Scurta introducere în Teoria numerelor

Ana Maria Bâcă

30.04.2020

Contents

1 Motivație **2**

2 Grupuri. Inele. Corpuri **2**

3 Aritmetică. Congruențe **3**
3.1 Introducere 3
3.2 Noțiuni matematice 4
3.3 Algoritmul Diffie-Hellman de schimb de chei 8

4 Lema chineză a resturilor **9**

5 Corpuri Galois.Algoritmul AES **10**

6 Algoritmul RSA **13**

7 Lecturi suplimentare, cursuri online **14**

1. Motivație

Acest mic curs își propune să facă o incursiune elementară în teoria numerelor folosită în doi algoritmi criptografici celebri: AES și RSA.

Studiul teoriei numerelor începe în jurul anului 300 î.Hr. când Euclid a demonstrat ca există o infinitate de numere prime și a dedus teoremele fundamentale ale aritmeticii, cum ar fi că orice număr poate fi descompus în factori primi. În jurul anului 972 d.Hr matematicienii arabi au formulat criteriul de congruență care ne permite să decidem dacă un anumit număr pozitiv întreg n poate reprezenta aria unui triunghi dreptunghic ale cărui laturi sunt exprimate prin numere raționale. În anul 1976, Diffie și Hellman au introdus primul sistem criptografic cu chei publice, sistem care a permis comunicarea securizată între persoane folosind un canal public. Acest sistem folosește concepte de congruență a numerelor pe care le vom studia și noi în acest curs. Criptografia anilor 80-90 a fost îmbunătățită tot prin intermediul teoriei numerelor, mai precis prin concepte precum curbe eliptice și teste de primalitate.

Punctul culminant al teoriei numerelor a fost atins în 1994 când matematicianul englez Andrew Wiles a demonstrat marea teoremă a lui Pierre de Fermat enunțată în 1637: Pentru $n > 2, n \in \mathbb{N}$ și $x, y, z \in \mathbb{Z}_+^*$ecuația $x^n + y^n = z^n$ nu are soluții.

Toate aceste aplicații din teoria numerelor au oferit criptografiei suportul necesar pentru comunicații sigure, făcând matematica să joace un rol extrem de important în acest domeniu al calculatoarelor.

2. Grupuri. Inele. Corpuri

Definiția 2.1. Un grup G este o mulțime, împreună cu o operație binară pe G, notată: $\cdot : G \times G \to G, (x, y) \to x \cdot y$, astfel încât:

- G_1 (asociativitate) $(x \cdot y) \cdot z = x \cdot (y \cdot z), \forall x, y, z \in G$.
- G_2 (element neutru) $\exists e \in G$ astfel încât $x \cdot e = e \cdot x = x, \forall x \in G$.
- G_3 (inversabilitate) $\forall x \in G, \exists x^{-1} \in G$ astfel încât $x \cdot x^{-1} = x^{-1} \cdot x = e$.

Dacă în plus are loc:

- G_4 (comutativitate) $x \cdot y = y \cdot x, \forall x, y \in G$

spunem că G este **comutativ sau abelian**.

Exemple de grupuri:

a) $(\mathbb{Z}, +), (\mathbb{Q}, +), (\mathbb{R}, +), (\mathbb{C}, +), (\mathbb{Z}_n, +), (\mathbb{Q}^*, \cdot), (\mathbb{R}^*, \cdot), (\mathbb{C}^*, \cdot)$.

b) $\mathcal{U}(\mathbb{Z}) = \{-1, 1\}$,$\mathcal{U}(\mathbb{Z}_n)$.

Definiția 2.2. Ordinul unui element a al unui grup $(G, \cdot)$ este cel mai mic număr întreg pozitiv k astfel încât: $a^k = a \cdot a \cdot \ldots \cdot a = e$, unde e este elementul neutru din grup.

Definiția 2.3. Un grup $(G, \cdot)$ care conține un element a al cărui ordin este egal cu cardinalul mulțimii G: $ord(a) = |G|$ se numește grup ciclic. Elementul a poartă denumirea de element primitiv (sau generator).

Definiția 2.4. Un inel R este o mulțime, împreună cu două operații binare pe G, notate: $+ : R \times R \to R$, $\cdot : R \times R \to R$, astfel încât:

- R_1 $(R, +)$ este grup abelian.
- R_2 $(R, \cdot)$ este monoid.
- R_3 (distributivitate):

$$x \cdot (y + z) = x \cdot (y + z), \forall x, y, z \in R$$
$$(x + y) \cdot z = x \cdot z + y \cdot z, \forall x, y, z \in R$$

Observație. R este inel integru: $\forall x, y \in R \setminus \{0\}, x \cdot y \neq 0$ (avem voie să simplificăm la dreapta sau la stânga).

Definiția 2.5. R este corp: $\forall x \in R \setminus \{0\}$, x este inversabil în raport cu operația $\cdot$.

Exemple de corpuri:

a) $(\mathbb{Q}, +, \cdot), (\mathbb{R}, +, \cdot), (\mathbb{C}, +, \cdot), (\mathbb{Z}_n, +, \cdot), n$ prim.

Observație. Orice corp este inel integru (domeniu de integritate).

3. Aritmetică. Congruențe

3.1. Introducere

De ce sunt importante aceste concepte? Să pornim de la o problemă simplă: o bancă vrea să îi transmită clientului ei date confidențiale. Banca alege și folosește o cheie de criptare. Cum intră clientul în posesia acestei chei? Telefonul iese din discuție, deoarece banca se teme de cei care interceptează apelurile telefonice. În anii 70 cheile guvernului SUA erau supravegheate și distribuite de COMSEC (Communications Security), transportându-se zilnic tone de chei sub formă de fișe, benzi magnetice sau dischete ce urmau să fie predate destinatarilor.

Cum ar putea doi indivizi, Alice și Bob, să comunice securizat fără a fi interceptați de Eve, dar și fără să se întâlnească sau să solicite un curier? Dacă Alice vrea să îi trimită un mesaj secret lui Bob ar putea încerca următorul lucru: pune mesajul într-o cutie de fier, o încuie și o trimite lui Bob. Bob când primește cutia o închide și el cu lacătul lui și o trimite înapoi la Alice. Când Alice primește cutia, are două lacăte. Alice scoate lacătul ei și o trimite înapoi la Bob, care va putea astfel cu propriul lacăt, la care are cheia, să descopere mesajul.

Pornind de la această anecdotă Whitfield Diffie și Martin Hellman au căutat înnebuniți de ideea de evitare a curierilor tot felul de funcții matematice. Provocarea era să găsească o funcție matematică neinversabilă (sau foarte greu inversabilă). Matematic, pentru o funcție $f : \mathbb{R} \to \mathbb{R}$ trebuie ca:

- Calculul $y = f(x)$ să fie simplu computațional.
- Calculul inversei $x = f^{-1}(y)$ să fie nefezabil chiar și cu un supercomputer.

Un răspuns oferit de matematică este aritmetica modulară pe care o vom studia în acest capitol.

3.2. Noțiuni matematice

Teorema 3.1. Dacă a și b sunt numere întregi, $a \neq 0$, atunci există o unică pereche de numere întregi q, r astfel încât

$$b = aq + r, 0 \leq r < |a| \tag{3.1}$$

Definiția 3.1. Fie numerele întregi m, a, b cu $m \neq 0$. Numerele a și b sunt congruente modulo m dacă $m|a - b$, iar m se numește modulul de congruență. Vom scrie

$$a \equiv b \pmod{m} \tag{3.2}$$

Prin intermediul numărului m s-a introdus pe $\mathbb{Z}$ o relație binară numită relație de congruență.

Definiția 3.2. Numărul natural $p > 1$ se numește prim dacă din $p|ab, a, b \in \mathbb{N}$, rezultă $p|a$ sau $p|b$.

Definiția 3.3. Numărul natural $n > 1$ se numește indecompozabil(ireductibil) dacă din $n = ab, a, b \in \mathbb{N}$, rezultă că $a = 1$ sau $b = 1$.

Teorema 3.2. Orice număr prim p este indecompozabil.

Definiția 3.4. Numerele întregi a și b sunt prime între ele, sau coprime, dacă nu au niciun factor prim comun. Vom scrie $(a, b) = 1$.

Teorema 3.3. Orice număr întreg pozitiv poate fi descompus ca produs de puteri de numere prime. Descompunerea este unică până la o permutare.

Teorema 3.4. Există o infinitate de numere prime.
Una dintre cele mai vechi demonstrații îi aparține lui Euclid.

Teorema 3.5. (Conjectura lui Goldbach)
Orice număr întreg pozitiv par $n, n \geq 2$ poate fi scris ca suma a două numere prime.

Teorema 3.6. (Numere prime Mersenne)
Există o infinitate de numere prime de forma $2^n - 1$.
Dacă $2^n - 1$ este prim, atunci n este prim.

Observație. Fie a, b două numere naturale. Notăm $g = (a, b)$ cel mai mare divizor comun al celor două numere.

Teorema 3.7. Fie $g = (a, b)$ cel mai mare divizor comun al celor două numere. Atunci există $x_0, y_0 \in \mathbb{Z}$ astfel încât $g = ax_0 + by_0$.

Observație. Fie d un divizor comun al numerelor a și b. Deoarece $d|a$ și $d|b$, obținem că $d|g$.

Propoziție 3.1. *Dacă $a = bq + r$, atunci $(a, b) = (b, r)$.*

Observație. Un procedeu practic pentru găsirea c.m.m.d.ca două numere naturale este dat de algoritmul lui Euclid. Acesta constă în aplicarea succesivă a teoremei împărțirii cu rest:
$a = bq_1 + r_1, 1 \leq r_1 < b$
$b = r_1q_2 + r_2, 1 \leq r_2 < r_1$
$\dots$
$r_{n-2} = r_{n-1}q_n + r_n, 1 \leq r_n < r_{n-1}$
$r_{n-1} = r_nq_{n+1} + r_{n+1}, r_{n+1} = 0.$
Finititudinea acestui lanț este dată de inegalitățile $b > r_1 > r_2 > \dots r_n$.

Propoziție 3.2. *Ultimul rest nenul din algoritmul lui Euclid este c.m.m.d.c al numerelor a și b.*

Exemple :

a) $(43, 27)$
$43 = 27 \cdot 1 + 16$
$27 = 16 \cdot 1 + 11.$
$16 = 11 \cdot 1 + 5$
$11 = 5 \cdot 2 + 1$
$5 = 5 \cdot 1 + 0.$
$\Rightarrow (43, 27) = 1.$

b) $(2261, 1275)$
$2261 = 1275 \cdot 1 + 986$
$1275 = 986 \cdot 1 + 289.$
$986 = 289 \cdot 3 + 119$
$289 = 119 \cdot 2 + 51$
$119 = 51 \cdot 2 + 17.$
$51 = 17 \cdot 3 + 0$
$\Rightarrow (2261, 1275) = 17.$

Observație. O consecință importantă a algoritmului lui Euclid este rezolvarea ecuațiilor de gradul întâi cu coeficienți întregi.

Teorema 3.8. Ecuația $ax + by = c$ are soluții în numere întregi dacă și numai dacă (a, b) divide c.

Exemplu:

a) $130x + 61y = (130, 61).$
$130x + 61y = 1.$
Folosim algoritmul lui Euclid pentru a determina soluțiile ecuației.
$130 = 61 \cdot 2 + 8 \Rightarrow 8 = 130 - 2 \cdot 61.$
$61 = 8 \cdot 7 + 5 \Rightarrow 5 = 15 \cdot 61 - 7 \cdot 130.$
$8 = 5 \cdot 1 + 3 \Rightarrow 3 = 8 \cdot 130 - 17 \cdot 61.$
$5 = 3 \cdot 1 + 2 \Rightarrow 2 = 5 - 3 \cdot 1 = 5 \cdot 61 - 7 \cdot 130 - 8 \cdot 130 + 17 \cdot 61 = 22 \cdot 61 - 15 \cdot 130$
$3 = 2 \cdot 1 + 1 \Rightarrow 1 = 23 \cdot 130 - 49 \cdot 61.$
$\Rightarrow x = 23, y = -49$ o soluție a ecuației $130x + 61y = 1$.

b) $281x - 133y = 3$.
Ecuația se mai scrie $281x + 133(-y') = 3$
$281 = 133 \cdot 2 + 15 \Rightarrow 15 = 281 - 2 \cdot 133$.
$133 = 15 \cdot 8 + 13 \Rightarrow 13 = 133 - 15 \cdot 8 = 133 - (281 - 2 \cdot 133) \cdot 8 = 17 \cdot 133 - 8 \cdot 281$.
$15 = 13 \cdot 1 + 2 \Rightarrow 2 = 15 - 13 = 281 - 2 \cdot 133 - (17 \cdot 133 - 8 \cdot 281) = 9 \cdot 281 - 19 \cdot 133$.
$13 = 2 \cdot 6 + 1 \Rightarrow 1 = 13 - 2 \cdot 6 = 17 \cdot 133 - 8 \cdot 281 - 6 \cdot (9 \cdot 281 - 19 \cdot 133) = 131 \cdot 133 + (-62) \cdot 281$.
$2 = 1 \cdot 2 + 0$
$\Rightarrow x_0 = 3 \cdot (-62) = -186, y_0 = -3 \cdot 131 = -393$.
Soluția generală a ecuației este $x = 133\lambda - 186, y = 281\lambda - 393$.

Algoritmul extins al lui Euclid
Input: Fie r_0 și r_1 întregi pozitivi cu $r_0 > r_1$.
Output: (r_0, r_1) = c.m.m.d.c , unde $(r_0, r_1) = s \cdot r_0 + t \cdot r_1$.
Pas inițial:

$$s_0 = 1, t_0 = 0$$
$$s_1 = 0, t_1 = 1$$
$$i = 1$$

Algoritm: DO

$$i = i + 1$$
$$r_i = r_{i-2} \pmod{r_{i-1}}$$
$$q_{i-1} = (r_{i-2} - r_i)/r_{i-1}$$
$$s_i = s_{i-2} - q_{i_1} \cdot s_{i-1}$$
$$t_i = t_{i-2} - q_{i-1} \cdot t_{i-1}$$

WHILE $r_i \neq 0$
RETURN

$$(r_0, r_1) = r_{i-1}$$
$$s = s_{i-1}$$
$$t = t_{i-1}$$

Algoritmul extins al lui Euclid ne permite să calculăm inversele elementelor. Această ecuație se poate rezolva cu ajutorul algoritmului lui Euclid, după cum s-a arătat mai sus. Găsirea inversului multiplicativ este un pas esențial în algoritmul RSA, folosit pe scară largă în comerțul electronic; anume, ecuația determină întregul utilizat pentru a decripta mesajul. Deși algoritmul RSA utilizează inele și nu corpuri, se poate folosi algoritmul lui Euclid pentru găsirea inversului multiplicativ acolo unde el există. Algoritmul lui Euclid are și alte aplicații în codurile corectoare de erori; de exemplu, el se poate folosi ca alternativă la algoritmul Berlekamp–Massey pentru decodificarea codurilor BCH și Reed–Solomon, coduri bazate pe corpuri Galois.

Observație. O altă consecință importantă este determinarea inversului unui element modulo m folosind $ax \equiv 1 \pmod{m}$ (sau folosind notațiile dinainte $s \cdot r_0 + t \cdot r_1 = 1 \Rightarrow t = r_1^{-1} \pmod{r_0}$)

În continuare vom reaminti proprietăților congruențelor: Pentru $m \geq 1$ congruența modulo m este o relație de echivalență și deci mulțimea $\mathbb{Z}$ este împărțită în m clase de echivalență notate $\widehat{0}, \widehat{1}, \ldots \widehat{m-1}$. Mulțimea acestor clase se notează $\mathbb{Z}_m$.

Teorema 3.9. $\mathbb{Z}_m$ este inel comutativ cu element unitate.

Definiția 3.5. Fie $m \geq 1$. Atunci $\phi(m) =$ numărul întregilor pozitivi $\leq m$ și primi cu m. $\phi(m) =$ se numește funcția lui Euler.

Teorema 3.10. Fie $m \geq 1$. Dacă descompunerea lui m în produs de puteri de factori primi este $m = p_1^{e_1} \cdot p_2^{e_2} \cdot \ldots p_n^{e_n}$, unde p_i sunt factori primi distincți, iar e_i întregi pozitivi, atunci:

$$\phi(m) = \prod_{i=1}^{n} (p_i^{e_i} - p_i^{e_i - 1}) \tag{3.3}$$

Exemplu: Fie $m = 240$. Atunci $m = 2^4 \cdot 3 \cdot 5 = p_1^{e_1} \cdot p_2^{e_2} \cdot p_3^{e_3}$.
$\Rightarrow \phi(m) = (2^4 - 2^3) \cdot (3^1 - 3^0) \cdot (5^1 - 5^0) = 8 \cdot 2 \cdot 4 = 64$.
Există 64 de numere prime cu 240, iar funcția lui Euler este cea mai rapidă metodă.

Teorema 3.11. (Euler) Dacă $m \geq 1$ și $(a, m) = 1$, atunci $a^{\phi(m)} \equiv 1 \pmod{m}$.

Teorema 3.12. (Fermat) Dacă p este prim și $(a, p) = 1$, atunci $a^{p-1} \equiv 1 \pmod{p}$.

Teorema 3.13. (Fermat) Dacă p este prim și $(a, p) = 1$, atunci $a^p \equiv a \pmod{p}$.

Observație. Fie p prim. Atunci $a^{-1} \equiv a^{p-2} \pmod{p}$ (calculul inversului).

Exemplu:
Fie $p = 7$ și $a = 2$. Atunci inversul lui a este $a^{p-2} = 2^5 = 32 \equiv 4 \pmod{7}$.
Verificăm rezultatul: $2 \cdot 4 \equiv 1 \pmod{7}$.
Exemplu

a) $12^{-1} \pmod{67}$
Folosim algoritmul lui Euclid.
$(12, 67) = 1 \Rightarrow 67 = 12 \cdot 5 + 7 \Rightarrow 7 = 67 - 5 \cdot 12$.
$12 = 7 \cdot 1 + 5 \Rightarrow 5 = 12 - 7 \cdot 1 = 12 - 67 + 5 \cdot 12 = -67 + 6 \cdot 12$.
$7 = 5 \cdot 1 + 2 \Rightarrow 2 = 7 - 5 = 67 - 5 \cdot 12 + 67 - 6 \cdot 12 = 2 \cdot 67 - 11 \cdot 12$.
$5 = 2 \cdot 2 + 1 \Rightarrow 1 = 5 - 2 \cdot 2 = -67 + 6 \cdot 12 - 2 \cdot (2 \cdot 67 - 11 \cdot 12) = -5 \cdot 67 + 28 \cdot 12$.
$\Rightarrow 12^{-1} \equiv 28 \pmod{67}$.

Teorema 3.14. (Wilson) Dacă p este prim, atunci $(p-1)! \equiv -1 \pmod{p}$.

Teorema 3.15. Dacă $x^2 \equiv 1 \pmod{p} \Rightarrow x \equiv \pm 1 \pmod{p}$.

Teorema 3.16. Dacă ecuația $x^2 \equiv -1 \pmod{p}$ are soluții $\iff p = 2$ sau $p \equiv 1 \pmod{4}$.

Teorema 3.17. Există o infinitate de numere prime de forma $4k + 1$.

Definiția 3.6. Pentru numerele întregi a, b, m cu $m \geq 1$ vom considera congruența $ax \equiv b \pmod{m}$ pe care o vom numi congruență de gradul întâi cu necunoscuta x.

Exemple:

a) $3x \equiv 2 \pmod{7}$

Congruența se verifică pentru $x_0 = 3$ deoarece $3 \cdot 3 - 2 \vdots 7$.
Numerele $10, 17, -4$ verifică de asemenea congruența. Toate aceste numere se află în clasa $\hat{3}$ din $\mathbb{Z}_7$.

b) Congruența $3x \equiv 2 \pmod{6}$ nu are soluție. Dacă $3x_0 \equiv 2 \pmod{6}$, atunci $6|3x_0 - 2$, deci $3|3x_0 - 2$, adică se obține contradicția $3|2$.

Amintim că dacă o congruență $ax \equiv b \pmod{m}$ este verificată de numărul x_0, atunci soluția este $\hat{x_0} \in \mathbb{Z}_m$. Condițiile în care o astfel de congruență are soluție sunt date de următoarea teoremă.

Teorema 3.18. Congruența $ax \equiv b \pmod{m}$ are soluție dacă și numai dacă $d|b$,unde $d = (a, m)$. Dacă $d|b$, atunci congruența are d soluții.

Teorema 3.19. Dacă $(a, m) = 1$, atunci congruența $ax \equiv b \pmod{m}$ este verificată de $x_0 = ba^{\phi(m)-1}$.

Exemplu

a) $35x \equiv 14 \pmod{28}$.
$d = (35, 28) = 7 \Rightarrow 7$ soluții.
$35x \equiv 14 \pmod{28} | : 7$.
$5x \equiv 2 \pmod{4}$.
$\Rightarrow 2, 6, 10, 14, 18, 22, 26.$ (cele 7 soluții ale congruenței)

Definiția 3.7 (Problema logaritmului discret)**.** Fie un grup ciclic finit $\mathbb{Z}_p$ de ordin $p-1$ și elementele primitive (generatoare) $a, b \in \mathbb{Z}_p$. Problema logaritmului discret este determinarea unui între $x \in [1, p-1]$ astfel încât $a^x \equiv b \pmod{p}$

3.3. Algoritmul Diffie-Hellman de schimb de chei

Ideea algoritmului de schimb de chei are la bază rezolvarea unei funcții neinversabile de forma $a^x \equiv b \pmod{p}$. Numărul prim p trebuie să fie cât mai mare (cel puțin 512 biți).

Alice și Bob care doresc să comunice, cad de acord asupra numărului prim p și asupra generatorului grupului a. Astfel $ord(a) = p$. Protocolul Diffie-Hellman este descris în continuare

1. Alice alege un element x, calculează $X = \alpha^x \pmod{p}$ și îl trimite lui Bob
2. Bob alege un element y, calculează $Y = \alpha^y \pmod{p}$ și îl trimite lui Alice
3. Alice și Bob determină cheia calculând X^y și Y^x

Observăm în protocolul de mai sus că Eve care ar dori să intercepteze mesajul are acces la X, Y, p (și implicit și la α, deoarece poate determina generatorul grupului ciclic), dar fiind numere prime mari pentru a calcula logaritmul discret (adică x și y) are nevoie de putere foarte mare de procesare.

Să luăm un exemplu simplu: fie grupul $\mathbb{Z}_{47}^*$ care are ordinul 46. Elementul $\alpha = 5$ este generator al grupului. Pentru $\beta = 41$ problema logaritmului discret pentru Eve ar fi: găsiți x astfel încât $5^x \equiv 41 \pmod{47}$. Printr-o metodă "brute-force" de încercări repetate ale valorilor lui x găsim soluția $x = 15$.

4. Lema chineză a resturilor

Problema rezolvării sistemelor de congruențe cu o singură necunoscută este tratată cu ajutorul lemei chineze a resturilor.

Teorema 4.1. Fie $m_1, m_2, \ldots m_s$ numere întregi nenule astfel încât $(m_i, m_j) = 1$ pentru $i, j \in \overline{1, s}, i \neq j$ și numerele întregi $a_1, a_2, \ldots a_s$. Sistemul

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} && (4.1)\\ x &\equiv a_2 \pmod{m_2} && (4.2)\\ &\ldots && (4.3)\\ x &\equiv a_s \pmod{m_s} && (4.4)\\ & && (4.5) \end{aligned}$$

are soluții și toate soluțiile sunt congruente modulo M, unde $M = m_1 m_2 \ldots m_s$.

Exemplu: Să se rezolve sistemul:

a)

$$\begin{aligned} 3x &\equiv 2 \pmod{5}\\ 4x &\equiv 1 \pmod{3}\\ 5x &\equiv 3 \pmod{7}\\ 3x &\equiv 1 \pmod{4} \end{aligned}$$

Soluție:
Sistemul se scrie

$$\begin{aligned} x &\equiv 4 \pmod{5}\\ x &\equiv 1 \pmod{3}\\ x &\equiv 2 \pmod{7}\\ x &\equiv 3 \pmod{4} \end{aligned}$$

Folosind notațiile folosite anterior avem $m_1 = 5, m_2 = 3, m_3 = 7, m_4 = 4, M = 5 \cdot 3 \cdot 7 \cdot 4 = 420$. $M_1 = 84, M_2 = 140, M_3 = 60, M_4 = 105$.

Ecuațiile $84y \equiv 1 \pmod 5), 140y \equiv 1 \pmod 3), 60y \equiv 1 \pmod 7), 105y \equiv 1 \pmod 4)$ au soluțiile $y_1 = 4, y_2 = 2, y_3 = 2, y_4 = 1$ și deci $e_1 = 336, e_2 = 280, e_3 = 120$ și $e_4 = 105$.

O soluție a sistemului este $x_0 = 4 \cdot 336 + 1 \cdot 280 + 2 \cdot 120 + 3 \cdot 105 = 2179 \equiv 79 \pmod{420}$.

Numerele care verifică sistemul sunt $79 + 420\lambda, \lambda \in \mathbb{Z}$.
Am introdus lema chineză a resturilor deoarece aceasta este folosită de algoritmul RSA pentru a face mai rapid operația de decriptare. Un exemplu este aici https://www.di-mgt.com.au/crt_rsa.html Implementări găsim în calculele de verificare a unei semnături folosind standardele PEM (Privacy Enhanced Email) și standardul PKCS (Public Key Cryptographic System).

5. Corpuri Galois.Algoritmul AES

O scurtă recapitulare a corpurilor Galois.

Definiția 5.1. Un corp finit cu p^n elemente se mai numeste și corp Galois și se notează $GF(p^n)$ sau F_{p^n}, unde $p, n \in \mathbb{N}, p$ prim.

În criptografie lucrăm cu corpuri finite(cu un număr finit de elemente) - corpuri Galois. Numărul elementului se numește ordinul corpului.

Teorema 5.1. Un corp de ordin m există dacă m poate fi exprimat ca puterea unui număr prim (i. e $m = p^n$), pentru n întreg pozitiv și p prim.
Numărul p se numește caracteristica corpului finit.

Putem deduce astfel că există corpuri finite cu $11, 13$ elemente sau cu 25 de elemente sau cu 256 elemente. Cu toate acestea, nu există corpuri finite cu 15 elemente.
Cele mai simple exemple de corpuri finite sunt corpurile cu ordin prim.

Fie $GF(p), p$ prim. Elementele lui $GF(p)$ sunt de forma $\{0, 1, \ldots p-1\}$. Corpul este înzestrat cu operațiile de adunare și înmulțire modulo p.
Am reamintit proprietățile corpurilor și proprietăților congruențelor în prima parte a cursului.

Unul dintre cele mai importante corpuri finite este $GF(2)$ - cel mai mic corp finit. $GF(2) = \{0, 1\}$. Aritmetica modulo 2 este trivială. În $GF(2)$ adunarea modulo 2 este echivalentă cu o poartă logică XOR, în timp ce înmulțirea este echivalentă cu poarta logică AND. Corpul $GF(2)$ este important pentru AES.

Extindere a unui corp finit $GF(2^m)$
Pentru AES (un algoritm de criptare simetrică) corpul finit utilizat are 256 de elemente și se notează $GF(2^8)$. Corpul aceste reprezintă cea mai bună alegere deoarece fiecare element al său poate fi considerat echivalentul unui byte.
Pentru transformările $S-Box$ și $MixColumn$ AES interpretează datele primite cu ajutorul operațiilor

aritmetice din corpul Galois.
Trebuie să remarcăm faptul că ordinul corpului nu este prim 2^8. Astfel de corpuri se vor numi extinderi .
Elementele unei astfel de extinderi sunt polinoame, iar elementele extinderii lui $GF(2^m)$ sunt polinoame cu coeficienți în $GF(2)$ (de grad maxim $m-1$ și cu m coeficienți).
Spre exemplu, un polinom A din $GF(2^8)$ este reprezentat astfel:

$$A(x) = a_7x^7 + \dots a_1x + a_0, a_i \in GF(2) = \{0, 1\} \tag{5.1}$$

Există 256 astfel de polinoame, iar un mod foarte simplu și util de reprezentare a polinomului este un vector de 8 biți:

$$A = \{a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0\} \tag{5.2}$$

Reamintim cum funcționează adunarea și înmulțirea în $GF(2^m)$.

Definiția 5.2. Fie $A(x), B(x) \in GF(2^m)$. Atunci suma lor este polinomul
$C(x) = A(x) + B(x) = \sum_{i=0}^{m-1} c_ix^i, c_i \equiv a_i + b_i \pmod 2$
iar diferența este polinomul $D(x) = A(x) - B(x) = \sum_{i=0}^{m-1} c_ix^i, c_i \equiv a_i - b_i \pmod 2$

Exemplu
Calculați suma polinoamelor $A(X) = x^7 + x^6 + x^4 + 1$ și $B(x) = x^4 + x^2 + 1$ în $GF(2^8)$.
$A(x) + B(x) = x^7 + x^6 + 2x^4 + x^2 + 2 = x^7 + x^6 + x^2$.
Observăm că suma celor două polinoame este egală cu diferența lor.

Definiția 5.3. Fie $A(x), B(x) \in GF(2^m)$ și polinomul ireductibil $P(x) \equiv \sum_{i=0}^{m} p_ix^i, p_i \in GF(2)$.
Atunci rezultatul înmulțirii celor două polinoame este $C(x) \equiv A(x) \cdot B(x) \pmod{P(x)}$.

Deducem că în fiecare corp $GF(2^m)$ avem nevoie de un polinom ireductibil $P(x)$ de grad m cu coeficienți în $GF(2)$. Pentru AES, polinomul ireductibil este $P(x) = x^8 + x^4 + x^3 + x + 1$.

Exemplu
Fie polinoamele $A(X) = x^3 + x^2 + 1$ și $B(x) = x^2 + x$ în $GF(2^4)$.
În plus, $P(x) = x^4 + x + 1$ este polinomul ireductibil din corpul Galois.
Atunci $C'(x) = A(x) \cdot B(x) = x^5 + x^3 + x^2 + x$.
$x^4 = 1 \cdot P(x) + (x + 1)$.
$x^4 \equiv x + 1 \pmod{P(x)}$.
$x^5 \equiv x^2 + x \pmod{P(x)}$.
Astfel $C(x) \equiv x^5 + x^3 + x^2 + x \pmod{P(x)}$.
$C(x) \equiv (x^2 + x) + (x^3 + x^2 + x) = x^3$.
$A(x) \cdot B(x) \equiv x^3$.

Este foarte important să nu confundăm înmulțirea polinoamelor din extinderea lui $GF(2^m)$ cu înmulțirea numerelor întregi . Polinoamele pot fi identificate cu vectori.

La nivelul biților exemplul anterior ar fi arătat astfel:
$A \cdot B = C \Rightarrow (x^3 + x^2 + 1) \cdot (x^2 + x) = x^3 \Rightarrow (1101) \cdot (0110) = (1000)$.
Dar această înmulțire nu este identică cu aritmetica factorilor întregi .
Să presupunem că polinoamele ar fi fost de fapt $(1101)_2 = 13_{10}, (0110)_2 = 6_{10}$. Atunci rezultatul înmulțirii ar fi fost $(1001110)_2 = 78_{10}$, complet diferit de cel obținut în extinderea corpului Galois.
În afară de operațiiloe de adunare și de înmulțire, vom studia și operația de determinare a inversului în $GF(2^m)$.
Determinarea inversului în $GF(2^8)$ este principala operație utilizată de Byte Substitution Transformation

Fie $GF(2^m)$ un corp finit și $P(x)$ polinomul ireductibil asociat. Atunci inversa A^{-1} a unui element nenul $A \in GF(2^m)$ îndeplinește condiția $A^{-1}(x) \cdot A(x) = 1 \pmod{P(x)}$.
Pentru corpurile cu cel mult 2^{16} elemente există tabele în care sunt calculate inversele elementele.
Elementul 0 nu are invers. Pentru AES există convenția ca inputul 0 să aibă drept output tot valoarea 0.

Observație. (EEA în corpuri Galois)
Fie $P(x)$ un polinom ireductibil și $A(x)$ un element din corpul finit $GF(2^m)$.
Atunci $(A(x), P(x)) = 1$

$$s(x)P(x) + t(x)A(x) = (A(x), P(x)) = 1 \Rightarrow s(x) \cdot 0 + t(x) \cdot A(x) \equiv 1 \pmod{P(x)}$$
$$t(x) \equiv A^{-1}(x) \pmod{P(x)}$$

Exemplu: Determinăm inversul lui $A(x) = x^2$ în $GF(2^3)$, unde $P(x) = x^3 + x + 1$

- $t_0(x) = 0, t_1(x) = 1$
- Pasul 1: $x^3 + x + 1 = x \cdot x^2 + x + 1$, $t_2 = 0 - x \cdot 1 \equiv x$
- Pasul 2: $x^2 = x \cdot (x + 1) + x, t_3 = 1 - x \cdot x \equiv 1 + x^2$.
- Pasul 3: $x + 1 = 1 \cdot x + 1, t_4 \equiv 1 + x + x^2$.
- Pasul 4: $x = x \cdot 1 + 0, r_5 = 0$
- Rezultă: $A^{-1}(x) = t(x) = t_4(x) = x^2 + x + 1$.

Puteți citi algoritmul AES, varianta avansată către NIST, în care cei doi autori ai algoritmului Rijndael au definit un algoritm de criptare pe blocuri in care lungimea blocului si a cheii puteau fi independente, de 128 de biți, 192 de biți, sau 256 de biți de aici: https://www.securitatea-informatiilor.ro/solutii-de-securitate-it/algoritmul-de-criptografie-aes/

6. Algoritmul RSA

Acest algoritm pentru criptare asimetrică (i. e cu cheie publică) folosește funcții din teoria numerelor.
Pentru a înțelege cum funcționează acest algoritm, reamintim cum funcționează un algoritm pentru criptarea simetrică.
Pentru criptare și decriptare(decodificare) este folosită aceeași cheie secretă.
În plus, funcțiile de criptare și decriptare sunt foarte asemănătoare.
Algoritmul AES este sigur, rapid și foarte util. Totuși, prezintă și anumite aspecte problematice: existența unui mod sigur de comunicare a cheii, numărul cheilor posibile (exemplu: dacă fiecare utilizator are o pereche diferită de chei într-o rețea cu n utilizatori, atunci există $\frac{n \cdot (n-1)}{2}$ chei, iar fiecare utilizator trebuie să se asigure că celelalte $n-1$ chei sunt sigure), posibilitatea fraudelor (mod nesigur de a transmite cheia).

Pentru a depăși aceste neajunsuri, au apărut algoritmii de criptare asimetrică. La bază se află următoarea idee: cheia de criptare nu trebuie să fie neapărat secretă. Singurul aspect important este ca pentru decriptare să se folosească o cheie secretă. Un exemplu pe care l-am văzut este schimbul de chei Diffie-Hellman.
RSA folosește o cheie publică și o cheie privată.
Matematica din spatele schemei de criptare poate fi redusă la algoritmul lui Euclid, funcția ϕ a lui Euler, mica teoremă a lui Fermat și teorema lui Euler.
Toate aceste concepte au fost studiate mai sus.
Pentru a descifra un mesaj cu ajutorul algoritmului RSA folosim descompunerea în factori primi a numerelor și determinăm numărul elementelor coprime cu acesta prin intermediul funcției lui Euler.

Pentru generarea a două chei (publică și privată) se aleg aleatoriu două numere prime mari p și q, care au același ordin de mărime (e.g. ambele sunt pe 2048 de biți). Se va calcula produsul $n = p \cdot q$. Se va alege apoi, aleatoriu, cheia de criptare e astfel ca e și $(p-1)(q-1)$ să fie relativ prime. Utilizând algoritmul extins al lui Euclid vom calcula exponentul de descifrare d astfel ca

$$ed \equiv 1 \pmod{(p-1)(q-1)} \tag{6.1}$$

Cu alte cuvinte

$$d \equiv e^{-1} \pmod{(p-1)(q-1)} \tag{6.2}$$

Remarcăm faptul că d și n sunt relativ prime. Numerele e și n constituie cheia publică, iar d este cheia privată. Cele două numere p și q nu sunt necesare, dar nu vor fi niciodată făcute publice.
Exemplu:

a) Să considerăm $p = 5, q = 11, e = 17$. Determinăm valorea lui d folosind algoritmul lui Euclid. Fie $N = p \cdot q \Rightarrow N = 5 \cdot 11 \Rightarrow N = 55$. Calculăm $\phi(N) = (p-1) \cdot (q-1) \Rightarrow \phi(55) = 40$. Rezolvăm congruența $ed \equiv 1 \pmod{\phi(N)} \Rightarrow 17d \equiv 1 \pmod{40}$, folosind algoritmul lui Euclid.
$40 = 17 \cdot 2 + 6 \Rightarrow 6 = 40 - 17 \cdot 2$

$17 = 6 \cdot 2 + 5 \Rightarrow 5 = 17 - 6 \cdot 2 = 17 - 2 \cdot (40 - 17 \cdot 2) = 17 \cdot 5 - 40 \cdot 2.$
$6 = 5 \cdot 1 + 1 \Rightarrow 1 = 6 - 5 \cdot 1 = 1 \cdot 40 - 17 \cdot 2 - 17 \cdot 5 + 2 \cdot 40 \Rightarrow 1 = 3 \cdot 40 - 7 \cdot 17.$
Deducem că $d = -7$ sau $d \equiv 33 \pmod{40}$.

b) Fie $p = 13, q = 7$ și $e = 5$. Determinați d folosind exemplul anterior.

Pentru a cripta un mesaj m îl vom diviza în blocuri de lungime mai mică n (cu date binare vom alege cea mai mare putere a lui 2 mai mică decât n). Dacă p și q sunt numere prime de 100 cifre, atunci n va avea sub 200 de cifre, iar fiecare mesaj bloc m_i va avea sub 200 de cifre. Dacă trebuie criptate blocuri de lungime fixă, atunci vom apela la operația de padding cu zero. Mesajul criptat c se va obține prin concatenarea mesajelor c_i care au aproximativ aceiași lungime. Formula de criptare va fi:

$$c_i \equiv m_i^e \pmod{n} \tag{6.3}$$

Pentru a descifra un mesaj se calculează:

$$m_i \equiv c_i^d \pmod{n} \tag{6.4}$$

pentru că:

$$c_i^d \equiv (m_i^e)^d \equiv m_i^{ed} \equiv m_i^{k(p-1)(q-1)+1} \equiv m_i m_i^{k(p-1)(q-1)} \equiv m_i \pmod{n} \tag{6.5}$$

7. Lecturi suplimentare, cursuri online

- https://ocw.mit.edu/courses/mathematics/18-781-theory-of-numbers-spring-2012
- https://crypto.stanford.edu/pbc/notes/numbertheory/
- Algoritmul AES, cursuri video:
  - https://www.youtube.com/watch?v=x1v2tX4_dkQ
  - https://www.youtube.com/watch?v=NHuibtoL_qk
- Algoritmul RSA și Euclid, cursuri video:
  - https://www.youtube.com/watch?v=QSlWzKNbKrU
  - https://www.youtube.com/watch?v=fq6SXByItUI