Kryptografie a počítačová bezpečnost

Matematický background

Literatura

- https://www.cs.vsb.cz/ochodkova/courses/kpb/cryptography-and-network-security_-principles-and-practice-7th-global-edition.pdf
- Kapitoly 9.1, 9.2
- https://www.cs.vsb.cz/ochodkova/courses/kpb/mzka.pdf
- Kapitoly 1.2, 1.3, 4

Motivace

- Cílem je pracovat s diskrétní množinou „prvků“ a manipulovat s nimi zejména pomocí co nejmenšího počtu operací.
  - Prvky množiny (abecedy):
    - celá č. (přirozená): $5 + 9 = 14$; $5 * 3 = 5 + 5 + 5 = 15$
    - polynomy: $(x^2+1) + x = x^2+x+1$; $(x^2+1) * x = x^3+x$
- Množina prvků musí být konečná, potřebujeme tedy provádět operace s:
  - s celými č. **modulo** nějaké c.č. (prvočíslo),
  - s polynomy mod nějaký polynom,
  - umocňování mod c.č., mod polynom (ireducibilní polynom),
  - výpočet inverzních prvků mod …

  → modulární aritmetika
- Dále potřebujeme výpočet Eulerova čísla, NSD, NSN, …
- http://www.cs.vsb.cz/ochodkova/courses/kpb/mzka.pdf

Motivace

- Shift Cipher
  - $M=C=K=Z_{26}$,
  - $Z_{26}=\{0,1,2,\ldots,25\}$ je konečná podmnožina množiny celých čísel, množina celých čísel modulo 26
  - zpráva m se šifruje znak po znaku

    $e_k(m)=(m+k) \mod 26$

    $d_k(c)=(c-k) \mod 26$
  - 25 (26) možných klíčů
- →Aritmetické operace s prvky konečné podmnožiny nekonečné množiny celých čísel →
- Kolik různých operací potřebujeme? Chceme jich co nejméně, časově a implementačně efektivní operace, …

Motivace - Affine Cipher

- **Affine Cipher,** monoalfabetická substituce
  - M=C=K=Z_{26}, ‘A’=0, …, ‘Z’=25
  - zpráva m se šifruje po blocích (délka bloku je 1 znak)
  - klíč k$\in${0, 1, …, 25}, k=(a,b), jaký je prostor klíčů?
  - $c = e_k(m)=(a*m + b) \bmod 26$
  - $m = d_k(c)=(a^{-1}*(c +(-b))) \bmod 26$
  - Problémem je existence a^{-1}, aby existoval, musí platit NSD(a,26)=1. Např. a=3, a^{-1}=9
  - Pro a = 1 dostáváme šifru Shift

Motivace - RSA šifrování - příklad

- **Volba velkých prvočísel** (alespoň 100 dekadických cifer) p, q (soukromá hodnota), p=7, q=17
- **Výpočet n**=p*q (veřejná hodnota) n=7*17=119
- **Volba veřejného klíče** e nesoudělného s $\Phi(n)$, $\Phi(n)=(p-1)*(q-1)=6*16=96$, $NSD(e,\Phi(n))=1$, $1 < e < \Phi(n)$ (veřejná hodnota), e=5
- **Výpočet soukromého klíče** $d \equiv e^{-1} \bmod \Phi(n) \rightarrow d^* e \equiv 1 \bmod \Phi(n)= 77$
- p, q lze zapomenout
- Zpráva M = 197 se rozdělí na bloky menší než n, tj. na bloky 19 a blok 7.
- Šifrování prvního bloku $19^5 \bmod 119 = 66$
- Dešifrování prvního bloku $66^{77} \bmod 119 = 19$

- Ale nejprve úvod do ASK ☺ a algoritmus RSA

Kryptografie a počítačová bezpečnost

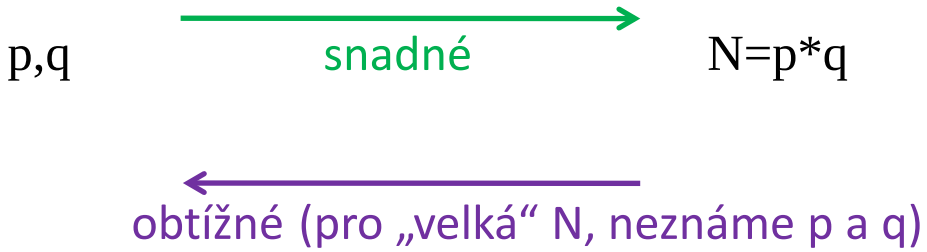
Asymetrická kryptografie

Asymetrická kryptografie

- Asymetrická kryptografie (ASK) vznikla
  - jako reakce na obtížnou správu klíčů symetrické kryptografie
    - idea ustavení klíče pro symetrickou kryptografii bez nutnosti jeho distribuce
- První veřejně známý algoritmus 1976: Diffie–Hellman key exchange
- 1977: algoritmus RSA (Rivest–Shamir–Adleman)
- Algoritmy ASK jsou založeny na odlišných principech než alg. symetrické
- Jsou založeny na obtížnosti některých známých výpočetních problémů (např. celočíselná faktorizace nebo např. problém diskrétního logaritmu)

Jednocestná funkce

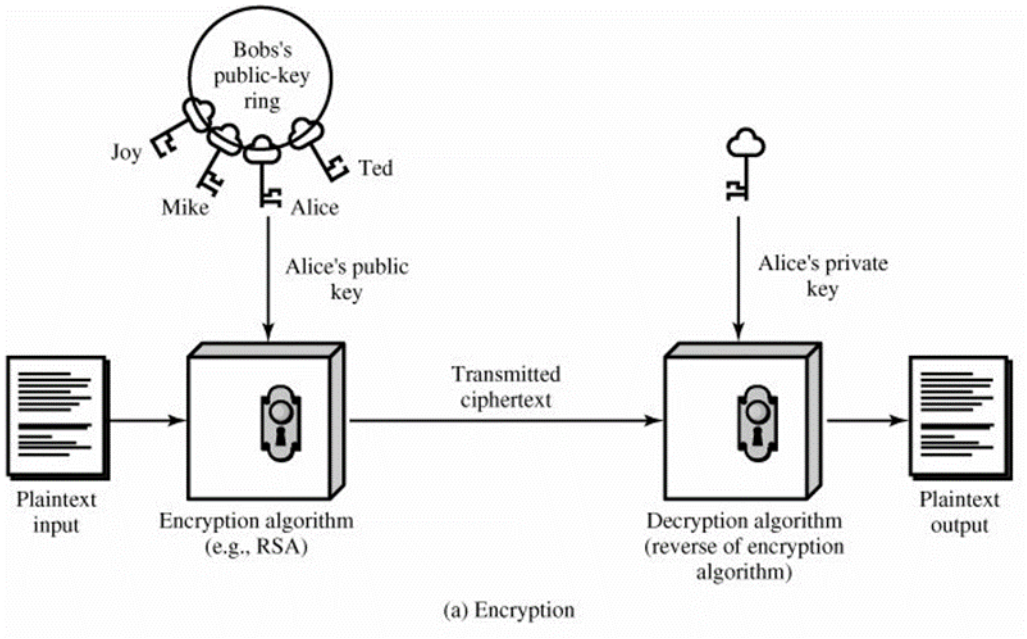
- ASK je založena na konceptu jednocestné funkce (používá se i „jednosměrná“ funkce)
- **Jednocestná funkce** (one-way function): Funkce f: X→Y se nazývá jednocestnou funkcí, jestliže je snadné spočítat f(x) pro všechna x∈X, ale v podstatě pro všechna y∈Y je výpočetně obtížné najít nějaké x∈X takové, že $f^{-1}(y) = x$.
- Co znamená „snadné“ a „výpočetně obtížné“?
  - „snadné“ – funkce je vyčíslitelná v polynomiálním čase
  - „výpočetně obtížné“ (těžké) – funkce je invertovatelná jen pro velmi zanedbatelnou část vstupů.

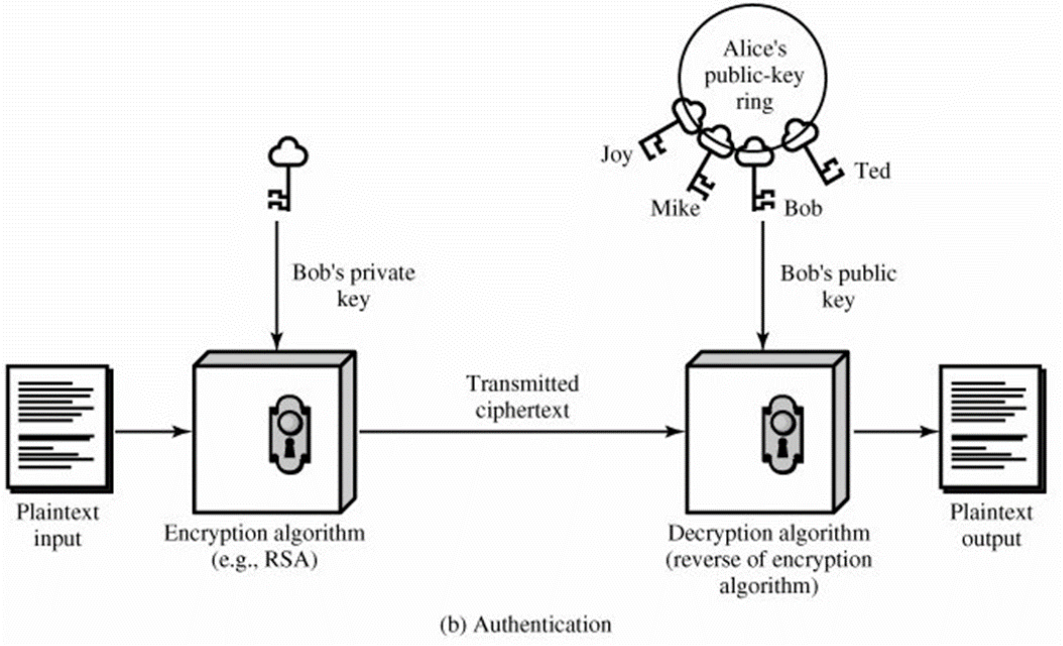


Jednocestná funkce se zadními vrátky

- **Jednocestná funkce se zadními vrátky** (trapdoor one-way function): je jednocestná funkce f_k: X→Y jestliže navíc splňuje následující vlastnost:
  - pro dané $y \in Y$ lze snadno spočítat $x \in X$ takové, že $x = f_k^{-1}(y)$ tehdy, když je známa hodnota nějakého tajemství k (trapdoor information), v kryptografii tímto tajemstvím bude klíč k (bude to např. soukromý klíč)
- Předpokládejme, že nějaká funkce f_k je spojena s určitým konkrétním subjektem - uživatelem A, f_k je jeho veřejný klíč.
- Informace o zadních vrátkách k je známa pouze subjektu A a je to jeho soukromý klíč.
- Z vlastností jednocestné funkce se zadními vrátky f_k plyne, že z pouhé znalosti f_k nelze najít výpočetně schůdnou inverzní funkci f_k^{-1}
  - => ze znalosti veřejného klíče f_k nelze nalézt klíč soukromý k
  - => ten, kdo zná pouze veřejný klíč, dokáže zašifrovat libovolnou zprávu, ale nedokáže náhodně vybraný šifrový text sám dešifrovat.

ASK – šifrování a podepisování





ASK

- ASK a utajení
  - $C = E_{KVB}(M)$, $M = D_{KSB}(C)$
- ASK a autentizace + integrita
  - $C = E_{KSA}(M)$, $M = D_{KVA}(C)$ – podpis s obnovou zprávy
  - $C = M \,||\, E_{KSA}(H(M))$ – podpis v dodatku ke zprávě (ověření podpisu později)
- Algoritmy ASK se používají pro:
  - RSA – šifrování, digitální podpis, distribuce klíčů,
  - Diffie-Hellman - distribuce klíčů (dohoda na klíči),
  - DSS - digitální podpis,
  - ElGamal - šifrování, digitální podpis, distribuce klíčů,
  - ECC (Elliptic curve cryptography) - šifrování, digitální podpis, distribuce klíčů.

ASK

- Výhody
  - Nepředává se tajemství
    - Komunikující entity si nemusí předávat žádné tajemství, nemusí se předem znát, nemusí spolu předem komunikovat
    - Správa klíčů jednodušší než v SK, ale ne jednoduchá (viz PKI později)
  - Není podstatná délka klíče (v jistém smyslu je irelevantní)
    - Jestliže je veřejný klíč publikován, není vlastně podstatná délka soukromého klíče (protože ho nemůžeme z veřejného klíče odvodit, ale ten musí být vlastníkem bezpečně uschován a chráněn).
- Nevýhody
  - Efektivita - výrazně pomalejší
- Hybridní kryptosystém kombinuje pohodlí kryptografie veřejného klíče s účinností kryptografie symetrické.
  - K zašifrování tajného klíče K_{AB} (pro nějaký symetrický alg., např. AES) se použije šifrování asymetrickým algoritmem (např. pomocí RSA s Bobovým veřejným klíčem K_{VB}) a k šifrování otevřeného textu M se použije právě klíč K_{AB} a příslušný symetrický algoritmus (např. AES).
    - TLS, https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/

Kryptografie a počítačová bezpečnost

RSA

RSA

- The Rivest-Shamir-Adleman (RSA) algorithm
  - Ron Rivest, Adi Shamir, Len Adleman z MIT
  - 1978
  - Založen na výpočetní obtížnosti problému prvočíselné faktorizace velkého celého čísla
    - **Prvočíselnou faktorizací** daného celého čísla n rozumíme jeho rozklad na součin menších prvočísel (faktorů), tj. $n = p_1^{e_1} * p_2^{e_2} * \ldots * p_k^{e_k}$, kde p_i jsou po dvou různá prvočísla a každé $e_i \geq 1$.
    - Nechť n je celé složené číslo, hledáme prvočísla p, q taková, že n = p * q.
    - $2^{113}-1 = 3391*23279*65993*1868569*106681813286820 7$
    - Asymetrie složitosti
      - 14 713*14 783 = 217 502 279, výpočetně snadné
      - Rozklad na prvočíselné dělitele 217 502 279 = 14 713*14 783, výpočetně obtížné

Faktorizace celého čísla

- RSA Challenge
  - http://en.wikipedia.org/wiki/RSA_numbers
  - https://en.wikipedia.org/wiki/RSA_Factoring_Challenge
  - RSA-576, 576 bitů nebo 174 dekadických cifer – faktorizováno 3.12.2003,
  - RSA-640 faktorizováno 5.11.2005, 193 dekadických cifer,
  - RSA-704 faktorizováno, 2/2012
  - RSA-768 ano, 12.12.2009, 232 cifer
  - RSA-896 ne (270 číslic)
  - Pozn. 1024 bitů představuje 309 dekadických cifer – nejmenší doporučená délka n (modulu) u algoritmu RSA

Algoritmus RSA

- OT i ŠT se reprezentují jako celá čísla z intervalu 0,..,n-1
- Např. pro šifrování se OT rozdělí na bloky < n, tj. velikost bloku je ≤ $\log_2 n$, tj. je to k bitů, kde $2^k < n \leq 2^{k+1}$

- Soukromý klíč d,
- Veřejný klíč e, a veřejná hodnota modulu n
- Šifrování (utajení)
  - Šifrování: $C = M^e \bmod n$
  - Dešifrování: $M = C^d \bmod n \rightarrow (M^e)^d \bmod n \rightarrow M^{ed} \bmod n = M$
- Podpis (autentizace, integrita, nepopiratelnost)
  - Podpis: $C = M^d \bmod n$
  - Ověření podpisu: $M = C^e \bmod n$

RSA šifrování

- Podmínky:
  - Lze nalézt e,d,n takové, že $M^{ed} \bmod n = M$ pro $\forall M < n$
  - Je relativně snadné vypočítat M^e a C^d pro $\forall M < n$
  - **Je nemožné určit d, známe-li {e,n}**
- Postup
  - Volba velkých prvočísel p, q (tajná hodnota), velikost obou přiměřeně stejná, např. pro n o velikosti 2048b. budou obě prvočísla mít cca 1024b.
  - Výpočet n=p*q (n je veřejná hodnota)
  - Volba veřejného klíče e nesoudělného s $\Phi(n)$, NSD(e,$\Phi(n)$)=1, 1 < e < $\Phi(n)$ (veřejná hodnota), kde $\Phi(n)$ je Eulerova funkce https://cs.wikipedia.org/wiki/Eulerova_funkce
  - Výpočet soukromého klíče $e*d \equiv 1 \bmod \Phi(n)$ (číslo d je multiplikativní inverzní prvek k číslu e)
  - p, q lze zapomenout! Nesmí být kompromitovány!

RSA šifrování - příklad

- **Volba velkých prvočísel** p, q (soukromá „tajná“ hodnota)

  p=7, q=17
- **Výpočet n**=p*q (veřejná hodnota)

  n=7*17=119
- **Volba veřejného klíče** e nesoudělného s $\Phi(n)$, (veřejná hodnota)

  $\Phi(n)=(p-1)*(q-1)=6*16=96$, NSD(e,$\Phi(n)$)=1, $1 < e < \Phi(n)$, e=5
- **Výpočet soukromého klíče** (soukromá „tajná“ hodnota)

  $d \equiv e^{-1} \bmod \Phi(n) \rightarrow d^* e \equiv 1 \bmod \Phi(n) = 77$
- p, q lze zapomenout
- Zpráva M = 197 se rozdělí na bloky s (hodnotou) menší než n, tj. na bloky 19 a blok 7.
- Šifrování prvního bloku $19^5 \bmod 119 = 66$
- Dešifrování prvního bloku $66^{77} \bmod 119 = 19$

RSA

- Potřebné algoritmy
  - Rozšířený Eukleidův algoritmus
  - Repeated square and multiply algorithm
  - Testování prvočíselnosti

Postup

- Volba velkých prvočísel p, q (tajná hodnota), velikost obou přiměřeně stejná, např. pro n o velikosti 2048b. budou obě prvočísla mít cca 1024b.
- Výpočet n=p*q (n je veřejná hodnota)
- Volba veřejného klíče e nesoudělného s $\Phi(n)$, NSD(e,$\Phi(n)$)=1, 1 < e < $\Phi(n)$ (veřejná hodnota), kde $\Phi(n)$ je Eulerova funkce https://cs.wikipedia.org/wiki/Eulerova_funkce
- Výpočet soukromého klíče $e*d \equiv 1 \bmod \Phi(n)$ (číslo d je multiplikativní inverzní prvek k číslu e)
- p, q lze zapomenout! Nesmí být kompromitovány!

Prvočísla

- Stránka o prvočíslech http://primes.utm.edu/
- Největší známé prvočíslo (Mersennovo, má tvar 2^p-1, kde p je prvočíslo) $2^{82\,589\,933}$-1, má 24,862,048 desítkových číslic, objeveno 7/12/2018 (51. Mersennovo číslo)
  - http://www.mersenne.org/ - GIMPS (Great Internet Mersenne Primes Search)
- Finding very small primes
  - https://primes.utm.edu/prove/prove2_1.html
  - https://www.youtube.com/watch?v=klcIklsWzrY
- Fermat, probable-primality and pseudoprimes
  - https://primes.utm.edu/prove/prove2_2.html
  - https://www.youtube.com/watch?v=oUMotDWVLpw
- Strong probable-primality and a practical test
  - https://primes.utm.edu/prove/prove2_3.html

Generování prvočísel

- Vygenerování (pseudo)náhodného čísla n požadované délky.
- Zjištění, je-li číslo n liché. Není-li liché, zpět na generování čísla.
- Je-li liché, provedení testu prvočíselnosti.
- Jestliže n projde úspěšně dostatečným počtem testů, bude n akceptováno jako prvočíslo.
- Testy prvočíselnosti
  - Testy pravděpodobnostní – Fermat (nepoužívá se), Miller-Rabin, …
  - Testy dokazující prvočíselnost – Lucas-Lehmer, …
  - později

Eulerova funkce

Postup

- Volba velkých prvočísel p, q (tajná hodnota), velikost obou přiměřeně stejná, např. pro n o velikosti 2048b. budou obě prvočísla mít cca 1024b.
- Výpočet n=p*q (n je veřejná hodnota)
- Volba veřejného klíče e nesoudělného s $\Phi(n)$, NSD(e,$\Phi(n)$)=1, 1 < e < $\Phi(n)$ (veřejná hodnota), kde $\Phi(n)$ je Eulerova funkce https://cs.wikipedia.org/wiki/Eulerova_funkce
- Výpočet soukromého klíče e*d ≡ 1 mod $\Phi(n)$ (číslo d je multiplikativní inverzní prvek k číslu e)
- p, q lze zapomenout! Nesmí být kompromitovány!

- Def.: **Euelerovo číslo** (funkce) udává počet čísel nesoudělných s m a menších než m (včetně 1).
  - pokud m je prvočíslo, potom $\phi(m)=m-1$, $\phi(19)=18$,
  - E.f. je multiplikativní funkce, $\phi(m*n)= \phi(m)*\phi(n)$, $\phi(21)= \phi(7)*\phi(3)=6*2=12$, pokud m a n jsou nesoudělná
  - $\phi(m)= \phi(p^k)=(p-1)*p^{k-1}$, pokud m je k-tou mocninou prvočísla p (př.: $\phi(25)= \phi(5^2)=(4)*5^{2-1}=20$)
  - obecně: mějme kanonický rozklad č. $m = p_1^{a1} * p_2^{a2} *... p_k^{ak}$, pak $\phi(m)= (p_1-1)p_1^{a1-1} * (p_2-1)p_2^{a2-1} *... (p_k-1)p_1^{ak-1}$, zapisuje se často jako

$$\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_k}\right).$$

Největší společný dělitel

- Celé číslo c je NSD (greatest common divisor, GCD) čísel a, b jestliže
  - c je dělitelem obou celých čísel a, b
  - Libovolný dělitel čísel a, b je také dělitelem čísla c
- Dvě celá čísla a, b jsou nesoudělná (relatively prime) jestliže jejich NSD = 1.
- Pokud je NSD(a,b) > 1, pak jsou čísla a, b soudělná.

Největší společný dělitel a EA

- **Eukleidův algoritmus (EA) (pro určení NSD** https://en.wikipedia.org/wiki/Euclidean_algorithm
- NSD(a,b)=NSD(b, a mod b) pro a ≥b
- r dělí a, b, a také platí a=d*b+r, pro nějaké c.č. d
  - Vstup: dvě přirozená čísla a, b kde a ≥b
  - Výstup NSD(a,b)
  - Dokud (b ≠ 0) opakuj:
    - {r←a mod b, a←b, b←r}
  - Vrať(a).

- Příklad: NSD(4864, 3458) = 38

```
4864 = 1 * 3458 + 1406
3458 = 2 * 1406 + 646
1406 = 2 * 646 + 114
646 = 5 * 114 + 76
114 = 1 * 76 + 38      ←——— NSD(4864,3458)
76 = 2 * 38 + 0.
```

EA

- $a_{i+1} = a_{i-1} \pmod{a_i}$:

a_0	$a_0 = 98$
a_1	$a_1 = 56$
$a_2 = a_0 \pmod{a_1}$	$a_2 = 98 \pmod{56} = 42$
$a_3 = a_1 \pmod{a_2}$	$a_3 = 56 \pmod{42} = \mathbf{14} = \mathsf{GCD}\ (98, 56)$
$a_4 = a_2 \pmod{a_3}$	$a_4 = 42 \pmod{14} = 0$

- $a_{i-1} = q_i \cdot a_i + a_{i+1}$:

$a_0 = q_1 \cdot a_1 + a_2$	$98 = 1 \cdot 56 + 42$
$a_1 = q_2 \cdot a_2 + a_3$	$56 = 1 \cdot 42 + \mathbf{14} = \mathsf{GCD}\ (98, 56)$
$a_2 = q_3 \cdot a_3 + a_4$	$42 = 3 \cdot 14 + 0$

- $a_i = a_{i-1} - q_i \cdot a_i$:

a_0	$a_0 = 98$
a_1	$a_1 = 56$
$a_2 = a_0 - q_1 \cdot a_1$	$a_2 = 98 - 1 \cdot 56 = 42$
$a_3 = a_1 - q_2 \cdot a_2$	$a_3 = 56 - 1 \cdot 42 = \mathbf{14} = \mathsf{GCD}\ (98, 56)$
$a_4 = a_2 - q_3 \cdot a_3$	$a_4 = 42 - 3 \cdot 14 = 0$

Extended Euclidean algorithm

- Rozšířený Eukleidův algoritmus (EEA) je určen pro řešení
  GCD(a,b) = x*a + y*b
- Příklad

  a_0 = 98

  a_1 = 56

  a_2 = 42 = 98 – 1*56

  a_3 = 14 = 56 – 1*42 = 56 – 1*(98 – 56) = 2*56 – 1*98, x = -1, y = 2

EEA pro určení multiplikativního inverzního prvku

- Jestliže GCD(a,b)=1, pak existuje multiplikativní inverzní prvek (pokud a je prvočíslo, pak exituje vždy (kromě 0)) Můžeme tedy použít EEA k jeho nalezení GCD(a,b) = x*a + y*b
- Příklad

a_0 = 25

a_1 = 18

a_2 = 7 = 25 – 1*18

a_3 = 4 = 18 – 2 * 7 = 18 – 2 * (25 – 18) = 3 * 18 – 2 * 25

a_4 = 3 = 7 – 4 = 25 – 18 – (3 * 18 – 2 * 25) = 3 * 25 – 4 * 18

a_5 = GCD(25, 18) = 1 = 4 – 3 = 3 * 18 – 2 * 25 – (3 * 25 – 4 * 18) = (– 5) * 25 + 7 * 18

→ y = 7, x = −5.

- Multiplikativní inverzní prvek k 18 je 7 (mod 25). Musí platit $a * a^{-1} \equiv 1 \pmod{n}$ pro for $a \in Z_n$, a != 0, a to platí: 18 * 7 ≡ 1 (mod 25).

Nejmenší společný násobek

- NSN (least common multiple, LCM)
- Celé číslo c je NSN čísel a, b jestliže
  - Když obě celá čísla a, b jsou dělitelem čísla c
  - Jestliže jsou a, b dělitelé libovolného c.č. c‘, pak je c‘ také dělitelem čísla c.
- Platí: NSD(a,b)= a*b/ NSN(a,b)
- U algoritmu RSA platí, že se místo Eulerovy funkce $\phi(N)=(p-1)\cdot(q-1)$ může použít Carmichaelova funkce $\lambda(N)=NSN(p-1,q-1)$, kde $\phi(N)=\lambda(N)\cdot NSD(p-1,q-1)$
  - https://en.wikipedia.org/wiki/Carmichael_function
  - https://crypto.stackexchange.com/questions/29591/lcm-versus-phi-in-rsa

Modulární aritmetika - operace

- **Sčítání** a+b mod n
- Odečítání a-b mod n = a+(-b) mod n
  - tj. potřebujeme sčítání s **opačným** prvkem vzhledem ke sčítání,
  - Pro prvek a a k němu opačný (-a) musí platit, že $a+(-a) \equiv 0 \bmod n$
- **Násobení** a*b mod n
  - odvozeno z opakovaného sčítání
- Dělení a/b mod n
  - je násobení s multiplikativním **inverzním** prvkem k b:
    $a/b \bmod n = a*b^{-1} \bmod n$
  - je-li n prvočíslo, $b^{-1} \bmod n$ existuje (ale neexistuje k 0), $b*b^{-1} \equiv 1 \bmod n$, pro čísla n složená nemusí b^{-1} existovat

Modulární aritmetika

$+$	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

$\cdot$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

a	$(-a)$	a^{-1}
0	0	–
1	5	1
2	4	–
3	3	–
4	2	–
5	1	5

Operace $(+)$, $(\cdot)$ a opačný a inverzní prvek v $\mathbb{Z}_6$

Určení inverzního prvku jinak

- Fermatova věta
  - Nechť p je prvočíslo, a je celé číslo a NSD(a,p)=1 potom $a^{p-1} \equiv 1 \bmod p$
- Několik důležitých vět je založeno na $\Phi(n)$
- Eulerova věta (zobecnění Fermatovy věty)
  - Pro libovolné $n \geq 2$ a pro celé číslo a takové, že NSD(a,n)=1 platí $a^{\phi(n)} \equiv 1 \bmod n$
- Určení inverzního prvku:
  - Je-li n malé, hledej 1,..., n-1 dokud není nalezen a^{-1} takový, že

    $a * a^{-1} \equiv 1 \bmod n$
  - Je-li známo $\Phi(n)$, potom z Eulerovy věty máme (pokud NSD(a,n) = 1):

    $a^{-1} \equiv a^{(\phi(n)-1)} \bmod n$
  - →Jinak se použije EEA