Întrebări şi exerciţii

1 Coduri

1. Fie codul:

Mesaj	Cuvânt de cod
a	0
b	101
c	11
d	100

   Se cere să se decodifice sirul 0101010111110101011

2. Se consideră mulţimea mesajelor $M = \{m_1, m_2, \ldots, m_n\}$ si mulţimea simbolurilor de cod $S = \{0, 1\}$. Se cere să se construiasca un cod cu proprietatea de prefix având lungimile cuvintelor de cod date mai jos. Să se argumenteze cazurile imposibile.

   (a) $l_1 = 2$, $l_2 = 1$, $l_3 = 2$, $l_4 = 2$;

   (b) $l_1 = 2$, $l_2 = 2$, $l_3 = 3$, $l_4 = 3$, $l_5 = 2$.

3. Aceeaşi cerinţa ca şi la punctul precedent, dar cu mulţimea simbolurilor de cod $S = \{x, y, z\}$:

   (a) $l_1 = 1$, $l_2 = 2$, $l_3 = 2$, $l_4 = 2$, $l_5 = 1$;

   (b) $l_1 = 1$, $l_2 = 2$, $l_3 = 2$, $l_4 = 2$, $l_5 = 1$, $l_6 = 3$;

4. Să se calculeze codul optimal pentru mulţimea simbolurilor de cod $S = \{0, 1\}$, pentru mulţimea de mesaje, cu frecvenţele de apariţie, date: $p_1 = 0.15$, $p_2 = 0.55$, $p_3 = 0.05$, $p_4 = 0.01$, $p_5 = 0.15$, $p_6 = 0.09$;

   Apoi să se calculeze lungimea medie a cuvântului de cod obţinut, şi entropia sursei (H(M)).

5. Acelaşi enunţ ca şi la problema precedentă, dar pentru $S = \{x, y, z\}$.

6. Presupunând probabilitatea ca un bit să fie transmis eronat egala cu 10^{-4}, calculaţi probabilitatea ca un şir de 1000 de biţi să conţină 4 sau mai multe erori individuale.

7. Considerăm polinomul generator $g(X) = X^3+X^2+1$. Luaţi ca informaţie utilă şirul 1001 (4 biti).

(a) Calculaţi biţii de control, şi scrieţi cuvântul de cod (complet);

(b) Modificaţi un bit oarecare, şi verificaţi biţii de control.

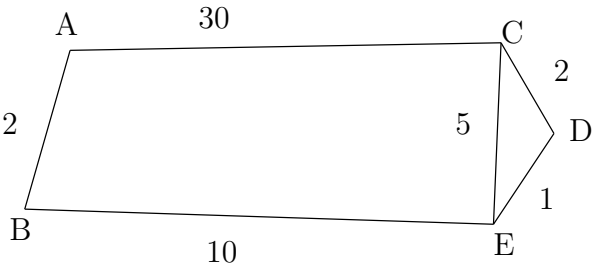
8. Demonstraţi că dacâ polinomul generator este $g(X) = X + 1$, bitul de control este bit de paritate. Arătati că CRC-ul astfel obtinut este capabil să detecteze o eroare.

9. Demonstraţi că dacă polinomul generator divide polinomul X^n+1, unde n este numărul de biţi ai cuvântului de cod, atunci orice permutare circulară a unui cuvânt de cod este cuvânt de cod.

10. Demonstraţi că restul împărţirii cuvântului recepţionat la polinomul generator depinde numai de poziţiile erorilor, nu şi de informaţia transmisă.

2 Nivelul reţea şi nivelul transport

1. Fie reţeaua din figura:



Simulaţi funcţionarea algoritmului de dirijare cu vectori distanta (cost). Simulati comportamentul dupa căderea nodului E. Costurile muchiilor sunt scrise lânga muchii.

2. Modificaţi algoritmul de la punctul 1, astfel ca în tabelul costurilor sa figureze şi drumul de cost minim (cel ce realizeaza costul din tabel). Dacă un drum optim de la un nod la alt nod ar reveni în nodul iniţial, algoritmul va pune cost infinit şi drum inexistent pentru acea pereche de noduri.

   Urmariti comportamentul noului algoritm in cazul penei de la punctul 1.

3. Pentru un protocol de tip fereastră glisantă, gasiti toate cazurile de numere de secvenţă invalide (care nu ar trebui receptionate daca partenerul de comunicatie ar functiona corect) pentru emitator si pentru

receptor. Presupuneţi ca avem la dispoziţie infinit de multe numere de secventa.

3 Criptografie şi aplicaţii criptografice

1. Următorul text este scris in limba româna, numai cu liere mari, fără diacritice, fără spaţii între cuvinte şi fără punctuaţie, după care a fost cifrat prin substituţie monoalfabetică (fiecare literă este substituită cu altă literă din alfabet, corespondenţa fiind aceeaşi la fiecare apariţie):

   ```
   VIDYGGDJGODEHTYGPDVNVKYKPXTGVYUVPQDPXXKT
   KXIXFPKODGXUYKZYGXGKGXDEHTHVSFVZDGXIV
   ```

   Se cere să se descifreze textul.

2. Descărcaţi de pe pagina proiectului Putty cheia publică a proiectului, şi unul din programe împreună cu fişierul semnătură. Verificaţi semnătura (folosiţi comanda `gpg` din linux).

3. La punctul precedent, presupuneţi cheia publică descărcată ca fiind corectă, dar presupuneţi că fişierul conţinând programul *putty* şi fişierul conţinând semnătura au fost descărcate de pe un site mirror suspect. E posibil ca programul descărcat să nu fie cel creat de autorul *putty*? Justificaţi.

4. Folosind gpg (Gnu Privacy Guard), generaţi-vă o pereche de chei pentru semnături digitale şi o pereche de chei pentru cifrare. Cereţi unui coleg cheile lui publice şi daţi-i cheile voastre publice.

   Trimiteţi apoi colegului un mesaj cifrat cu cheia lui publică şi semnat cu cheia voastră secretă, şi cereţi-i să procedeze la fel. Descifraţi şi verificaţi semnătura mesajului primit.

5. Configuraţi, pentru conectarea folosind *putty* la `linux.scs.ubbcluj.ro`, un sistem de autentificare cu cheie publică. Cifraţi cu parolă cheia privată şi utilizţi *pageant* pentru a ţine cheia publică în clar pe durata unei sesiuni.

6. La semnarea documentelor electronice folosind criptografie asimetrică, valoarea funcţiei de dispersie aplicată documentului se trece prin funcţia de descifrare. De ce nu se poate folosi în loc funcţia de cifrare?

7. Dacă A şi B nu au un canal sigur pentru a-şi transmite o cheie, pot totuşi să-şi stabilească un canal de comunicaţie confidenţial fie folosind criptografie asimetrică, fie stabilindu-şi o cheie de cifrare prin protocolul Diffie-Hellman. Arătaţi cum un intrus activ T poate să comunice cu A dându-se drept B şi cu B dându-se drept A, putând obţine efectiv interceptarea canalului dintre A şi B (atacul se numeşte man-in-the-middle, omul-din-mijloc). De asemenea, justificaţi de ce o apărare impotriva acestui atac nu este posibilă.

8. Pe baza punctului anterior, explicaţi mesajul de avertizare dat de orice program client *ssh* la prima conectare la un server. Explicaţi de ce, dacă la prima conectare clientul s-a conectat la serverul adevărat, la cea de-a doua conectare nu mai este nici un pericol.

4 Protocoale de nivel aplicaţie

1. Explicaţi de ce protocolul SMTP nu are prevăzută autentificarea clientului. Ce dificultăţi ar întâmpina introducerea autentificării?

2. Arătaţi de ce protocolul SMTP nu este capabil să transmită (direct) mesaje binare (cu conţinut oarecare).

3. Pentru a putea ataşa fişiere la un mesaj de poştă electronică, ar exista posibilitatea reprezentării fiecărui octet din fişerul ataşat ca o secvenţă de două cifre hexa. Arătaţi de ce se preferă codificarea „în baza 64“ şi cât de mare este avantajul.

4. De ce protocolul HTTP nu necesită codificarea specială a fişierelor binare. Comparaţi cu SMTP.

5. De ce în protocolul FTP la aducerea unui fişier nu este necesară informarea prealabilă a clientului cu privire la lungimea fişierului? Cum determină clientul lungimea fişierului adus?