UNIVERSITATEA POLITEHNICA BUCURESTI

FACULTATEA DE ELECTRONICA,TELECOMUNICATII SI TEHNOLOGIA INFORMATIEI

Protocoale de securitate pentru sisteme distribuite

Minea Maria-Adriana
Sararu Stefania – Alexandra
Grupa 442A

Cuprins

1.1 Introducere

1.1.1 Obiective ale securitatii informative

Securitatea se poate manifesta in functie de cerinte si situatie, in mai multe feluri. Toate entitatile dintr-o tranzactie trebuie sa fie convinse de indeplinirea obictivelor asociate cu securitatea informatiei.

S-au elaborate protocoale si mecanisme cu scopul de a rezolva problemele legate de securitatea informatiei, atunci cand transimisia informatiei se realizeaza prin documente fizice. Exista si situatii cand obiectivele securitatii informatiei nu se pot asigura prin algoritmi matematici si protocoale, in aceste situatii avem nevoie de proceduri tehnice si respectarea unor legi pentru asigurarea rezultatului dorit. In acest sens putem da un exemplu legat de confidentialitatea scrisorilor; aceasta este asigurata de plicurile sigilate si de legea devenita conventie a deschiderii neautorizate a plicurilor , astfel limitandu-se securitatea oferita de un plic.

Sunt si situatii in care securitatea se realizeaza prin documentul fizic pe care este inregistrata. Un exemplu sunt bancnotele care au nevoie de anumite meteriale si cerneluri speciale pentru a nu putea fi falsificate.

Felul in care se face inregistrarea informatiei nu s-a schimbat semnificativ o data cu trecerea timpului. Acum ceva timp informatia era transmisa pe hartie, in zilele noastre aceasta se transmite prin sisteme de telecomunicatii si este stocata pe medii magnetice. In schimb s-a modificat substantial posibilitatea de copiere si chiar modificare a acesteia. Acest lucru implica necesitatea mijloacelor pentru asigurarea securitatii care sa nu depinda de mediul fizic pe care informatia este asigurata sau transmisa, iar obiectivele securitatii informatiei se bazeaza doar pe informatia digitala.

Obiectivele fundamentale ale securitatii informatiei:

1. **Confidentialitatea**- informatia nu este accestibila pentru cei care nu sunt autorizati sa o vada. Se mai foloseste si termenul “secret”.

Sunt mai multe metode de realizare a confidentialitatii (incepand cu protectia fizica, si ajungand la algoritmi matematici care fac datele neinteligibile).

2. **Integristatea datelor**- datele nu se modifica fara sa se autorizeze

cest lucru. Pentru a se asigura aceasta integritate a datelor, trebuie mai intai sa avem capacitatea de a detecta modificarile datelor de catre unele entitati neautorizate. Datele se pot modifica prin substitutie, stergere si inserare.

3. **Autentificarea –** este in stransa legatura cu identificarea.

Se aplica atat in cazul entitatilor cat si la informatie. Daca doua entitati sunt implicate intr-un protocol atunci ele ar trebui sa se identifice mai intai pentru a face si alte schimburi de mesaje. Informatia transmisa printr-un canal ar trebui identificata in legatura cu originea, data la care s-a creat, ora la care a fost trimisa si continutul. Datorita acestul fapt, acest aspect al securitatii informatiei, se imparte in 2 clase :

- Autentificarea entitatilor
- Autentificarea originii datelor

Integreitatea datelor este asigurata de autentificarea originii datelor.

4. **Nerepudierea –** obiectiv al securitatii informatiei ca are rol in

prevenirea negarii a unor actiuni anterioare ale entitatilor. In momentul in care apar unele controverse cu privire la anumite actiuni, trebuie ca acestea situatii sa se rezolve prin diferite mijloace. In generat aici intervine tertul de incredere, care este o a treia entitate, si va rezolva problema.

Plecand de la aceste obiective fundamentale ale securitatii informatiei, se pot obtine si alte obiective precum:

- **Semnatura** – este un mijloc de legare a unei informatii de o entitate;
- **Autorizarea –** reprezinta transferul unei altei entitati a dreptului de a face ceva;
- **Controlul acesului –** presupune restrictionarea accesului la resurse pentru autoritatile care sunt privilegiate;

- **Certificarea –** se refera la girarea informatiei de catre o entitate de incredere;
- **Confirmarea** – se refera la confirmarea furnizarii unui srviciu;
- **Anonimatul** – este obiectivul care are ca scop ascunderea identitatii unei entitati care este implicata intr-un anumit proces.

1.1.2 Protocoale si mecanisme

Protocolul criptografic este algoritmul distribuit definit de o diferenta de pasi care au rolul de a specifica exact actiunile unor entitati (doua sau mai multe) pentru realizarea unui anumit obiectiv de securitate.

Comparativ cu protocoalele, mecanismul de securitate este un concept mai general care cuprinde protocoale, algiritmi si tehnici non-criptografice (cum ar fi protectia hardware) in ideea realizarii unor obiective de securitate.

Exemplu: *"(un protocol simplu pentru stabilirea unei chei) Alice si Bob aleg o schema de criptare cu cheie simetrica pentru a o folosi in comunicarea printr-un canal nesigur. Pentru criptarea informatiei ei au nevoie de o cheie. Protocolul comunicatie este urmatorul:*

1. *Bob alege un sistem de criptare cu chei publice, genereaza pereche de chei (private si publica) si ii trimite lui Alice cheia publica.*
2. *Alice genereaza o cheie pentru cheia de criptare simetrica.*
3. *Alice cripteaza cheia generata la pasul precedent cu cheia publica a lui Bob si ii trimite mesajul criptat.*
4. *Bob decripteaza mesajul primit folosind cheia lui privata si obtinand cheia simetrica (secreta).*
5. *Alice si Bob incep comunicatia securizata folosind criptosistemul cu cheie simetrica ales initial si cheia secreta."*[1]

In multe situatii criptarea cu cheie publica are rolul prezentat in acest protocol: acest tip de criptare (cu cheie publica) se foloseste pentru a face schimb de chei care voi fi ulterior utilizate intr-o schema de criptare simetrica. Argumentul pentru aceasta este diferenta de performanta intre criptarea cu cheii simetrice respectiv

[1] "Protocoale de securitate pentru sisteme distribuite"

criptarea cu chei publice. Criptografia cu chei publice foloseste mai multe chei decat criptografia cu chei simetrice, si este cu mult mai lenta.

Aceste este se intampla atunci cand un mecanismul nu resuseste sa asigure scopul pentru care a fost realizat, astfel un adversar obtine un avantaj prin manipularea mecanismului sau a protocolului.

1.1.3 Esecul protocoalelor si al mecanismelor

Exemplu: *"(esecul unui mecanism) Alice si Bob comunica prin intermediul unui cifru secvential (cifru stream). Se stie ca mesajele pe care le cripteaza au o forma speciala: primi 20 de biti contin o informatie in legatura cu o suma de bani. Un adversar activ poate folosi o simpla operatie XOR intre primii 20 de biti ai textului cifrat si o valoare aleasa aleator pentru a modifica suma respectiva. Cu toate ca adversarul nu reuseste sa descopere depsre ce suma a fost vorba si la ce valoare s-a modificat, totusi a reusit sa alerteze transmisia. Criptarea nu a fost compromisa si cu toate acestea protocolul nu a reusit sa ruleze corect; presupunerea ca o schema de criptare asigura integritatea datelor este evident gresita in acest caz."*[2]

Exemplu : *"(atac prin cautare inainte) Presupunem ca intr-o tranzactie electronica bancara campul de 32 de biti care contin valoarea tranzactiei este criptat folosind o schema de criptare cu chei publice. Acest protocol simplu are scopul de a furniza confidentialitatea valorii din acel camp. Un adversar poate cripta cu usurinta toate secventele binare de 32 de biti, in numar de 2^{32}, folosind cheia publica. Comparand fiecare din cele 2^{32} valori calculate cu cea care este criptata intr-o anumita instanta a protocolului, adversarul poate determina valoarea transmisa in instanta respectiva."*[3]

In exemplu de mai sus este compromise modul in care se foloseste schema de criptare cu chei publice. Mecanismele si protocoalele esueaza din cauze diferite:

[2] "Protocoale de securitate pentru sisteme distribuite"
[3] "Protocoale de securitate pentru sisteme distribuite"

1. Se ingnora principiile care se aplica la o casa larga de primitive (de exemplu scriptarea).
2. Slabiciuni ale unei primitive criptografice ce pot fi accentuate de modul in care sunt utilizate.
3. Presupunerea ca mecanismul sau protocolul de securitate ofera garantii de securitate fara sa fie intelese total toate aspectele acestui protocol sau mecanism.

1.2 Protocoale de identificare si autentificare

1.2.1 Obiective si aplicatiile identificarii

Pentru protocoalele de identificare avem, in ceea ce priveste modelul general, un pretendent (reclamant) A si un verificator B. Verificatorul va primi identitatea pe care reclamantul o pretinde. Scopul este de a oferi autentificarea entitatilor.

Definitie "*Autentificarea entitatilor este procesul prin care o entitate este asigurata (prin obtinerea de dovezi de confirmare) in legatura cu identitatea unei a doua entitati implicate in protocol si ca aceasta entitate a participat efectiv la process(este active in momentul sau imediat inaintea obtinerii dovezii).*"[4]

Obiectivele protocoalelor de identificare

Rezultatul ce poate fi obtinut de pe urma unui protocol de identificare a entitatilor, din perspectiva verificatorului, este acceptarea identitatii ca si autentica din partea reclamantului, sau respingerea, caz in care nu se accepta identitatea. Protocoalele de identificare au urmatoarele obiective:

[4] "Protocoale de securitate pentru sisteme distribuite"

1. Daca A si B sunt entitati oneste, atunci A are capacitatea de autentificare in fata lui B, astfel B va termina protocolul acceptand identitatea lui A.
2. (Transferabilitate) in aceasta sitiatie B nu are voie sa reutilizeze mesajele schimbate cu A pentru a-l impersona pe A in fata tertului C.
3. (Impersonare) aici, probabilitatea este foarte mica ca o entitate C ce difera de A, ce joaca rolul lui A urmand protocolul, sa-l determine pe B sa accepte identitatea lui A.
4. Punctele anterioare vor ramane valabile si in situaita in care au fost inregistrate sau observate un numar polinomial de autentificari care au avut loc intre A si B. C a luat parte initial in executii ale protocolului A si/sau B; mai multe instante ale protocolului care pot fi initiate de care C, ruleaza simultan.

Executia protocol nu dezvaluie nicio informatie partial care ar putea sa usureze sarcina lui C, in cazul protocoalelor cu informatie zero.

Protocolul de autentificare de entitati ofera certitudinea ca entitatea autentificata este prezenta la momentul executiei protocolului (fiind un proces in timp real), si ca a executat unele operatii de la inceputul protocolului. Doar in situatia in care executia protocolului are success, protocoalele de identificare pot oferii garantii.

Facilitarea controlului de acces la o resursa reprezinta unul din scopurile de baza ale identificarii, in situatia in care preivilegiul de acces se leaga de o anumita identitate. Schema parolelor este utilizeaza cu scopul pentru a perimite accesul la contul unui utilizator, putand fi privita ca fiind cea mai simpla forma de matrice pentru controlul accesului: fiecare sursa are o lista de identitati asociate cu ea si confirmarea cu success a unei identitati permite accesul la resursele autorizate. Exista multe aplicatii in care motivatia pentru identificare este aceea de a permite sa se factureze utilizarea resurselor catre entitatile care le folosesc, un exemplu elocvent este in cazul telefoniei mobile. O cerinta esentiala in protocoalele pentru stabilirea autentica a cheilor, este identificarea .

Proprietatile protocoalelor de identificare sunt urmatoarele:

1. **Reciprocitatea indentificarii** se realizeaza atunci cand una /ambele entitati reusesc sa-ti demonstreze indentitatea celeilalte, acorzand identificare unilaterala/bilaterala. Exista si tehnici ce pot fi suspectate ca sunt folosite de catre o entitate, numita entitate adversar, ce are rol de verificator, pentru a obtine parolele utilizatorilor legitimi (de exemplu schemele cu parole fixe).

2. **Eficienta computationala** se refera la cate operatii sunt necesare pentru executia protocolului.

3. **Eficienta comunicationala** se refera la numarul care se schimba si de asemenea la latimea de banda care este necesara pentru efectuarea acestui lucru, altfel spus, la numarul de octeti ce s-au transferat in total.

4. **Implicarea in timp real a unui tert.** In acest caz, tretul poate fi privit ca un tert ce poate participa activ in momentul rularii protocolului distribuind chei simetrice entitatilor implicate; un serviciu de directoare ce participa activ in momentul rularii protocolului, care nu este de incredere si care distribuie certificate de chei publice, bazandu-se pe autoritati de certificare ce nu participa activ in momentul rularii protocolului.

5. **Natura increderii in tert**. In acest sens se poate da ca exemple increderea intr-un tert ce autentifica si leaga de o cheie publica, numele unei entitati; de asemenea , increderea intr-un tert ce stie cheia privata a unei entitati.

6. **Natura garantiilor de securitate.** Aici este vorba despre securitatea demonstrabila dar si despre proprietatile de informatie zero.

7. **Stocarea secretelor**. Se refera la metodele folosite dar si la locul folosit pentru stocarea informatiilor critice pentru a genera cheile. Acest lucru se intampla in software, pe dicuri locale dar si pe elemente hardware.

Schemele de semnatura digitala si schemele de indentificare se aeamana dar in general schemele de identificare sunt mai simple pe cand schemele de semnatura digitala implica mesaje de lungime variabila si au proprietatea de non-repudiere, permitand ca disputele sa fie rezolvate ulterior de catre un tert. In cazul schemelor de identificare, semantica mesajelor este stabilita, este o identitate pretinsa la un moment dat. Pretinderea este confirmata sau este respinsa imediat, privilegiile asociate sau accesul sunt oferite sau sunt respinse. Spre deosebire de semnaturi, identificarile nu au o perioada de valabilitate

1.2.2 Autentificare slaba

Autentificarea slaba rezulta in urma schemelor de parole conventionale care implica parametrii ce nu depind de timp. Fiecarei entitate (verificator) i se asociaza o parola si de obicei este un sir de 6-10 caractere pe care utilizatorul trebuie sa le memoreze. Parola este secretul dintre utilizator si sistem. Aceste scheme de parole conventionale apartin categoriei de tehnici cu cheie simetrica care ofera o autentificare unilaterala. Accesul la o resursa sistem, de exemplu la o imprimanta, se face de catre utilizator , introducand o pereche -id_utilizator, parola- specificand in mod implicit sau explicit o resursa. In cazul nostru id_utilizator este o pretindere a unei identitati, parola fiind dovada pentru acea pretindere. In sistem se verifica daca parola este corecta cu datele primite de la acel id_utilizator, iar entitatea poate accesa resursa. Daca se recunoaste parola atunci ea este acceptata de sistem ca reprezentand o confirmare a identitatii pentru respectiva entitate.

Unele scheme de parole se deosebesc prin modul in care acestea isi stocheaza informatia in sistem permitand verificarea parolei si prin metoda de verificare.

Se ve discuta despre o serie de idei care sunt la baza schemelor de parole care trebuie sa ofere protectie, acestea sunt: dezvaluirea parolei, acest lucru se intampla in afara sistemului, si ascultarea linilor de comunicatii, in cadrul

sistemului, ambele permit atacuri ulterioare prin repetare a parolei, ghicind parola si atac pe baza de dictionar.

Tehnici pentru schemele de parole fixe

Fisiere de parole stocate reprezinta o modalitate simpla de a inregistra parole in sistem se face prin existenta unui fisier protejat la citire si scriere , acest lucru realizandu-se prin privilegii de control al accesului oferit de sistemul de operare, unde parolele sunt trecute in clar. Cand parola este introdusa de utilizator, sistemul verifica parola introdusa cu cea stocata in fisierul de parole pentru acel utilizator. Tinand cont ca nu se folosesc chei secrete, aceasta cheie este una non-criptografica, fiind un mare dezavantaj pentru ca nu ofera protectie impotriva unor utilizatori priveligiati cum ar fi administrator, root, etc., iar stocarea fisierelor pe medii de backup este o mare problema de securitate.

Fisiere de parole „criptate”- in acest caz se prefera folosirea unei functii greu inversibile, in defavoarea stocarii parolelor utilizatorilor intr-un fisier protejat la citire si la scriere. O alta posibilitate este reprezentata de criptarea parolei, dar in acest caz este necesara o cheie. Atat in cazul functiei greu ireversibile cat si in cazul functiei de criptare, parola care rezulta se numeste ca este “criptata”, in ciuda faptului ca in cazul functiei greu ireversibile acest lucru nu este valabil. Sistemul calculeaza functia greu ireversibila pentru parola introdusa, iar dupa aceea, o compara cu valoarea ce este stocata in sistem pentru acel utilizator, acest lucru se face pentru a verifica o parola introdusa de utilizator. Acest lucru se poate observa in Fig.1 de mai jos, in care fisierul de parole trebuie sa fie protejat doar pentru scriere.

Reguli pentru parole - publicate de catre Departamentul Apararii al SUA. Din cauza succesului atacurilor pe baza de dictionar impotriva parolelor slabe in sisteme s-au introdus „reguli pentru parole” in ideea de a preveni utilizatorii care folosesc parole slabe. Una dintre aceste reguli este limitarea minima pentru parole cu 8 caractere, sa aiba cel putin un caracter din fiecare categorie: litere mici, majuscule, cifre, non-alfa-numerice, iar utilizarea unor cuvinte care se gasesc in dictionar este interzisa, dealtfel si parolele legate de contul utilizatorului. Scopul acestor reguli pentru parole este acela de a creste a entropiei parolelor utilizator.

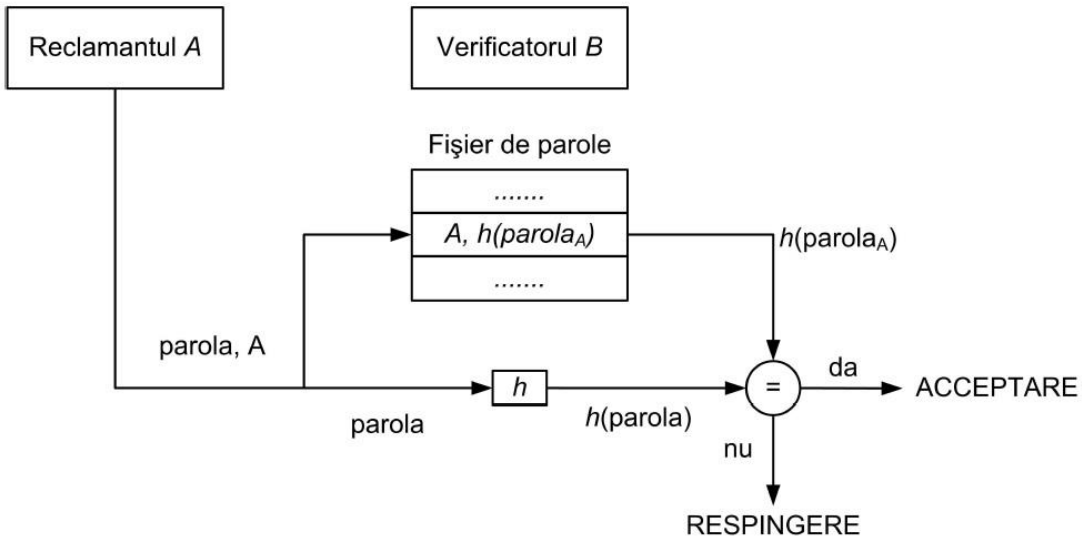


Fig.1[5]- schema prezinta folosirea functiei greu ireversibile pentru verificarea parolei.

Alta modalitatea de a imbunatati securitatea parolelor este de a stabili o perioada de valabilitatea a parolei dupa care este necesara schimbarea ei, acesata se numeste imbatranirea parolelor. Aceasta perioada este de la 30 pana la 90 de zile.

Incetinirea procesului de "criptare" a parolei – se realizeaza functia ce are rolul de a verifica parola, numita functia one-way, iar calcularea acesteia trebuie sa fie un proces mai intens din punct de vedere computational, avand nevoie de timp mai mult. Aceasta functie se realizeaza cu scopul incetinirii atacurilor ce implica testarea unui numar mare de parole. O tehnica prin care se poate realiza acasta functie este de a itera o functie mai simpla de $t > 1$ori. Intrarea pentru iteratia i, reprezinta ceea ce s-a calculate la iteratia precedenta. Pentru a nu aparea intarzieri pentru utilizatorii legitimi, numarul iteratiilor trebuie restrictionat.

Parole cu salt – aceasta notiunea a fost introdusa de Morris si Thompson in raportul facut asupra parolelor UNIX. O parola este extinsa cu un sir de t biti aleatori (acest sir se numeste salt) unde functia one-way este aplicata la parola modificata, in acest mod atacurile pe baza de dictionar nu mai au succes. In fisierul de parole se salveaza rezultatul ce se obtine in urma functiei one-way si salt-ul pentru a putea verifica parola ce a fost introdusa de utilizator. Prin salt, doi utilizatori ce au aceeasi parola dar salt diferit, vor primi in fisierul de parole inregistrari diferite in campul ce contine parola "criptata".

[5] "Protocoale de securitate pentru sisteme distribuite"

Fraze pentru parole – parolele pot si extinse in fraze de parole cu scopul de a avea o entropie mai mare, dar fara depasirea limitei de memorare ale unei persoane. Astfel utilizatorul va memora o fraza scurta in locul unui cuvant. Cu ajutorul unei functii hash, fraza se va transforma intr-o valoare de lungime fixa, ce joaca rolul parolei. Astfel utilizatorii vor putea memora mai simplu o fraza decat o secventa aleatoare de caractere ce rezulta in urma functiei hash.

Atacuri asupra schemelor de parole fixe

Reluarea unei parole fixe - in momentul in care un utilizator tasteaza parola, aceasta poate fi interceptata de catre un adversar, datorita securitatii slabe prin care este transmisa parola in clar pe linia de telecomunicatii catre sistem.

Astfel de parole fixe sunt bune doar daca liniile de telecomunicatii sunt sigure si nu pot fi monitorizate. In Fig. 2 avem un reclamant A(utilizatorul) de la distanta se conecteaza catre un site B. Intr-o astfel de situatie parole trece printr-un canal de comunicatii nesigur.

Cautarea exhaustiva a parolei – atacul nativ este acela in care un adversar incearca in mod aleator si simetric parole, pe rand, in speranta ca va gasi parola corecta. Daca parolele se pot alege dintr-un spatiu mare, atunci acest atac poate fi prevenit, se limiteaza numarul de incercari invalide intr-un interval de timp stabilit.

Se de un fisier cu parole "criptate", un adversar testeaza parolele rand pe rand si compara parola "criptata" cu cela din fisier. Acest lucru se poate intampla din punct de vedere teoretic daca functia one-way si texul clar este cunoscut. Atacul reuseste in functie de numarul de parole care trebuie verificate.

Exemplu *"(entropia parolelor) Presupunem ca parolele sunt formate din caractere ASCII pe 7 biti. Fiecare caracter are o valoare numerica cuprinsa intre 0 si 127.(Atunci cand se folosesc caractere pe 8 biti, valorile 128-255 formeaza setul de carectere extins, care de regula nu este accesibil pe tastaturi standard). Codurile ASCII cuprinse intre 0 si 31 sunt rezervate pentru caractere de control; 32 este caracterul spatiu; 33-126 sunt caractere tiparibile accesibile pe tastatura iar 127 este un caracter special. Tabelul de mai jos ne da numarul de parole distincte formate din n caractere, parole formate din combinatii obisnuite de*

caractere, indicand o limita superioara pentru securitatea oferita de astfel de spatii de parole.”[6]

$\rightarrow c$ $\downarrow n$	26 (litere mici)	36 (litere mici alfanumerice)	62 (litere mici şi mari alfanumerice)	95 (caractere tastatură)
5	23,5	25,9	29,8	32,9
6	28,2	31,0	35,7	39,4
7	32,9	36,2	41,7	46,0
8	37,6	41,4	47,6	52,6
9	42,3	46,5	53,6	59,1
10	47,0	51,7	59,5	65,7

Fig.2[7] Tabelul arata dimensiunea in biti a spatiilor de parole pentru diferite combinatii de caractere.

$\rightarrow c$ $\downarrow n$	26 (litere mici)	36 (litere mici alfanumerice)	62 (litere mici şi mari alfanumerice)	95 (caractere tastatură)
5	0,67 ore	3,4 ore	51 ore	430 ore
6	17 ore	120 ore	130 zile	4,7 ani
7	19 zile	180 zile	22 ani	440 ani
8	1,3 ani	18 ani	1400 ani	42000 ani
9	34 ani	640 ani	86000 ani	$4,0 \times 10^6$ ani
10	890 ani	23000 ani	$5,3 \times 10^6$ ani	$3,8 \times 10^8$ ani

Fig. 3[8] Acest tabel indica timpul necasar pentru cautarea intregului spatiu de parole.

Ghicirea parolei si atacurile pa baza de dictionar. Un adversar poate cauta parolele intr-o ordine descrescatoare a probabilitatii lor, pentru imbunatarirea probabilitatii succesului unei cautari exhaustive. Parolele slabe cu entropii mici sunt usor de ghicit.

[6] “Protocoale de securitate pentru sisteme distribuite”
[7] “Protocoale de securitate pentru sisteme distribuite”
[8] “Protocoale de securitate pentru sisteme distribuite”

PIN si passkey

PIN - personal identification number - face parte din categoria parolelor fixe, fiind folosite pentru un element fizic (de exemplu un card bancar, chipcard), ca sa se poate demonstra identitatea posesorului de card si pentru a putea beneficia de privilegiile cardului. In general parolele de tip PIN sunt scurte, de regula formate intre 4 si 8 cifre, iar din motive de securitate s-au introdus proceduri suplimentare cum ar fi confiscarea cardului daca PIN-ul este introdus gresit de trei ori consecutiv.

PIN-ul este folosit atat in sistemul on-line cat si off-line, unde insistemul on-line verifica comparand PIN-ul introdus cu cel stocat, iar la sistemul off-line,unde nu are acces la o baza de date PIN-ul trebuie sa fie stocat pe un card pentru a permite verificarea parolei aleasa de utilizator.

Parola one-time

Parolele de tip one-time sunt folosite o singura data pentru a oferi protectie fata de alte perosoane care intercepteaza parola si o refolosesc ulterior.

Tipuri de parole one-time:

- liste partajate cu parole one-time folosite secvential;
- parole one-time actualizate secvential;
- secvente de parole one-time bazate pe functii one-way.

Parole one-time bazate pe functii one-way – Schema Lamport

In aceasta schema, utilizatorul incepe cu un secret *w*. Secventa de parole *w, H(w), H(H(w)), ...,* $H^t(w)$ este definite de functie one-way *H.* $w_i = h^{t-1}(w)$ este parola pentru sesiunea $i, 1 \leq i \leq t$.

Protocol (Schema Lamport)

„Se identifica in fata lui *B* folosind parole one-time dintr-o secventa.

1. Initializare

a) Utilizatorul *A* incepe cu un secret *w*. Fie *H* o functie one-way.

b) Se fixeaza o constanta *t (t=100 sau t=1000),* care defineste numarul de identificari permise .(Dupa epuizarea acestui numar sistemul este reinitializat cu un alt *w* pentru a preveni atacurile prin reluare).

c) *A* transfera lui *B* (secretul partajat initial $w_0=H^t(w)$),intr-un mod care garanteaza autenticitatea. *B* initializeaza contorul pentru *A* la $i_A=1$.

2. (Mesajele protocolului) La identificarea numarul i, $1 \leq i \leq t$, se procedeaza astfel:

$$A \rightarrow B: A, i, w_i \ (= H^{t-1}(w))$$

Aici $A \rightarrow B : X$ semnifica faptul ca *A* transmite lui *B* mesajul *X*.

3. (Actiunile protocolului) Pentru a se identifica in sesiunea *i*, *A* executa urmatoarele:

a) *A* se calculeaza $w_i = H^{t-1}(w)$ (ceea ce se poate face simplu pornind de la *w* sau de la o valoare intermediara salvata initial la calculul lui $H(w_i)$) si transmite mesajul lui *B*.

b) *B* verifica faptul ca $i=i_A$, si ca parola primita w_i satisface relatia: $H(w_i)=w_{i-1}$. Daca ambele verificari reusesc, *B* accepta parola, incrementeaza pe i_A ($i_A \leftarrow i_A+1$) si salveaza w_i pentru urmatoarea sesiune de identificare."[9]

1.2.3 Autentificare puternica

In cazul protocoalelor criptografice care au la baza provocare-raspuns, unde o entitate, cu rol de reclamant, trebuie sa isi demonstreze identitatea fata de o alta entitate, care joaca rolul verificatorului, prin adresarea unui raspuns la o provocare ce variaza in functie de timp, de secretul entitatii si de provocare. Needham si Schroeder sunt cei care au initiat o parte semnificativa din munca in domeniul protocoalelor de identificare.

Parametrii variabili in timp

Acesti parametri se folosesc in protocoale de identificare, scopul fiind prevenirea atacurilor prin reluare si intercalare, intr-o unitate de timp dar si impotriva atacurilor ce se bazeza pe text ales. Se mai folosesc si pentru schimbul autentificat de chei in cadrul protocoalelor.

[9] "Protocoale de securitate pentru sisteme distribuite"

Trecerea in revista a parametrilor variabili in timp si a reluarii mesajelor realizata de Gong presupune ca parametrii care au rolul de distingere a unei instante a unui protocol de alta se mai numesc si *nonce* de la number once.

Ca definitie un *nonce* este o valoare care este folosita cel mult o data pentru acelasi scop. Are rolul de prevenire a atacurilor prin reluare ce sunt nedetectabile.

Pentru siguranta parametrilor variabili in timp intalnim urmatoarele clase:

- numere aleatoare
- marci de timp
- numere de secventa

Cerintele protocoalelor pentru acesti parametri este unicitatea.

Numere aleatoare

Numerele aleatoare sunt des intalnite in mecanismele de tip provocare-raspuns, asigurand unicitatea si incadrarea intr-o secventa de timp, scopul fiind sa nu aiba parte de atacuri prin reluare si intercalare. Cele mai dese atacuri sunt cele de tip „text ales”, tocmai de aceea numerele aleatoare sunt nepredictibile pentru un adversar deoarece protocoalele de identificare si autentificare include si numere pseudoaleatoare. Numerele aleatoare fixeaza un punct relativ in timp pentru entitatile implicate, analog cu un ceas partajat. Protocoalele ce implica numere aleatoare necesita numere aleatoare sigure din punct de vedere criptografic.

Numere de secventa

Pentru a identifica si a detecta reluarea unui mesaj este necesar un numar de secventa (numar serial sau valoare contor). In privinta fisierelor stocate, numerele de secventa au rolul de numere de versiune a fiecarui fisier. Fiecarei pereche de entitati ii este atribuita un numar de secventa si trebuie asociat atat implicit cat si explicit cu originatorul si cu receptorul mesajului. Pentru numerotarea mesajelor entitatile folosesc o politica predefinita. Mesajele sunt acceptate doar daca numarul de secventa pe care il contine nu a mai fost folosit. Din cauza unor erori de comunicatii mesajele se pot pierde si pentru ca aceastea sa nu fie interceptate de adversari se impune ca numerotarea mesajelor sa fie crescatoare.

Marci de timp

Marcile de timp au rolul de a detecta reluarea mesajelor, ele sunt incadrate intr-o secventa de timp au unicitate si prin intermediul lor se pot implementa privilegii de acces pe o perioada determinata. Entitatea care trimite un mesaj obtine o marca de timp de la ceasul local si o leaga de mesaj in mod criptografic; atunci cand se receptionaza un mesaj ce contine marca de timp, o a doua entitate obtine ora locala de la sistemul sau, si o compara cu ora primita de la marca de timp. Mesajul va fi valid daca:

- intre mesajul trimis si cel primit sunt de cel mult 20 milisecunde;
- nu a primit un mesaj cu aceeasi marca de timp;

Un rol important la baza marcilor de timp il au ceasurie din sistem ce nu pot fi modificate, este esential ca un adversar sa nu poata da un ceas inapoi sau inainte, pentru a pregati mesaje pe care le poate folosi ulterior.

Dezavantaje:

- spatiu mare pentru stocare si timp mare pentru verificarea mesajelor este considerat un mare dezavantaj pentru protocoalele bazate pe marci de timp;

- depinde de sincronizarea si securizarea ceasurilor din sistem.

Provocare - raspuns prin tehnici simetrice

Aceste tehnici necesita ca reclamantul si verificatorul sa aiba in comun o cheie simetrica. Daca se folosesc tehnici cu chei simetrice in sisteme mari este necasar un servel on-line de incredere, acesta oferind o cheie de sesiune comuna pentru fiecare pereche de entitati care doresc sa se autentifice una in fata celeilalte, in timp ce la sistemele inchise tehnica este mult mai simpla deoarece fiecare pereche de utilizatori poate partaja o cheie simetrica.

Provocare-raspuns bazat pe criptare cu cheie simetrica

Protocoalele Kerberos si Needham-Schroeder au rol de a oferi autentificare de entitati pe baza de criptare simetrica implicand folosirea unui tert on-line de incredere.

In criptare cu cheie simetrica se regasesc tehnici bazate pe ISO/IEC 9789-2 (aceasta tehnica este bazata pe o cheie secreta si este necesar un server on-line), ce preuspune autentificarea a doua entitati printr-un singur mesaj folosind marci de timp sau prin doua mesaje daca se folosesc numere aleatoare sau numere de secventa.

Remarca "(integritatea datelor) Atunci cand in protocoalele de autentificare se foloseste criptare, de regula trebuie garantata si integritatea datelor pentru a asigura securitatea. De exemplu, la mesajele care au lungime mai mare de un bloc, rearanjarea blocurilor cifrate nu poate fi detectata in modul de operarea ECB, chiar si criptatea in modul de operarea CBC ofera doar o solutie partiala. "[10]

Mecanisme 9789-2: avand in vedere notatia, r_A si t_A semnifica un numar aleator, mai exact o marca de timp care este generata de *A*. In aceste tip de mecanisme, t_A se poate fi inlocui de un numar de secventa n_A care prezinta garantii oarecum diferite. E_K este un algoritm de criptare simetrica avand cheia *K* partajata de *A* si *B*. Se folosesc cheile K_{AB} si K_{BA} pentru comunicatie unidirectionala. Se presupune ca ambele entitati cunosc identitatea pretinsa a celelilalte, fie din acest din contex fie datorita folosirii unor campuri de date aditionale transmise in clar. Asterisc (*) este folosit pentru a marca mesajele aditionale, iar virgla semnifica concatenare.

1. autentificare unilaterala este bazata pe marci de timp:

   $A \rightarrow B: E_K(t_A, B^*)$ mesaj (1);

Dupa ce a primit si a decriptat mesajului (1), *B* va face o verificare asupra marcii de timp daca este valida si daca identificatorul primit corespunde cu identitatea sa. Identificatorul *B* impiedica un adversar mesajul ulterior lui *A*, atunci cand se foloseste o cheie bidirectionala *K*.

2. autentificare unilaterala cu ajutorul numere aleatoare: pentru a nu folosi marci de timp, exista posibilitatea folosirii numerelor aleatoare, cu costul unui mesaj suplimentar:

[10] "Protocoale de securitate pentru sisteme distribuite"

$A \leftarrow B : r_B$ (1)

$A \rightarrow B : E_K(r_B, B^*)$ (2)

B realizeaza descriptarea mesajului ce a fost primit (2) si face verificarea daca numarul aleator este egal cu cel transmis lui *A* in cadrul mesajului (1), iar in mod optional *B* poate verifica daca identificatorul din mesajul (2) este al sau. Pentru a realiza prevenirea atacurilor asupra schemei de criptare E_K prin text ales, *A* are posibilitatea sa includa in raspunsul sau un numar aleator suplimentar sau forma provocarii ar putea fi restrictionata; in aceasta situatie este important este ca numerele ce se folosesc drept provocari sa nu se repete.

3. autentificare mutuala cu ajutorul numere aleatore:

$A \leftarrow B : r_B$ (1)

$A \rightarrow B : E_K(r_A, r_B, B^*)$ (2)

$A \leftarrow B : E_K(r_A, r_B)$ (3)

Cand se receptioneaza mesajul (2) *B* realizeaza aceleasi teste ca si in cazul autentificarii unilaterale, iar provocarea r_A va fi inclusa in mesajul (3). Cand se realizeaza descriptarea mesajului (3), *A* va face o verificare daca ambele numere aleatoare corespund cu cele din mesajele precedente. Numarul aleator ales de *A*, joca doua roluri: reprezinta o provocare pentru *B* si deasemenea este un mijloc de prevenire ale atacurilor prin text ales.

Pentru a se putea obtine autentificare mutuala este necesara rularea de doua ori a protocoalelor descrise mai sus, procedura considerata incompleta din moment ce nu exista legaturi intre cele doua autentificari si nu pot fi asociate logic cu o singura rulare a protocolului.

Provocare-raspuns ce se bazeaza pe functii one-way cu cheie.

In mecanismele prezentate anterior algoritmul de criptare poate fi inlocuit cu o functie one-way sau o functie neinversabila, o astfel de solutie se foloseste cand nu putem folosi algoritmi de criptare. Se vor prezenta cateva modificari aduse mecanismului 9789-2 prezentat mai sus:

- functia E_K de criptare se va inlocui de o functie MAC h_K
- receptorul unui mesaj va calcula valorile MAC avand la intrare valori cunoascute iar daca valoarea calculata corespunde cu valoarea primita , va accepta autentificarea.

- valoarea t_A va trebui transmisa in clar catre mesajul (1) al mecanismului cu un singur mesaj pentru ca receptorul sa poate realiza calcularea independent a valorilor MAC. r_A trebuie sa fie transmis in clar in mesajul (2) al mecanismului cu trei mesaje.

Acest mecanism care tocami a fost revizuit compus din trei mesaje ce se vor baza pe functia MAC h_k, va oferi o autentificare mutuala. Protocolul SKID3 are mesajele :

$A \leftarrow B : r_B$ (1)

$A \rightarrow B : r_A, r_k\,(r_A, r_B, B)$ (2)

$A \leftarrow B : h_k(r_B, r_A, A)$ (3)

Campul *A* este introdus in mesajul (3). Protocolul SKID2 ce se obtine prin eliminarea mesajului (3) ofera autentificare unlilaterala.

Provocare-raspuns prin tehnici asimetrice

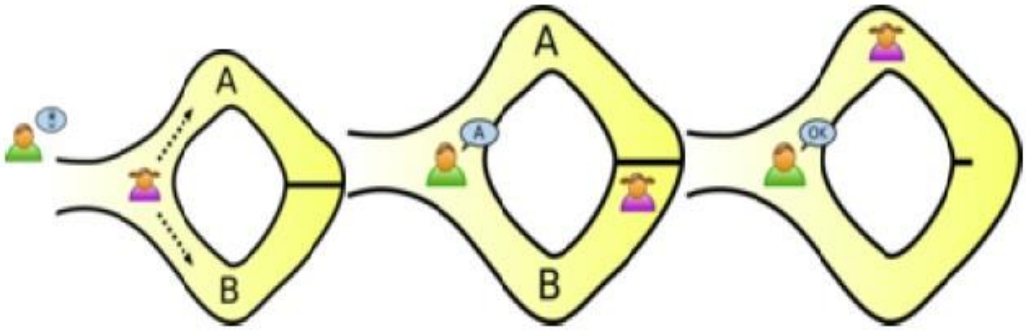
Aceasta tehnica este bazata pe chei publice, unde reclamantul arata ca stie cheia privata:

- reclamantul realizeaza o decripteare a unei provocari criptate cu ajutorul cheii sale publice;

- reclamantul va semna o provocare digital.

Sistemul cu cheie publica este vulnerabil la atacurile prin text cifrat ales, iar adversarul poate extrage informatii prin impersonarea unui verificator, alegand in mod strategic aleator un numar folosit drept provocare.

1.2.4 Protocoale de identificare particularizate, cu informatie zero[11]



Protocoalele cu informatie zero IZ

Aceste protocoale nu se degradeaza prin utilizare, nu utilizeaza algoritmi de cripatare, au nevoie de o latime de banda mare, au o demonstratie probabilistica, se poate demonstra cunoasterea unui secret fara ca el sa fie dezvaluit. Securitatea se baseaza doar pe presupuneri (fara ca acestea sa fie demonstrate).

Protocolul Fiat-Shamir de identificare

Are urmatorul obiectiv: *A* isi realizeaza identificarea in fata lui *B* demonstrand cunoasterea secretului *s* ce este asociat cu *A* prin date publice autentice, fara sa dezvaluie nici o informatie cu privire la *s*.

Securitatea are la baza ei dificultatea de extragere a radacinilor patrate modulo a numerelor intregi compuse mari, pentru care nu se cunosc factorii.

“Initializare: Un TTP alege $n = pq$ - intreg de tip RSA (p, q secrete); *A* alege s secret $(s, n) = 1$, calculeaza $v = s^2 \bmod n$ cheia publica

Mesajele protocolului: (t runde)

[11] “http://webhost.uoradea.ro/horos/files/PSSD-2%20slide.pdf”

$A \rightarrow B$: $x = r^2 \bmod n$, r – angajament,
x - martor
$A \leftarrow B$: $e \in \{0,1\}$,provocare
$A \rightarrow B$: $y = r \cdot s^e \bmod n$, raspuns

B accepta identitatea lui A daca $y^2 \equiv x \cdot v^e \bmod n$ "[12]

Structura generala a protocoalelor IZ este:

$A \rightarrow B$ martor
$A \leftarrow B$ provocare
$A \rightarrow B$ raspuns

Martorul are rol in definirea unei multimi de intrebari pentru care *A* sustine ca stie sa raspunda, asadar raspunsurile sunt fixate apriori.
Martorul este calculate pe baza unui angajament secret.
Deoarece cunoaste secretul, doar *A* poate raspunde la toate provocarile.

Exemple de protocoale de identificare IZ:

- Protocolul de identificare Schnorr- in cazul acestul protocol, securitatea de bazeaza pe logaritmul discret.

- Protocolul de identificare Feige-Fiat-Shamir- reprezinta o generalizare a protocolului Fiat-Shamir. Acest protocol poate fi implemantat in mecanisme ce au putere mica de calcul deoarece acest protocol necesita putine calcule, iar securitatea se bazeaza pe extragerea radacina patrata modulo $n=pq$

- Protocolul de identificare Guillou-Quisquater- extensie a protocolului Fiat-Shamir, acest protocol are un numar redus de mesaje si o memorie necesara redusa, iar securitatea in acest caz se bazeza pe extragerea radacina de ordin v modulo $n = pq$.

[12]" http://webhost.uoradea.ro/horos/files/PSSD-2%20slide.pdf"

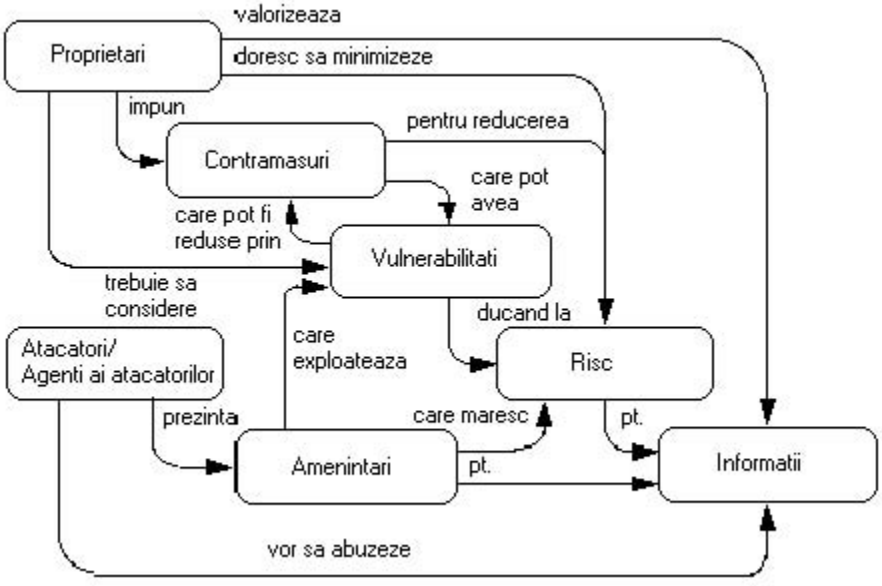
2.1 Atacuri asupra protocoalelor de identificare

Un atac se poate traduce ca fiind o disimulare a unei entitati (entitatea se da drep o alta entitate),iar astfel de atac mai este cunoscut sub numele de impersonare.

Amenintarile posibile la adresa protocoalelor de securitate contin impersonarea prin unul sau mai multe atacuri.Urmatorul tabel ne arata o clasificarea atacurilor si solutiile pentru prevenirea acestora.

Tipuri de atac	Solutii pentru prevenirea atacului
Reluare	Se vor folosi atat tehnici provocare-raspuns, cat si includerea identitatii tinta în raspuns
Intercalare	Se vor lega toate mesajele dintr-o executie a unui protocol
Reflexie	Se va include identificatorul entitatii tinta în raspunsurile la provocari; Se vor evitarea simetriilor între mesaje Se vor utiliza cheile unidirectionale
Text ales	Se va include în fiecare raspuns la o provocare un numar aleator
Intârziere fortata	Se vor combina numerele aleatoare cu timpii scurti de raspuns; Se vor folosi marcile de timp

Urmatoarea figura ne detaliaza comceptele privind securitatea informatiilor:[13]



- [13] http://www.securitatea-informatica.ro/securitatea-informatica/riscurile-de-atac-asupra-securitatii-sistemelor-informationale/

O remarca importanta este aceea ca trebuie mentinuta autenticitatea.

Protocoalele de identificare au rolul de a oferi identitate unei entitati doar la un anumit moment in timp,caci pentru a se mentine continua aceasta garantie este nevoie de tehnici suplimentare.Scopul acestor tehnici suplimentare este de a preveni atacurile din partea unor adversari activi.Pentru a preveni ca un atac sa se intample se pot lua urmatoarele masuri:

- Se vor efectua re-autentificari periodice sau cel putin la fiecare utilizare a unei resurse.
- Se va lega procesul de identificare la un serviciu de integritate permanent.

Nivelul de securitate necesar pentru protocoalele de identificare depinde exclusiv de mediul in care se ruleaza aplicatia. Securitatea pentru atacuri se face atat la nivel on-line cat si off-line, astfel ca pot aparea urmatoarele atacuri:

- Locale:
- De la distanta
- Off-line sau non-interactive

Atacurile locale sunt acele atacuri care are scopul de a selecta parametrii de securitate care limiteaza probababilitatea unei impersonari reusire printr-un atac de ghicire (in care entitatea isi da seama doar de secretul entitatii legitime) la $1/2^{20}$.

Insa,aceasta probabilitate este buna doar daca, pentru fiecare tentativa de impersonare este necesar prezenta locala si daca la fiecare esuare de identificare va exista o penalizare,care de obicei este o penalizare de timp.

Atacurile de la distanta: in cazul sistemelor in care posibilitatea de a se face un numar nelimitat de tentative de identificare este mare,atunci avem nevoie de un nivel mai mare de securitate.

Atac prin reluare se realizeaza atunci cand se foloseste informatie dintr-o executie precedenta a protocolului.

Atac prin intercalare reprezinta folosirea informatiei combinate selectiv din una sau mai multe executii ale protocolului.

Atac prin reflexive reprezinta intercalarea in care se trimite informatia dintr-o instanta a protocolului inapoi la sursa.

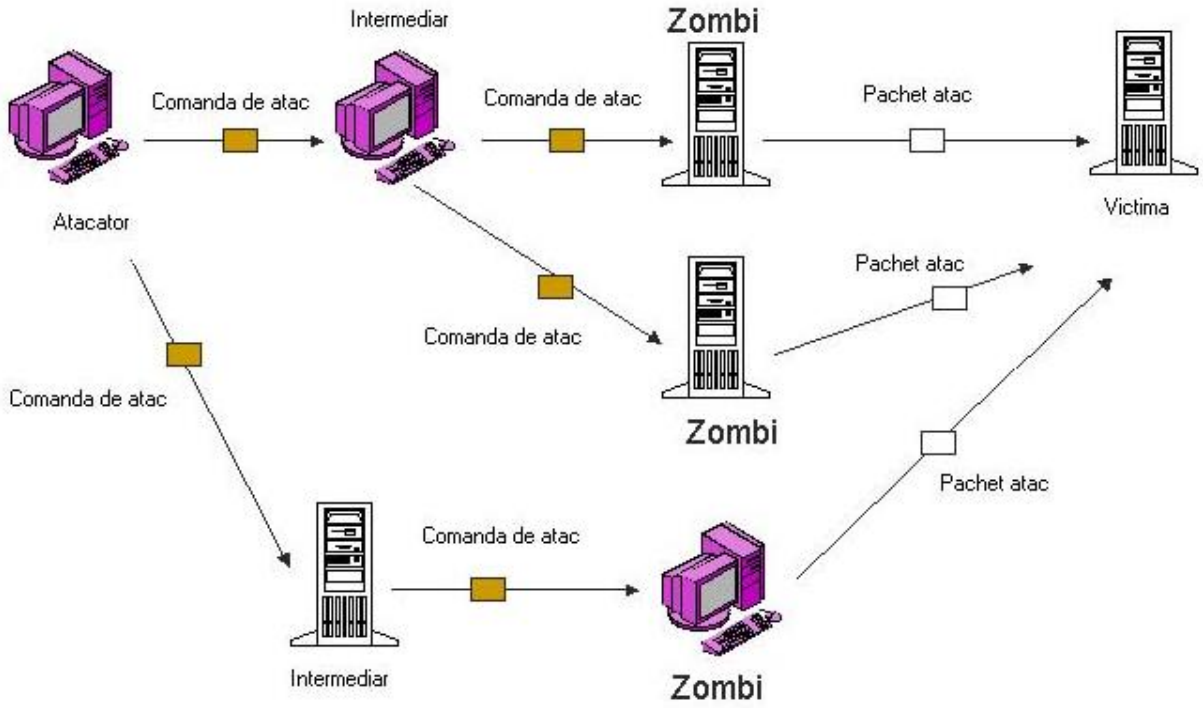
Intarziere fortata este atunci cand un adversar intercepteaza un mesaj (ce contine un numar de secventa) si il transmite mai tarziu.

Atac prin text ales: se aleg in mod strategic provocarile pentru a extrage informatii in legatura cu cheia pe termen lung a demonstratorului. Demonstratorul este folosit ca un oracol, pentru obtinerea informatiei ce nu poate fi calculata cunoscand doar cheia publica.

Atacuri off-line sau non-interactive: vom alege paramentrii cu conditiile:

- atacul sa necesite 2^{40} operatii in timp real
- limita mai mare ,2^{80},va fi impusa doar cand calculele se pot efectua off-line si atacul este verificabil.

Figura de mai jos ne exemplifica un atac Denial – of – Service distribuit[14]



In astfel de atacuri sunt folosite foarte multe calculatoare compromise care au rol de a automiza transmiterea de date care vor acapara sistemele ce vor sa fie atacate,deci doar cele vizate.Astfel ca,aceste calculatoare compromise sunt controlate de la distanta de niste entitati numite “zombi” .Aceste atacuri sunt extreme de greu de stapanit si de oprit.

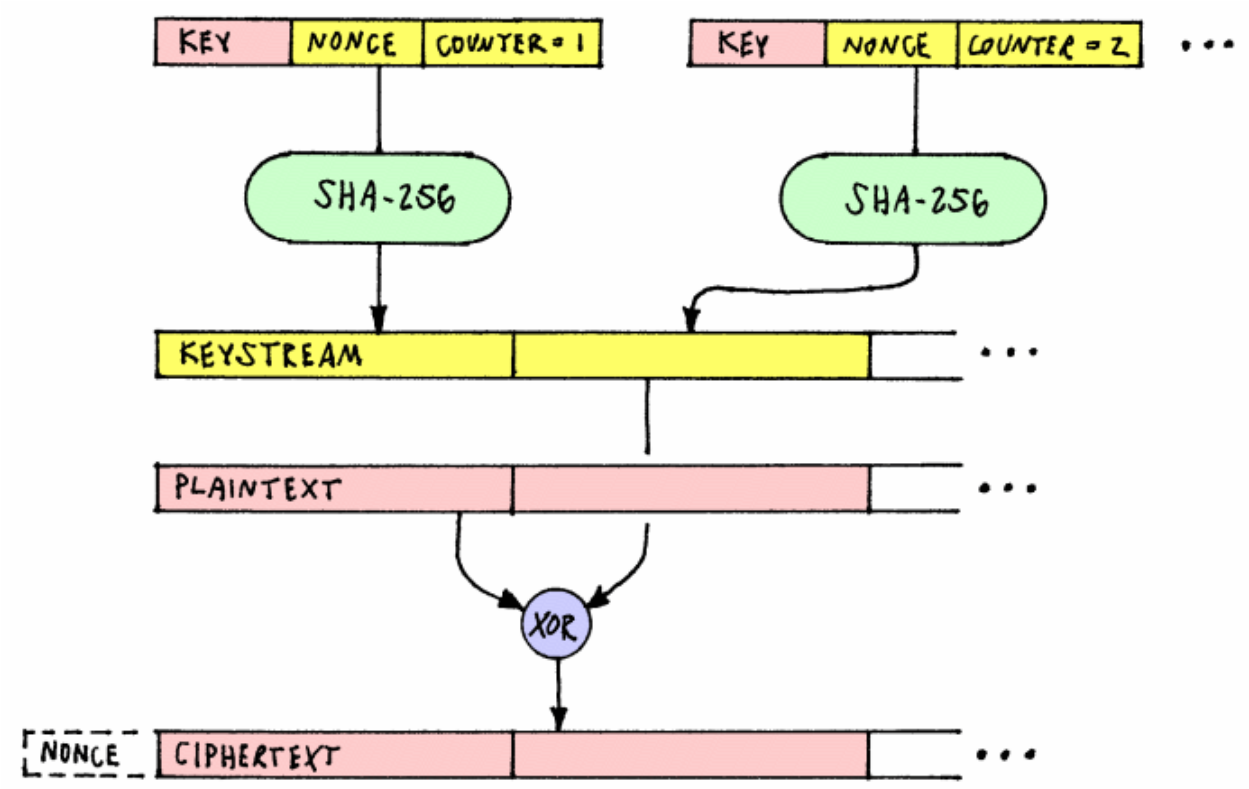
[14] http://www.securitatea-informatica.ro/securitatea-informatica/riscurile-de-atac-asupra-securitatii-sistemelor-informationale/

2.2 Analiza protocoalelor pentru stabilirea de chei

Stabilirea de chei reprezinta procesul prin care se da unei multimi de participanti o capabilitate partajata (nu doar un simplu secret partajat).

Rolul capabilitatii oferite este de a permite anumite operatii ce nu pot fi executate de alti participant,ceea ce ne aduce in prim plan notiunea de securitate fizica(persoanele fara autorizatie nu pot accesa sistemul),caci aceasta transformare a datelor cu scopul de a împiedica accesul persoanelor neautorizate se numeşte criptare.

Deci,un procesul care are rolul de a codifica se mai poate numi si cifrare sau criptare (encryption), iar procesul invers codificarii se numeşte descifrare sau decriptare *(*decryption). Pentru generarea unei keystream care face criptarea si decriptarea se poate folosi o functie Hash.Aceasta posibilitate e descrisa cu ajutorul schemei urmatoare:



15

[15] http://www.cs.rit.edu/~ark/lectures/onewayhash/HashFig5.png

Cheia pe care doua sau mai multe entitati o folosesc în comun se numeşte cheie de sesiune şi este generată aleator.

O cheie de sesiune are scopul de a fi efemera din urmatoarele motive:

- Restrictionarea textului cifrat (creat cu o singura cheie) disponibil atacurilor criptanalitice
- Reducerea expunerii unei chei compromise,atat din punct de vedere al timpului cat si al cantitatatii de date
- Crearea cheilor doar atunci cand sunt necesare, cu scopul evitarii stocarii pe un interval prea mare de timp a unui numar mare de chei distincte
- Crearea unei independente între sesiuni de comunicatie sau aplicatii

2.2.1 Ipoteze si adversarii in protocoale in stabilirea cheilor

In vederea stabilirii de chei este necesar un model informal cu rolul de a clarifica amenintarile la adresa acestora si cu scopul de a motiva caracteristicile protocoalelor.Se vor prezenta doar protocoale care implica doua entitati.

In protocoalele pentru stabilirea de chei avem entitati legitime cu nume unic,dar si terti neautorizati.

Entitate legitima este entitatea care comunica in vederea obtinerii unui obiectiv

Tertii neautorizati se refera in special la intrusi,adversari,inamici ,respective atacatori.

Mecanismele criptografice pe care se bazeaza protocoalele de securitate sunt create in ideea de a fi sigure.Insa,sunt cazuri in care siguranta lor este foarte scazuta si protocolul este nesigur,deci protocolul se va gasi in imposibilitatea de a oferi siguranta.

Adversarul protocolului este cel care incearca sa schimbe obiectivele protocolului atacandu-l.

Atacurile adversarului sunt de doua feluri:

- Pasive,atunci cand se incearca sa se distruga o tehnica criptografica prin simpla inregistrarea a unor date si prin analizarea lor ulterioara
- Active, atunci cand se modifica sau se injecteaza mesaje.

Mesajele protocolului sunt transmise prin retele deschise, neprotejate,deci se poate modifica,insera si refolosi mesaje vechi,curente sau chiar se pot injecta mesaje noi,astfel ca entitatile legitime sunt modelate ca primind exclusiv de la adversari,adversari care nu au intarzieri sesizabile.Astfel ca un adversar poate sa determine anumite entitati legitime,care sunt in imposibilitatea de a-si da seama de inselatorie,sa initieze noi instante de protocol.

Strategiile folosite de adversari:

- Deducerea cheii de sesiune folosind informatia obtinuta prin ascultarea liniilor de comunicatie
- Participarea ascunsa la un protocol si influentarea acestuia
- Initierea uneia sau a mai multor executii de protocol

In protocoalele cu scop de schimbare de chei neautentificat,impersonarea se face prin definitie,iar in autentificarea de entitati,pentru ca nu exista nicio cheie de sesiune care sa fie atacata,obiectivul adersarului este sa convinga entitatea,printr-o serie de mesaje,ca protocolul a fost rulat cu success cu o entitate legitima diferita de cea detinuta de adversary.

Din punctual de vedere al tipului de informative care este disponibila, adversarii pot fi de tipul:

- Outsider= adversar care cunoaste doar datele disponibile in mod public,sau cel mult informatiile care pot fi interceptate prin liniile de comunicatie
- Insider = adversar care are acces la informatii suplimentare, si este de doua feluri:
    - Insider one-time=este cel care obtine informatii suplimentare la un moment de timp in scopul de a le folosi ulterior
    - Insider permanent = este cel care are un acces nelimitat la informatia privelegiata.

In analiza protocoalelor pentru stabilirea de chei trebuie luat in considerarea impactul pe care il poate avea daca anumite tipuri de chei sunt compromise,chiar daca un astfel de compromise este posibil sa nu aiba loc.Deci trebuie considerat:

- Compromisul cheilor secrete pe termen lung
- Compromisul unor chei de sesiune vechi

Securitatea perfecta se intampla atunci cand compromiterea unei chei secrete folosita pe termen lung nu duce la efectarea cheilor de sesiune mai vechi.

Atunci cand se compromite cheile de sesiune vechi sau un adversar poate realiza impersonari ,putem spune ca protocolul este vulnerabil la o cheie cunoscuta.

2.2.2 Obiectivele analizei si metode de analiza

Scop: stabilirea unei legaturi de incredere cu securitatea criptografica oferita de protocol.

Prin analiza protocoalelor se confirma daca un protocol respecta toate obiectivele date spre indeplinire.

Beneficiile analizei:

- Se identifica ipotezele pe care se bazeaza securitatea protocolului
- Se identifica proprietatiel protocolului
- Se examineaza eficienta protocolului

Un protocol pentru stabilire de chei

- se numeste operational daca,atunci cand lipsesc atat adversarii activi ,cat si erorile de comunicatie,participantii onesti care respecta specificatiile protocolului vor putea incheia protocolul cu succes,doar calculand o cheie comuna,obtinand astfel identitatea entitatii cu care se partajeaza cheia respective.
- Se numeste rezilient atunci cand un adversar se afla in imposibilitatea de a insela participantii onesti in legatura cu rezultatul.

Elemente necesare metodelor de analiza:

- Protocolul specificat- mesajele protocolului la trimitere si la primire trebuie specificate clar si concis,in niciun caz ambiguu
- Obiectivele-trebuie specificate clar garantiile care sunt oferite la incheierea protocolului
- Ipoteze si starea initiala – trebuie enuntate atat ipotezele cat si conditiile initiale
- Demonstratia- pasii protocolului care duc la indeplinirea obiectivelor protocolului.

Abordarile folosite in analiza protocoalelor criptografice sunt:

- Analiza ad-hoc si analiza practica =metoda ne spune ca un atac reusit impotriva protocolului are nevoie de resurse mult mai mari decat a adversarului
- Reducerea de la probleme dificile=metoda demonstreaza ca un atac reusit duce la rezolvarea unei probleme dificile din teoria numerelor.
- Analiza din punctul de vedere al complexitatii calculului = metoda ne defineste un model corespunzator de calcul,unde adversarii sunt medelati si au putere de calcul polinomial.
- Analiza din punctul de vedere al teoriei informatiei= aceasta metoda ne demonstreaza ca protocoalele sunt sigure neconditionat si ca adversarii sunt modelati ca avand resurse de calcul nelimitate.
- Metode formale=metode care au rolul de a verificare si evaluare pentru a se ajunge la convingerea ca au fost atinse obiectivele protocolului.

Avantajele analizei ar putea fi urmatoarele:

- identificarea detaliata a ipotezelor pe care se bazeaza securitatea protocolului
- identificarea proprietatilor protocolului ,enuntarea prompta a obiectivelor sale ,dar si determinarea aplicabilitatii protocolului în diverse situatii
- verificarea eficientei protocolului,atat pentru latimea de banda folosita, cat si a complexitatii computationale

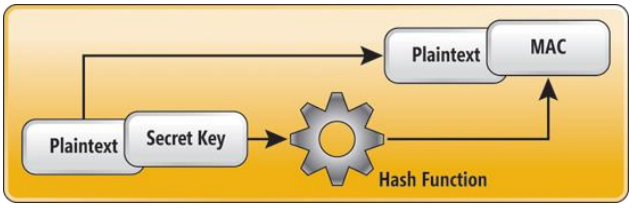
2.3 Functii hash si integritatea datelor
2.3.1 Definitia si clasificarea functiilor hash

Rol:maparea secventelor binare de lungime aleatoare pe secvente binare de lungime n.

Functia h:D → R nu este o functie injective,deci exista coliziuni,adica perech de valori care detin aceeasi valoare hash.

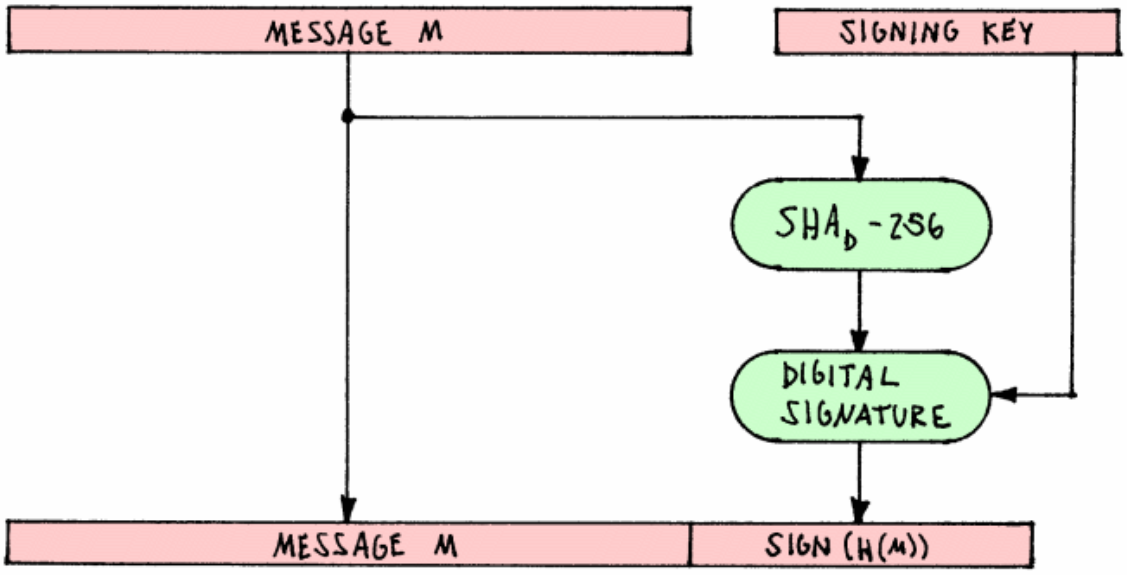
Valorile hash sunt cele care reprezinta exprimarea compacta a unei secvente binare de o lungimi foarte mari ,iar functia hash,poate fi in acest caz folosita ca identificator pentru acea secventa binara.

Urmatoarea figura exemplifica functia Hash:[16]



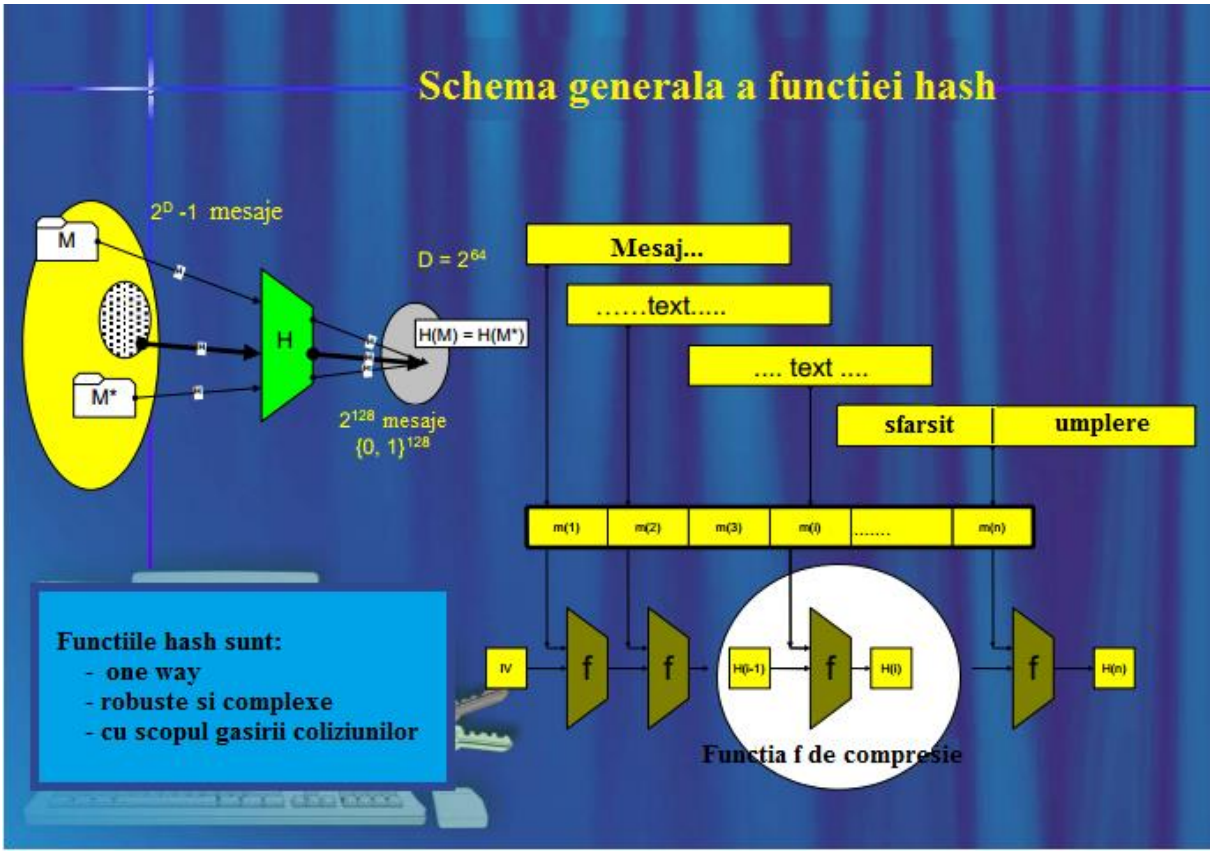
Tot functiile hash sunt acelea care asigura integritatea datelor.

O schema de semnatura digitala nu se aplica asupra mesajului,ci asupra unei valori hash a acestuia din considerente de eficienta si pentru ca nu este la fel de sigur, și, de obicei, mai repede, pentru a calcula o semnătură digitală de hash a unui mesaj în loc de mesajul în sine.Aceasta idee este reprezentata in imaginea de mai jos[17]:



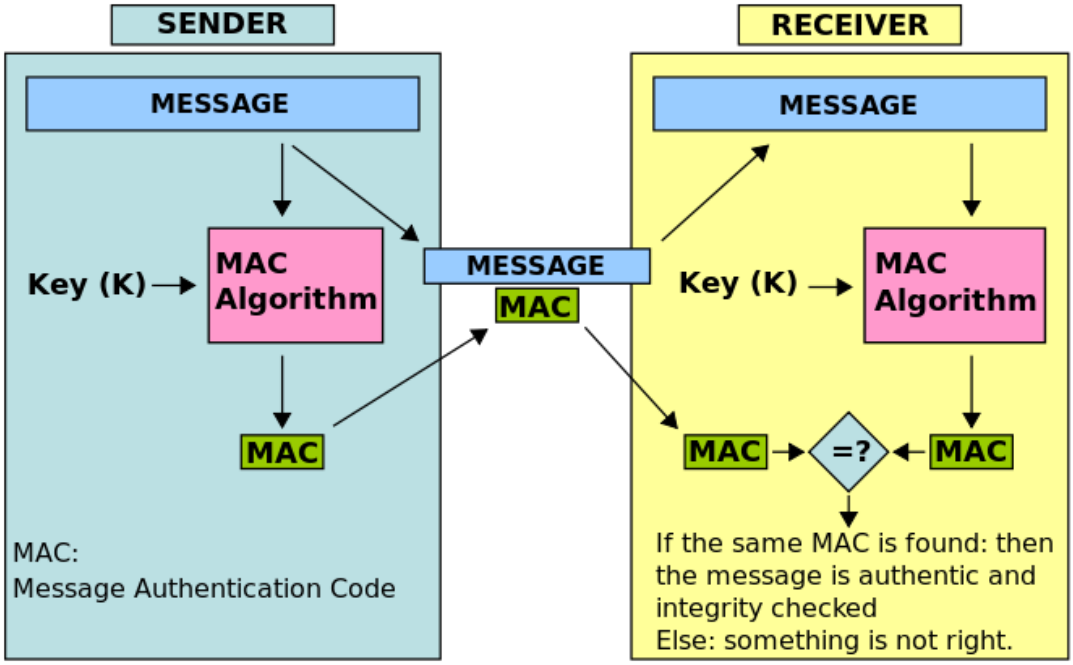
[16] http://i.msdn.microsoft.com/ff797918.Sullivan_Figure3_hires(en-us,MSDN.10).png
[17] http://www.cs.rit.edu/~ark/lectures/onewayhash/HashFig4.png

Schema generala a functie hash este prezentata in figura urmatoare:[18]



O clasa speciala de functii hash se numeste MAC(Message Authentication Code = coduri pentru autentificarea mesajelor).Scopul lor este de a permite autentificarea mesajelor prin niste tehnici simetrice.

[18] Poza originala este de pe site-ul http://andrei.clubcisco.ro/cursuri/f/f-sym/5master/aac-atcns/Hash%20functions.pdf

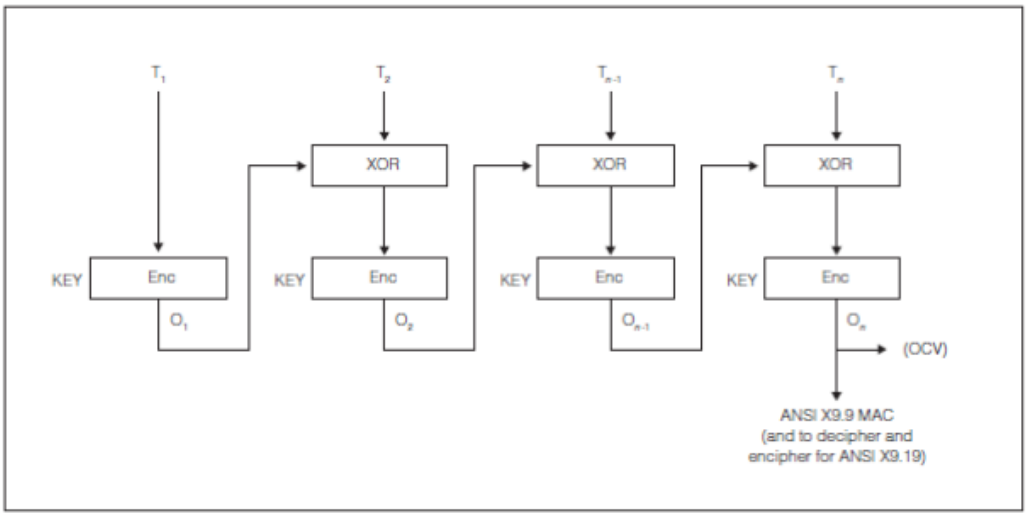


19

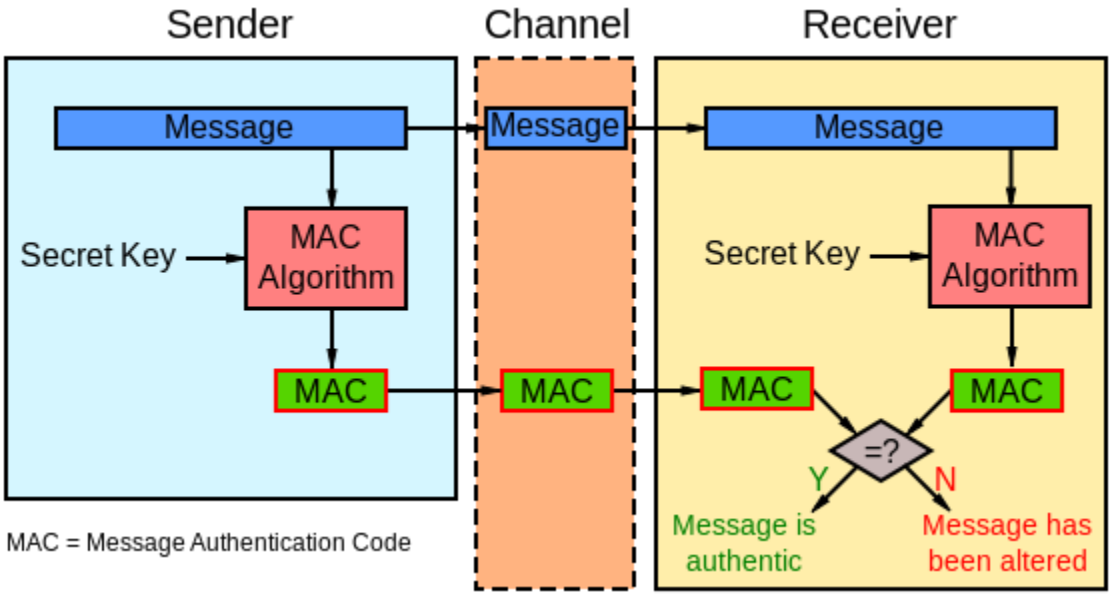
Instituția financiară (en-gros) Mesaj de autentificare Standard (ANSI X9.9-1986) definește un proces de autentificare a mesajelor de la inițiator la destinatar, iar acest proces este numit de autentificare Codul Mesaj (MAC) metoda de calcul.

Figura de mai jos[20] ne arata metoda de calculare MAC pentru date binare.In acesta schema ,KEY este de o cheie de 64 biti,iar T_1,$T_2 \ldots T_n$ sunt blocuri de date pe 64 de biti de text.In cazul in care T_n este are o lungime mai mica de 64 biti,zerourile binare sunt anexate in dreapta lui T_n.Blocurile de date $T_1 \ldots T_n$ sunt criptate cu toate iesirile eliminate , cu exceptia ultimului bloc de iesire O_n.

[19] http://upload.wikimedia.org/wikipedia/commons/thumb/0/08/MAC.svg/661px-MAC.svg.png
[20] http://pic.dhe.ibm.com/infocenter/lnxinfo/v3r0m0/topic/com.ibm.linux.z.wskc.doc/l0wskc04.pdf



Functiile MAC sunt functii care primesc la intrare un mesaj si o cheie secreta si produc la iesire o secventa fixa,concept explicate in figura de mai jos:[21]



Aceste functii sunt folosite pentru:

- Asigurarea integritatii datelor
- Autentificarea originii datelor prin tehnici simetrice
- Autentificare prin scheme cu chei simetrice

[21] http://upload.wikimedia.org/wikipedia/commons/thumb/b/b8/Message_Authentication_Code.svg/600px-Message_Authentication_Code.svg.png

O utilizare tipica a functiilor hash fara cheie este in asigurarea integritaii datelor in felul urmator.se calculeaza valoarea hash pentru un mesaj x la momentul T_1. Integritatea acestei valori hash,dar nu si al mesajului e asigurata intr-un anumit mod la transfer.

Ulterior , la momentul T_2 pentru a verifica daca un mesaj x' este identic cu x se calculeaza valoarea hash pentru x' si daca aceasta valoare hash este egala cu cea calculate initial,atunci se poate considera ca mesajul nu s-a modificat.

In felul acesta ,problema pastrarii integritatii unui mesaj s-a redus la pastrarea integritatii valorii hash,care este un mesaj mult mai scurt de dimensiune fixate.

Datorita faptului ca functiile hash nu sunt functii injective ,aceasta asociere intre mesaje si valorile hash ale lor se face doar in sens computational.

In practica,valorile hash ar trebui sa fie identificabile in mod unic cu o singura valoare,iar gasirea unor coliziuni trebuie sa fie nefezabila computational.

Clasificarea functiilor hash:

- Fara cheie=functii care primesc la intrare doar un argument(mesaj);
- Cu cheie = functii care primesc la intrare un mesaj si o cheie secreta

Proprietatile functiei hash

- Compresie= h are scopul de a asocial secvente binare x de o lungime aleatoare,unor secvente h(x) ,care au o lungime fixa n.
- Faciliteaza calculul=daca se cunoaste functia h si o valoare x,h(x) este simplu de calculat.

Acestea sunt cateva proprietati pe care le au functiile hash.Pentru a cunoaste si mai multe proprietati ale acestor functii vom face urmatoarea clasificare:

1. MDC=Modification Detection Codes = coduri pentru detectarea modificarilor,o reprezentarea a functiilor hash fara cheie

    - Sunt greu inversabile: rare situatiile in care vom gasi o valoare al carui hash este egal cu unul deja prestabilit
    - Rezistente la coliziuni : rare situatiile in care doua valori au acelasi hash.

2. MAC = coduri pentru autentificarea mesajelor face parte din clasa functiilor hash cu cheie

    - Asigura autenticitatea mesajului
    - Asigura integritatea mesajului

Toate entitatile implicate in process cunosc functiile hash,deci in cazul MDC-urilor functiile hash sunt cunoscute si deci calcularea lor nu implica greutate,iar in cazul MAC-urilor , valoarea hash se poate calcula de toti cei care cunosc cheie.

Dar ,proprietatile aceastea generalizate pentru cazurile MDC si MAC nu sunt singurele,astfel ca,alte proprietati sunt:

- Rezistenta la preimagine=pentru orice valoare hasn nu se poate gasi preimaginea acesteia.O exceptie de la aceasta proprietate face parte o lista de valori care pot fi precalculate.
- Rezistenta la a doua preimagine,daca se da x si h(x) atunci nu se poate gasi o a doua valoare x' astfel incat $h(x)=h(x')$
- Rezistenta la coliziune=nu exista o cale de a se gasi doua valori distincte x, x' care sa detina o aceeasi valoare hash, $h(x)=h(x')$

2.3.2 Atacuri asupra functiilor hash

Un adversar care vrea sa atace un MDC are urmatoarele obiective de indeplinit:

- Daca se da cunoscuta o valoare hash notate cu y,atunci trebuie gasita o a doua preimagine x ,astfel incat y =h(x).
- Daca se da cunoscuta o pereche (x,h(x)), atunci trebuie gasita o a doua preimagine x' ,astfel incat $h(x')=h(x)$.
- Gasirea a doua valori x si x' astfel incat functiile lor hash sa fie egale,adica $h(x)=h(x')$.

Din punctul de vedere al puterii pe care o are adversarul atacurile asupra MAC-urilor le putem clasifica astfel:

- Atac cu text cunoscut – una sa mai multe perechi mesaj – MAC (x_i , $h_k(x_i)$) sunt disponibile
- Atac cu text ales- una sa mai multe perechi mesaj – MAC (x_i ,$h_k(x_i)$) sunt disponibile,insa valorile x_i sunt alese strict de adversar
- Atac cu text ales in mod adaptive- valorile x_i sunt alese strict de adversary,doar ca alegerea se face in functie de MAC-urile obtinute pentru valorile anterioare

In cazul in care se face falsificarea MAC ,producerea consecintelor practice difera in functie de gradul d control pe care adversarul il are asupra valorii x,vinovata de producerea falsificarii.

Exista doua tipuri de falsificari:

- Selectiva=adversarul produce perechi mesaj-MAC ,iar mesajele si le alege
- Existentiala=adversarul produce perechi mesaj – MAC,insa fara control asupra mesajului care calculeaza valoarea MAC

2.3.3 Constructia functiilor hash

Cele mai multe functii fara cheie (MDC) sunt proiectate ca fiind procese iterative care primesc la intrare mesaje de lungime variabila .Pentru aceasta variabila se calculeaza functia hash, procesand niste blocuri cu o dimensiune deja fixate.

Exemplu:

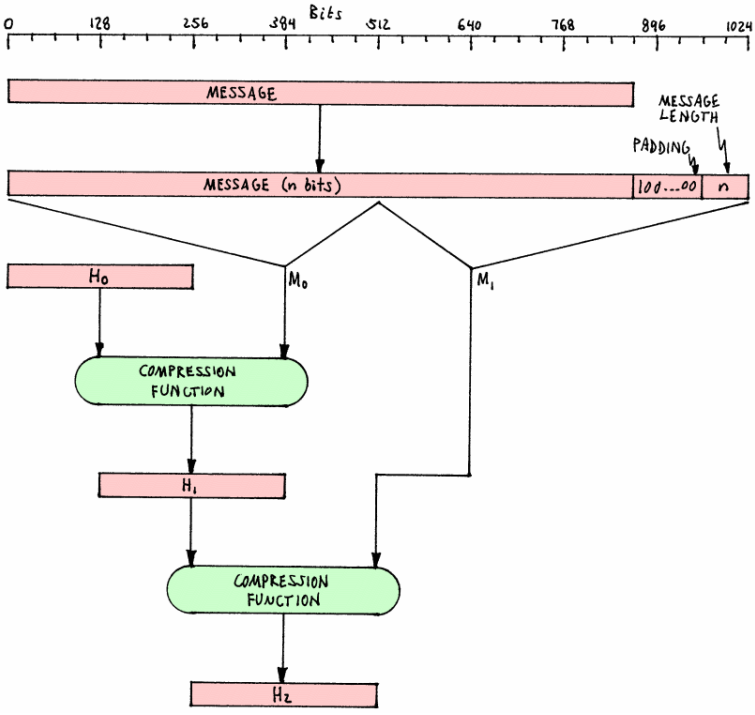
Fie:

- x=mesajul pentru care trebuie calculate functia hash,deci $x=x_1x_2\ldots x_t$
- x_i= blocurile de lungime egala r (biti) folosite ca valoare de intrare pentru o functie interna f
- f= functia de compresie a lui h.Rolul lui f este de a calcula un rezultat intermediar de lungime binara n si care depinde de rezultatele intermediare anterioare si de urmatorul bloc x_i.
- H_i = rezultatul intermediar dupa pasul i

Modelarea vectorului x se face astfel:

- H_0=IV; unde, IV=vector de initializare
- $H_i=f(H_{i-1},x_i)$, unde , 1<=i<=t
- $h(x)=g(H_t)$, unde, g=realizeaza ultima transformare asupra rezultatului final al procesului iterativ

Structura functiei Hash este reprezentata in figura de mai jos[22]:



[22] http://www.cs.rit.edu/~ark/lectures/onewayhash/SHA256Fig2.png

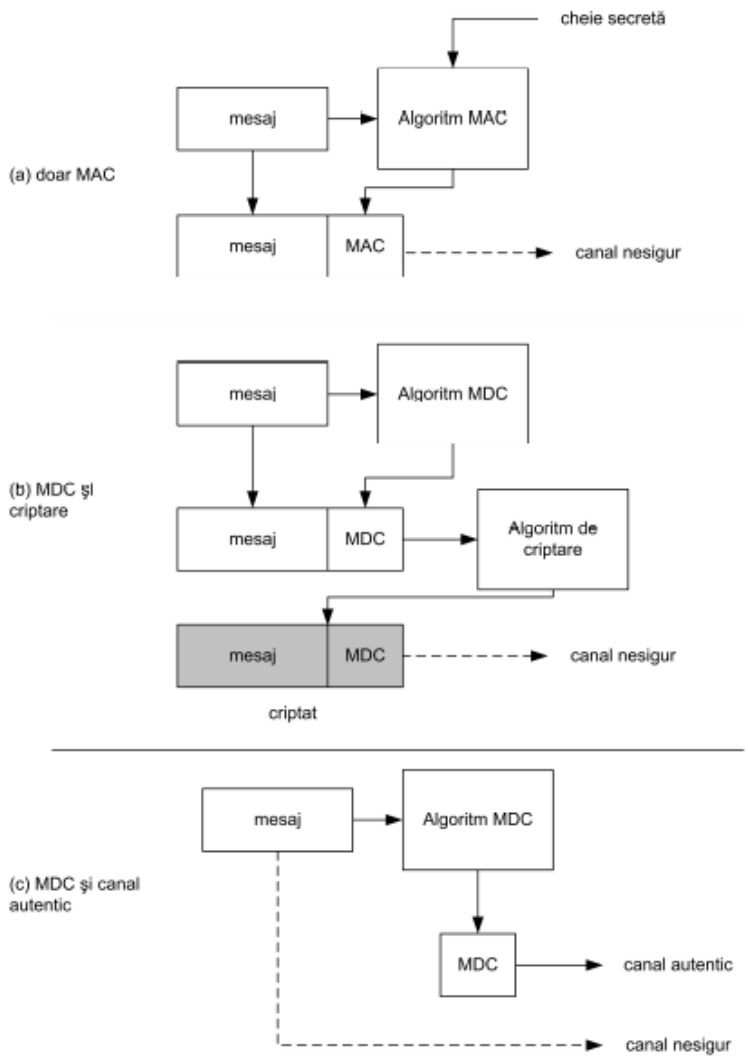
2.3.4 Metode pentru asigurarea integritatii datelor

Integritatea datelor se defineste ca fiind cea care asigura faptul ca datele nu au fost modificate neautorizat dupa crearea ,transmiterea si stocarea lor de sursa autorizata.

In vederea asigurarii datelor exista urmatoarele abordari:

a) doar MAC
b) MDC si criptare
c) MDC si canal autentic

Iar urmatoarea schema este cea care le exemplifica:[23]



[23] http://webhost.uoradea.ro/horos/files/PSSD-S.pdf

Criteriile ce permit recunoasterea integritatii datelor include de asemenea si:

- Redundanta datelor
- Formatul predefinit al dateor

Tehnicile criptografice se bazeaza pe:

- Informatia secreta
- Canal autentic

Asupra integritatii datelor pot fi doua tipuri de amenintari:

- Intentionate
- Neintentionate,care se mai numesc si erori de transmisie

Pentru a verifica integritatea datelor trebuie sa verificam:

- Integritatea originii datelor,adica sa ne asiguram ca datele le primim de la aceeasi sursa care trebuie sa trimita
- Integritatea datelor,adica trebuie sa ne asiguram ca datele nu au fost modificate pe parcurs

Aceste doua aspect lucreaza impreuna:daca nu se poate stabili sursa datelor , atunci o rezolvare a lor nu este posibila.

Operatiile care au rolul de a invalida integritatea datelor sunt urmatoarele:

- Inserarea bitilor
- Includerea unor date noi dun surse frauduloase
- Stergerea bitilor
- Reordonarea bitilor
- Completarea sau inlocuirea bitilor

Metodele care au rolul de a asigura autentificaarea originii datelor includ:

- Coduri MAC pentru autentificarea mesajelor care au rolul de a nu lasa sa se realizeze distinctii intre entitatile care cunosc o cheie,astfel ca nu ofera o acceptare a originii datelor
- Schemele de semnaturi digitale
- Adaugarea unei valori de autentificare secrete la text inainte de criptare

Deci, ca sa ne asiguram de unicitatea si de asigurarea momentului crearii datelor vom apela la urmatorii parametrii care au scopul de a autentifica mesajele si entitatile,si acestia sunt:

- Numerele aleatoare in protocoale provocare-raspuns au rolul de a preveni reluarea mesajelor,daca orice valuare MAC reluata ar fi incorecta .
- Numerele de secventa(acestea permit detectarea reluarii mesajelor)
- Marcile de timp

Bibliografie :

1. “Protocoale de securitate pentru sisteme distribuite” - Horea Gavril Oros – Oradea 2009
2. https://ics.ubbcluj.ro/data/phdThesis/tezaHoreaOros.pdf
3. http://webhost.uoradea.ro/horos/files/PSSD-2%20slide.pdf
4. http://en.wikipedia.org/wiki/Lamport_signature
5. http://en.wikipedia.org/wiki/Needham%E2%80%93Schroeder_protocol
6. http://en.wikipedia.org/wiki/Feige%E2%80%93Fiat%E2%80%93Shamir_identification_scheme#Security
7. http://www.securitatea-informatica.ro/securitatea-informatica/riscurile-de-atac-asupra-securitatii-sistemelor-informationale/
8. http://andrei.clubcisco.ro/cursuri/2pc/auxiliare/5_Analiza%20performantelor.pdf
9. http://andrei.clubcisco.ro/cursuri/f/f-sym/5master/aac-atcns/Hash%20functions.pdf
10. Linux for System z: Secure Key Solution with the Common Cryptographic Architecture Application Programmer's Guide : http://pic.dhe.ibm.com/infocenter/lnxinfo/v3r0m0/topic/com.ibm.linux.z.wskc.doc/l0wskc04.pdf
11. http://fmi.unibuc.ro/ro/pdf/2013/doctorat/RezumatOlimid.pdf