Securitatea Sistemelor Informatice

Lab1
Adversar - O entitate (inclusiv un insider) care acționează rău intenționat pentru a compromite un sistem.

Securitate - O condiție care rezultă din stabilirea și menținerea măsurilor de protecție care permit unei organizații/sistem să își îndeplinească misiunea sau funcțiile critice, în ciuda riscurilor reprezentate de amenințări.

Risc - O măsură a gradului în care o entitate este amenințată de o eventuală circumstanță sau eveniment.

Vulenrabilitate - Slăbiciune într-un sistem informațional, proceduri de securitate ale sistemului, controale interne sau implementare care ar fi putea fi exploatate sau declanșate de o sursă de amenințare

Securitate cibernetica - Capacitatea de a proteja / apăra spațiul cibernetic de atacuri cibernetice

ex. mai de la capat
malware = software care fura date dintr un calculator sau ii provoaca daune lui ori sistemului său

virus = portiune de cod modificata din programe / din sectiunea de boot a computerului care strica functionarea corecta a computerului

dropper = program cu ajutorul caruia se aduce un virus in calculator
downloader = program de instalare a unui software tip de malware care instaleaza in calculator fisiere malitioase fara ca userul sa stie

trojan= un tip de software care mimeaza alte software uri adevarate
spyware = soft care spioneaza si preia datele unui utilizator pentru a le partaja cu altcineva fata de cu cine dorea utilizatorul

riskware = orice program legitim care prezinta riscuri pentru calculator din cauza incompatibilitatii software, vulnerabilitatii securitatii, sau violarii legalitatilor

ransomware = tip de malware care cripteaza parti din memoria unui calculator iar apoi, pentru a le revedea in totalitate, utilizatorul trebuie sa plateasca
adware = malware ce afiseaza reclame nedorite
worm = malware ce se extinde singur fara a fi ajutat de useri sau alte programe

obfuscare = tehnica de scriere a codului care implica scrierea acestuia sub o forma mai greu de inteles pentru a nu fi descifrat cu usurinta

Lab2

Criptologie - Știința care se ocupă de criptanaliză și criptografie.

Criptografie - Disciplina care studiază principiile, mijloacele și metodele de transformare a datelor pentru a ascunde conținutul lor semantic, a preveni utilizarea lor neautorizată sau a preveni modificarea lor nedetectată.

Criptanaliză - Încercarea de a înfrânge protecția criptografice fără o cunoaștere inițială a cheii utilizate în furnizarea protecției.

Confdențialitate - Asigurarea că informațiile nu sunt dezvăluite entităților neautorizate.

Integritate - Protejarea împotriva modificării sau distrugerii necorespunzătoare a informațiilor.

Disponibilitate - Asigurarea accesului și utilizării informațiilor în timp util și fiabil.

Primitivele criptografice sunt algoritmi criptografici care construiesc protocoale criptografice.

Adversar Probabilistic Polinomial în Timp (PPT) = actioneaza pe baza probabilitatilor in timp polinomial --- are si dreptul de a ghici cheia

Securitate computațională
Securitatea computat¸ional˘a mai slab˘a dec ̂at securitatea informat¸ional-teoretic˘a;
l Prima se bazeaz˘a pe prezumpt¸ii de securitate; a doua este necondit¸ionat˘a;
l ̂Intrebare: de ce renunt¸˘am la securitatea perfect˘a?
l Raspuns: datorit˘a limit˘arilor practice!
l Prefer˘am un compromis de securitate pentru a obt¸ine construct¸ii practice.

Ideea de baz˘a: principiul 1 al lui Kerckhofs
Un cifru trebuie s˘a fie practic, dac˘a nu matematic, indescifrabil.
l Sunt de interes mai mare schemele care practic nu pot fi sparte de¸si nu beneficiaz˘a de securitate perfect˘a;

Lab3
Sisteme de criptare istorice -
metoda substitutiei = Rail Fence
metoda permutarii - monoalfabetic = Cezar
- polialfabetic = Vigenere

Lab4

Inginerie social - O încercare de a păcăli pe cineva să dezvăluie informații (de exemplu, o parolă) care pot fi folosite pentru a ataca sisteme sau rețele.

Phishing - O tehnică pentru încercarea de a achiziționa date sensibile, cum ar fi numerele de cont bancar, printr-o solicitare frauduloasă prin e-mail sau pe un site web, în care făptuitorul se maschează ca o afacere legitimă sau o persoană de încredere.

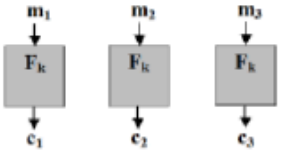
Whaling - Un tip specific de phishing care vizează membrii de rang înalt ai organizațiilor.

Pharming - Utilizarea mijloacelor tehnice pentru a redirecționa utilizatorii către accesarea unui site Web fals, mascat drept unul legitim și divulgarea informațiilor personale.

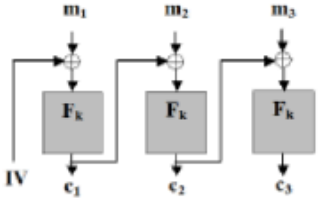
Spear phishing - Un termen colocvial care poate fi folosit pentru a descrie orice atac de phishing foarte vizat.

Spoofing - Falsificarea adresei de trimitere a unei transmisii pentru a obține intrarea ilegală într-un sistem securizat.

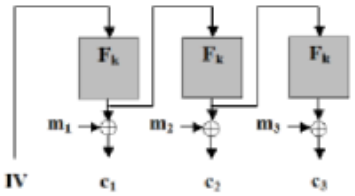
Lab 6



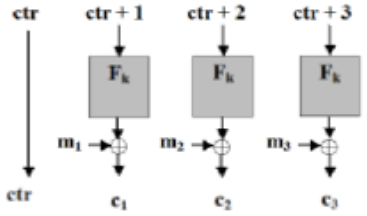
(a) ECB



(b) CBC



(c) OFB

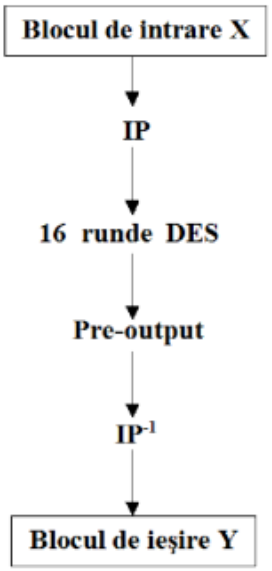


(d) CTR

DES

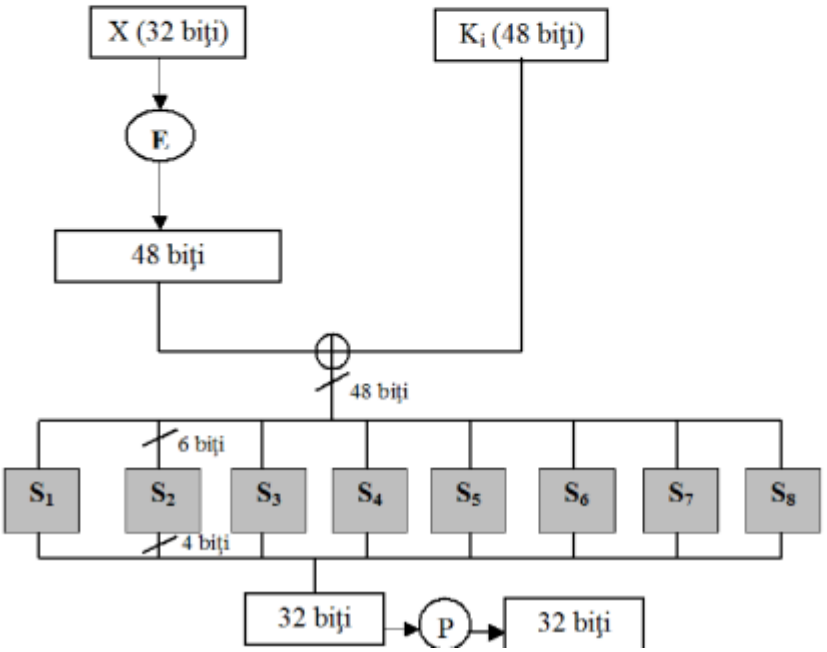
Descriere DES

- DES este o rețea de tip Feistel cu 16 runde și o cheie pe 56 biți;
- Procesul de derivare a cheii (*key schedule*) obține o sub-cheie de rundă k_i pentru fiecare rundă pornind de la cheia master k;
- Funcțiile de rundă $f_i(R) = f(k_i, R)$ sunt derivate din aceeași funcție principală $\widehat{f}$ și nu sunt inversabile;
- Fiecare sub-cheie k_i reprezintă permutarea a 48 biți din cheia master;
- Întreaga procedură de obținere a sub-cheilor de rundă este fixă și cunoscută, singurul secret este cheia master .



S_5		Cei 4 biţi din mijloc															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Biţii din margine	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

- Modificarea **unui bit** de la intrare întotdeauna afectează cel puţin **doi biţi** de la ieşire.



-avem o cheie pe 56 de biti din care luam subchei cu 48 de biti
-la fiecare runda luam mesajul, ii punem padding ca sa aiba 48 de biti
- il xoram cu subcheia de runda
- obtinem 8 blocuri x 6 biti pe care le trecem prin SBox => 8x4 biti
- acesti 32 de biti sunt permutati si apoi mergem la runda urmatoare

Spatiul cheilor este prea mic (2^56) deci alg e vulnerabil la atacuri cu forta bruta
DES are efect de avalansa

Atacul Meet in the Middle asupra DES

- Fie F un cifru bloc (in particular ne vom referi la DES); definim un alt cifru bloc F' astfel

$$F'_{k_1,k_2}(x) = F_{k_2}(F_{k_1}(x))$$

cu k_1, k_2 chei independente;

- Lungimea totală a cheii este 112 biți, suficient de mare pentru căutare exhaustivă;
- Insă, se poate arăta un atac în timp 2^{56} unde $|k_1| = 56 = |k_2|$ (fața de $2^{2\cdot 56}$ cât necesită o căutare exhaustivă);
- Atacul se numeste meet-in-the-middle;

Atacul meet-in-the-middle

- Iată cum funcționează atacul dacă se cunoaște o pereche text clar/text criptat (x, y) cu $y = F_{k_2}(F_{k_1}(x))$:
  1. Pentru fiecare $k_1 \in \{0,1\}^n$, calculează $z := F_{k_1}(x)$ și păstrează (z, k_1);
  2. Pentru fiecare $k_2 \in \{0,1\}^n$, calculează $z := F^{-1}_{k_2}(y)$ și păstrează (z, k_2);
  3. Verifică dacă există perechi (z, k_1) și (z, k_2) care coincid pe prima componentă;
  4. Atunci valorile k_1, k_2 corespunzătoare satisfac

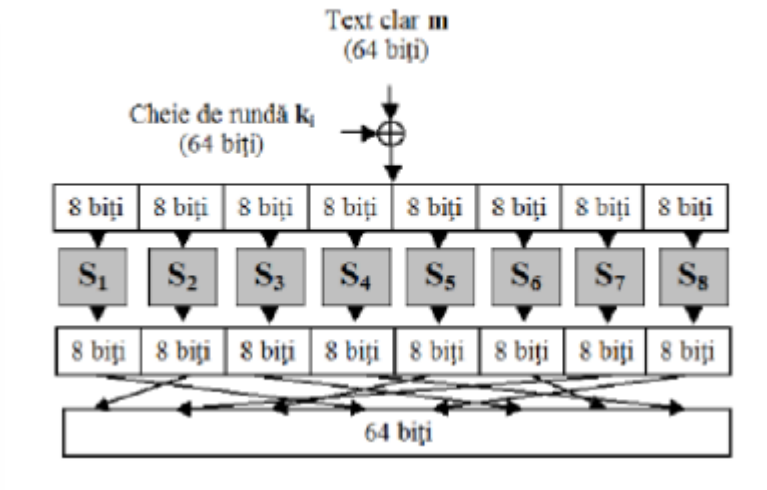
$$F_{k_1}(x) = F^{-1}_{k_2}(y)$$

adică $y = F'_{k_1,k_2}(x)$

- Complexitatea timp a atacului este $O(2^n)$.

AES inlocuieste DES in 2001 dupa ce e spart in 97
-avem un mesaj
-in functie de lungimea lui avem un anumit nr de runde (ex 64 biti - 8 runde)
-o runda:
avem o cheie de runda
xor mesaj si cheie de runda AddRoundKey
=> 8 blocuri de 8 biti
fiecare bloc trece prin Sbox SubBytes

obtinem iar blocuri de 8 biti pe care le permutam si astfel obtinem msj criptat
ShiftRow si MixColumns



Nu exista atac mai eficient decat cautarea exhaustiva pentru AES cu numar complet de runde.

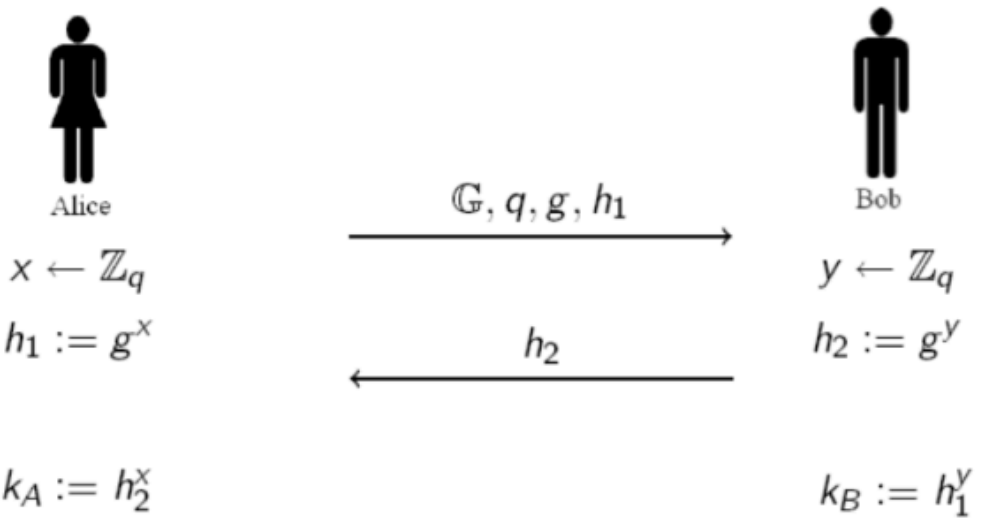
RSA

CA LA PROSTI
avem un N nr mare
vrem sa l factorizam in 2 nr prime p si q (N=p*q)

primim si o cheie publica e
daca nu am reusi sa factorizam N, ar fi foarte greu sa calculam fi(N)=(p-1)(q-1)
, deci nu am putea afla cheia privata d, deoarece d=e^(-1) mod(fi(N)) .

Schimbul de chei Diffie Hellman

Schimbul de chei Diffie-Hellman



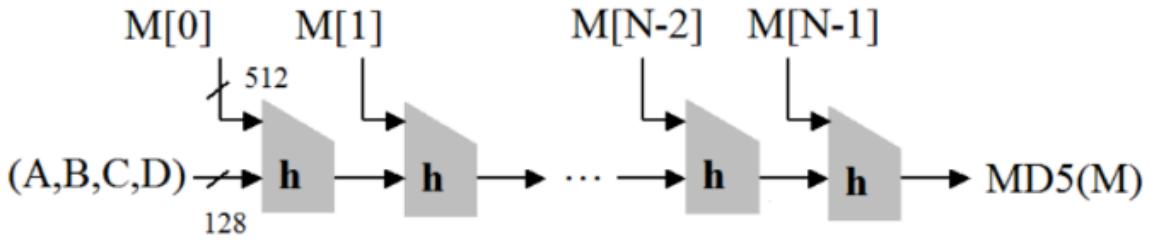
El Gamal
e algoritmul de criptare bazat pe schimbul de chei DiffieHellman

MAC
- coduri de autentificare a mesajelor
- nu ofera protectie la atacurile prin replicare, sau daca atacul e intre partile replicante (atunci e recomandat sa se foloseasca timestamp sau secv de numere)
- CBC MAC e pentru lungimi mai mari

1. criptare si autentificare = nesigur
2. autentificare apoi criptare = nesigur
3. criptare apoi autentificare = mereu sigur!
Este important sa se foloseasca chei simetrice diferite pentru a atinge scopuri diferite.

MD5 - Message Digest 5

- Folosește construcția Merkle-Damgård pentru blocuri de 512 biți și vector de inițializare de 128 biți:



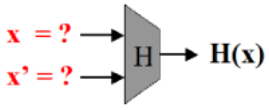
A	=	01	23	45	67
B	=	89	ab	cd	ef
C	=	fe	dc	ba	98
D	=	76	54	32	10

MD5 nu e o functie hash sigura la coliziuni (e inutilizabila in practica).

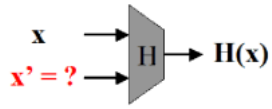
Rezistenta la coliziuni pentru o functie hash

Rezistenta la coliziuni pentru o functie hash

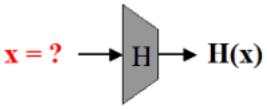
1. functia e rezistenta la coliziuni <=> vrem x!=x` a.i. H(x)=H(x`) (e cea mai puternica notiune de securitate)



2. functia e rezistenta la a doua preimagine <=> stim x si H(x) si vrem sa gasim x` pt care H(x)=H(x`)



3. functia e rezistenta la prima preimagine <=> stim H(x) si vrem sa gasim x (One way function)



a) Amestecarea ingredientelor pentru realizarea unei prăjituri poate fi considerată one-way function. TRUE

b) Funcția hash MD5 este considerată sigură la coliziuni. FALSE

c) SHA256 este o funcție hash cu output pe 256 biți. TRUE

d) Valoarea hash SHA-1 pentru cuvantul „laborator” este 0x4bcc6eab9c4ecb9d12dcb0595e2aa5fbc27231f3. TRUE

e) Este corect să afirmăm că „o funcție hash criptează”. FALSE

f) O funcție hash folosită pentru stocarea parolelor trebuie să fie rapidă (i.e., să se calculeze rapid H(x) pentru x dat). FALSE

g) Hash-ul (fără salt) - 095b2626c9b6bad0eb89019ea6091bd9 – corespunde unei parole sigure, care nu ar fi susceptibilă spre exemplu la un atac de tip dicționar. FALSE
parola123

Lab12

a) Ca să analizați/testați securitatea aplicației, ajută să gândiți ca un atacator.
b) Pentru că sunt foarte multe, din punct de vedere al logicii/design-ului aplicației, nu încercați să acoperiți toate cazurile posibile pentru a preveni un comportament neașteptat.
c) Întotdeauna validați câmpurile de input, atât ca format (tip de date, protejare împotriva SQL injection, etc.) dar și ca valori (dimensiuni, valori minime/maxime, verificări între diferite câmpuri de input; ex. data de început a unei activități anterioară datei de final, prețurile să aibă valori pozitive, etc.)
d) Aveți în vedere vulnerabilități de tip buffer overflow.
e) În general nu e o practică bună să stocați log-uri, pentru că ocupă spațiu și cresc timpul de așteptare al utilizatorului.
f) Oferiți cât mai multe detalii posibile utilizatorilor când eșuează autentificarea prin username și parolă sau când implementați mecanisme de recuperare a parolei, pentru a facilita accesul acestora (spre exemplu menționați „Adresa de e-mail nu corespunde unui cont activ” la încercarea de a recupera parola prin e-mail).
g) Nu rețineți parole în clar.
h) Hardcodați parole în cod.