Kryptografia s verejným kľúčom

- základný princíp, šifrovanie a autentizácia
- Diffieho-Hellmanov algoritmus na výmenu kľúčov
- El Gamalov algoritmus
- RSA algoritmus
- kryptografia na báze eliptických kriviek
- veľkosti kľúcov, porovnanie so symetrickými šiframi

Primárny zdroj informácii k dnešnej prednáške:
[1] prof. Ing. Dušan Levický, CSc.
APLIKOVANÁ KRYPTOGRAFIA
od utajenia správ ku kybernetickej bezpečnosti
Elfa, Košice, 2018 (str.139-163, str.220-223)

Princíp kryptografie s verejným kľúčom

Kryptografia s verejným kľúčom (**public–key cryptography)** je založená na **asymetrickom šifrovaní**, teda používa dva kľúče. Jeden kľúč sa používa na šifrovanie, druhý na dešifrovanie, čo je zásadný rozdiel oproti symetrickému šifrovaniu, ktoré používa jeden tajný kľúč na šifrovanie aj dešifrovanie. V symetrickom šifrovaní je potrebné zabezpečiť utajenú distribúciu kľúčov dvom komunikačným stranám a proces výmeny šifrovaných dát sa môže začať až keď je distribúcia tajného kľúča ukončená. Distribúcia tajných kľúčov v symetrickom šifrovaní prinášala v praktickom používaní značné problémy, najmä z dôvodu bezpečnej distribúcie kľúčov utajenými kanálmi, ktoré predstavovali slabé miesta v koncepcii symetrického šifrovania.

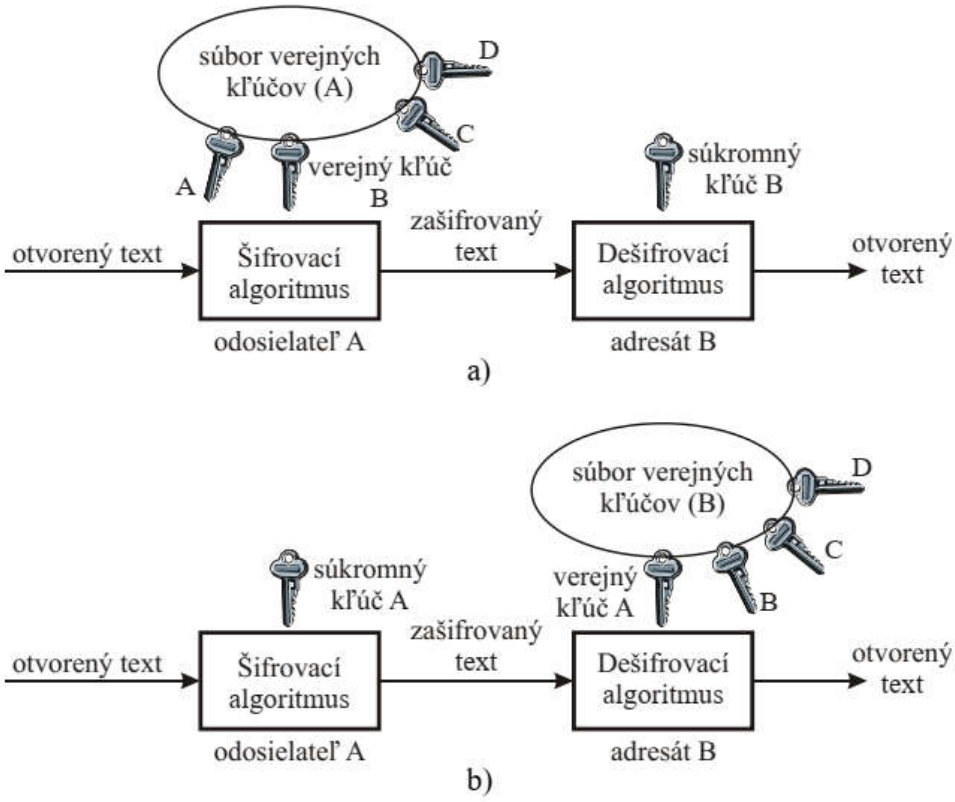
Nový prístup v distribúcii kľúčov priniesol **algoritmus Diffieho-Hellmana**, ktorý umožnil realizovať výmenu tajných kľúčov cez verejný prenosový kanál. Tento algoritmus, znamenal počiatok kryptografie s verejným kľúčom, ale stále ešte vyžadoval postupnosť interaktívnych krokov medzi účastníkmi. Veľkou výhodou kryptografie s verejným kľúčom je to, že nevyžaduje žiadnu interakciu medzi účastníkmi pred výmenou zašifrovaného textu, resp. dát. Každý účastník v kryptografickom systéme s verejným kľúčom vlastní dva kľúče. Jeden kľúč, ktorý sa označuje ako **súkromný kľúč** (**private key**), druhý sa označuje ako **verejný kľúč** (**public key**). Súkromný kľúč sa utajuje a verejný kľúč sa môže zverejniť. Použitie dvoch kľúčov ovplyvňuje stupeň bezpečnosti, spôsob distribúcie kľúčov a autentizáciu používateľov. Kryptografia s verejným kľúčom je univerzálna a umožňuje realizovať základné funkcie ako sú utajenie obsahu správy, autentizácia používateľov a autentizácia dát.

Kryptografia s verejným kľúčom zabezpečuje tieto funkcie:

• **šifrovanie**
• **autentizáciu**.

Šifrovanie v kryptografii s verejným kľúčom sa realizuje verejným kľúčom adresáta. Ak napr. odosielateľ A chce zaslať zašifrovanú správu (text) adresátovi B, na šifrovanie použije verejný kľúč B. Dešifrovanie sa realizuje súkromným kľúčom B, ktorý vlastní iba adresát B (Obr. 7.1a).

Autentizácia v kryptografii s verejným kľúčom sa rieši inverzne. Ak otvorený text zašifruje odosielateľ A svojim súkromným kľúčom, dešifrovanie možno realizovať iba verejným kľúčom A čo znamená, že správu zašifroval účastník A (Obr. 7.1b).



Obr. 7.1 Kryptografia s verejným kľúčom, a) šifrovanie, b) autentizácia

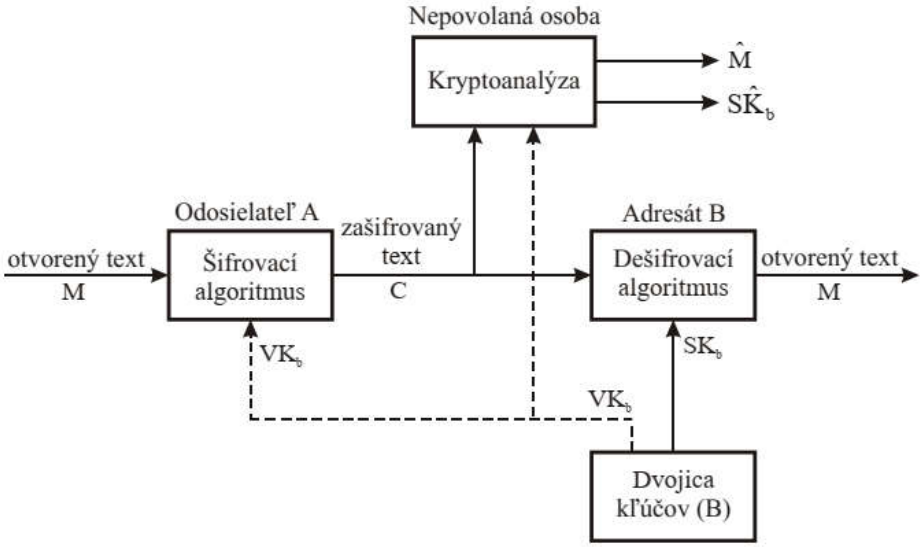
Kryptografický systém s verejným kľúčom – utajenie

Odosielateľ A **zašifruje** otvorený text M **verejným kľúčom** VK_b adresáta B, ktorý je dostupný (zverejnený). Šifrovanie možno vyjadriť v tvare

$$C = E_{VK_b}(M) \tag{7.1}$$

Zašifrovaný text C sa prenáša k adresátovi B, ktorý prijatý zašifrovaný text **dešifruje** svojim **súkromným kľúčom** SK_b. Dešifrovanie má potom tvar

$$M = D_{SK_b}(C) = D_{SK_b}\left(E_{VK_b}(M)\right) \tag{7.2}$$



Obr. 7.2 Kryptografický systém s verejným kľúčom (utajenie)

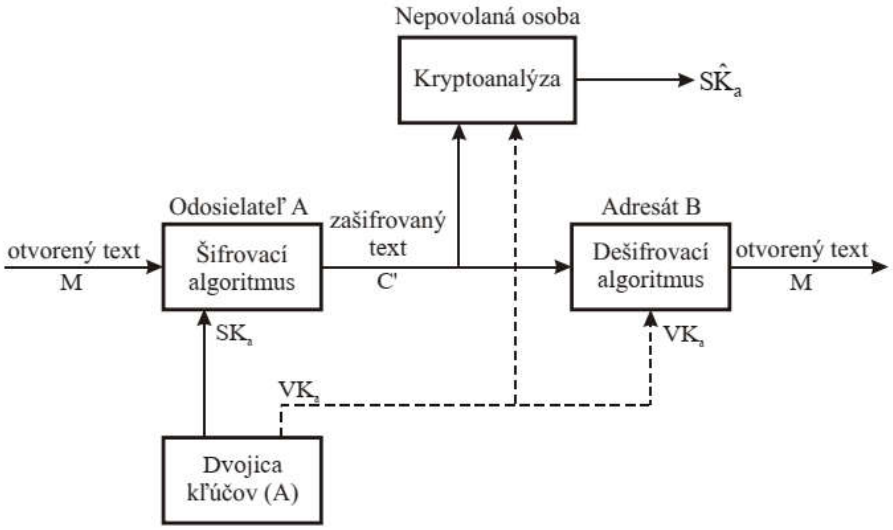
Kryptografický systém s verejným kľúčom – autentizácia

Odosielateľ **zašifruje** otvorený text M svojím **súkromným kľúčom** SK_a, čím sa získa zašifrovaný text C'. Šifrovanie možno zapísať v tvare

$$C' = E_{SK_a}(M) \tag{7.3}$$

Zašifrovaný text C' sa prenáša k adresátovi B, ktorý prijatý zašifrovaný text **dešifruje verejným kľúčom** VK_a. Tento kľúč je dostupný a dešifrovanie má tvar

$$M = D_{VK_a}(C') = D_{VK_a}\left(E_{SK_a}(M)\right) \tag{7.4}$$



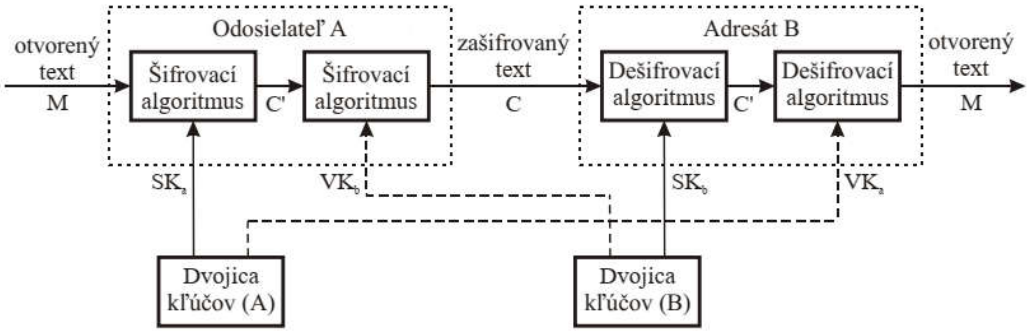
Obr. 7.3 Kryptografický systém s verejným kľúčom (autentizácia)

Autentifikátor správy a digitálny podpis

Pretože otvorený text bol zašifrovaný súkromným kľúčom SK_a a možno ho dešifrovať iba verejným kľúčom VK_a, autorom správy je A. Zašifrovaný text C' má v tomto prípade charakter **digitálneho podpisu** (**digital signature**). Zároveň z toho vyplýva, že modifikácia správy (otvoreného textu) bez prístupu k súkromnému kľúču SK_a je nemožná, teda tento kryptografický systém s verejným kľúčom zabezpečuje autentizáciu zdroja správy aj integritu prenášaných dát.

Nevýhodné je aj **šifrovanie celej správy**, ktoré sa realizuje súkromným kľúčom SK_a, najmä z hľadiska časových a pamäťových nárokov. Efektívnejší je postup, pri ktorom sa šifruje **len určitý výťah** zo správy, ktorý má podstatne menší rozsah. Tento výťah je funkciou úplného dokumentu a je zašifrovaný súkromným kľúčom SK_a. Uvedený výťah zo správy sa nazýva **autentifikátor** (**autentificator**) a má takú vlastnosť, že každá modifikácia pôvodného dokumentu vyvolá modifikáciu tohto identifikátora. **Zašifrovaný identifikátor slúži ako digitálny podpis**, ktorým sa verifikuje pôvod a obsah pôvodnej správy. Uvedená technika autentizácie bude podrobnejšie opísaná v časti o **hašovacích funkciách** (4. týždeň) a **digitálnych podpisoch** (6. týždeň).

Kryptografický systém s verejným kľúčom – súčasné utajenie a autentizácia



Obr. 7.4 Kryptografický systém s verejným kľúčom (utajenie a autentizácia)

Uvedený systém realizuje šifrovanie v dvoch krokoch, teda platí

$$C = E_{VK_b}\left(E_{SK_a}(M)\right) \tag{7.5}$$

Proces dešifrovania, ktoré tiež vyžaduje dva kroky, možno vyjadriť v tvare

$$M = D_{VK_a}\left(D_{SK_b}(C)\right) \tag{7.6}$$

Nevýhodou uvedeného kryptografického systému s verejným kľúčom je to, že proces šifrovania a dešifrovania je zložitejší a vyžaduje štyri kroky.

Podmienky realizovateľnosti kryptografického systému s verejným kľúčom

Asymetrické algoritmy používané v kryptografických systémoch s verejným kľúčom by mali splňovať podmienky, ktoré možno formulovať takto:

1. výpočtovo **jednoduché generovanie dvojice kľúčov** pre odosielateľa A (SK_a a VK_a) a pre adresáta B (SK_b a VK_b)

2. výpočtovo **jednoduchá realizácia šifrovania** správy M odosielateľom A pre adresáta B, na základe dostupnosti VK_b, teda jednoduché generovanie zašifrovaného textu

$$C = E_{VK_b}(M)$$

3. výpočtovo **jednoduchá realizácia dešifrovania** prijatého zašifrovaného textu C adresátovi B s použitím súkromného kľúča SK_b, teda jednoduché získanie pôvodnej správy M

$$M = D_{SK_b}(C) = D_{SK_b}\left(E_{VK_b}(M)\right)$$

4. výpočtovo **zložité získanie súkromného kľúča** SK_b zo známeho verejného kľúča VK_bnepovolanou osobou

5. výpočtovo **zložité získanie pôvodnej správy** M zo známeho zašifrovaného textu C a známeho verejného kľúča VK_b nepovolanou osobou.

Voliteľná podmienka:
6. funkcie šifrovania a dešifrovania sú **vzájomne zameniteľné**, t. j. môžu byť vykonané v ľubovoľnom poradí

$$M = E_{VK_b}\left(D_{SK_b}(M)\right) = D_{VK_b}\left(E_{SK_b}(M)\right)$$

Uvedené podmienky spĺňajú v plnom rozsahu najmä dva algoritmy a to **algoritmus RSA** a **algoritmy na báze eliptických kriviek**.

Pojem jednocestnej funkcie

Vo všeobecnosti možno konštatovať, že všetky algoritmy pre kryptografické systémy s verejným kľúčom využívajú **špeciálnu triedu matematických funkcií**, ktoré sa označujú ako **jednocestné funkcie** (**one-way functions**). Vlastnosti uvedených funkcií možno sformulovať takto:

1. jednoduchý výpočet funkčnej hodnoty $Y = f(X)$ zo známeho argumentu X
2. obťažný výpočet argumentu X zo známej funkčnej hodnoty Y, t. j. výpočet $X = f^{-1}(Y)$.

Osobitnú triedu jednocestných funkcií tvoria **jednocestné funkcie so skrytým vstupom** (**trap-door one-way functions**), v ktorých je výpočet inverznej funkcie podmienený znalosťou **utajeného parametra** k. V **opačnom prípade** je výpočet inverznej funkcie **obťažný**. Vlastnosti jednocestných funkcií so skrytým vstupom možno zapísať takto:

1. jednoduchý výpočet $Y = f_k(X)$, ak X a k sú známe
2. jednoduchý výpočet $X = f_k^{-1}(Y)$, ak Y a k sú známe
3. obťažný výpočet $X = f_k^{-1}(Y)$, ak Y je známe a k nie je známe.

Kategorizácia kryptografických systémov s verejným kľúčom

Z hľadiska použitých algoritmov možno kryptografické systémy s verejným kľúčom rozdeliť do troch kategórií a to na kryptografické systémy, ktoré realizujú:

- **šifrovanie**, resp. **dešifrovanie**
- **digitálne podpisy**
- **výmenu kľúčov**.

Na **šifrovanie** sa využíva verejný kľúč adresáta, dešifrovanie sa realizuje súkromným kľúčom adresáta.

Digitálny podpis správy je šifrovanie správy súkromným kľúčom odosielateľa. Šifrovanie sa aplikuje buď na celú správu alebo na výťah zo správy, t. j. malý blok dát, ktorý je funkciou celej správy.

Výmena kľúčov sa realizuje pomocou **kľúča relácie**, ktorý je k dispozícii obom stranám. Niektoré prístupy využívajú súkromný kľúč jednej alebo oboch strán.

Vo všeobecnosti platí, **že nie všetky algoritmy** používané v kryptografických systémoch s verejným kľúčom **sú univerzálne**, t. j. sú schopné zabezpečovať všetky uvedené funkcie.

Medzi univerzálne algoritmy s verejným kľúčom patria algoritmy **RSA**, **El Gamal** a algoritmy **na báze eliptických kriviek**. Algoritmus **Diffie-Hellman** je použiteľný len na výmenu kľúčov. Uvedenú situáciu ilustruje Tab. 7.1.

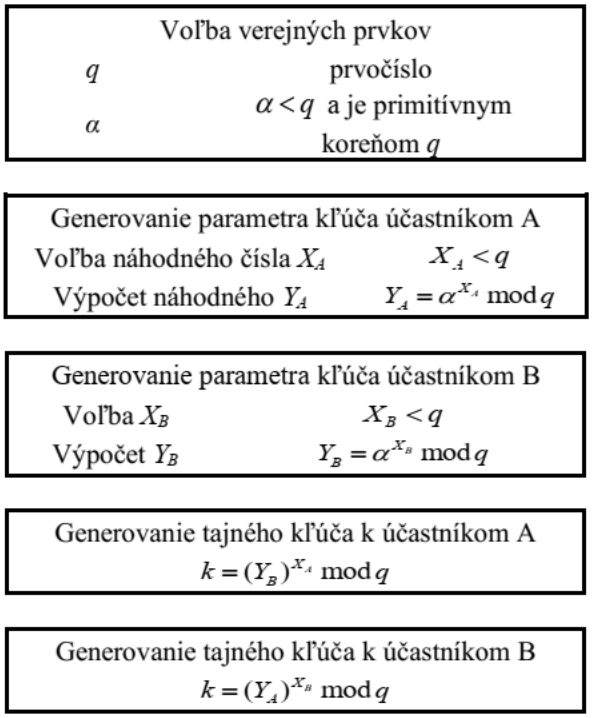
Tab. 7.1 Prehľad vybraných algoritmov s verejným kľúčom

Typ algoritmu	Šifrovanie/Dešifrovanie	Digitálny podpis	Výmena kľúčov
Diffie – Hellman	Nie	Nie	Áno
El Gamal	Áno	Áno	Áno
RSA	Áno	Áno	Áno
Eliptické krivky	Áno	Áno	Áno

Algoritmus na výmenu kľúčov Diffie-Hellman

Algoritmus označovaný ako Diffie-Hellman bol **prvým publikovaným algoritmom** s verejným kľúčom, pričom **umožňuje výmenu tajných kľúčov** v kryptografickom systéme s verejným kľúčom. Uvedený algoritmus vytvoril predpoklady pre vznik kryptografie s verejným kľúčom a odstránil ťažiskový problém kryptografie s tajným kľúčom, ktorým bola distribúcia tajných kľúčov. Algoritmus Diffie- Hellamn je **založený na obťažnosti výpočtu diskrétnych logaritmov**, ktorých princíp bol vysvetlený na cvičení.

Bezpečnosť algoritmu je založená na tom, že je relatívne **ľahké vypočítať modulárnu mocninu** α, ktorá je primitívnym koreňom prvočísla q, ale je **obťažné pre veľké prvočísla vypočítať diskrétny logaritmus**.



Obr. 7.5 Princíp algoritmu Diffie – Hellman

Poznámka:
primitívny koreň α je **generátorom** v poli GF(q)

Použité vzťahy na výpočet tajného kľúča k vytvárajú identické výsledky, pretože platí

$$\begin{aligned} k &= (Y_B)^{X_A} \bmod q \\ &= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\ &= (\alpha^{X_B})^{X_A} \bmod q \end{aligned}$$

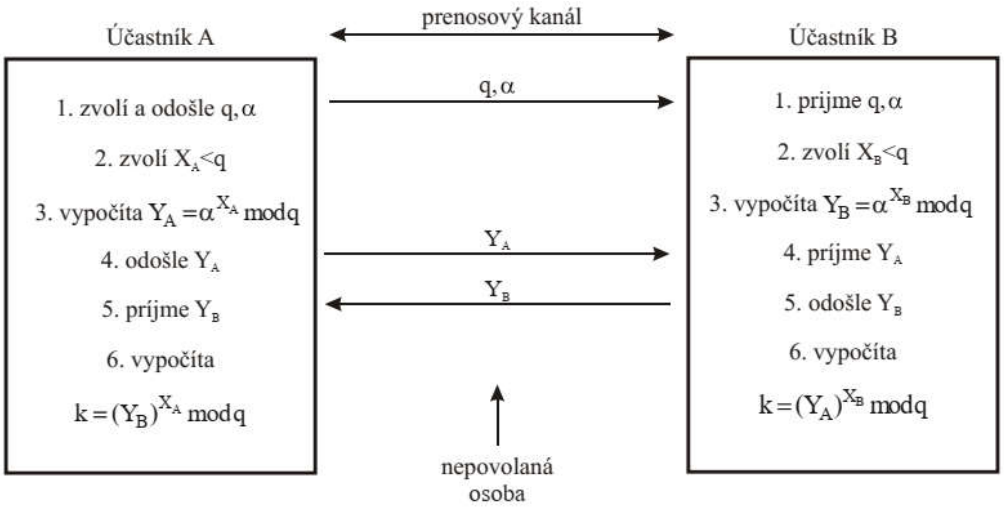
$$\begin{aligned} &= (\alpha^{X_B X_A}) \bmod q \\ &= (\alpha^{X_A})^{X_B} \bmod q \\ &= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\ &= Y_A^{X_B} \bmod q \end{aligned}$$

Teda platí

$$(Y_B)^{X_A} \bmod q = (Y_A)^{X_B} \bmod q$$

Protokol algoritmu Diffie-Hellman (zverejnený v roku 1977)

Na Obr. 7.6 je uvedený jednoduchý **protokol** algoritmu Diffie-Hellman v prípade, že účastník chce nadviazať komunikáciu s účastníkom B s využitím **tajného kľúča** *k*, ktorý musia mať k dispozícii obaja účastníci.



Obr. 7.6 Protokol algoritmu Diffie–Hellman

Účastník A potom zvolí **jednorazový súkromný kľúč** X_A a vypočíta hodnotu Y_A, ktorú pošle účastníkovi B. Účastník B analogicky zvolí **jednorazový súkromný kľúč** X_B a vypočíta hodnotu Y_B, ktorú odošle účastníkovi A. Po výmene hodnôt Y_A a Y_B, ktoré možno považovať za **verejné kľúče** môžu účastníci A aj B vypočítať hodnotu **tajného kľúča** *k*. Účastníci A a B potom môžu komunikovať **šifrovanou komunikáciou s využitím tajného kľúča** *k*.

Uvedený algoritmus má **interaktívny charakter**, teda **výpočet tajného kľúča** *k* možno realizovať na oboch stranách **až po výmene hodnôt** Y_A a Y_B medzi účastníkmi A a B. Táto vlastnosť môže byť **nevýhodná** napr. pri komunikácii dvoch strán **s rozdielnou výpočtovou kapacitou** (napr. komunikácia v senzorovej sieti medzi výkonným serverom a energeticky limitovaným senzorom).

Algoritmus El Gamal (publikovaný v roku 1985)
Je tiež založený na obťažnosti výpočtu diskrétnych logaritmov v konečnom poli a možno ho použiť univerzálnejšie, ako algoritmus Diffie-Hellman.

Princíp použitia algoritmu El Gamal v kryptografickom systéme s verejným kľúčom je uvedený na Obr. 7.7.

Generovanie kľúčov	
Vyber p	p – prvočíslo
Zvoľ g,x	náhodné čísla $g<p$ $x<p$
Vypočítaj y	$y = g^x \bmod p$
Verejný kľúč	$VK=\{y, g, p\}$
Súkromný kľúč	$SK=\{x\}$

Šifrovanie	
Vyber k	náhodné číslo k, $1 \le k \le p-2$
Otvorený text	M
Zašifrovaný text	$a = g^k \bmod p$
(dvojica a,b)	$b = y^k M \bmod p$

Dešifrovanie	
Zašifrovaný text	a,b
Otvorený text	$M = b / a^x \bmod p$

Obr. 7.7 Princíp algoritmu El Gamal

Algoritmus El Gamal je v podstate zhodný s algoritmom Diffie – Hellman na distribúciu kľúčov s tým rozdielom, že y je časťou verejného kľúča a operácia šifrovania zahŕňa násobenie činiteľom y^k. Je potrebné tiež poznamenať, že algoritmus El Gamal **má „pravdepodobnostný“ charakter**, t. j. vzhľadom na náhodný výber k **rovnakým otvoreným textom** zodpovedajú **rôzne zašifrované texty**.

Algoritmus El Gamal je využitý napr. v štandarde **DSA** (**Digital Signature Algorithm**) pre digitálny podpis, podrobnejšie v prednáške 6. Digitálne podpisy, certifikáty.

Algoritmus RSA (publikovaný v roku 1978)

Algoritmus RSA je najznámejším algoritmom s verejným kľúčom a je pomenovaný podľa jeho autorov, ktorými sú **Ron Rivest**, **Adi Shamir** a **Leonard Adleman**. Tento algoritmus zatiaľ úspešne odoláva pokusom o jeho prelomenie, preto zaznamenal široké použitie v kryptografických systémoch s verejným kľúčom.

Bezpečnosť algoritmu RSA je založená na **obťažnosti faktorizácie veľkých čísel**. Verejný a súkromný kľúč sa odvodzuje z **dvoch veľkých** (200 a viac miestnych) **prvočísel**. Z hľadiska implementácie algoritmus RSA je **bloková šifra**, v ktorej otvorený aj zašifrovaný text sú celé čísla v rozmedzí 0 až (n–1), ak n je zvolené číslo. Typická veľkosť n je 1024 bitov alebo 309 dekadických čísel, prípadne viac. Lúštenie otvoreného textu so znalosťou verejného kľúča a zašifrovaného textu sa zakladá na odhade správnej faktorizácie súčinu dvoch veľkých prvočísiel.

Na **vygenerovanie** verejného a súkromného kľúča je potrebné zvoliť dve, približne rovnako veľké, **prvočísla** p a q, pre ktoré platí

$$p \cdot q = n$$

Potom sa zvolí šifrovací kľúč e tak, aby čísla e a $\phi(n) = (p-1)\cdot(q-1)$ boli navzájom nesúdeliteľné, t. j. $\gcd(\phi(n), e) = 1$. Dešifrovací kľúč d sa potom vypočíta pomocou rozšíreného Euklidovho algoritmu tak, aby platilo

$$e \cdot d \equiv 1 \bmod \phi(n)$$

alebo

$$d = e^{-1} \bmod \phi(n)$$

Čísla e a d sú teda multiplikatívne inverzné čísla modulo $\phi(n)$ čo platí iba, ak $\gcd(d, \phi(n)) = 1$. Je potrebné tiež poznamenať, že čísla d a n sú tiež nesúdeliteľné.

Čísla e a n predstavujú verejný kľúč a čísla d a n súkromný kľúč, teda

$$VK = \{e, n\}$$
$$SK = \{d, n\}$$

Poznámky:
a) **Euklidov algoritmus** je algoritmus pre nájdenie najväčšieho spoločného deliteľa **GCD** (**Greatest Common Divisor**) a bude preberaný na cvičení. **Rozšírený Euklidov algoritmus** umožňuje vypočítať počas výpočtu aj ďalšie parametre (napr. **modulárnu inverziu**) a bude tiež preberaný na cvičení.
b) Prvočísla p a q sa už v ďalšom postupe **nepoužívajú**, ale musia ostať utajené. Možno ich však použiť **na zrýchlenie dešifrovania** s vužitím tzv. **čínskej vety o zvyškoch** (**Chinese Remainder Theorem**) sformulovanej okolo roku 100 pred našim letopočtom.

Proces šifrovania, resp. dešifrovania prebieha po blokoch a možno ho vyjadriť v tvare

$$C = M^e \bmod n$$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

kde M – je blok otvoreného textu

C – je blok zašifrovaného textu.

Postup implementácie algoritmu RSA je uvedený v *Tab. 7.2*.

Tab. 7.2 Algoritmus RSA

Generovanie kľúčov	
Vyber p,q	p,q – prvočísla $p \neq q$ a $\phi(n) = (p-1)\cdot(q-1)$
Vypočítaj $n = p \cdot q$	
Zvoľ celé číslo e	$\gcd(\phi(n), e) = 1$ $1 < e < \phi(n)$
Vypočítaj d	$d = e^{-1} \bmod \phi(n)$
Verejný kľúč	$VK = \{e,n\}$
Súkromný kľúč	$SK = \{d,n\}$

Šifrovanie	
Otvorený text	$M<n$
Zašifrovaný text	$C = M^e \bmod n$

Dešifrovanie	
Zašifrovaný text	C
Otvorený text	$M = C^d \bmod n$

Generovanie RSA kľúčov
Generovanie kľúčov zahrňuje proces generovania dvojice kľúčov každého účastníka, teda dvojice VK a SK, pričom je potrebné:

• **zvoliť** dvojicu prvočísiel *p* a *q*
• **zvoliť** celé číslo *e* a **vypočítať** *d*, resp. zvoliť *d* a vypočítať *e*.

Výberom prvočísel *p* a *q* je zároveň dané aj číslo $n = p.q$, ktoré je súčasťou verejného kľúča. Teda číslo *n* môže byť známe aj nepovolanej osobe, ktorá sa snaží z čísla *n* určiť *p* a *q*. Prvočísla *p* a *q* sa teda musia vyberať z veľkej množiny, resp. *p* a *q* **musia byť veľké prvočísla**. Zároveň ale metóda výberu prvočísel *p* a *q* by mala byť **dostatočne efektívna**. V súčasnosti totiž nie je známa jednoznačná (dostatočne efektívna) metóda výberu veľkých prvočísel. Obvykle pri výbere veľkých prvočísel sa postupuje tak, že sa **náhodne zvolí nepárne číslo** potrebnej veľkosti a **realizuje sa test**, ktorý potvrdí, resp. nepotvrdí, či dané zvolené číslo je prvočíslo. V prípade neúspešného textu sa náhodne vyberie ďalšie nepárne číslo, pričom postup sa opakuje až test potvrdí, že dané číslo je prvočíslo.

Kľúčovým aspektom je **testovanie** zvoleného čísla, resp. **potvrdenie**, že zvolené číslo je prvočíslo. Vo väčšine prípadov známe testy zaručujú kladný výsledok, t. j. že číslo je prvočíslo, len **s určitou pravdepodobnosťou**. Napriek tomu nedostatku možno opakovaním testu dosiahnuť, že táto pravdepodobnosť je blízka hodnote 1. Veľmi efektívny a populárny test je **algoritmus Miller-Rabin**.

Poznámka:
Veľmi často je číslo e zvolené ako 4-te **Frematovo číslo** (https://cs.wikipedia.org/wiki/Fermatovo_číslo):

Fermatovým číslem se v matematice rozumí takové přirozené číslo, které je rovno

$$F_n = 2^{2^n} + 1$$

pro nějaké přirozené číslo n. Svoje jméno tato čísla získala podle matematika Pierra de Fermata, který je zkoumal jako jeden z prvních.

Prvních devět Fermatových čísel je:

$F_0 = 2^1 + 1 = 3$
$F_1 = 2^2 + 1 = 5$
$F_2 = 2^4 + 1 = 17$
$F_3 = 2^8 + 1 = 257$
$F_4 = 2^{16} + 1 = 65\,537$

Táto voľba je často aj súčasťou šifrovacích noriem a protokolov. Toto číslo ma v binárnom vyjadrení **len dva jednotkové bity** a je pomerne malé, čo výrazne zrýchľuje modulárne umocnenie. **Operácia šifrovania** v RSA je tak zvyčajne **podstatne rýchlejšia**.

Bezpečnosť algoritmu RSA
Bezpečnosť algoritmu RSA môžu ohroziť tri druhy útokov. Sú to:
• útok metódou totálnych skúšok (brute–force attack)
• matematické útoky (mathematical attacks)
• časové útoky (timing attacks).

Útok metódou totálnych skúšok predstavuje cyklické prehľadávanie celej množiny možných súkromných kľúčov. Ochrana proti tejto metóde je rovnaká ako ochrana všetkých kryptografických algoritmov, resp. systémov a spočíva v zabezpečení veľkého priestoru kľúčov. Zároveň je potrebné poznamenať, že veľký počet bitov prináša vyššie časové nároky a teda predlžuje dobu šifrovania.

Matematické útoky na algoritmus RSA možno založiť na týchto postupoch:

- faktorizácia čísla n na prvočísla p a q. To umožní určiť $\phi(n)$ a vypočítať $d = e^{-1} \bmod \phi(n)$
- priame určenie $\phi(n)$ bez určenia p a q. To opäť umožní výpočet $d = e^{-1} \bmod \phi(n)$
- priame určenie d bez určenia $\phi(n)$.

Vzhľadom na rýchly vývoj výpočtovej techniky sa zdokonaľujú aj faktorizačné technológie, takže na dosiahnutie **dostatočnej kryptografickej bezpečnosti** má byť číslo n v rozsahu **1024 až 2048 bitov, resp. 4096 bitov**.

Vo všeobecnosti možno formulovať pre algoritmus RSA takéto odporúčania:

• Skupina používateľov nemá **nikdy používať rovnaké** n. Použitie rovnakej hodnoty n v skupine používateľov **uľahčuje kryptoanalýzu** a znižuje odolnosť voči vybraným typom útokov na RSA.

• hodnota e ovplyvňuje **efektívnosť implementácie šifrovania**, často sa volí malá hodnota (4-te Fermatovo číslo).

Slabou stránkou algoritmu RSA sú **značné časové nároky** na šifrovanie a dešifrovanie. Aj keď uvedené parametre sa vývojom technických prostriedkov zlepšujú, algoritmus **RSA nikdy nedosiahne časové parametre symetrických šifier**.

Časové útoky na algoritmus RSA sú založené na spôsobe výpočtu modulárnej mocniny *a^k* mod *n*, v ktorom postupne spracovávajú jednotlivé bity súkromného kľúča, t. j. exponentu. Výpočet prebieha v cykle, pričom dobu výkonu cyklu je rôzna pre hodnoty príslušného bitu kľúča. Ak je daný bit 0, výpočet je rýchlejší, ak je bit 1, výpočet trvá dlhšie. Z uvedených časových parametrov možno teda dedukovať hodnoty bitov kľúča. Tieto typy útokov patria do kategórie tzv. **útokov s využitím postranných kanálov** (**Side-Channel Attacks**), ktoré sú predmetom aktívneho výskumu a vývoja vhodných implementačných protiopatrení pre **vstavané kryptografické aplikácie**.

Hodnotenie bezpečnosti kryptografických algoritmov a ich kombinácie

Bezpečnosť kryptografických algoritmov (**security strengths**) s definovanou dĺžkou kľúčov sa obvykle určuje ako množstvo práce, vyjadrenej vo vhodných jednotkách (napr. počet elementárnych operácií alebo počet hodinových cyklov), potrebnej na prehľadanie celej množiny kľúčov. Kvantitatívne sa bezpečnosť kryptografického algoritmu vyjadruje počtom bitov kľúča daného algoritmu. Ak má napr. kľúč *k* dĺžku **56 bitov**, potom množina všetkých kľúčov obsahuje 2^*k* = **2^56 kľúčov** a **bezpečnosť tohto algoritmu je 56 bitov**.

V ďalšom výklade budeme predpokladať, že dva kryptografické algoritmy budú mať porovnateľnú kryptografickú bezpečnosť, ak množstvo práce potrebnej na ich prelomenie alebo na určenie kľúča je pri rovnakých výpočtových výkonoch približne rovnaká. Útoky na bezpečnosť kryptografických algoritmov môžu znižovať túto bezpečnosť zmenšením potrebného počtu preskúmaných kľúčov, ktorý je menší ako počet všetkých kľúčov z množiny kľúčov. Parameter, ktorý zohľadňuje vplyv známych útokov na bezpečnosť sa označuje ako **ekvivalentná bezpečnosť kryptografických algoritmov** a vyjadruje sa v bitoch.

Tab. 11.6 Ekvivalentná bezpečnosť kryptografických algoritmov

Ekvivalentná bezpečnosť [b]	Symetrické algoritmy	Algoritmy DSA DH	Algoritmus RSA	Eliptické krivky	Hašovacie funkcie
80	2DES	VK=1024 SK=160	$n = 1024$	$n = 160 - 223$	SHA-1
112	3DES	VK=2048 SK=224	$n = 2048$	$n = 224 - 255$	SHA-224
128	AES–128	VK=3072 SK=256	$n = 3072$	$n = 256 - 383$	SHA-256
192	AES–192	VK=7680 SK=384	$n = 7680$	$n = 384 - 511$	SHA-384
256	AES–256	VK=15360 SK=512	$n = 15360$	$n > 512$	SHA-512

Prvý stĺpec *Tab. 11.6* udáva ekvivalentnú bezpečnosť kryptografických algoritmov v rozsahu 80 až 256 bitov.

Poznámka:
Na stránke https://crypto.stackexchange.com/questions/8687/security-strength-of-rsa-in-relation-with-the-modulus-size sú napr. uvedené **vzorce a diskusia**, ktoré umožňujú vyjadriť **ekvivalentnú kryptografickú bezpečnosť RSA** algoritmu (pomocou aproximácie zložitosti **GFNS algoritmu** (https://en.wikipedia.org/wiki/General_number_field_sieve) na faktorizáciu veľkých čísel).

Tab. 11.7 poskytuje prehľad odporúčaných algoritmov a minimálne dĺžky kľúčov ako aj odhad pre ich použitie na obdobie do a po roku 2030 podľa dokumentu **NIST SP 800-57** z roku **2005**.

Tab. 11.7 Odporúčané kryptografické algoritmy a minimálne dĺžky kľúčov

Obdobie (ekvivalentná bezpečnosť)	Symetrické algoritmy	Algoritmy DSA DH	Algoritmus RSA	Eliptické krivky
do roku 2010 (min 80 bitov)	3DES AES–128 AES–192 AES–256	min VK=1024 SK=160	min $n = 1024$	min $n = 160$
2011 – 2030 (min 112 bitov)	3DES AES–128 AES–192 AES–256	min VK=2048 SK=224	$n = 2048$	min $n = 224$
po roku 2030 (min 128 bitov)	AES–128 AES–192 AES–256	min VK=3072 SK=256	$n = 3072$	min $n = 256$

Ak sa v **zloženom kryptografickom algoritme** používa **kombinácia** viacerých základných kryptografických algoritmov, potom **výsledná ekvivalentná bezpečnosť** zloženého kryptografického algoritmu je **určená najmenšou hodnotou** ekvivalentnej bezpečnosti použitých základných kryptografických algoritmov.

Pri použití algoritmu **ECC s dĺžkou kľúča 160 bitov** v kombinácii s kryptografickým algoritmom **AES-128** je výsledná ekvivalentná bezpečnosť tejto kombinácie algoritmov na **úrovni 80 bitov**. Na dosiahnutie **ekvivalentnej bezpečnosti 128 bitov** je potrebné použiť algoritmus **ECC s dĺžkou kľúča minimálne 256 bitov** v kombinácii napr. s algoritmom AES-128.

Kryptografia na báze eliptických kriviek a súvislosť s konečnými poľami

Kryptografia na báze eliptických kriviek **ECC** (**Elliptic Curve Cryptography**) je novým a perspektívnym smerom v modernej kryptografii. Jej **hlavnou prednosťou** v porovnaní s existujúcimi kryptografickými systémami a algoritmami (napr. RSA) je to, že **umožňuje dosiahnuť rovnakú kryptografickú bezpečnosť pri menšej dĺžke kľúča**, čo jasne demonštrujú aj predchádzajúce tabuľky.

Kryptografické systémy na báze **ECC** sa stali **súčasťou viacerých kryptografických štandardov** a predstavujú alternatívu k systémom na báze RSA aj DSA, pričom hlavnou **výhodou systémov ECC** je **väčšia rýchlosť a menšie nároky** na technické prostriedky.

Podrobnejšie sa budeme kryptografii na báze ECC zaoberať v **inžinierskom štúdiu**. Na záver tejto prednášky **naznačíme súvislosť medzi eliptickou kryvkou a konečným poľom** (GF(p) resp. GF(2^m)), ktoré sme prebrali v predchádzajúcom cvičení.

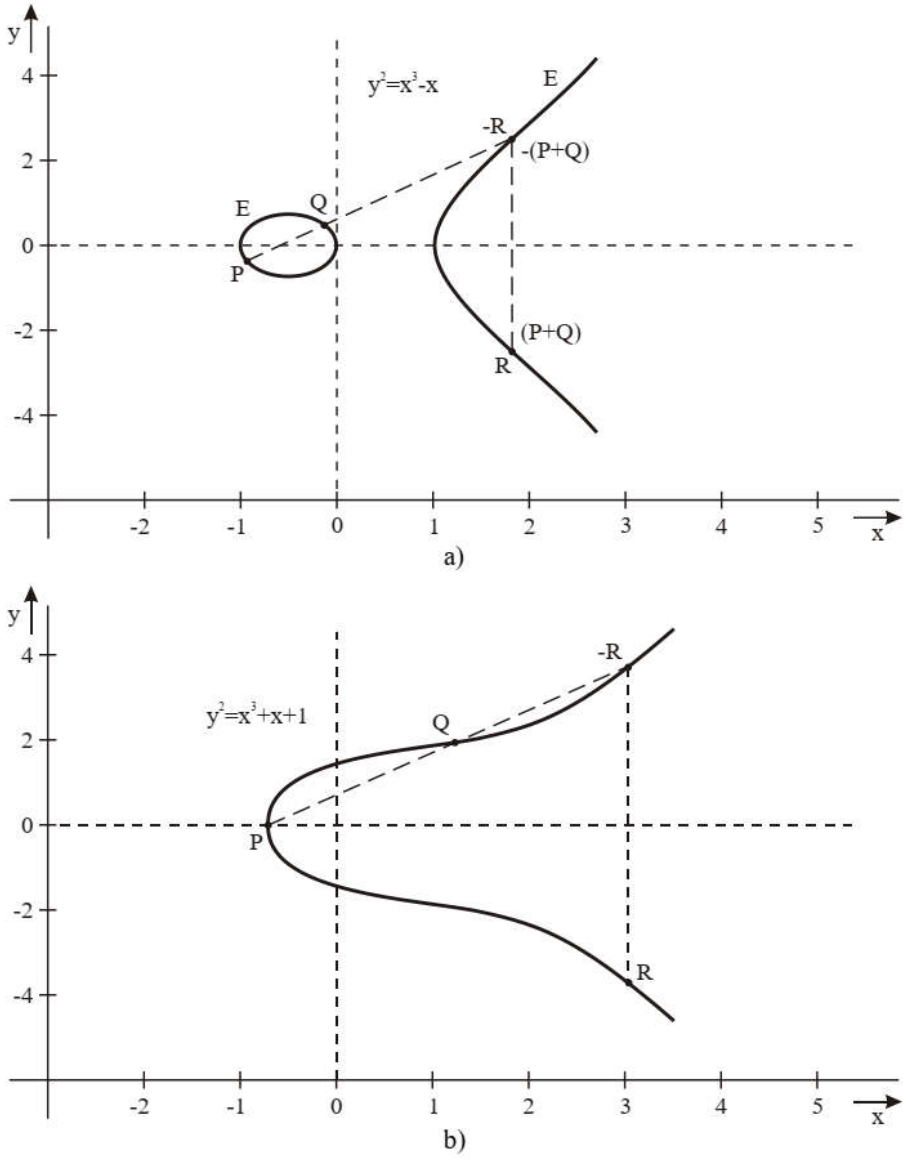
Eliptická krivka nad reálnymi číslami

Eliptické krivky sú špeciálnou **triedou kubických kriviek**. Skúmaním vlastností eliptických kriviek sa najviac zaoberal Karl Theodor Wilhelm Weierstrass. Názov eliptické krivky sa zaužíval preto, lebo kubické rovinné funkcie sa v minulosti používali k výpočtu obvodu elipsy.

Eliptická krivka je množina bodov, ktoré vyhovujú eliptickej rovnici

$$y^2 = x^3 + ax + b \tag{7.7}$$

Pre dané hodnoty *a* a *b*, graf eliptickej krivky obsahuje kladné a záporné hodnoty *y* pre každú hodnotu *x*, teda každá eliptická krivka je symetrická podľa osi *x*.



Obr. 7.9 Príklady eliptických kriviek

Uvažujme, že **množina bodov E(a,b)** obsahuje všetky body (x,y), ktoré spĺňajú rovnicu (7.7) a **prvok O** (tzv. bod v nekonečne). Ak sa použijú rôzne hodnoty dvojice (a,b), potom sa získajú rôzne množiny *E(a,b)*, teda rôzne eliptické krivky. Napr. eliptické krivky na Obr. 7.9 zodpovedajú množinám *E(*-1,0*)* a *E(*1,1*)* a sú spojité pre všetky reálne hodnoty x, y.

Definujme operáciu, ktorú nazveme sčítaním na množine *E(a,b)* a označíme ju symbolom +, pričom prvky a,b vyhovujú podmienke (7.8). Z dôvodu názornosti operáciu sčítania dvoch prvkov množiny *E(a,b)* definujme pomocou geometrických pojmov, teda formulujme geometrickú interpretáciu sčítania takto: **ak tri body eliptickej krivky ležia na jednej priamke, ich súčet je rovný O**.

Z uvedenej definície sčítania vyplýva:

1. Bod O slúži ako neutrálny prvok vzhľadom na sčítanie a teda platí $O=-O$, resp. $\mathrm{P}-O=\mathrm{P}$, pričom $\mathrm{P}\neq O$.
2. Opačný bod k bodu P je bod, ktorý má rovnakú súradnicu x ako bod P, ale y súradnica má hodnotu $-y$. Pre opačný bod $-$P teda platí, že ak P=(x,y), potom $-$P=(x,$-y$). Je potrebné tiež poznamenať, že body P a $-$P ležia na vertikálnej priamke a zároveň platí P+($-$P)=P$-$P=O.
3. Sčítanie dvoch rôznych bodov P a Q realizujeme geometricky tak, že bodmi P a Q preložíme priamku, ktorej tretí priesečník s eliptickou krivkou označíme $-$R. Sčítanie dvoch rôznych bodov P a Q teda možno zapísať v tvare P+Q=R. Bod R je symetrickým bodom k bodu $-$R podľa osi x, teda leží na rovnobežke s osou y, ktorá prechádza bodom $-$R. Uvedená konštrukcia je ukázaná na *Obr. 7.9*.
4. Geometrickú interpretáciu predchádzajúceho pravidla možno aplikovať aj na body P a $-\mathrm{P}$, t. j. body s rovnakými súradnicami x. Uvedené body možno spojiť vertikálnou priamkou, ktorá pretína eliptickú krivku v nekonečne, teda v bode O. Teda platí $\mathrm{P}+(-\mathrm{P})=O$, čo je konzistentné s pravidlom (2).

Z uvedených pravidiel a z preverenia asociatívneho zákona vyplýva, že množina E(a,b) tvorí **abelovskú grupu**.

Eliptické krivky nad konečným poľom

Z dôvodov, ktoré sme diskutovali pri GF(p) a GF(2^m), t.j. predovšetkým **možnosť realizovať výpočty bez zaokrúhľovacích chýb**, sú v kryptografii využívané ECC nad GF(p) alebo GF(2^m). Tieto eliptické krivky sú krivky, ktorých premenné a koeficienty sú prvkami konečných polí.

Uvedené eliptické krivky možno rozdeliť do dvoch skupín. Sú to:
• **prvočíselné** eliptické krivky (**prime** curves)
• **binárne** eliptické krivky (**binary** curves).

Prvočíselné eliptické krivky sú definované nad konečným poľom GF(*p*) a **binárne eliptické krivky** nad konečným poľom GF(2^*m*). Pre **softvérové aplikácie** sú výhodnejšie **prvočíselné eliptické krivky**, **binárne eliptické krivky** sú výhodné pre **hardvérové riešenia**.

Pre **ilustráciu využijeme** jednoduchú prvočíselnú krivku.

Prvočíselné eliptické krivky nad konečným poľom GF(p) sú opísané kubickou rovnicou, v ktorej premenné a koeficienty nadobúdajú hodnoty z množiny celých čísel v rozsahu od 0 do (p–1). Operácie sa realizujú **modulo *p***, teda rovnica (7.7) prechádza na tvar

$$y^2 \bmod p = \left(x^3 + ax + b\right) \bmod p \qquad (7.13)$$

Uvažujme, že množina $E_p(a,b)$ obsahuje všetky dvojice celých čísel (x,y), ktoré spĺňajú rovnicu (7.13) a bod O. Zvoľme napr. p=23 a eliptickú krivku $y^2=x^3+x+1$, teda $a=b=1$.

Pre množinu $E_{23}(1,1)$ je potrebné uvažovať iba nezáporné celé čísla x,y z kvadrantu od (0,0) až $\left(p-1, p-1\right)$, ktoré sú výsledkom operácie modulo p, teda $x, y \in \left\{0,1,2,\ldots, p-1\right\}$.

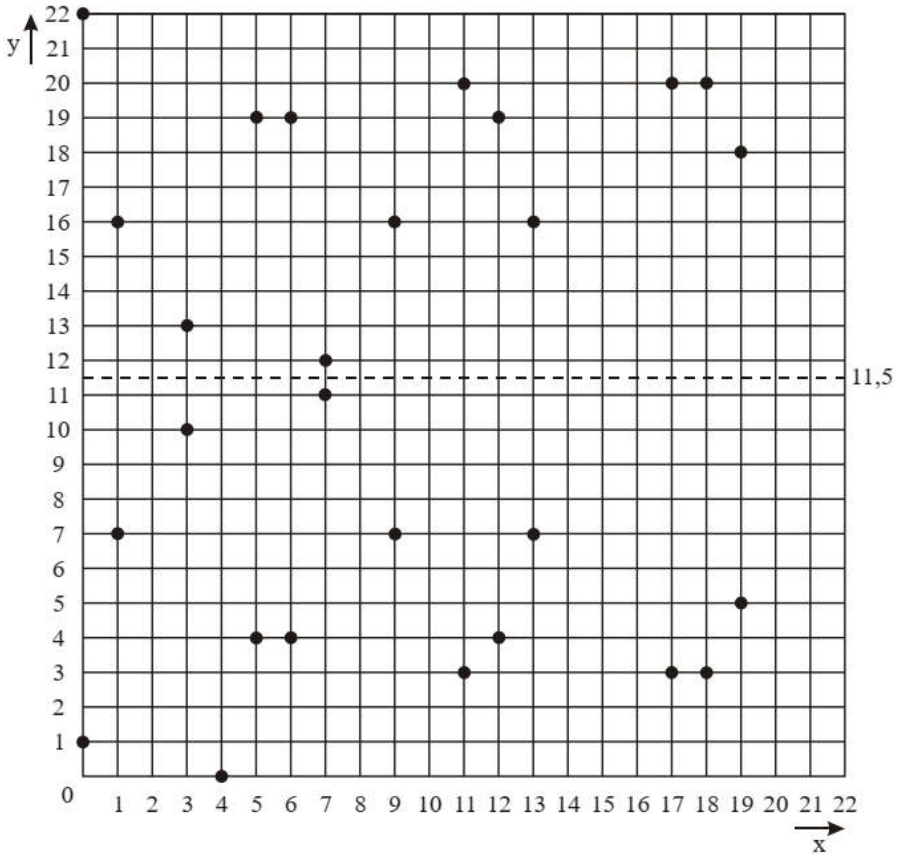
Zoznam bodov, ktoré tvoria množinu $E_{23}(1,1)$ spolu s bodom O sú uvedené v *Tab. 7.3* a ich rozloženie v kvadrante je znázornené na *Obr. 7.10*. Je potrebné poznamenať, že s jedinou výnimkou sú všetky body rozložené symetricky okolo $y = \frac{23}{2} = 11{,}5$.

Možno tiež ukázať, že konečná abelovská grupa môže byť definovaná na množine $E_p(a,b)$ ak platí

$$\left(4a^3 + 27b^2\right) \bmod p \neq 0 \bmod p \qquad (7.14)$$

Tab. 7.3 Zoznam bodov množiny $E_{23}(1,1)$

(0,1)	(6,4)	(12,19)	(0,22)
(6,19)	(13,7)	(1,7)	(7,11)
(13,16)	(1,16)	(7,12)	(17,3)
(3,10)	(9,7)	(17,20)	(3,13)
(9,16)	(18,3)	(4,0)	(11,3)
(18,20)	(5,4)	(11,20)	(19,5)
(5,19)	(12,4)	(19,18)	*O*



Obr. 7.10 Rozloženie bodov eliptickej krivky $y^2 = x^2 + x + 1$ pre p=23

Ak máme definovanú abelovskú grupu (t.j. množinu bodov P na eliptickej krivke a operáciu sčítania bodov $P+Q$) a vieme na krivke definovať aj operáciu násobenia bodu P skalárom k ako k-násobné sčítanie:

$$Z=kP = P + P + \dots P \qquad (1)$$

Pripomeňme, že **geometrickú interpretáciú** sčítania bodov (Obr.7.9) je možné vyjadriť aj pomocou **algebraických rovníc**, v ktorých vystupujú **súradnice bodov** a **koeficienty krivky**. Keďže všetky uvedené objekty sú **prvky konečného poľa**, je možné v týchto rovniciach použiť všetky **bežné operácie** (sčítania, odčítanie, násobenie, delenie).

Vidíme, že konečné polia sú využité pri vytvorení algebraického systému (abelovskej grupy) so zložitejšou štruktúrou, ktorá napr. umožňuje definovať **problém diskrétneho algoritmu** pre body Z, P na eliptickej krivke v tvare:

Nájdi k ak je dané Z, P a platí (1).

Ak je zvolené GF(p) pre **dostatočne veľké** p, je **problém diskrétneho logaritmu** pre body na eliptickej krivke **príkladom jednocestnej funkcie**. Eliptická krivka tak umožňuje tento problém využiť na vytvorenie kryptografie s verejným kľúčom, ktorý nazývame termínom **ECC** (**Elliptic Curve Cryptography**).

ECC umožňuje realizovať plnohodnotné **šifrovanie**, **autentizáciu** aj **výmenu kľúčov** s **podstatne menšími veľkosťami kľúčov** ako algoritmy DH, RSA, El Gamal (Tab.11.6).