

④

Integritatea datelor

~~H~~
hash

- se realizează (astăzi) prin fct hash → plecând de la un mesaj și se face un refrunt (digest), care în seara unei atacuri → modificări ușoare în mesajul original produc în mod probabil modificări ușoare în refrunțul hash
- pot apărea coliziuni doar spălând fct hash e ușoare.

$$\exists m_1 \neq m_2 \text{ cu } h(m_1) = h(m_2)$$

$$|H(K, m)| < |m|$$

Probleme fct hash:

- 1) - reîncercarea coliziunii - adu. poate consta în doar o colizie.
- 2) Reîncercarea de pe calea pre-image attack (impersonificare). - a căuta un semnificație diferită dorind să rețină același refrunț hash.
- 3) Rez. la preimage -

→ Tehnici de construcție pt fct hash - (iterative; reduc unui număr de biți, apoi se repetă operația)

HMAC = construcția de MAC cu funcții hash. -

MAC = message authentication code.

NMAC = nested MAC → se plasează de la H_K - fct hash cu cheie, un vector de inițializare OT și

$$OT \rightarrow H_K(m_1, OT) \xrightarrow{m_2} H_K(m_1, m_2, C) \rightarrow \dots \xrightarrow{C_1} H_{K_1}$$

$$NMAC = H_{K_1}(H_K(m))$$

HMAC = NMAC în care cheile K_1, K_2 sunt XOR cu ipad.

(o secvență de podare fixată; 4 octetii zeci, 4 octetii 1).

Dacă fct hash e sigură → ac metoda e sigură.

⇒ Criptografie cu chei publice (astăzi).

- rezolvă prob. distribuției cheii -

$$\begin{array}{ccc} P_{KU} \{m\}_{P_{KU}} & \{ \{m\}_{P_{KU}} \}_{P_{KU}} & = m \\ \leftarrow S_{KU} & & \end{array}$$

- Se generează o pereche de chei, uno publică, una privată

IND CPA - reîncercarea atac de plaintext aleș.

* Eu criptografia cu chei publice, atacul ~~IND~~ CPA nu poate fi evitat.

metoda IND CPA e sigură dacă păr. adres. de a ghica care din cele 2 mesaje e cel criptat e neglijabilă.

- același mesaj criptat de mai multe ori duc la refuz. diferențe

- Schemele deterministe nu sunt sigure la atac de plaintext aleș.
- IND-CCA = registrant în atac de criptotext aleș.
- IND-CPA = adresațul nu poate distinge între 2 mesaje clare având un singur criptotext, de la care din cele 2 mesaje clare provine acest criptotext.
- IND-CCA \Rightarrow IND-CPA dar nu și invers.

• Pt. părțea mare de mesaj se criptează cu cheia simetrică pt că e foarte rapidă; (criptografia cu chei publice e foarte lentă)

Dacă două scheme sunt IND-CPA sigure \Rightarrow compunerea lor tot iată sigură.

RSA nu e probabilist \rightarrow nu e IND-CPA sigur.

IND = indistinguishability

log discret: G = grup ciclic, α generător (rezident) al lui G ;

 $y = \text{element al lui } G, \log_{\alpha} y = a, \text{unic}, a \in \mathbb{Z}, \text{ și } y = \alpha^a$

Factorizarea: $n = \text{latura PNTU}$, cu $n = p_1^{e_1} p_2^{e_2} \cdots p_m^{e_m}$, $p_i \neq p_j$ prime factori

EL GAMAL \rightarrow sol. prob. logaritmului discret.

$P_K = (G, \alpha, g, *)$ - cheia secretă; $g^x \rightarrow x$? problema intractabilă.

$x = \log_g g^x$ soluția logaritmului discret.

\rightarrow EL GAMAL e IND-CPA sigur pt că PCD e intractabilă.

EL GAMAL și RSA nu sunt rezistenți la CCA. \rightarrow

Challengerul $G(1^n) \rightarrow (P_K, S_K)$, consultă protocolul P_K și trimită

c (mes. criptat) challengerul trimite mesajul $\rightarrow (m_0, m_1, y) \rightarrow$ te

\rightarrow trimită C își obține \rightarrow ca și trimită b' ; Adăugă că și

RSA - sigur la IND-CCA.

\rightarrow se utilizează construcție hibridă \rightarrow fct HASH \rightarrow se expandă

o cheie unică \rightarrow fct fără să fie că mai random

Semnătura digitală = metodă de autentificare a mesajelor.

Diffie Hellman: A trimită lui B $\alpha^x \bmod p$
B calculează $(\alpha^x)^y \bmod p$
B trimită lui A $\alpha^y \bmod p$
A calculează $(\alpha^y)^x \bmod p$

$\alpha^x \bmod p$ este de
probabil log discret

Aloc: C se interupe. Între A și B, intercepteză α^x , transmite
lui B α^x , B transmite lui C α^y pentru că poate calcula α^{xy}
dor C poate calcula α^{xy}

DNS SEC - extensie de securitate pt DNS

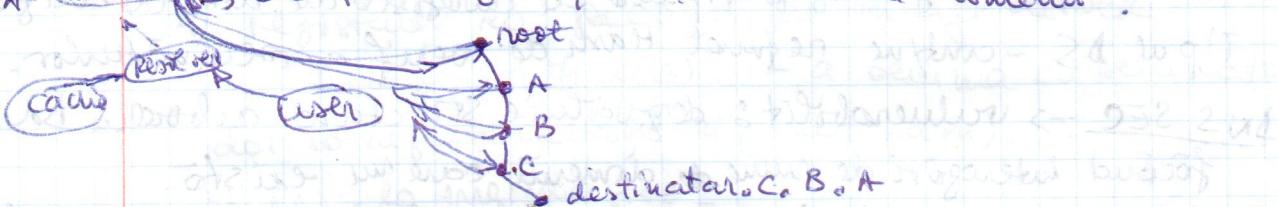
Problema: - lipsa autenticității

- nu se verifică integritatea -

DNSSEC adaugă autenticitate + integritate. (cu) digitale
lanturi
de încredere

NU se adaugă confidențialitate -

- Fiecare host (gadget) are asociat un IP, DNS-ul face mapare nume \rightarrow IP; (fisierul hosts din sistemul săz)
- arborele DNS - important în domeniu și subdomeniu -
 → zone de autoritate → servere de nume { primar secundare}
 - local network
 - cache
 - user
 → ele toc reprezintă nivelul de domeniu :



La DNS standard, răspunsul nu conține autentic și integritate.

Securitate la DNS :

- se combină semnătura digitală cu un lant de încredere
- la DNS-ul standard, transportul răspunsului pt utilizator se face prin UDP; cind se face o delegare de autoritate, de la o zonă la alta, se folosește TCP.

Cind se cere ID-ul pt destinatar, inf de semnată -
 → se căuta la nivel superior și se ajunge la Root → elem de autenticitate (autoritate).

→ fiecare nod are asociate informații → structură sub forma înregistrăriilor de resurse (Resource Record)

Struct generală : (RR) :

- 1) nume domeniu
- 2) TTL time to live al RR.
- 3) clasa de protocoale utilizate.
- 4) Tip înregistrare (IP-address, SOA)
- 5) R DATA - resource data.

DNS Sec vine cu 4 tipuri noi de RR

1) DNS Key - cheia pt semnată digitală.

2) RR SIG - semnatura digitală pt o multă de RR de același tip

3) NSEC → zona de delegare autoritate pt autentic răspuns.

4) DS → pt verif cheia publică - de autentic a semnată

algoritmul : pt semnarea digitală : RSA, MD5, SHA1/DSA;

RSA/SHA1; RSA/SHA2; RSA/SHA3; ECDSA/RSA2; ECDSA/SHA256

Public key = base 64 -

Cum se lucrașă în Base 64?

Se încolțește și se imparte în grupe de 6 biți de la stg la dreapta. Grup de 6 biți + unul din între 0 și 63 - \Rightarrow un codificat cu tabelă base 64 \rightarrow dacă ultimul byte nu are 6 biți se pun 4 de 0 și se padează cu 2 de 0 (egal).

6 biți

$$\boxed{26 \text{ } 00 \text{ } 00} = 8 \text{ biți finali sau } \boxed{16 \text{ biți } 46 \text{ } 00} = \begin{matrix} \text{un singur} \\ \text{6 biți} \\ \text{câtun = pt Hec @ pus.} \end{matrix}$$

RR SIG - conține semnătura digitală pe un RR de tip de același nume de domeniu care urmărește delegația de autoritate.

NSEC \rightarrow furnizează numele de domeniu și urmărește delegația de autoritate.

\rightarrow furnizează IPuri și deținătorii ale domenilor în cadrul

Tipul DS - conține reținut Hash de verificare a semnăturilor.

DNS SEC \rightarrow vulnerabilitatea dezvoltării structurii arborescente, făcând interogatori de nume de domeniu să existe.

(zone enumeration) \rightarrow se defuzează structura unei zone de autoritate.

NSEC3 \rightarrow numele de domeniu să introduces în varianta HASH - ordinea de afișare nu mai e crucială, pe domeniu, ci pe hash.

DNS SEC nu asigură confidențialitatea și protecție pt. denial of service -

- asigură doar autentificare + integritate

TLS - obiectivul principal standardizarea și imbunătățirea SSL.

SSL = secure socket layer.

TLS = transport layer security

NU se oferă neîmpărțire

ci doar: confidențialitate, autentificare, integritate.

2 layer SSL \rightarrow ① prelucrarea astelor \rightarrow confidențialitate și integritate.

② celelalte protocoale \rightarrow handshake \rightarrow autentificare -

- mediu oferă parametri criptografici -

SSL = este un protocol bogat pe stare = o stare a unei conexiuni furnizată parametrui conexiunii

- e compus din 2 layeruri: { SSL Record \rightarrow prelucrare date.

- asigură servicii sigure TCP { layer 2: 4 protocoale { SSL handshake
SSL change cipher spec
SSL alert
SSL application data.

Sistemul SSL mută boala pe stare

- sunt 4 stări: { current read (receive)
{ current write (send)
{ pending read (receive)
{ pending write (send)

Picături corespunzătoare sunt asociate cu o scrisoare. Mai multe scrisori pot exista simultan în trei același entități.

⑤

Panouetii SSL

Suport criptografic

Session State

- ID de sesiune -
- Certificat Peer -
- Metoda de comprimare -
- Cipher Spec → algoritmul de criptare (NULL, DES, --- etc)
- Master secret
- Flag (is resumable)

Panouetii de stare ai conexiunii

states
connection
(connection state)

- Nc, Ns - monice client, server.
- $K_{MAC}^C, K_{MAC}^S \rightarrow$ secret key folosit în op MAC -
- Kc, ks - cheie de scrisoare -
- Vector de initializare fol în CBC (code block chaining)
- Nr de octecte (ordinea mesajelor)

Master Secret - generat din { pre-master secret }

{ nounce - c, s. } for MD5 of SHA1
constant.

secretul Pre Master → stabilit de

pontul la handshake → număr { RSA

DH; 3 variante { Fixed DH. - { D.H.
{ Fortezza

One Time DH. → se generează fol cu RSA sau DSS -

Anonim DH. (folosește autentificare).

Generare master Secret: MD5 (pre-master-secret || sha(A, premaster-sec)) ||

|| MD5 (pre master || sha(B, premaster-)) ||

|| MD5 (pre master || sha(C, premaster)).

Generare paranteți criptografici

- client server write MAC

- client server write Key

- c s write iv (init vector) → dim

keyblock = MD5(master secret || sha(A, master))

se face dim keyblock cîte

chei e nevoie -

|| MD5 (master secret || sha(B, master))

|| MD5 (master secret || sha(C, master))

etc.

SSL Handshake

- cel mai complex; permite pontificare:

{ alegă un protocol

autentificare

negociază suport criptografic

stab. premaster

schimb de monce

2 faze: stabilește o nouă sesiune -
de la 1 pînă la 2 noi sesiuni existente

- Resuming → sau duplicate an existing one

Mesaje handshake: serverHello; certificate; ServerKey Exchange; Certif Request;
serverHello Done; Client Key Exchange; Certif Verify;

Change Cipher Spec; Finished;

6 em bit care spune să se schimbe specificările de criptare (algoritru, etc)

- cind CCS e pe 1 se copie staușe pending în staușe curente
- se face update la suita criptografică
- SSL ALERT - descrie alerta pt provizorii atacatorilor
 - la nivel maxim → se termină conexiunea instantaneu.
 - alertă de sechidere → cel puțin un miliun de minute.
 - eroare → failure

SSL Appl. Data → permite comunicarea între parteneri pt a schimba date diverse din aplicație. → SSL record

TLS - transport layer protocol.

+ un SSL standardizat, implementat.

Sch Majorat folosește → generație de Master Secret nou PRF

- se folosește HMAC
- SHA2
- folosire explicită de vectorii de initializare în CBC mode
- suite criptografice noi (pe baza de AES)

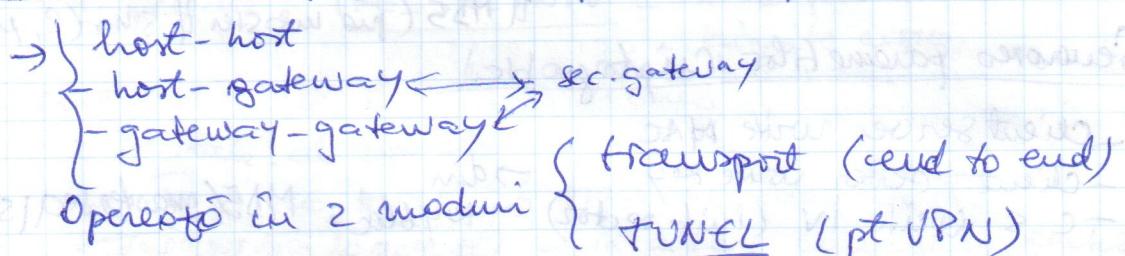
TLS 1.1 → 1.2 → se trece la SHA2-, o nouă fct PRF, HMAC + SHA256,

$$\begin{aligned} P\text{-hash}(\text{secret}, \text{seed}) = & \text{HMAC_hash}(\text{secret}, A(1) \parallel \text{seed}) \parallel \\ & \text{HMAC_hash}(\text{secret}, A(2) \parallel \text{seed}) \parallel \\ & \text{HMAC_hash}(\text{secret}, A(3) \parallel \text{seed}) \parallel \\ & \parallel \text{etc} \end{aligned}$$

Aplicare SSL și TLS

→ HTTPS (HTTP cu SSL/TLS + TCP); FTPS (FTP cu TLS)

IPSEC - oferă securitate pt IPv4 și IPv6 - (internet Protocol).



Nod = loc în rețea unde mesaj poate fi creat, recepționat, transmis

→ ex: orice device cu IP address. în rețea TCP/IP.

Host - nod care este un calculator

Security Gateway = sistem care implementează IPsec protocolul (router sau firewall)

COMPONENTE

IP SEC ① Protocoale de securitate

AH = authentication header - (associat la adresa de origine IP)

- pt a autentifica conținutul datelor primite

ESP = encapsulating security payload -

→ oferă protecție datagramelor IP prin encriptare pt (autentificare optională)

- 2) Asocieri de securitate -
- 3) Protocoale de managementul cheilor.
- 4) Algoritmi de autentif. și criptare

→ Rf că aceste protocoale sunt oferite la nivelul IP, tot fiind la nivale superioare (TCP, UDP, etc)

- ESP cu Auth → oferă: acces control, integrat date, autentif. origini datelor, confidențialitate, reacția pachetelor neadmiti, confid. la trafic limitat.

Mod Transport

- folosind comunicarea între 2 hosts. (client-server)
- Nu e necesar să urmări din partea Gateways.
- oferă protecție și supervizare (TCP, UDP)

☺: costuri mici de procesare

bad: compunile nefixe nu sunt autentificate

AH în mod transport → autentifică payload IP - părțile header (fără compunile mutabile)

ESP în Mod transport - ESP criptat și optional autentifică IP payload - al doilea fără header.

Mod TUNEL : tunneling înseamnă încapsulare →

împachetarea unui pachet în altul nou.

folosind ceea ce din găzduie gateway.

(TRANSPORT Mode e folosit în special între 2 găzdui (host))

protectie e totală, dor costurile de procesare cres.

AH în mod tunel: se autentifică tot pachetul IP părții din IP-ul extern + intern -

ESP în mod tunel % ESP cu autentif. criptat și auten-
tifică pachetul IP intern.

Atacuri: Sequence Number Field →

→ Protecție împotriva spoofing. (pentru adresa saură și dest
nu sunt autentificate)

RFC 4835 → autentif. și criptare pot fi NULL dor că stimulări

Asocieri de securitate -

- sunt conexiuni logice unicatoare - între 2 echipaje - definite în mod unic printr-un triplet:
 - (SPF, IP destinație, protocol de securitate)
 - ↓unicast, broadcast, multicast
 - security param index pe 32 bit -

→ AH sau ESP.

Asoc de securitate sunt urmăriile de urmă:

- o Asoc de sec poate fi făcută cu AH sau cu ESP, doar nu au ambele simultan.
- și o cizăruină care necesită și AH și ESP → se face 2 asocieri de sec separat pt fiecare direcție

SA bundle -

Asoc de securitate este astfel numit selectori de securitate.

Protocolul IKE = Internet Key Exchange.

SA bundle - e o secvență de asocii de securitate prin care se aranjă securitatea dintre -

- 2 moduri
- transport adjacency → combină în mod transport {AH + ac. ESP - datagram}
 - tunneling iterat → mai multe layeruri de securitate (nu mai mult de 3, altfel e imposibil).

Security end to end : ori prin ESP, ori AH, ori ambele, ori transport

- Fiecare asoc. de securitate are o intrare în SAD (Security Association Database)

SPD = security policy Database = specifică serviciu și este astfel în ce mod.

IKE - protocol - component IPsec.

- stabilește o asoc de securitate (cu părțile de securitate)
- autentificare mutuală -
- AH și ESP (SA) și stab. alg. criptografică folosită

Schimbul și perioada de merge → cerere, răspuns.

I perioada ; IKE-SA-INIT - stab. param. IKE SA

core de peștește și pe toate comunicările - tunurile noastre.

→ se stabilește sch.

II → IKE-AUTH → tunurile identitate -

(necesită autentificare) - se dovedește cunoșterea secretelor - entități - stab SA pt AH și apoi Child SA.

Create Child SA -

se face pt creație sau modificarea de asociații de securitate.

- Re-key → cheia veche și se stergă, se adaugă la cheia nouă.

Informațional - stergere o asoc de securitate, report de eroare, etc

- schimbări de jocuri de A

- schimbări într-o rețea și înțelegeri întreținute -

folosind un set de chei sau baze de date

(schimbări în securitate, schimbări în IP, 192)

- schimbări în securitate

- schimbări în securitate

⑥ EMAIL. - metoda de schimb de mesaj de pe Internet
Sistem - FTP based - → prima linie din mesaj conține adresa destinator.
→ de la un singur loc:
- grup de fizic broadcast
- mesajele nu au o structură standard
- expeditorul nu stă deocamdată mesaj și primul destinatar
- mesajele nu pot conține text, desen, vocă, etc.

Un sistem email:

- comuniere
- transfer
- raportare
- afisare
- stocare

alte fct: CC, BCC, liste de adrese; priorități,
cuptor; semnături; etc.

Structura email system:

} User agents (client de mail).

} message transfer agents (demon)

Structura mail

: envelope (pliș) - incapsulează mesajul → conținut
necesar transport (destinație, prioritate, nivel de securitate)

: mesaj

} header
body

Nu sunt același lucru.

envelope ≠ message
header

RFC 821 → SMTP -

RFC 822 → formatare mesajului (nu face diferență {envelope})
mesaj: { header + empty-line + body. }

O linie de mesaj: caractere ASCII pe 7 biți - . (decar!)

: fiecare linie de termen cu CR + LF → ca să fie OK
; fiecare linie are maxim 78 caractere, fără CR LF
dar maxim 998 (deci 1000 cu totul cu CR LF)
altele nu se transmit în sig.

headerul: <name>: <value> nou <header name> & <header val part>
spatiu <header val part>
etc <header part>

Format mesaj: în header: { FROM }

{ TO }

{ subject ; Date ; }

header custom: [X] - în fapt -

- Se crează procesare: compoște - procesare de către clientul de mail;
- procesare server SMTP; (se adaugă Received of Return Path)
- procesare de acceptare; accesare.

MIME - multi purpose email extensions

- permite adăugarea multimediei, fișiere xsl; mesaje unicode.
- se adaugă o structură la body-ul mesaj.
- se codifică datele NON ASCII ca text *SCH

headers MIME (not)

- MIME version
- Content description
- content ID -
- Content type
- content transfer encoding

Body : audio, video, pdf, imagine JPG, text, etc.

Multipart message: → mixed, alternative, parallel a digest.

Metode de codare body

- 7 bit - ASCII - (RFC 822)

- 8 bit binary → 8 bit char -

= quoted-printable - folositi cind aveam text ASCII majoritar

= base 64 → respectă blocuri de 6 biti la blocuri de 8 biti
de 8 biti → toate caracterele printabile ASCII (Radix 64)

header: = ? < charset > ? < encoding > ? < encoded text >

 ↓
unicode

 ↑ B - base 64

 Q - quoted printable.

SMTP - simple mail transfer protocol - pt transfer email prin
TCP/IP - ; serverele pot fi următoare de moduri (receiver, sender)

. SMTP foloseste TCP -

- serverul ascultă la portul 25 continuu -

Un server SMTP - care vrea să trimită email:

- DNS LookUp → la MX record → pt a obține SMTP server destinat -
- motor; → se obține adresa IP a destinatarului -
- se stabilește o conexiune SMTP între cele 2 servere SMTP -
- cind SMTP server termină, se opresc conexiunile

SMTP NU e folosit pt accesa email, ci doar transport

de la surse la destinație -

Model de acces EMAIL - online - conexiune permanentă la rețea -

{ offline → se download. mail și apoi îl
sterge din server -

- disconnected - download mail,

dacă nu este conectat de pe rețea.

Protocol de acces la mail -

POP - (post office protocol)

- implementarea accesului offline -

IMAP - Internet Message Access Protocol → împărțite în 3 modele, doar

e folosit în special la online și disconnected.

Email acces :- direct server email acces - mailul e un filtre
manipulat de clientul de mail

- web based email acces - via web browser

PGP - pretty good privacy. - oferă autenticitate → și conținut
confidențialitate (criptare cădere) ZTB
- se mai spune complete,
- segmentare;
- conversie base 64 -

Management key : generare chei;
stocare chei → inele de chei → 1 pt chei private
- certificate chei → 1 pt chei publice
stocate pe cald. hot

Public key ring: - timestamp; key ID; public key; user ID;
→ cheia 2: owner trust, și semnare, și signature's trust, etc.

- folosire inele chei :
 - ca să userul semneze un mesaj
 - ca să userul cripteze un mesaj
 - userul scrie că o semnătură
 - userul decriptează un mesaj

S/MIME -

- oferă : - împachetare date (envelop)
 - semnare date
 - semnătură în clor (+ base 64) → :
 - semnătură pt împachetare date

semnat în clor, mesaj poate fi vorba de orice anume.

criptare și decriptare = or. cu key private: cript + decrypt
de același cheie

- arhitectură = ch. publică, cript cu k pub. decr cu k privată.
Cele 2 key = cheile arh.