň však výpočet tohoto výsledku nijak nekomplikovat. To znamená, že je výpočet jedním směrem snadný, v opačném směru však obtížný, což je přesně to, co požadujeme po jednosměrné funkci.

Šifra RSA

Skutečnosti, že je funkce $e(x, k_e) = x \circledcirc_n k_e$ vhodným kandidátem na jednosměrnou funkci, si všiml v sedmdesátých letech 20. století americký informatik Ronald Rivest, který v té době pracoval v laboratoři počítačových věd na Massachusettském technologickém institutu.[4)] Při hledání této funkce a při zamítání nevhodných kandidátů mu byli nápomocni jeho dva kolegové z téže laboratoře, Adi Shamir a Leonard Adleman. Roku 1977 tak společně publikovali článek, ve kterém navrhli asymetrickou šifru založenou právě na zmíněné funkci, která se dnes nazývá RSA (název je zkratkou iniciál jmen autorů). Nedávno však vyšlo najevo, že stejnou šifru navrhl už roku 1973 anglický kryptograf Clifford Cocks. Je to tentýž Clifford Cocks, který „předběhl“ Ralpha Merkleho, Whitfielda Diffieho a Martina Hellmana v objevu bezpečné výměny klíče (byla o tom řeč v předchozím dílu). V tomto smyslu se historie zopakovala.

Ušli jsme dlouhou cestu, abychom byli schopni, aspoň v prvním přiblížení, šifru RSA popsat. Pustíme se tedy do toho. Šifrovací a dešifrovací funkce šifry RSA vypadají takto:

$$e(x, k_e) = x \circledcirc_n k_e, \tag{4}$$

$$d(y, k_d) = y \circledcirc_n k_d. \tag{5}$$

Šifrovací funkcí je, nám dobře známé, umocnění v modulární aritmetice. Veřejným klíčem je exponent k_e spolu s modulem n.[5)] Soukromým klíčem je pak exponent k_d. Všimněme si dobře, že dešifrovací funkce je také

[4)] Massachusetts Institute of Technology, známý spíše pod zkratkou MIT, je prestižní soukromou americkou univerzitou, která se do dějin informatiky i jiných oborů zapsala mnohými významnými objevy a vynálezy. Svědčí o tom už fakt, že mezi absolventy MIT je více než 70 laureátů Nobelovy ceny.

[5)] V předchozím dílu, ve kterém jsme se bavili obecně o asymetrickém šifrování, jsme

založena na umocnění, stejně jako šifrovací funkce. O umocnění víme, že jej umíme rychle vypočítat. Umíme tedy rychle vypočítat hodnotu $y \circledcirc_n k_d$; exponent k_d proto představuje zadní vrátka, o kterých jsme mluvili v předchozím dílu.

Zbývá zodpovědět otázku, jak zvolit čísla k_e, n a k_d, aby vše správně fungovalo. V následujících odstavcích si proto uvedeme podrobný popis šifrovacího a dešifrovacího procesu. Šifrovací proces probíhá v těchto krocích:

1. Bob, který figuruje jako příjemce, si nejprve zvolí dvě prvočísla. Označme si tato prvočísla jako p a q. Dále vypočítá jejich součin. Tento součin si označíme písmenem n, platí tedy $n = p \cdot q$. Číslo n bude modulem v modulární aritmetice, na které jsou založeny výpočty hodnot šifrovací a dešifrovací funkce.

2. Dále Bob vypočítá číslo, které se zpravidla označuje jako $\phi(n)$, a pro které platí
$$\phi(n) = (p-1) \cdot (q-1).$$
Poznamenejme, že hodnota $\phi(n)$ udává počet čísel, která jsou menší než n a která jsou s n nesoudělná.

3. V dalším kroku si Bob zvolí číslo k_e. Tato volba může být náhodná, důležité pouze je, aby k_e a $\phi(n)$ byla nesoudělná čísla.

4. Čísla k_e a n Bob zveřejní.

5. Alice může získat veřejný klíč. Zprávu, kterou chce Bobovi odeslat, nějakým způsobem zakóduje do posloupnosti čísel.[6)] Tuto posloupnost čísel pak rozdělí na stejně dlouhé bloky takovým způsobem, že pro každé číslo tvořící daný blok, bude platit, že je mezi 0 a $n-1$ (to z toho důvodu, že se šifrování i dešifrování provádí v modulární aritmetice s modulem n).

za veřejný klíč uvažovali pouze jedno číslo. Skutečnost, že veřejný klíč šifry RSA je složen ze dvou částí, exponentu k_e a modulu n, však myšlenku asymetrického šifrování nijak nenarušuje. Je zjevné, že obě komunikující strany musí vědět, který modul mají ve svých výpočtech použít, proto musí být n také zveřejněno.

[6)] Toto kódování může být celkem libovolné. Pouze musí být zajištěno, aby Bob používal stejné kódování.

6. Alice pak postupně zašifruje všechny bloky pomocí šifrovací funkce a veřejného klíče složeného z čísel k_e a n. Pokud je tedy x číslo odpovídající danému bloku, pak vypočítá $e(x, k_e) = x \circledcirc_n k_e$.

7. Na závěr Alice odešle Bobovi takto vypočtená zašifrovaná čísla.

Dešifrovací proces je pak proveden v těchto krocích:

1. Bob obdrží zašifrovaná čísla.

2. Poté vypočítá převrácenou hodnotu k_d k číslu k_e s využitím modulu $\phi(n)$.[7)] Proč je využito jako modul právě číslo $\phi(n)$ je spíše technickým (i když pro správné fungování RSA významným) detailem; zdůvodněním se tedy nebudeme zabývat.

3. Pokud je y jedno ze zašifrovaných čísel, pak Bob použije dešifrovací funkci a vypočítá tak $d(y, k_d) = y \circledcirc_n k_d$. Je možné ukázat, že takto vypočtené číslo $d(y, k_d)$ je stejné jako číslo x, které Alice zašifrovala.

Ukažme si šifrovací proces na následujícím příkladu. Předpokládejme prvočísla $p = 71$ a $q = 47$. Z těchto hodnot vypočítáme:

$$n = p \cdot q = 71 \cdot 47 = 3337,$$
$$\phi(n) = (p - 1) \cdot (q - 1) = 70 \cdot 46 = 3220.$$

Číslo k_e, které je součástí veřejného klíče, zvolíme rovno 79. Tato volba je v pořádku, protože čísla 79 a 3220 jsou nesoudělná (jediným společným dělitelem těchto čísel je jednička). V modulární aritmetice s modulem 3220 proto existuje k číslu $k_e = 79$ převrácená hodnota. Prozradíme, že touto převrácenou hodnotou je $k_d = 1019$, jelikož $79 \odot_{3220} 1019 = 1$ ($79 \cdot 1019$ děleno 3220 dává zbytek roven 1).

Zprávu, kterou chceme zašifrovat převedeme nějakým způsobem na posloupnost čísel. Dejme tomu, že tato posloupnost vypadá třeba takto:

$$1235371029471$$

Tuto posloupnost musíme rozdělit na stejně dlouhé bloky tak, že každý blok bude menší než $n = 3337$. Rozdělení tedy provedeme následovně:

$$123|537|102|947|001$$

[7)] Zde se nám jasně vyjevil důvod, proč jsme požadovali, aby byla čísla k_e a $\phi(n)$ nesoudělná. Tato podmínka totiž zajistí, že převrácená hodnota k číslu k_e skutečně existuje.

Každé číslo pak samostatně zašifrujeme pomocí vztahu (4), například tedy první dvě čísla zašifrujeme na hodnoty:

$$e(123, 79) = 123 \circledcirc_{3337} 79 = 1433,$$
$$e(537, 79) = 537 \circledcirc_{3337} 79 = 1385.$$

Celkem pak dostaneme posloupnost zašifrovaných čísel:

$$14331385323022411$$

Šifrování RSA si může čtenář samostatně vyzkoušet prostřednictvím následujícího úkolu (řešení je opět uvedeno na konci článku). Alice poslala Bobovi zašifrovanou zprávu, ve které uvedla místo jejich plánované schůzky. K utajení zprávy se rozhodla použít šifru RSA s modulem $n = 33$. K zakódování textu zprávy použila následují tabulku:

a	b	c	d	e	f	g	h	i	j
00	01	02	03	04	05	06	07	08	09
k	l	m	n	o	p	q	r	s	t
10	11	12	13	14	15	16	17	18	19
u	v	w	x	y	z	␣	.	,	:
20	21	22	23	24	25	26	27	28	29
+	-	@							
30	31	32							

Protože je modul malý, rozdělila posloupnost čísel (kódujících zprávu) na dvojice. K šifrování si zvolila exponent $k_e = 7$. Zašifrovanou posloupnost čísel převedla pomocí zmíněné tabulky opět na znaky a obdržela tak následující text:

`HAF,UIHCMFHAMQGNCFVFULUMU␣:CD`

Jak zní původní zpráva? V úloze figuruje skutečně velmi malý modul. V této souvislosti musíme udělat dvě důležité poznámky o bezpečnosti šifry RSA:

1. Kvůli malému modulu musela Alice rozdělit posloupnost čísel na dvojice, takže vlastně šifrovala každé písmeno zprávy samostatně. Důsledkem toho je, že se například písmeno `a` vždy zašifruje na jeden a týž znak. Tím ovšem z RSA udělala obyčejnou substituční

(i když asymetrickou) šifru. Pokud by zvolila větší modul a mohla tak rozdělit posloupnost na větší bloky, tak ke zmíněnému problému nedošlo. Pro účely našeho příkladu jsme malý modul zvolili záměrně, abychom se nezahltili náročnými numerickými výpočty.

2. Záměrně jsme v zadání úlohy neuvedli, z jakých prvočísel je modul $n = 33$ složen. Bez těchto prvočísel nejsme schopni vypočítat exponent k_d. Řešitel této úlohy se proto ocitl v roli poněkud žárlivé Evy, které se podařilo zachytit zašifrovanou zprávu, a pokouší se ji s maximálním nasazením dešifrovat. Příklad je samozřejmě nastaven tak, aby se dal rozklad modulu n lehce uhádnout. Docházíme tak k jednomu důležitému závěru: bezpečnost šifry RSA není založena pouze na jednosměrnosti šifrovací funkce, ale také na obtížnosti rozkladu čísla na součin prvočísel. O tomto problému byla napsána spousta tlustých knih a mnoho vynikajících informatiků a matematiků si láme hlavu nad tím, jestli existuje nějaká rychlá metoda, která by tento rozklad byla schopná získat a to i v případě, že je rozkládané číslo hodně velké. Prozatím je toto hledání neúspěšné.

Vidíme tedy, že pro bezpečné použití šifry RSA je nutné, aby byl modul skutečně hodně velkým číslem. Ve výše uvedené úloze byl modul směšně malý, v praktických aplikacích se v dnešní době používají moduly, které mají, zapsány v binárním tvaru, délku od 1024 do 4096 bitů.

Dost dobré soukromí

V předchozím dílu jsme vyzdvihli všechny výhody asymetrického šifrování oproti šifrování symetrickému. Existuje však jedna vlastnost symetrických šifer, za kterou asymetrické šifry značně pokulhávají. Symetrické šifry jsou totiž založeny na jednoduchých matematických operacích; zašifrovat zprávu symetricky je proto vždy o něco rychlejší než zašifrovat ji asymetricky (například pomocí RSA). Tento malý rozdíl v rychlosti se pochopitelně zvětšuje při šifrování dlouhých zpráv. Pokud budeme chtít šifrovat objemné multimediální soubory, tak tento rozdíl může komplikovat plynulost komunikace.

Nedostatečnou rychlostí asymetrických šifer se zabýval americký programátor Philip R. Zimmermann, který si položil jednoduchou otázku: nebylo by možné zkombinovat symetrickou a asymetrickou šifru tak, aby byly zvýrazněny výhody a naopak potlačeny nevýhody obou těchto typů šifrování? V devadesátých letech 20. století vytvořil Zimmermann program,

ve kterém tuto kombinaci jednoduchým způsobem zrealizoval. Tento program, který nazval PGP (z angličtiny Pretty Good Privacy – dost dobré soukromí, jak je uvedeno v názvu této sekce), funguje následovně.

Šifrování zprávy, která může být značně objemná, je prováděno rychlou symetrickou šifrou. Symetrická šifra, jak víme, používá pouze jeden tajný klíč, který si musí Alice s Bobem vyměnit. Tato výměna je však slabým článkem komunikace, tajný klíč totiž může být odhalen. Pro výměnu tajného klíče však můžeme použít bezpečnou asymetrickou šifru, která problémem výměny klíče netrpí. Jako symetrickou šifru použil Zimmermann šifru zvanou IDEA, jako asymetrickou šifru použil právě RSA.

Podobným způsobem jako v programu PGP je šifra RSA použita pro šifrování tajného klíče v protokolu SSH. Tímto protokolem se v současné době ve velké míře zajišťuje bezpečnost komunikace na Internetu.

Na závěr se podepíšeme

Ukázali jsme si různé metody utajení obsahu zpráv. Šifrování má však i výrazně jiné použití – stojí v základu takzvaného elektronického podpisu, který v elektronické komunikaci nahrazuje klasický, ručně psaný podpis. V závěru si proto tuto problematiku stručně popíšeme.

Připojení elektronického podpisu k dané zprávě by mělo zejména zajistit následující dvě skutečnosti: *autentičnost* – Bob by měl být schopen ověřit, kdo zprávu podepsal a odeslal; *nepopiratelnost* – Alice by neměla mít možnost popřít, že zprávu podepsala.

Vidíme, že autentičnost i nepopiratelnost zajišťuje v klasické komunikaci vlastnoruční podpis jednoduše tím, že rukopis každého člověka je jedinečný. Jak ovšem tuto jedinečnost přenést do světa elektronické komunikace? Autentičnost je v elektronické komunikaci realizována použitím takzvaného *digitálního certifikátu*.[8)] Nepopiratelnost je pak možné zajistit, možná trochu překvapivě, pomocí šifry RSA (nebo pomocí jiné asymetrické šifry). Stačí pouze vyměnit roli veřejného a soukromého klíče. Pro šifrování proto použijeme soukromý klíč a pro dešifrování naopak klíč veřejný. Pokud totiž Alice zašifruje zprávu soukromým klíčem,[9)] jejímž

[8)] Více informací o digitálních certifikátech a certifikačních autoritách, které tyto certifikáty vydávají a zaručují se za jejich správnost, je možné najít třeba na stránkách Wikipedie.

[9)] Ve skutečnosti se nešifruje celá zpráva, ale pouze její „výtah“, který má oproti podepisované zprávě malou velikost. Tento výtah se získává tak, že se na zprávu aplikuje takzvaná *kryptografická hashovací funkce*.

je jediným vlastníkem, pak nemůže popřít, že jej podepsala. Na druhou stranu má Bob, nebo kdokoliv jiný, možnost zprávu dešifrovat, podobně jako u vlastnoručně psaného podpisu má kdokoliv možnost si tento podpis přečíst.

Jak jsme si ukázali, se šifrou RSA se setkáváme (zejména kvůli Internetu) dnes a denně, aniž by si toho byla většina uživatelů vědoma. Je historickou zajímavostí, že v době, kdy Ronald Rivest, Adi Shamir a Leonard Adleman pracovali na článku popisujícím princip fungování této šifry, Adleman navrhnul, aby nebyl uveden mezi autory. Domníval se totiž, že se bude jednat v jeho životě o nejméně významnou práci, na které se podílel…

Řešení úkolů

1. Řešení prvních čtyřech příkladů je následující:

$$\begin{aligned} &15 \ominus_{17} (10 \oplus_{17} 14 \oplus_{17} 0) = 8, \\ &14 \odot_{25} 24 = 11, \\ &3 \oslash_9 3 = 0, \\ &20 \oslash_6 5 = 4. \end{aligned}$$

   Poslední příklad $(10 \oslash_{10} 5)$ nemá řešení, protože k číslu 5 neexistuje převrácená hodnota (5 není nesoudělné s modulem 10).

2. Nejprve je potřeba rozložit modul $n = 33$ na součin prvočísel. Na první pohled je vidět, že tato prvočísla jsou $p = 3$ a $q = 11$. Dále vypočítáme hodnotu

$$\varphi(n) = (p-1) \cdot (q-1) = 2 \cdot 10 = 20.$$

   Soukromý klíč k_d je převrácenou hodnotou veřejného klíče $k_e = 7$, přičemž modulem je hodnota $\varphi(n) = 20$. Snadno zjistíme, že $k_d = 3$.

   Kryptogram

`HAF,UIHCMFHAMQGNCFVFULUMU␣:CD`

   zakódujeme pomocí tabulky uvedené v zadání na posloupnost čísel

07 00 05 28 20 08 07 02 12 05 07 00 12 16 06
13 02 05 21 05 20 11 20 12 20 26 29 02 03

Tato čísla budeme postupně dosazovat za y do vztahu pro dešifrování:

$$y \oslash_{33} 3.$$

Obdržíme tak posloupnost čísel

13 00 26 07 14 17 13 08 12 26 13 00 12 04 18
19 08 26 21 26 14 11 14 12 14 20 02 08 27

Dekódováním těchto čísel se konečně dostaneme k otevřenému textu:

`na␣hornim␣namesti␣v␣olomouci.`

Literatura

[1] Whitfield Diffie and Martin Hellman 1976. New directions in cryptography. *IEEE Transactions on Information Theory*. 22 (6): 644–654.

[2] Steven Levy 2001. *Crypto: How the Code Rebels Beat the Government Saving Privacy in the Digital Age*. Penguin Books.

[3] Simon Singh 2000. *Kniha kódů a šifer: Tajná komunikace od starého Egypta po kvantovou kryptografii*. Argo, Dokořán.