

EXAMEN

1) a) i) Stocarea parolelor folosind o funcție hash se face pentru a asigura securitatea lor. Utilitatea ei este că nu putem inversa rezultatul unei funcții hash.

ii) Putem scrie parolele în cazăl în care este necesară obținerea lor în text clar (De ex. o aplicație care permite utilizatorului să-și înveadă parola în text clar în cazăl în care a uitat-o și poate să o deodă că el este utilizatorul printr-o altă metodă)

b) Diferența majoră este că hash-ul este one-way function, în timp ce criptarea este two-way function. Pentru hash nu se poate inversa efectul (obținerea textului clar din loc), dar la criptare acest lucru este posibil.

c) Un atacator cu puteri nemărginite poate construi un dicționar cu toate hash-urile posibile și astfel pentru o parolă stocată printre-un hash, ar putea fi spartă doar verificând hash-ul respectiv în dicționar.

2) a) Formule de criptare:

$$C_0 = \text{ctr}_0$$

$$C_i = P_i \oplus E(k, \text{ctr}_i)$$

P_i - plaintext i
 C_i - ciphertext i

b) Decriptare:

$$P_i = C_i \oplus E(K, \text{ctr}_i)$$

P_i - plaintext

C_i - ciphertext

c) Considerăm împărțirea mesajului de 320 biți în 3 blocuri de 128 biți astfel:

$$128 + 128 + 64 + 64 = 384$$

lungimea mesajului padding

d) Lungimea tag-ului este egală cu cea blocului, deci 128 biți.

e) GCM aduce un plus ^{partea de} autentificare mesajelor datorită tagului "Auth Tag"

~~+) Nu este securitate~~

~~+) Afirmația este falsă deoarece~~

3) a) $\ell = 65537$

N este pe 2048 biți

b) Chiar dacă se observă că avem egalitate între l-uri, acest lucru nu înseamnă că vom avea egalitate între N-uri. $N = p \cdot q$ p, q prime foarte mari

Pt să existe relații $1 < \ell < \lambda(N)$ unde $\lambda(N) = \text{cmmc}(p-1, q-1)$

$$\phi(\ell, \lambda(N)) = 1$$

dici $65537 < \text{cmmc}(p_1-1, q_1-1)$
 $65537 < \text{cmmc}(p_2-1, q_2-1)$

$$\text{viii} \quad (\cancel{65537}, \lambda(u_1)) = 1$$

$$(\cancel{65537}, \lambda(N_2)) = 1$$

v) Avem do dispozitie un challenger care are ca input o mesaj m_0, m_1 , $|m_0| = |m_1|$ și urmări cryptarea uneia dintre ele și un orocel de criptare și decriptare.

$$\vec{m} = \vec{o}^8 \parallel v \parallel m$$

$$v_c = \bar{m}^e \bmod N$$

Né fatoriu de faptul că $c = \bar{m}^e$ și

$$(2\bar{m})^e = 2^e \cdot \bar{m}^e = 2^e \cdot c$$

Introducem în protocol de decriptare rezultatul 2^e. c
 și obținem un m' . Se cunoaște e (este public), iar rezultatul
 este $2^{m_0} \bmod n$. Compar m' cu ~~m~~ ce am introdus
 în challenger și obținu că mesaj criptat a returnat challenger
 = NU e CCA sigur.

Apare problema de înmulțirea cu 2 a mesajului. În momentul înmulțirii cu 2 modul text, privit în binar, este shiftat la stânga cu un bit, deci e posibil ca verocobule să întoarcă o eroare deoarece nu mai este respectată structura $0^8 \| 12 \| m$ un copil în care primul bit al lui v este 1. În acest caz, repetăm procedeu de verificare este o probabilitate de $\frac{1}{2}$ că un bit să fie $\frac{1}{2}$, iar ~~se~~ după n poziții, probabilitatea să am eroare este de $(\frac{1}{2})^{n \rightarrow 0}$.

d) Atacul nu funcționează. Înmulțind cu 2, nu mai este respectată structura celui de-al \underline{ii} -lea bloc de biți (al 2-lea octet) care nu devine 00000100, deci oracolul de decriptare va întoarce eroare de fișiere doboră.

4) a) Se poate utiliza o funcție hash SHA-256 pentru integritatea datelor, dar numai atunci este folosită în cadrul unui HMAC (deasemenea MAC-urile sunt cele care asigură integritatea datelor)

b) Ne folosim de faptul că XOR este comutativ, deci $h(x) = f(x' \oplus x'') = f(x'' \oplus x')$, $x = x' \parallel x''$. În cazul în care $x' \neq x''$, considerăm $y = x'' \parallel x'$ $\Rightarrow h(y) = f(x'' \oplus x') = f(x' \oplus x'') = -h(x)$. Cum $x \neq y \Rightarrow h$ nu este rezistentă la a 2-a preimage.

c) Elăm că h nu este rezistentă la coliziuni, deci pentru un document x poate exista un alt document y care să își acorde semnatură, deci un atacator poate înlocui documentul x cu documentul y și nu s-ar semnaliza niciun problemă.

d) Se poate rezolva prin ~~adăugarea~~ a unui salt.
Se nu modifică funcția f astfel: Fie s un salt aleator ~~unic~~ generat, $|s|=2m$. Putem aplica

$h(x) = f(x' \oplus x'' \oplus S)$. Prin utilizarea lui, $h(x) \neq h(y)$.

e) Cum sistemul de criptare este fluid și se folosește un PRG, determină apoi următoarea vulnerabilitate:

Ei nu folosesc cheie fluidă $\checkmark K = G(\text{key})$, iar cum G determinist, unui text clar îi va corespunde unul singur mesaj criptat. În consecință, un atacator poate observa că nu există aleatorism în sistem, sistemul devenind vulnerabil.

f) Citește următorul principiu lui Kerckhoffs :

Sistemul nu trebuie să fie secret, poate să codă ușor în mâinile adversarului (i.e. securitatea unui sistem de criptare nu constă decât în menținerea secreții a cheii)

5) a) $S = K \oplus a \quad (1)$

$$m = S \oplus b \quad (2)$$

$$w = m \oplus a \quad (3)$$

$$w \oplus b \stackrel{(3)}{=} m \oplus a \oplus b \stackrel{(2)}{=} S \oplus b \oplus a \oplus b = S \oplus a \stackrel{(1)}{=} K \oplus a \oplus a = K$$

b) Nu este sigurană faza de um adversar posiv.
Acesta poate intercepta S, m, w (ce se transmite pe canalele de comunicație).

$$\cancel{S = K \oplus a} \Leftrightarrow$$

$$\text{Iată că } w \oplus b = K \Rightarrow w = K \oplus b$$

$$\Rightarrow w \oplus S = K \oplus b \oplus S \stackrel{(1)}{=} K \oplus b \oplus K \oplus a = a \oplus b$$

$$\begin{aligned}
 s \oplus w &= a \oplus b \quad | \oplus u \\
 s \oplus w \oplus u &= a \oplus b \oplus u \stackrel{(2)}{=} a \oplus b \oplus s \oplus b \\
 &\stackrel{(1)}{=} \underbrace{a \oplus b}_{= K} \oplus k \oplus \underbrace{a \oplus b}_{= K} \\
 &= K
 \end{aligned}$$

Deci $s \oplus w \oplus u = K \Rightarrow$ schema nu este sigura.

- c) Alice și Bob se pot folosi de schimbul Diffie-Hellman (care autentifică prin certificate digitale semnate de entități autorizate sau recunoscute reciproc pentru a preveni un atacator să intrepteze cheile). Pentru a combate un ~~atac~~ atacator ~~activ~~.