Rezolvare Examen

Created	@January 23, 2022 11:19 PM
Class	SSI
Topic	Exam
Materials	18_pages_on_security.pdf Examen 2021.pdf RSA_2021.txt
Reviewed	☐
Property	

Probabil nu e o rezolvare corecta, dar e una sincera

Legenda:

 Sunt destul de sigur ca asta e rezovare, formularea nu e neaparat cea mai buna

 O parte din rezolvare e ok si are sens, dar am impresia ca omit niste chestii

🍎 Spala varza, scriem lemne

1. (Asta parca a fost si intr-un lab)
    a.
        i. Atunci cand stocam parolele intr-o baza de date vrem sa facem ca aflarea acestora (in cazul unei bresede securitate) sa fie cat mai grea, doeoarece oamenii tind sa foloseasca aceeasi parola pe mai multe site-uri
        ii. Un caz ar fi acela al unui manager de parole, avem un “master password” cu care ne conectam, dar parolele sunt criptate deoarece vrem sa utilizam parola initiala cand ne conectam la un site
    b. Un sistem de criptare foloseste o cheie de cripatre, care poate fi folosita pentru a reveni la mesajul initial, o functie hash trebuie sa tina mesajul initial secret, de aceea ii mai spunem “one way function”.
    c. Securitatea perfecta ar fi in cazul in care am putea mapa fiecare input intru output unic, a i sa nu existe coliziuni, daca consideram posibilitatile de input

infintite, asta ar insemna sa generam toate posibilitatile dintr-un spatiu infinit pentru hash-ul acestora, ceea ce e computational imposibil

2. You've got to

   Primele doua sunt similare cu ce e in Pagens on security - 18

   a. ✅ Criptare: $c_0 = ctr\ c_i = E(k, ctr + i) \oplus m_i$

   b. ✅ Decriptare: $m_i = E(k, crt + i) \oplus c_i$

      (Aici e ✅ doar pentru ca nu stiu daca trebuie cumva inclus si partea de Auth tag aici, dar nu cred, formulele strict pt encrypt ar trebui sa fie ✅)

      (Putine explicatii: c se refera la "ciphertext", ctr la "counter", iar m se refera la "mesajul clar", "E" e pur si simplu functia de criptare bloc iar "k" e cheia)

   c. ✅ Cand primim un text clar mai mic decat lungimea blocului de criptare, facem padding pana ajungem la lungimea blocului, in cazul nostru **128+128+64**+64 = **320** +64 = 384 → lungimea mesajului criptat

      (Tind sa cred ca ✅ deoarece Wiki)

   d. 🍎Avem 384 de biti de text criptat $< 2^6$, conform wiki, trebuie sa ne uitam in tabelul din Appendix C in NIST SP 800-38D, aici vedem ca putem sa folosim un tag de lungime 32 pentru pachete de maxim 2^6 biti daca folosim cheia de cel mult 2^20 ori,

      (Exista doua optiuni, am gasit pe wikipedia practic fix ce cerea exercitiul, link-uit intr-un fisier de research, e f posibil ca asta sa vrea profa, caz in care ✅, sau optiunea 2 in care n-am habar ce fac 🤡)

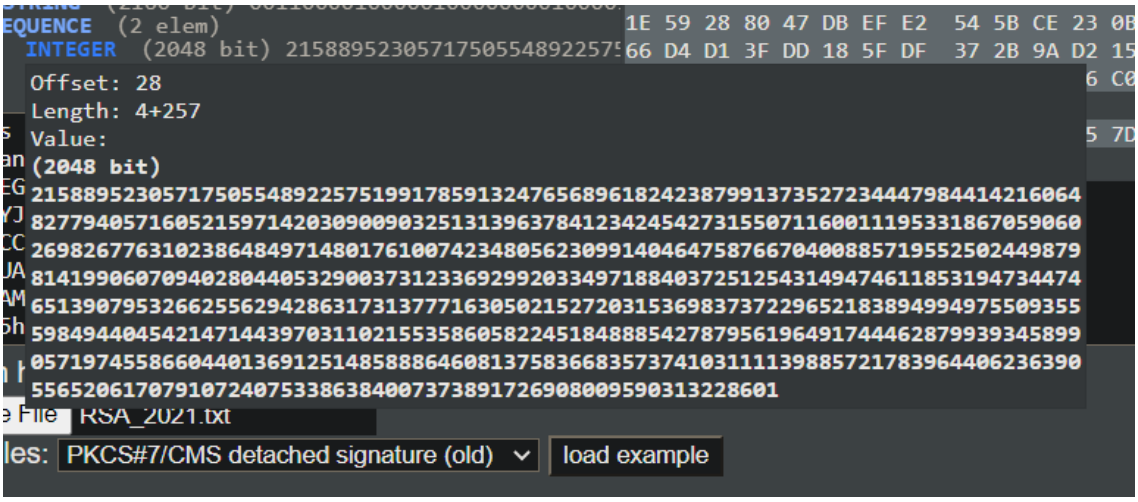
   e. ✅ GCM se foloseste de CTR in operatiile lui, in plus, acesta calculeaza si un tag de autenticitatea a mesajului (notat aici cu Auth Tag), care ne asigura ca mesajul nostru nu a fost alterat de catre un adversar activ

   f. 🍎 Nu, sa avem un sistem PERFECT sigur, ar insemna ca un adversar nelimitat in resurese si timp, sa nu poate extreage niciun fel de informatie, in timp ce un cifru bloc poate fi spart prin brute force

      (al doilea raspuns de aici ar putea sa schimbe raspunsul, but idk)

3. Asta e primul rezultat

   a. ✅Acolo vedem un integer de lungime 2048, trebuie sa fie N,



nu stiu cum plm sa copiez valoare de acolo, e-ul este fix udpa si este 65537

b. N-ul trebuie sa fie mai mare decat e-ul ales, adica N> 65537

c. 🍎

d. 🍎

4.

a. Un atacator cae doreste sa altereze un mesaj poate modifica pachetul iar mai apoi sa calculeze chiar el rezultatul functiei SHA256 pe pachetul alterat, deci doar algoritmul SHA nu este destul pentru a asigura integritatea datelor

b. de aici

> Second preimage resistance is the property of a hash function that it is computationally infeasible to find any second input that has the same output as a given input. This property is related to preimage resistance and one-wayness; however, the later concept is typically used for functions with input and output domain of similar size (see one-way function). Second preimage resistance

$f : \{0,1\}^m \to \{0,1\}^m$, bijectiva, rezisteanta la prima preimagine

$h : \{0,1\}^{2m} \to \{0,1\}^m$ astfel

$h(x) = f(x' \oplus x'')$, unde $x = x'||x''$ si $x', x'' \in \{0,1\}^m$

Cred ca

- rezistenta la prima preimagine insemna ca pentru un anumit hash, ne e imposibil sa localizam care a fost inputul

- rezistenta la a doua preimagine inseamna ca e imposibil sa gasim un al doilea input care sa dea aceelasi hash ca un alt input

Consideram un x de lungime $2k$ unde k este un numar intreg

alegem x' si x'' de lungimi egale

Daca punem conditia in plus ca x sa fie un numar care isi repeta bitii dupa jumatate, atunci ajunglem la $x' = x''$, astfel, pentru orice numar care satisaface conditiile, functia noastra h o sa furnizeze f(0), deci nu este rezistenta la a doua preimagine

(Mai pe scurt, alegem niste numer oarecare (care e binar) si ii repetam bitii, obtinem astfel un numar nou, daca alegem x' si x'' sa fie fix jumatate, functia h o sa fie f(0) pentru fiecare, deci putem sa gasim mai multe numere care sa aiba hasu-ul f(0)

c. Un adversar activ ar putea sa acelasi algoritm sa semneze fisierul iar dupa aceea sa recalculeze hashul

d. Ar trebui sa avem un mecanism de autentificare, unde mesajele sunt semnate de catre cheia secreta a unui user

e. Sistemul PRG sufera de un "Predictible seed" deoarece folosirea datii curente in acel format este previzibila.

f. (Vezi rezolvarile lui Majeri) Incalca principiul nu stiu cui care zice ca securitatea nu ar trebui sa fie prin obscurarea sistemelor si a algorimilor

5.

Alice alege $k, a \leftarrow \{0,1\}^n$ iar Bob primeste $s = k \oplus a$

Bob alege $b \leftarrow \{0,1\}^n$ si ii trimite lui Alice $u = s \oplus b$

Alice calculeaza $w = u \oplus a$ si ii trimite w lui Bob

Alice considera k cheia comuna iar bob $w \oplus b$

a. Extindem $w \oplus b$

- $(u \oplus a) \oplus b$
- $(s \oplus b \oplus a) \oplus b$
- $(k \oplus a \oplus b \oplus a) \oplus b$

Stim ca $\oplus$ este si asociativa si comutativa, deci putem sa “dam cancel” la b si a iar in final ramanem cu k

b. 🍎Nu, deoarece un adversar pasive Eve poate sa ia s-ul calculat de alice iar dupa ce Bob transmite w, Eve poate sa calculeze b, mai mult la urmatorul pas Eve poate sa preia w si sa calculeze si a

c. 🍎 Probabil ca se refera la diffie hellman cu autentificare: https://en.wikipedia.org/wiki/Diffie–Hellman_key_exchange#:~:text=In the original description%2C the Diffie–Hellman exchange by itself does not provide authentication of the