Nume: Buhai Darius

Grupă: 334

Subiecte netratate: -

Exercițiul 1

(a): **Fals** - Decriptarea, folosind OTP, a textului criptat 0x253505ba folosind cheia 0x717056ee este mesajul clar TEST.

(b): **Adevărat**

(c): **Adevărat**

(d): **Fals** - Un PRP presupune ca pentru fiecare intrare, iesirea sa contina exact bitii de intrare, permutați pseudoaleatoar (ex. pentru o cheie fixata K si PRPK : $\{0,1\}^4 \rightarrow \{0,1\}^4$, PRPK(1101) = 1011 poate fi o atribuire corecta dar PRPK(1101) = 0101 este întotdeauna o atribuire incorectă).

(e): **Fals** - Este recomandat să se folosească RSA pentru transmiterea cheilor în mod criptat.

(f): **Fals** - Pentru a asigura integritatea unor fisiere personale, este suficient sa stocați pe calculatorul propriu fișierele semnate digital și valoarea SHA256 corespunzătoare fiecăruia sub forma (file1,SHA256(file1)), (file2,SHA256(file2)). . . . .

(g): **Fals** - SHA256(PAROLA) = 0x467b4a3eca61a4e6 2447400d93fc35d4 295c08ffa2b04ae9 42f4de03fa62f464

(h): **Adevărat** - Deși problema reală este Diffie-Hellman ce poate fi rezolvată foarte ușor dacă putem rezolva Problema Logaritmului Discret.

(i): **Adevărat**

(j): **Adevărat**

Exercițiul 2

(a): Principiul **Principle of (key) separation** este satisfacut deoarece site-ul foloseste protocolul TLS cu 2 chei pentru integritate și 2 pentru confidențialitate.

(b): **Principle of the weakest link** nu este satisfacut deoarece câmpurile de introducere a datelor nu sunt sanitizate, iar utilizatorii pot plasa comenzi cu prețuri modificate de ei (prețuri negative chiar). Nesanitizarea câmpurilor poate fi cel mai slab punct al website-ului prin care un posibil atacator accesa vulnerabilitățile site-ului (Ex: Comenzi cu prețuri modificate sau chiar și sql-inject).

(c):

Confidențialitate:

- Conexiunea dintre client și server este confidențială deoarece foloseste TLS.
- Fișierele stocate local nu sunt confidențiale, deoarece algoritmul AES-ECB poate fi foarte ușor spart.

Integritate

- Integritatea aplicației este verificată prin protocolul TLS ce stabilește conexiunea dintre client și server.
- Conturile utilizatorilor nu sunt integre, deoarece parolele lor pot fi schimbate foarte ușor de un atacator (descris la următorul subpunct) și utilizatorii pot fi impersonați.

(d): Un atacator poate genera un link de **resetare a parolei** folosind username-ul, ziua curentă și funcția PRNG cunoscută (folosită de aplicație) pentru a schimba parola unui utilizator și pentru a se autentifica. Astfel un atacator poate **impersona** orice utilizator ce are cont pe platformă.

Exercițiul 3

(a): Funcția verif() este definită astfel: - Input: **σ (semnatura)**, **pk = (N, e) și m (mesajul inițial)** - Calculăm **$m'' = \sigma^e \pmod N$** - Calculăm **$m' = 0^8 \,||\, 0^7 1 \,||\, FF^x \,||\, 0^8 \,||\, m$**, - Verificam dacă **$m'' = m'$** - Output: **Adevărat sau Fals** în funcție de verificarea anterioară

(b): Da, deoarece:

- Fie **$m_1 = 2m$ și $\sigma_1 = 2^d\sigma$**
- Știm că **$m'' = m'$** (semnătura σ este validă)
- Știm că **$d*e = 1$**
- **$m_1'' = 2^{de}\sigma^e_1 \pmod N = 2\sigma^e \pmod N = 2m''$**
- **$m_1' = 0^8 \,||\, 0^7 1 \,||\, FF^x \,||\, 0^8 \,||\, 2m_1 = 2m'$**
- Prin urmare, **$2m'' = 2m'$** (Adevărat)

(c): Observăm ca mesajul nostru m^- poate fi alterat astfel: - La mesajul m^- se pot adăuga multiplii de N, astfel încât mesajul m să rămână la fel. - **$m = lsb_{|N|/2-1}(m^- \bmod N) = lsb_{|N|/2-1}(m^- + \underline{x*N} \bmod N)$**, unde x este mesajul inserat de un atacator

(d): Pentru a preveni această inserție vom împărți m^- în grupuri de câte |N|/2-1 biți ($m^- = m^-_1 \,||\, m^-_2 \,...\, ||\, m^-_{|N|/2-1}$) și le vom concatena valorile mod 1 în m. - **Pasul 0.** m^- se transformă în $0 < |m| < |N|/2$ astfel: $m = (m^-_1 \bmod 1) \,||\, (m^-_2 \bmod 1) \,||\, ... \,||\, (m^-_{|N|/2-1} \bmod 1)$, unde m^-_i este împărțirea lui m^- în |N|/2-1 grupuri de *biți.*

Notă: m^- = noul nostru mesaj - cel din cerință.

Exercițiul 4

MAC-ul dat nu este sigur, deoarece valoarea lui **m AND NOT(m)** va fi intotdeauna **= 0**, iar functia **MAC** va returna mereu acelasi raspuns (mesajul primit in cele 2 functii este mereu la fel, egal cu 0).

```
Mac'(k, m) = Mac(k, m AND NOT(m)) = Mac(k, 0)
Vrfy'(k, m, t) = Vrfy(k, m AND NOT(m), t) = Vrfy(k, 0, t)
```