UF1.NF1 – 1. Introducció a la criptografia

CFGS Desenvolupament d'Aplicacions Multiplataforma

Mòdul 9: Programació de serveis i processos

Apunts

Índex

Bibliografia

1. Terminologia bàsica

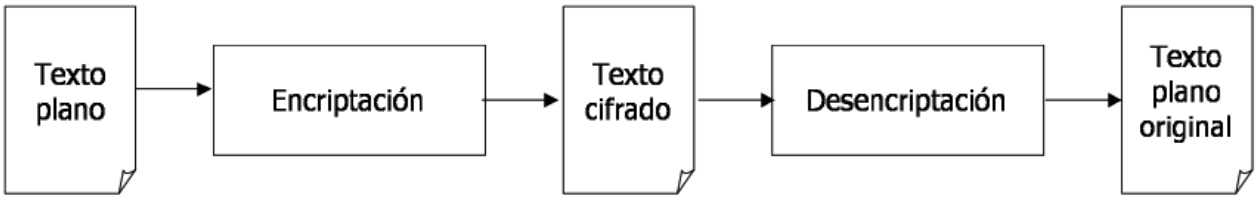
La **criptografia** és un conjunt de tècniques que ens permeten enviar un missatge des d'un emissor a un receptor sense que ningú que intercepti el missatge en el camí pugui interpretar-ho.

Els **criptògrafs** són les persones que estudien i utilitzen la criptografia. L'objectiu dels criptògrafs és poder enviar missatges de forma segura, és a dir, sense que cap altra persona pot descobrir què missatge és el que està enviant per un canal insegur.

La **criptoanàlisi** és una tècnica que busca poder desxifrar missatges xifrats. A les persones que realitzen la criptoanàlisi se'ls anomena **criptoanalistes**.

La **criptologia** és la branca de la matemàtica que s'encarrega d'estudiar tant la criptografia com la criptoanàlisi. Els **criptólegs** són les persones que realitzen aquesta tasca.

Elements que intervenen en un sistema criptogràfic:



El text pla (plaintext) és el missatge que volem transmetre de forma confidencial. El xifrat és el procés de transformar el text pla en un text xifrat que ningú pugui interpretar. El desxifrat és el procés de tornar a transformar el text xifrat en el text pla original.

Al text xifrat se li anomena també text encriptat, que de fet és un nom més correcte, ja que "xifrat" significa que està sent representat per un altre sistema de codificació (normalment numèric), mentre que "encriptat" significa que no es pot accedir a ell. No obstant això, en els llibres s'usa amb més freqüència el terme "xifrat" (ciphertext) i nosaltres ho anomenarem també així.

2. Tècniques criptogràfiques d'algoritme públic i privat

2.1. TÈCNIQUES CRIPTOGRÀFIQUES D'ALGORITME PRIVAT

Les tècniques criptogràfiques d'algoritme privat són tècniques en les quals la confidencialitat de les dades resideix en el ***desconeixement públic de la forma en què s'ha codificat un missatge***.

En principi podrien semblar una bona idea, ja que al no conèixer ningú l'algoritme de xifrat i desxifrat del missatge, ningú podria desxifrar el missatge. Per desgràcia, les tècniques de criptoanàlisis actuals descobreixen molt fàcilment l'algoritme emprat (especialment si coneixen un tros del text pla), amb el que en la pràctica han deixat d'utilitzar-se.

2.2. TÈCNIQUES CRIPTOGRÀFIQUES D'ALGORITME PÚBLIC

Les tècniques criptogràfiques d'algoritme públic són tècniques en les quals ***la confidencialitat de les dades resideix en la clau utilitzada***, i no en l'algorisme emprat, amb el que el sistema continua sent igual de segur si desvetllem l'algoritme.

De fet, en aquestes tècniques quan desvetllem l'algoritme sol augmentar la seguretat, ja que l'algoritme és estudiat per criptólegs de tot el món, i si en un temps prudencial cap ha trobat un defecte, podem estar segurs que l'algoritme és segur.

3. Algoritmes de clau secreta i algoritmes de clau pública

Les tècniques criptogràfiques d'algoritme públic (que són les que nosaltres anem a estudiar), són tècniques en les quals la seguretat de les dades resideix en la clau.

Els algoritmes públics al seu torn es subclassifiquen en dos:

- Algoritmes de clau secreta o simètrics.
- Algoritmes de clau pública o asimètrics.

3.1. ALGORITMES DE CLAU SECRETA O SIMÈTRICS

En aquests algoritmes és necessari que abans de començar a transmetre dades de forma segura, ***l'emissor i el receptor es posin d'acord en la clau a utilitzar***.

A aquests algoritmes se'ls anomena de clau secreta, i no de clau privada, perquè la clau no la coneix un només individu, sinó dos.

Aquests algoritmes són els més clàssics, i fàcils d'entendre, però tenen una limitació important, i és que ***l'emissor i el receptor s'han d'haver posat en contacte prèviament per acordar la clau secreta***, la qual cosa és possible en alguns casos (p.e. comunicació militar, cartes d'amor, xifrar un fitxer), però no en uns altres (p.e. comunicació segura entre un navegador i un servidor web).

Els algorismes de clau secreta se subdivideixen en dos tipus, en funció de la forma en què es codifica i descodifica el missatge:

- **Algoritmes de flux (stream algorithms)**. Són algoritmes en els quals les funcions de xifrat i desxifrat reben el missatge a codificar o descodificar bit a bit. En conseqüència, es fa una crida a funció per cada bit a codificar o descodificar. Per la seva naturalesa, són més apropiats per implementar en maquinari (circuits dissenyats per xifrar i desxifrar).
- **Algoritmes de bloc (block algorithms)**. Són algoritmes en els quals les funcions de xifrat i desxifrat no reben un només bit, sinó un bloc de bits (64 bits, 128 bits, etc.). En aquests algoritmes habitualment durant l'última crida cal ficar un farcit (padding) perquè completem els 64 o 128 bits. Són més apropiats per implementar en programari.

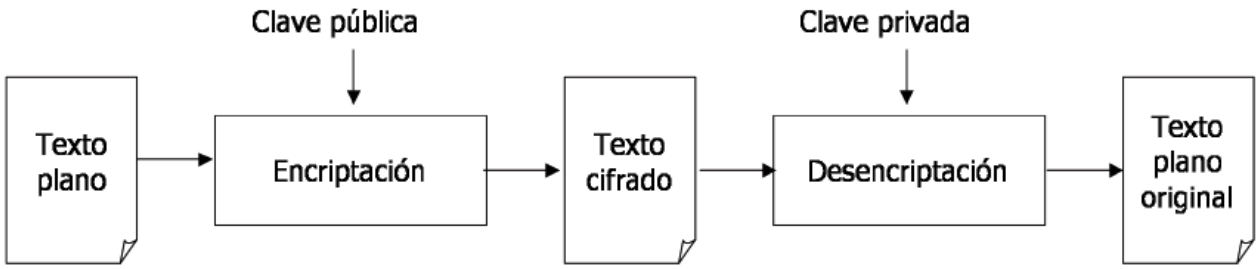
3.2. ALGORITMES DE CLAU PÚBLICA O ASIMÈTRICS

Aquesta nova forma de criptografia va ser proposada per primera vegada per Whitfield Diffie i Martin Hellman, dos professors de la Universitat de Stanford en 1976, aquests senyors van proposar un algoritme de xifrat anomenat algoritme de la motxilla, que encara que actualment està trencat, va posar les bases d'una nova forma de pensar, en la qual es van basar altres algoritmes, com per exemple l'algorisme RSA.

La gran novetat que introdueixen els algoritmes de clau pública és que permeten que emissor i receptor es comuniquin de forma segura ***sense haver-se posat d'acord prèviament en una clau secreta*** (p.e. un navegador es comunica amb un servidor web de forma segura a través d'HTTPS sense que prèviament es coneguessin per haver acordat una clau secreta).

Aquí, perquè l'emissor pugui enviar informació al receptor de forma segura, el receptor ha de disposar de dues claus:

- **Clau pública**. Aquesta clau l'ha de conèixer tothom, i només val per xifrar missatges.
- **Clau privada**. Aquesta clau només la coneix el receptor (a diferència de la clau secreta que la coneixien emissor i receptor), i només val per desxifrar missatges.

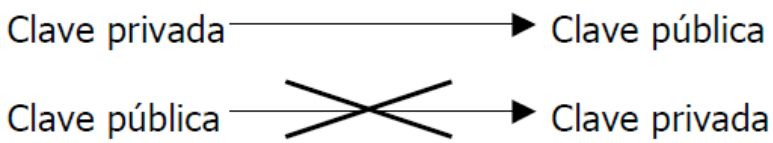


D'aquesta forma l'emissor pot xifrar missatges, i només el receptor pot desxifrar-los.

Per exemple, suposem que un banc té una clau privada i lliura la corresponent clau pública als seus clients. En aquest cas els clients poden enviar missatges al banc xifrats amb la clau pública del banc, i tenen la seguretat que només el banc els sabrà desxifrar.

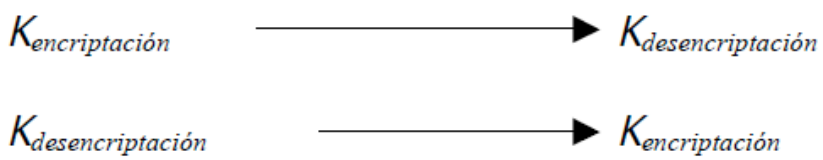
Perquè un possible espia que hi hagi al mig no pugui desxifrar els missatges xifrats amb la clau pública, però sí que els pugui desxifrar el receptor, l'única cosa que s'exigeix és que:

- Donada la clau privada sigui possible calcular la clau pública.
- Donada la clau pública sigui impossible deduir la clau privada.

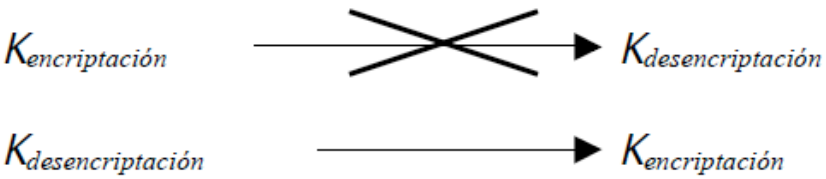


En concret, el receptor tria una clau privada, calcula la seva clau pública, i demana a l'emissor que li enviï el missatge codificat amb aquesta clau pública. Sota aquestes condicions, si l'espia (que només disposa de la clau pública) no pot deduir la clau privada, no pot desxifrar el missatge.

Als algoritmes de clau secreta se'ls anomena també algorismes simètrics, ja que, o bé utilitzen la mateixa clau per xifrar i desxifrar, o bé utilitzen claus diferents, però sempre es compleix que:



Als algoritmes de clau pública se'ls anomena també algorismes asimètrics, perquè:



4. Esteganografia

La esteganografia consisteix a ***enviar un missatge secret amagat en un altre missatge no secret***.

Aquesta tècnica la van inventar els poetes, que ficaven un missatge ocult usant cadascuna de les primeres lletres de cada vers (línia) d'un poema. Una versió moderna d'aquesta tècnica és utilitzar el bit menys significatiu d'una imatge, i hi ha programes freeware que ho fan. El principal problema d'aquesta tècnica és que, com a tècnica criptogràfica d'algoritme privat que és, si algú sospita que s'estan enviant missatges així, li resultarà fàcil desxifrar-los. Un altre problema és que necessitem un missatge gran per amagar el missatge secret. Per exemple, en un bitmap necessitaríem 8 píxels per amagar un byte.

5. Xifrat per substitució i transposició

5.1. SUBSTITUCIÓ

El xifrat per substitució consisteix a reemplaçar cada caràcter del text pla per un altre caràcter en el text xifrat, i per desxifrar se segueix el procés invers.

Un exemple de xifrat per substitució molt conegut és el "Xifrat del César", que ho va usar per enviar missatges als seus generals en el front. Consistia que substituïen cada lletra del missatge per tres lletres més endavant en el vocabulari en mòdul 26, tal com mostra el següent exemple:

Texto plano ——————————> H U Y E
Texto cifrado ——————————> K X B H

Per desxifrar-ho només calia substituir les lletres xifrades per tres lletres enrere en el vocabulari, és a dir:

Texto cifrado ——————————> K X B H
Texto plano ——————————> H U Y E

5.2. TRANSPOSICIÓ

Els xifradors per transposició simplement canvien l'ordre de les lletres.

El xifrat per transposició més comuna és el xifrat per columnes, que consisteix a col·locar el text en files d'ample predeterminat i llegir-ho per columnes.

P.ex. si volem transmetre:

Texto plano: NOS ATACAN CON CARBUNCO

N	O	S	_	A
T	A	C	A	N
_	C	O	N	_
C	A	R	B	U
N	C	O	_	_

Texto cifrado: NT CN OACACSCORO ANB AN U

Com ja havíem indicat, aquestes tècniques criptogràfiques es troben en l'actualitat superades. El llibre: "Cryptanalysis" H.F. GAINES Ed. American Photografic Press, mostra lo fàcil que és trobar l'algoritme privat utilitzat quan s'utilitzen tècniques de substitució, transposició o ambdues combinades.

Exercicis 1 i 2

6. Protocols

En criptografia anomenem protocol a ***l'ordre en què hem d'executar els algoritmes criptogràfics per resoldre un problema criptogràfic de forma segura***.

Els protocols existeixen perquè de vegades, si aquests passos s'executen de forma incorrecta, es pot trencar la seguretat del sistema criptogràfic, encara que tots els algoritmes utilitzats siguin algoritmes totalment segurs.

Els protocols han estat dissenyats per experts i publicats en molts llibres sense que ningú hagi trobat un forat de seguretat en un sistema que funcioni tal com diu el protocol.

6.1. ROLS QUE PARTICIPEN A UN PROTOCOL

6.1.1. Els interlocutors

A un protocol pot participar un únic individu A (p.e. xifrar un fitxer), dos individus A, B (p.e. en una comunicació xifrada), o fins i tot més de dos interlocutors, en aquest cas els anomenarem A, B, C, D…

6.1.2. Els àrbitres

Quan els individus que participen en un protocol no confien l'un en l'altre, poden necessitar l'ajuda d'un àrbitre, que és un ***tercer participant en el qual tots dos confien***, és a dir, l'àrbitre ha de ser imparcial, i tots dos se sotmeten a les resolucions de l'àrbitre.

P.ex. Si A vol vendre un cotxe a B, però, d'una banda B vol pagar amb xec a A, i A no vol donar les claus del cotxe a B fins que hagi comprovat que el xec és vàlid, pot executar un protocol amb àrbitre com el següent:

1. A dóna les claus del cotxe a l'àrbitre.
2. B dóna el xec a A.
3. A diposita en xec al banc.
4. Transcorregut el temps pactat l'àrbitre dóna les claus a B, però si en aquest temps A demostra a l'àrbitre que el xec no tènia fons, llavors l'àrbitre retorna les claus a A.

Un tipus d'àrbitre són els ***expenedors de certificats***, que són àrbitres que poden signar documents per demostrar la seva autenticitat.

P.ex. un expenedor de certificats pot ser un banc. El banc pot expedir un xec certificat, en el qual garanteix que aquests diners està en el compte de B, i que B no ho pot treure. En aquest cas el protocol per al problema que A vengui un cotxe a B podria ser:

1. B demana un xec certificat al banc.
2. A dóna les claus a B, i B dóna el xec certificat a A.
3. A cobra el xec sense problemes, ja que està certificat pel banc.

Un altre tipus d'àrbitre són els ***notaris*** la missió dels quals és prendre nota que una operació entre A i B s'ha realitzat, en cas que posteriorment hi hagués una disputa entre A i B, el notari diria què és el que realment va passar.

A causa que el cost econòmic d'un àrbitre sol ser gran, molts protocols utilitzen ***jutges***, que són àrbitres que només actuen en cas que hi hagi una disputa.

Si usem un jutge, el protocol general seria:

1. A i B negocien els termes del contracte.
2. A signa el contracte.
3. B signa el contracte.

Si més tard hi hagués una disputa:

4. A i B presenten el contracte al jutge.
5. El jutge resol la disputa.

6.1.3. Els atacants

Els atacants són individus que no haurien de formar part del protocol.

Un possible atacant és ***l'espia***, que és un atacant que no modifica el protocol, sinó que només intenta accedir a les dades intercanviades. A aquest tipus d'atacant també se li crida atacant passiu.

Un altre tipus d'atac més perillós és ***l'atac actiu***, en el qual l'atacant pot veure i modificar les dades que s'intercanvien en el protocol.

En cas que l'atacant actiu sigui un dels participants, se li anomena participant mentider, i també és un atac molt perillós.

6.2. PROTOCOL D'UNA COMUNICACIÓ SIMÈTRICA

La comunicació simètrica és la que es realitza utilitzant algoritmes criptogràfics de clau secreta. En aquest tipus de comunicació el protocol que s'utilitza té la següent forma:

1. A i B es posen d'acord en el sistema criptogràfic a utilitzar.
2. A i B es posen d'acord en la clau a utilitzar.
3. A xifra un missatge usant l'algoritme i clau acordats.
4. A envia el missatge xifrat a B.
5. B desxifra el missatge usant l'algoritme i clau acordats.

A l'actualitat els sistemes criptogràfics són prou segurs a una criptoanàlisi de text xifrat per força bruta, però si l'espia no és estúpid pot intentar espiar el pas 2, en aquest cas l'atac és més perillós. Per aquesta raó és pel que avui dia és tan important el que es denomina la ***gestió de claus*** (key management).

6.3. PROTOCOL DE PARTICIÓ D'UN SECRET

La partició d'un secret consisteix a ***dividir un missatge en peces***, aconseguint que, de forma individual no signifiqui gens, però si unim totes les peces tenim el missatge original.

P.es. si tenim la fórmula de la Coca-cola, un àrbitre la pot partir en dos, i donar una peça a A i una altra a B, seguint el protocol que anem a detallar:

1. L'àrbitre genera una sèrie aleatòria de bits R, que tindrà la mateixa longitud que el missatge M.
2. L'àrbitre fa un XOR de M i R per obtenir S: M^R=S.
3. L'àrbitre dóna R a A, i S a B.
4. Després, per reconstruir el missatge es reuneixen A, B i fan un XOR a les seves parts, per obtenir així el missatge original: M= R^S.

Si l'àrbitre és de confiança, la sèrie aleatòria és totalment aleatòria, i no es repeteix, estem usant un xifrat one-time pad, que és incondicionalment segur.

Lògicament, si el missatge és molt gran, és millor compartir una clau secreta que després s'usi per xifrar i desxifrar tot el missatge.

No obstant això, partir un secret té un seriós inconvenient: Si un dels membres mor (o es passa a la competència) el secret es perd per sempre.

6.4. PROTOCOL DE COMPARTICIÓ D'UN SECRET

L'inconvenient de la tècnica de partició d'un secret es pot solucionar amb tècniques de compartició d'un secret, en les quals ja ***no és necessari que tots els membres estiguin presents per desvetllar un secret***, sinó que només és necessari que s'aconsegueixi un llindar mínim.
P.ex. si estem dissenyant un sistema de llançament de míssils nuclears, podríem exigir que almenys 3 dels 5 generals del Pentàgon estiguin d'acord perquè els míssils es llancin. Fins i tot ho podríem complicar més i exigir que si un general no està disponible, el seu vot pugui ser substituït pel de 5 coronels. Per a això utilitzem un sistema de participacions, de manera que els generals tenen 5 participacions, i els coronels 1. Ara fixem un sistema de 20/30 participacions perquè s'iniciï el llançament dels míssils. Els algoritmes de compartició de secret estan descrits en el llibre: "An Introduction to Shared Secrets and/or Shared Control Schemes" GUS SIMMSONS. Ed IEEE Press.

Els algoritmes de partició i compartició de secrets tenen altres inconvenients com són:

1. És fàcil que un membre menteixi. P.ex. si C és un pacifista pot introduir un nombre erroni, i encara que tots els altres fiquin el nombre correcte, el resultat és incorrecte. Lògicament, també s'han inventat algoritmes que eviten això.
2. Quan es reuneixen per reconstruir el secret tots han de desvetllar els seus nombres. També s'han inventat formes que els nombres segueixin sent secrets.

7. Funcions Hash unidireccionals

Anem a explicar que són les funcions hash unidireccionals, també anomenades funcions de compressió o message digest.

- Aquestes funcions tenen moltes aplicacions en criptografia. Aquestes funcions reben un text de longitud variable i retornen un altre de longitud fixa (generalment més petit).
  Una funció hash molt simple seria una que rep un text i retorna el XOR de tots els seus bytes. Per desgràcia aquesta funció hash no seria unidireccional, ja que és molt fàcil trobar un altre text d'entrada que retorni la mateixa sortida.
- Una bona funció hash és aquella per la qual és molt difícil trobar un text d'entrada que doni la mateixa sortida.
- També cal tenir en compte que les funcions hash són públiques, és a dir, tothom sap com calcular el hash d'un text. La seva força resideix en la impossibilitat de trobar una altra entrada que produeixi aquesta sortida.

Les dues funcions hash més usades es diuen: MD5 i SHA1.

Un DOS (Denial Of Service) molt efectiu consisteix a canviar els missatges xifrats que viatgen per la xarxa amb la finalitat de que l'aplicació receptora perdi el control de flux i exploti. Una de les aplicacions de les funcions hash unidireccionals és evitar aquest problema, és a dir, **comprovar que el missatge xifrat arriba sense alteracions, i si les ha sofert, detectar-ho**.

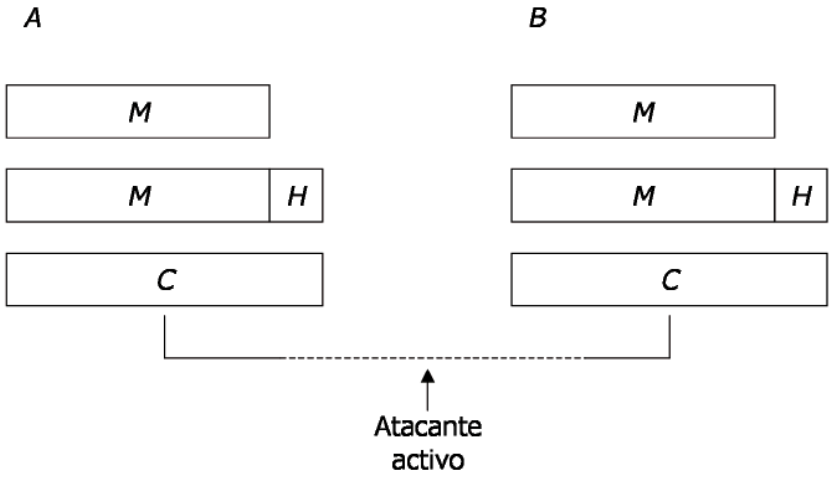
Para això, s'utilitza el següent protocol:

1. A escriu un missatge M.
2. A calcula el hash de M
3. A, usant la clau secreta, xifra M concatenat amb la seva hash per obtenir C.
4. A envia a B el missatge xifrat C.

Ara en recepció:

5. B desxifra C amb la clau secreta.
6. B comprova que el hash concatenat H correspongui al missatge M.
7. B utilitza el missatge pla M rebut.

Observi's que si ara un atacant actiu modifiqués el text xifrat C durant el seu viatge per la xarxa, B detectaria que el hash H del missatge rebut M no és correcte, i no ho usaria.



8. MAC (Message Authentication Codes)

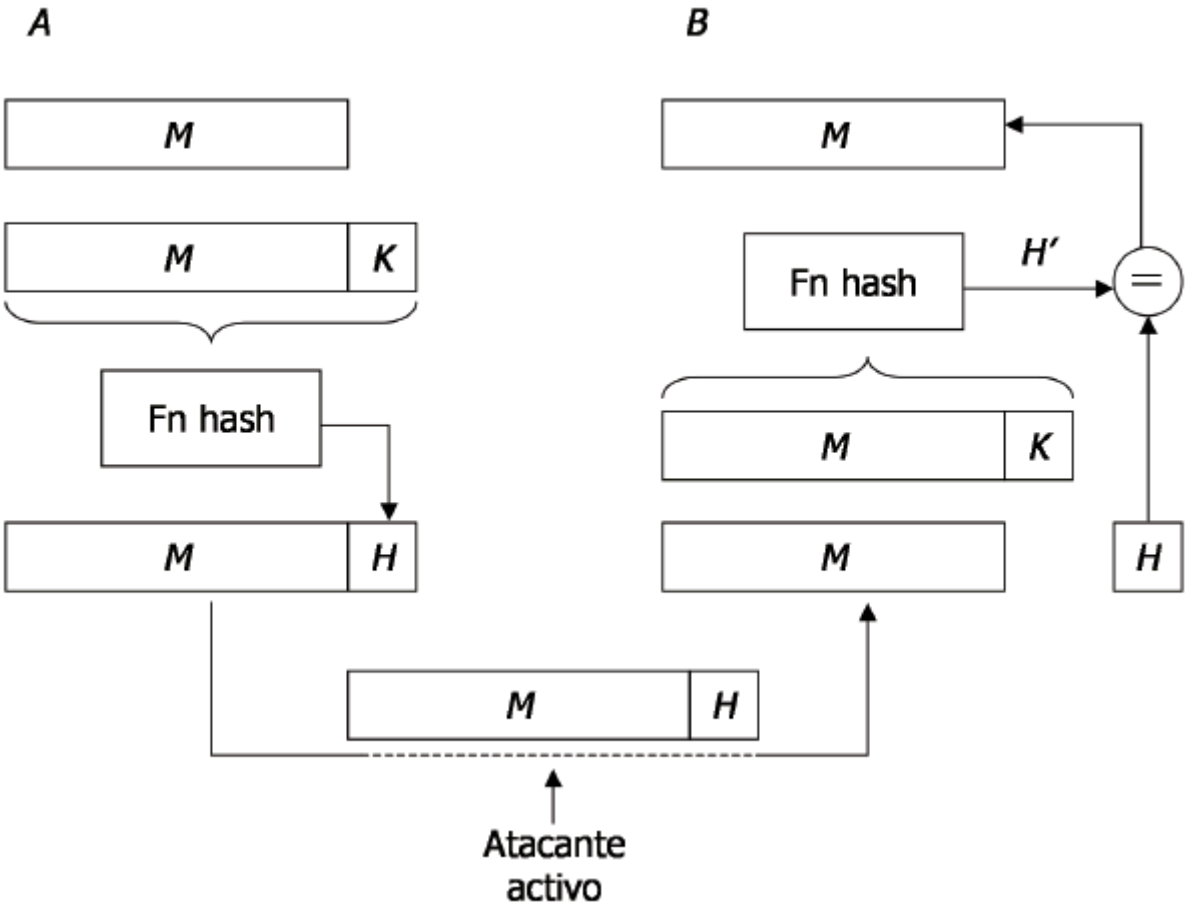
Les funcions hash les hem usat per garantir que el text xifrat no ha estat modificat en la transmissió. Els MAC van un pas més enllà i ens garanteixen que un text pla no ha estat modificat per un atacant actiu durant el seu transmissió. Els MAC s'utilitzen quan no ens interessa ocultar el missatge, ***només ens interessa que ningú el pugui modificar***.

Per garantir l'autenticitat, els MAC, utilitzen el següent protocol:

1. A escriu un missatge M.
2. A concatena al missatge M una clau secreta K i calcula la seva hash H.
3. A envia el missatge M i el seu hash H (però no la clau secreta K) a B.

Quan ho rep B fa el següent:

4. B concatena al missatge rebut M la seva clau secreta K i calcula la seva hash H'.
5. Si el hash calculat H' coincideix amb l'H rebut, B sap que el missatge no ha estat modificat.



Ara, si l'atacant modifica el missatge, no pot calcular el hash del nou missatge al no conèixer la clau secreta. En els MAC, per comprovar l'autenticitat d'un missatge cal conèixer la clau secreta. Existeix una altra tècnica anomenada ***signatures digitals***, en la qual podem comprovar l'autenticitat d'un missatge, fins i tot sense necessitat d'acordar una clau secreta, sinó coneixent només la clau pública de l'emissor.

Exercicis 3 i 4