Capitolo 8

1) Quali sono le differenze tra riservatezza e integrità del messaggio? Si può avere una senza l'altra? Argomentate la risposta.

La riservatezza si intende che solo il mittente e il destinatario sono in grado di comprendere il contenuto del messaggio che viene trasmesso, quindi è *necessario* cifrare il messagio per renderelo incomprensibile ad altri. Per integrità si intende che il contenuto del messaggio trasmesso non subisca, durante la trasmissione, alterazioni dovute a manipolazioni. Si può avere una senza l'altra ma in questo modo non verrà assicurata una comunicazione sicura tra diversi utenti.

2) Le entità di Internet, quali router,switch,DNS e web server nonché i sistemi periferici, hanno spesso la necessità di poter comunicare in modo sicuro. Date 3 esempi di entità con questa necessità.

1) Un consimatore che trasmette il suo numero di carta di credito a un web server pre effettuare un acquisto

2) Un cliente che interagisce online con la propria banca

3) Soggetti che richiedono comunicazione sicura potrebbero far parte integrsnte dell'infrastruttura di rete. Nel caso del DNS, un malintenzionato può attivamente interferire ,controllare o danneggiare la ricerca e aggiornamento di un DNS provocando danni su Internet.

3) Qual è un importante differenza fra un sistema a chiave simmetrica e uno a chiave pubblica dal punto di vista del servizio?

Nei sistemi a chiave simmetrica due utenti che vogliono scambiarsi i messaggi hanno un unica chiave privata che condividono tra di loro,che utilizzano per cifrare e decifrare il messaggio trasmesso.

Nei sistemi a chiave pubblica gli utenti hanno ognuno 2 chiavi: una pubblica e una privata. La chiave pubblica la conoscono tutti, invece la privata è unica per ogni utente e non è condivisibile. Se Alice deve inviare un messaggio a Bob, essa utilizzarà la chiave pubblica di Bob per cifrare il messaggio e Bob utilizzerà la sua chiave privata per decifrarlo.

Il sistema a chiave simmetrica risulta più costoso di quello a chiave

privata poiché se vi sono N utenti saranno necessarie N(N-1)/2 chiavi (quindi nell'ordine di N^2) , contro le 2N del sistema a chiave pubblica.

La crittografia a chiave pubblica però è più lenta della crittografia a chiave simmetrica in quanto la lentezza è data dalla lunghezza delle chiavi e dalla complessità degli algoritmi di crittografia utilizzati, come per esempio RSA, che deve applicare operazioni semplici, ma che se vengono utilizzati numeri di una certa grandezza aumenta il tempo di calcolo e quindi la complessità. Al fine di mantenere la sicurezza, la crittografia asimmetrica deve rendere troppo difficile per un hacker per rompere la chiave pubblica e scoprire la chiave privata. Infatti con RSA la fattorizzazione dei numeri primi non risulta banale, infatti p e q sono numeri molto molto grandi.

4) Supponete che un intruso disponga di un messaggio cifrato e della sua versione decodificata. Avvalendosi di queste conoscenze può organizzare un attacco al testo cifrato, un attacco con testo in chiaro noto o un attacco con testo in chiaro scelto?

Un attacco con testo in chiaro scelto, dato che in questo caso l'intruso è ha ottenuto la forma cifrata di un testo a lui noto. Per esempio se Trudy intercetta un messaggio di Alice che contiene tutte le lettere dell'alfabeto,allora può decifrare lo schema crittografico.

5) Considerate un cifrario a 8 blocchi. Quanti possibili blocchi ha? Quante associazioni esistono? Se consideriamo Ogni associazione come una chiave, quante possibili chiavi ha questo cifrario?

Numero possibile di blocchi 2^8=256

Numero associazioni 2^8=256

Numero di possibli chivavi di questo cifrario sono 2^8!=256!

6) Supponete che *N* individui vogliano comunicare tra di loro utilizzando la cifratura a chiave simmetrica e che il messaggio fra due persone,i e j,siano visibili a tutti i componenti del gruppo, che non sono però in grado di decodificarli. Quante chiavi sono necessarie nell'intero sistema? Considerate ora la chiave pubblica. Quante chiavi sono richieste in questo caso?

1) N(N-1)/2 chiavi dato che nella crittografia simmetrica due utenti

quando vogliono scambiarsi il messaggio condividono la stessa chiave segreta.

2) 2*N chiavi,ogni utente avrà un chiave pubblica e una privata, in questo modo tutti possono codificare messaggio con la chiave pubblica del destinatario e il destinatario può decodificare messaggio con la sua chiave privata

7) Supponete n=10000 a=10023 e b=10004. Usate una delle uguaglianze dell'aritmetica in modulo per calcolare velocemente (a*b) mod n

(a*b)mod n = [(a mod n) * (b mod n)] mod n = 92

8) Supponete di voler cifrare il messaggio 10101111, codificando il numero in notazione decimale corrispondente. Qual è?

10101111=1+2+4+8+32+128=175

9) In che modo l'hash di un messaggio fornisce miglior controllo dell'integrita rispetto ai checksum, come quello di Internet?

Le funzioni hash prendono in input un messaggio di lunghezza variabile e danno in output un stringa di lunghezza prefissata,digest, (risulta impossibile trovare 2 messaggi diversi che abbiamo come output della funzione hash la stessa stringa di lunghezza prefissata) che poi verrà cifrato con la chiave segreta del mittente inviando la firma digitale + messaggio m al destinatario che andrà a decifrare ritornando la funzione hash che dovrà,a sua volta, confrontare con la funzione hash creata a partire dal messaggio di invio (plaintext), verificandone cosi la corrispondenza.

Internet Checksum consiste nel trattare i caratteri come byte e sommarli a blocchi di 4. Il problema di questo checksum sta nel fatto che una stringa diversa potrà avere lo stesso checksum di una precendente, cosa che accade in maniera veramente rara con le funzioni hash.

10) Potete decifrare l'hash di un messaggio per ottenere il messaggio originale? Argomentate la vostra risposta.

La funzione hash è invertibile. L'unico modo per ricreare i dati di input dall'output di una funzione di hash ideale è quello di tentare una ricerca forza-bruta di possibili input per vedere se vi è corrispondenza. Ma è un modalità praticamente impossibile.

11) Considerate una variazione dell'algoritmo MAC nel quale il mittente invia (m,H(m)+s), dove H(m) + s è la concatenazione di H(m) +s. Tale variazione è corretta? Motivare la risposta.

No, perché non applicando la funzione hash anche sullo shared secret si è più vulnerabili ad attacchi da parte di un intruso, il quale potrà recuperare lo shared secret, invece questo non poteva succedere nel caso in cui andavo a concatenare (XOR) lo shared secret con il messaggio m che andavo a dare in pasto alla funzione hash. Questo perché la funzione hash è irreversibile, quindi non permette di poter risalire al messaggio originale da essa.

12) Che cosa significa che un documento firmato deve essere verificabile e non falsificabile?

Vuol dire che si deve consentire di dimostrare che un certo documento sia davvero stato firmato proprio da quella data persona (verificabile) e che solo lei poteva realizzarlo (non falsificabile).

13) In che modo l'hash di un messaggio firmato con chiave pubblica fornisce una firma digitale migliore rispetto al messaggio cifrato con chiave simmetrica?

Poichè l'hash di una funzione firmato con chiave pubblica(K^-_B (H(m))) fornisce sia l'autenticazione dell'utente, poiché solo lui può conoscere K^-_B, sia l'integrità del messaggio, dal momento che la firma digitale generata da Bob per m non sarà valida per nessun altro messaggio. Al contrario un msg cifrato con chiave simmetrica garantisce sì l'autenticità dell'utente dal momento che solo i due che partecipano alla comunicazione possono conoscere la chiave segreta, ma non fornisce integrità dei dati (no ripetizioni, seq corretta, contenuto msg non alterato, ecc).

14) Supponete che Alice abbia un messagio pronto per essere inviato a chiunque lo richieda. Migliaia di persone lo vogliono,ma desiderano essere sicure dell'integrità del messaggio. In questo caso pensate sia meglio uno schema basato su MAC o sulla firma digitale?

MAC, questo perché in questo caso dato che le persone sono molte, non conviene utilizzare la firma digitale che risulta essere più onerosa dato che deve richiedere l'infrastruttura di chiave pubblica PKI sottostante con le relative autorità di certificazione CA. Il MAC non utilizza ne chiave pubblica ne privata, ma basta aggiungere la

chiave autenticazione (share secret) e prendiamo l'hash del risultato.

15) Qual è lo scopo di un Nonce nei protocolli di autenticazione end-point?

Nonce è numero di protocollo usato per assicareal ricevente che il mittente è ancora attivo.

16) Perchè si dice che il Nonce è un valore che compare solo una volta? Nella vita di chi?

Supponiamo che Alice debba inviare un messaggio a Bob. Bob sceglie il Nonce, R, e lo trasmette ad Alice. Alice utilizza la chiave simmetrica segreta che condivide con Bob, K_{A-B} , per decodificare il Nonce,e gli ri-invia il valore risultante K_{A-B} (R). Bob decifra il messaggio ricevuto e se il nonce è quello da lui inviato allora Alice è autenticata. Nonce compare solo una volta, cioè per verificare che quell'utente sia attivo e di conseguenza per l'autenticazione di esso e come visto è richiesto dal destinatario del messaggio.

17) Supponete chinviato a chiunque lo richieda. Migliaia di persone lo vogliono,ma desiderano essere sicure dell'integrità del messaggio. In questo caso pensate sia meglio uno schema basato su MAC o sulla firma digitale?e Bob riceva messaggio PGP da Alice. Come fa Bob a essere sicuro che sia stata Alice a creare il messaggio, piuttosto che, per esempio, Trudy? PGP usa un MAC per l'integrità del messaggio?

PGP usa la crittografia a chiave asimmetrica, nella quale il destinatario del messaggio ha generato precedentemente una coppia di chiavi collegate fra loro; una chiave pubblica ed una privata. La chiave pubblica del destinatario serve al mittente per cifrare una chiave di sessione per un algoritmo di crittografia simmetrica; questa chiave viene quindi usata per cifrare il testo in chiaro del messaggio. Molte chiavi pubbliche di utenti PGP sono a disposizione di tutti dai numerosi key server PGP sparsi per il mondo, che operano come mirror reciproci.

Il destinatario di un messaggio protetto da PGP decifra prima la chiave si sessione inclusa nel messaggio usando la sua chiave privata. Decifra poi il testo usando la chiave di sessione con l'algoritmo simmetrico. L'uso di due cifrature è giustificato dalla notevole differenza nella velocità di esecuzione tra una cifratura a chiave asimmetrica ed una a chiave simmetrica.

PGP inoltre fornisce anche un meccanismo di certificazione della chiave pubblica, diverso da quello convenzionale CA. Le chiavi pubblice di PGP sono verificate attraverso una rete di fiducia.

18) Nei record SSL esiste un campo per numeri di sequenza SSL?

Falso, i numeri di sequenza vengono inseriti nel campo MAC insieme al MAC, cioè ora il MAC è un hash didati cui si aggiunge la chiave MAC M_C e il numero di sequeza corretto.

20) Qual è lo scopo dei nonce casuali nell'handshake di SSL?

I nonce dell'handshake di SSL vengono utilizzati per impedire la ripetizione delle connessioni, ma in particolare ache per la creazione delle chiavi si sessione.

21) Supponete di stare inviando un flusso di pacchetti dall'Host A all'Host B usando Ipsec. Generalmente viene stabilita una nuova SA per ogni pacchetto inviato nel flusso. Vero o Falso?

No perché Sa è persistente quindi rimane attiva durante tutta l'intera sessione, quindi non viene stabilita una nuova.

22) Supponete che sia in esecuzione TCP su IPsec tra il quartier generale e la filiale. Se TCP ritrasmette lo stesso pacchetto, i 2 pacchetti corrisondenti inviati da R1 avranno lo stesso numero di sequenza nell'intestazione ESP. Vero o falso?

No, perché anche se utilizzo TCP i numeri di sequenza relativi a IPsec sono indipendenti da esso; quindi in questo caso aumentano e non rimangono uguali

23) Una SA IKE e una SA IPsec sono la stessa cosa?

Assolutamente no. Prima di tutto la SA IKE è bidirezionale, invece la SA IPsec è unidirezionale. Inoltre sulla SA IKE vengono scambiate le chiavi per la cifratura e l'autenticazione della SA IKE e anche il master secret che verrà utilizzato per calcolare le chiavi relative a SA IPsec. Infine nella SA IKE vengono stabilit gli algoritmi che devono essere utilizzati dalla SA IPsec.

24) Considerate WEP per 802.11. Supponete che i dati siano 10101100 e che la sequenza di chiavi generate sia 1111000. Qual è il testo cifrato risultante?

Devo fare lo XOR → 10101100 XOR 1111000= 11010100

25) In WEP viene mandato in chiaro un IV per ogni frame. Vero o falso?

Vero, il valori di IV cambia da un frame all’altro e viene inserito nel plaintext nell’intestazione di ogni frame cifrato da WEP.

26) In cosa consiste il paradosso del compleanno e per quale motivo è importante per la sicurezza delle funzioni hash? Spiegarlo attraverso un esempio.

Il paradosso del compleanno afferma che la probabilità che in un insieme di persone ce ne siano almeno due che festeggiano il compleanno nello stesso giorno è molto più alta di quella che potrebbe sembrare intuitivamente, tanto da sembrare paradossale. Il paradosso è importante nel calcolo dell'impronta di un documento tramite funzione hash perché, sapendo che se ho un documento e ho una funzione hash come ad esempio DES a 64 bit, è abbastanza difficile trovare un documento con la stessa impronta, tuttavia se considero un insieme grande di documenti la probabilità che ve ne siano almeno due con la stessa impronta è troppo alta e rende la funzione inadeguata.

Il paradosso è usato per quantificare la validità della funzione hash.

26 bis)

1. Quante persone bisogna scegliere a caso affinché con prob.>0.5 i sia una persona con lo stesso mio compleanno? 253

2. Quante persone bisogna scegliere a caso affinché con prob.>0.5 ci siano almeno due persone con lo stesso compleanno? 23.

1. Calcoliamo la probabilità che una persona sia nata il mio stesso giorno: la probabilità che non sia nata il mio stesso giorno è: $1 - 1/365 = 364/365$ per cui $1-(364/365)=1/365$ è la probabilità che invece ce ne sia almeno una nata il mio stesso giorno.

Quindi scegliendo n=183 avrò $1-(364/365)^{253}=0.51>0.5$

2. Supponiamo di prendere in considerazione il fatto che due persone non compiano gli anni lo stesso giorno; utilizzando la probabilità contraria, si trova che vale: $1 - 1/365$, dato che vi è una sola possibilità su 365 che il compleanno di una persona coincida con quello di un'altra.

Possiamo dire che la seconda persona realizza la non coincidenza del proprio compleanno con la prima, con probabilità 364/365.

Generalizzando, consideriamo n persone e calcoliamo che in tale gruppo non ci siano due persone con lo stesso compleanno. Essendo in presenza di eventi indipendenti, *la probabilità p1 che tutti i compleanni cadano in date diverse è data da*: (365-n+1)/365

e dunque la probabilità p del suo evento complementare, cioè che *esistano almeno due compleanni uguali* è: 1- (365-n+1)/365

Se scego n=23 p= 0,51 che è > 0.5

27) Su cosa si basa la sicurezza di un algoritmo a chiave pubblica? Descrivere brevemente gli aspetti principali di RSA.

La sicurezza si basa sul fatto che la decodifica si compie con la conoscenza di ben due chiavi, di cui una è di dominio pubblico, mentre l'altra la possiede solo il ricevente delle informazioni. Dunque i due individui si scambiano messaggi crittografati senza essersi mai scambiati la chiave di codifica (come avviene per la crittografia a chiave simmetrica).

Kpri(Kpub(mess))=mess

RSA è il metodo più usato per realizzare algoritmi asimmetrici:

scelgo p

scelgo q

n=p*q

z=(p-1)(q-1)

encryption:

scelgo 'e' tale che e<n, MCD(e,z)=1 ('e' e 'z' non hanno fattori

comuni)

decryption:

scelgo 'd' tale che e*d mod z=1

la chiave pubblica è Kpub=(n,e)

la chiave privata è Kpri=(n,d)

cifratura: c=m^e mod n

decifratura: m= c^d mod n

28) Assumete che un KDC server o un CA server si guasti. Chi può comunicare in modo sicuro e chi no nei due casi?

Nel KDC le chiavi segrete sono esplicitamente di sessione quindi vengono usate in quella sessione e poi buttate. Nel KDC poi non ci sono i CA. Con KDC guasto non possiamo fare nulla anche possiedo le chiavi sengrete condivise con KDC.

Mentre se utilizziamo CA ormai la chiave pubblica l'abbiamo acquisita e non cambia poi così spesso. CA rimane in piedi e KDC no.

29) Dovete realizzare un sistema per l'autenticazione in una rete aziendale non connessa ad Internet a cui accedono soli 5 utenti. Che soluzione proporreste? Motivare la risposta.

Poiché ci sono solo 5 utenti, io lascerei stare l'autenticazione intesa in senso classico ed utilizzerei un sistema con chiavi private. Infatti il solo fatto che due utenti conoscano la chiave privata da loro condivisa, ci assicura l'autenticazione. L'unico problema delle chiavi private è lo scambio delle chiavi e il numero delle chiavi: problema aggirabile perché, appunto, ci sono solo 5 utenti!

30)Per quale motivo per firmare un documento è necessario prima ottenerne l'impronta attraverso una funzione di hash? Quali caratteristiche deve avere una funzione di hash?

Le firme hash possono essere utilizzate per la creazione di firme digitali in quanto permettono la rapida creazione di firme digitali anche per file di grossa dimensioni.

E’ infatti più conveniente eseguire con rapidità un hashing del testo da firmare e poi autenticare solo quello piuttosto che eseguire algoritmi complessi di crittografia asimmetrica su moli di dati molto grandi.

La funzione crittografica di HASH deve avere 3 caratteristiche fondamentali:

1. Deve essere estremamente semplice calcolare un HASH da qualunque tipo di stato.

2. Deve essere estremamente difficile o quasi impossibile risalire al testo che ha portato ad un dato HASH.

3. Deve essere estremamente improbabile che due messaggi differenti, anche se simili, abbiano lo stesso HASH.

31) In cosa consiste l'attacco man in the middle, e come può essere evitato?

→ MAN IN THE MIDDLE E PROTOCOLLO

La tipologia di attacco che va sotto il nome di man-in-the-middle consiste nel dirottare il traffico generato durante la comunicazione tra 2 host verso un terzo host (attaccante). Durante l'attacco è necessario far creder ad entrambi gli end-point della comunicazione che l'host attaccante è in realtà il loro interlocutore legittimo.

questo attacco avviene quando si adotta il protocollo ap5.0

1) Alice contatta Bob

2) Bob le manda un nonce R

3) Alice utilizza la chiave segreta per codificare messaggio il nonce, $K^-_A(R)$ e lo ri-invia a Bob

4) Bob si procura la chiave pubblica di Alice K^+_A

5) Bob calcola K^+_A ($K^-_A(R)$) e verifica che

K^+_A ($K^-_A(R)$) =R

Questo protocollo ha dei pro e dei contro:

Pro) i due estremi della comunicazione non condividono segreti.

Contro) Risulta più facile l'intrusione da parte di terzi.

Infatti Trudy può fingere di essere Alice procedendo in questo modo:

1) Trudy contatta Bob

2) Bob manda un nonce R ad Alice e Trudy lo intercetta.

3) Trudy manda K^-_T (R) a Bob

4) Bob richiede la chiave pubblica di Alice K^+_A , ma questa richiesta viene intercettata da Trudy che gli manda la sua: K^+_T

5) Bob senza saperlo calcola $K^+_T(K^-_T(R))$ e verifica che

$K^+_T(K^-_T(R))=R$

La falla del protocollo di autenticazione ap5.0 appena descritta può essere sfruttata da un malintenzionato per interporsi nella comunicazione in modo del tutto trasparente.

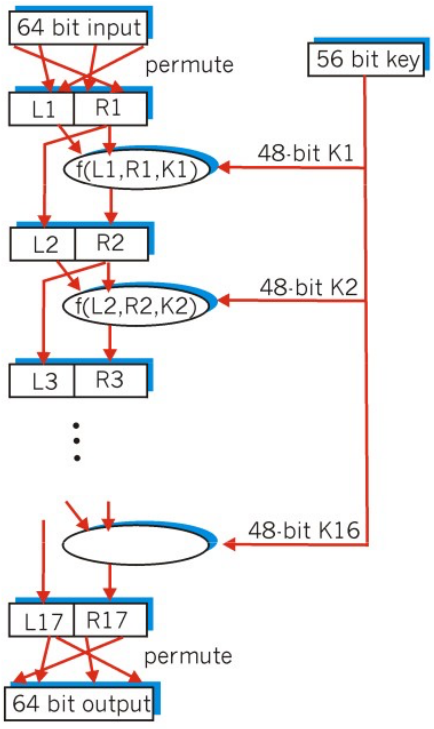
Per Alice e Bob l'autenticazione è andata a buon fine, così credono che ciò che uno spedisce l'altro riceve ma in realtà non è così. Possiamo rendere sicuro il protocollo di autenticazione ap5.0 tramite le chiave certificate.

32) Descrivere graficamente un metodo per il calcolo dell'impronta di un documento che utilizza DES (e calcola impronte di 64 bit). Infine spiegare in dettaglio perché impronte di 64 bit non sono considerate sicure (indipendentemente dal metodo utilizzato per il calcolo).

Il DES è a chiave simmetrica e questa è già una limitazione per le applicazioni di rete, in più, considerando che quello che lavora a blocchi di 64 bit ha una chiave di 56 bit, si capisce facilmente che è soggetto ad attacchi a forza bruta.

Essendo a 56 bit le possibili chiavi sono solo 2^56 = 72'057'594'037'927'936

Quando il DES fu approvato come standard nel 1976, realizzare allora un elaboratore capace di provare tutte le possibili chiavi in un tempo ragionevole sarebbe costato una cifra irragionevole.

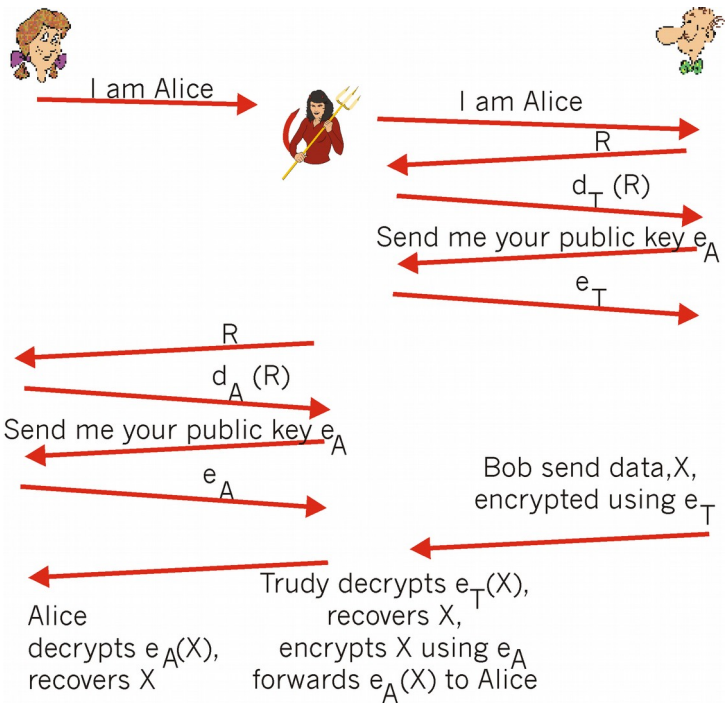


Impronte di 64 bit non sono sicure, perché in 2^33 tentativi si trovano due documenti con la stessa impronta, cosa che non vogliamo assolutamente.

33) Descrivere graficamente un protocollo di autenticazione basato su chiave pubblica che risulta vulnerabile rispetto ad un attacco del tipo "uomo nel mezzo". Il protocollo è più sicuro se l'autenticazione è richiesta da entrambi le parti? Discutere brevemente inoltre le modalità con cui è possibile rendere robusto il protocollo rispetto a questo tipo di attacchi

La tipologia di attacco che va sotto il nome di man-in-the-middle consiste nel dirottare il traffico generato durante la comunicazione tra 2 host verso un terzo host (attaccante). Durante l'attacco è necessario far creder ad entrambi gli end-point della comunicazione che l'host attaccante è in realtà il loro interlocutore legittimo.

La sicurezza del protocollo non sta nel fatto che sia Alice che Bob si inviano un nonce per autenticarsi ed identificarsi, perché il nonce non bast per l'autenicazione perché un esterno può intercettare il nonce e inviare un messaggio chiedendo autenticazione cifrando il nonce con la propria chiave privata, spacciandosi cosi per un'altro. Come esempio qui sotto.



La soluzione è quello di utilizzare chiavi certificate. Utilizzando uno standard X.509 per le infrastrutture a chiave pubblica PKI, il quale definisce i formati standard per i certificati a chiave pubblica.

Per poter autenticare 2 utenti:

- timestamp e il nonce

In questo caso quando Alice vorrà inviare messaggio a Bob dovrà firmare il timestamp (t_A), nonce (r_A) che verrà inviato a Bob

Bob riceverà il messaggio e lo decifrerà utilizzando la chiave pubblica di Alice. A questo punto Bob genererà un nonce (r_B) che andrà a firmare insieme al suo timestamp (t_B) e al nonce ricevuto da Alice (r_A) che inoltrerà ad Alice.

Questo implica che se Trudy usa messaggi di autentica usati precedentemente (replay di messaggi) per sostituirsi ad Alice allora il timestamp è cambiato. Trudy viene bloccata.

Posso anche rafforzare ancora di più l'autenticazione facendo firmare il nonce di Bob ad Alice, fornendo cosi un'ulteriore prova dell'autenticità di Bob

34) 1) Qual'è il vantaggio di usare una Key Distribution Center (KDC) rispetto ad avere una chiave per ogni coppia di utenti?

Quantificare la risposta nel caso di N utenti.

2) Qual è l'informazione deve essere condivisa tra il KDC e ciascun utente?

3) Come si fa a condividere una chiave di sessione usando un KDC?

4) Che differenza c'è tra una Key Distribution Center (KDC) ed una Certification Authority (CA)?

1) Il vantaggio consiste nel fatto che nel KDC ho N chiavi invece per ogni coppia di utenti ho N^2 chiavi

2) La chiave sessione condivisa, senza di essa ciascun utente non è in grado di comunicare con l'altro e bisogna per forza passare per il KDC per ottenerla. Tramite questa chiave segreta condivisa gli utenti sono in grado di comunicare in segretezza con KDC e tra di loro.

3) Un utente condivide una chiave segreta ($K_{A\text{-}KDC}$) con KDC che può utilizzare per dire con chi vuole comunicare. Per esempio abbiamo che Alice vuole comunicare con Bob. Alice chiede al KDC che vuole comunicare con Bob, il KDC invia richiesta di una chiave di sessione segreta. Generata la chiave di sessione la invia ad Alice, in particolare Alice riceve R (chiave di sessione) e la coppia cifrata ($K_{B\text{-}KDC}(A,R)$) con la chiave segreta condivisa tra Bob e KDC($K_{B\text{-}KDC}$) identità e la chiave di sessione (R).

Alice decifra il messaggio, memorizza la chiave di sessione R ed invia $K_{B\text{-}KDC}(A,R)$ a Bob

Bob decifra il messagio di Alice,memorizza la chiave di sessione R e l'identità di Alice.

4) *Key Distribution Center KDC*, i centri per la distribuzione delle chiavi effimere. Grazie al KDC, ogni nodo della rete protetta deve conoscere soltanto le credenziali proprie e del KDC, mentre il KDC stesso si incarica di tener traccia delle credenziali di tutti i nodi della rete protetta. Questo fa sì che il KDC sia un elemento fondamentale della rete, deve essere un nodo di fiducia per tutti coloro che si attestano alla rete, perché è l'unico vero intermediario tra i nodi deputato all'autenticazione di ogni singolo nodo e alla comunicazione delle chiavi temporanee.

Le *Certification Autority, CA* rispondono alla necessità di autenticare le chiavi pubbliche: il concetto è simile al KDC, ma si

opera nel mondo della crittografia asimmetrica. La CA si pone come un intermediario di fiducia tra Alice e Bob, che certifica l'autenticità d l'integrità delle loro chiavi pubbliche, in modo tale che essi siano in grado di riconoscersi senza ombra di dubbio.

35)Un compratore si rivolge ad un sito di commercio elettronico. Quali strumenti garantiscono il compratore dell'identità del sito di commercio elettronico e viceversa? Quale protocollo permette di mantenere il numero di carta di credito del compratore ignoto al sito di commercio elettronico?

Si potrebbe utilizzare la certificazione delle chiavi e i nonce, il compratore manda un nonce r1 con la chiave pubblica certificata al sito di commercio elettronico. Il sito cifra r1 e genera il nonce r2 che invia al compratore con la chiave pubblica certificata del compratore. Quando il compratore riceve K(r1,r2) risalendo a r1 autentica il sito. Il sito, con le stesse modalità risale a r2 e autentica il compratore. Il protocollo per mantenere ignoto il numero della carta di credito al sito di commercio elettronico è SET, si basa su SSL.

36) Si illustrino varie modalità di concordare su una chiave segreta di sessione nel caso di

- **i) un sito di commercio elettronico;**
- **ii) all’interno di una rete locale;**
- **iii) tra due router IPSEC.**

i) Una volta che vi è sicuri dell’idnetita dall’altra parte si può scambiare la chiave segreta di sessione utilizzando la chiave pubblica.

ii) Utilizzare KDC

iii) Metodo public Infrastructure

37) 1)llustrare e motivare un esempio in cui è utile calcolare il digest (l'impronta) di un messaggio.

2)Spiegare per quale motivo l'utilizzo del checksum (controllo di parità) usato dal protocollo IP non è una buona scelta per il calcolo del digest di un pacchetto.

1) Calcolare il digest risulta utile per quando si vuole verificare

integrità di un messaggio e l'autenticazione dell'utente. Utilizzo le funzioni hash sul messaggio, che per esempio Alice vuole inviare a Bob. H(m) è il risultato dalla funzione hash che verrà poi cifrato con la chiave privata di Alice, applicando cosi una firma. Invio il digest cifrato e il messaggio m di Alice che verranno cifrate poi con la chiave segreta condivisa K_S. Alla quale concatenerò la codifica della chiave simmetrica K_S con la chiave pubblica di Bob.

Quando Bob riceverà il messaggio, utilizzerà la sua chiave privata per ottenere la chiave segreta condivisa K_S che userà per ottenere la firma digitale e messaggio m ancora cifrati . Con la chiave pubblica di Alice otterà il H(m) e il messaggio m.

Infine Bob confronterà la funzione hash H(m) del messaggio e quella ottenuta con la sua funzione hash su m, se coincidono allora si ha la che il messaggio proviene veramente da Alice e che nessun intuso ha modificato parti del messaggio.

2) Il checksum usato dal protocollo IP non è sicuro in quanto è possibile che diverse stringhe abbiano come risultato lo stesso output, cosa che invece non succede con le funzioni hash.

38) Fornire una spiegazione intuitiva dei motivi che sono alla base della sicurezza della crittografia a chiave asimmetrica e motivare anche intuitivamente per quale motivo la crittografia a chiave asimmetrica è estremamente più lenta di quella a chiave simmetrica.

Nella crittografia asimmetrica è più difficile far si che si identifichino le fue chiavi perché, come nel caso dell'algoritmo RSA, bisognerebe fattorizzare numeri molto grandi questo implica che sia dispendiosa dato che decriptare numeri molto grandi comporta impiegare più tempo per decrptarli, quindi risulta essere più lenta.

Quindi la lentezza della crittografia a chiave asimmetrica sta nel fatto più le chiavi sono grandi più la crittografia a chiave asimmetrica sarà lenta, dovuta alla complessità che si va a generare dal momento in cui vado ad applicare operazione, anche semplici su numeri relativamente molto grandi.

39) Illustrare le principali caratteristiche di un sistema di impronta digitale ed un possibile attacco alla sua sicurezza.

La firma digitale garantisce integrità del messaggio ed

autenticazione dell’utente.

Supponiamo che Bob voglia inviare un messaggio ad Alice. Oltre al messaggio m, le invia un digest cifrato calcolato applicando prima la funzione hash sul messaggio m e poi utilizzando la chiave privata di Bob. Firma digitale ($K^{-}_{B}(H(m))$.

Alice riceve (m, $K^{-}_{B}(H(m))$ e per prima cosa applica la chiave pubblica di Bob sul digest cifrato per ottenere una hash(digest), H(m). Poi impiegherà una funzione hash al messaggio m in modo da porcurarsi una seconda hash che andrò a confrontare con la prima e se coincidono Alice può stare tranquilla sull’integrità del messaggio e Bob sull’autenticità di Alice.

Un possibile attacco da parte di Trudy consiste nell’andare a replicare un determinato messaggio, che però può essere risolto con l’aggiunta del Nonce.

40) Assumi di avere 2 file, uno di piccole dimensioni ed uno molto grande. Dovendo scambiare questi file in modo sicuro con un qualunque protocollo sincrono, che tipo di crittografia (simmetrica, asimmetrica o combinazione di esse) useresti e perchè? Qualora il protocollo fosse invece asincrono cambierebbe qualcosa? Argomentare le risposte.

La chiave simmetrica è più facile da generare e da decifrare rispetto alla chiave asimmetrica. Per un file di grandi dimensioni se si utilizza un protocollo sincrono dove devi attendere la risposta, è essenziale la velocità, ecco per cui si preferisce utilizzare la crittografia a chiave simmetrica.

Per quanto riguarda il file di piccole dimensioni, se la dimensione è relativamente piccola da non andare a creare problemi sulla velocità in un protocollo sincrono, allora potrò andare ad utilizzare qualsiasi tipo di crittografia (simmetrica, asimmetrica, combinazione di esse).

Invece se vado ad utilizzare protocollo di tipo asincrono, dove non è necessaria l’attesa di risposta da parte di un altra persona (per esempio le code AMQP) puoi utilizzare o la crittografia asimmetrica,simmetrica oppure la combinazione tra le due per garantire autenticazione, integrità e quindi una maggiore sicurezza. Questo perché non hai la necessità di avere un riscontro immediato.

41) Un tradizionale filtro di pacchetti senza memoria di stato può filtrare pacchetti basandosi sui bit di flag TCP o su altri campi dell'intestazione. Vero o falso?

Vero, se per esempio un'organizzazione non vuole alcuna connessione TCP entrante, se non quellle del proprio web server pubblico, può bloccare tutti i segmenti TCP SYNC in ingresso (dove SYN è un bit di flag), tranne quelli con porta destinazione 80 e indirizzo IP di destinazione corrispondente a quello del web server.

42) In un filtro di pacchetti tradizionale ogni interfaccia può avere la propria lista di controllo degli accessi. Vero o falso?

Vero,ogni interfaccia di router possiede una tabella di controllo degli accessi composta da una *azione,* permette di far passare o bloccare quel pacchetto, *indirizzo sorgente*, *indirizzo destinazione, protocollo, porta sorgente, porta destinazione, bit di flag*.

43) Perchè un gataway applicativo deve lavorare insieme al filtro di un router per essere efficace?

La configurazione del filtro del router blocca tutti i collegamenti eccetto quelli che riportano l'idirizzo IP del gateway, questo vuol dire che tutte le connessione di un certo tipo dovranno passare attraverso il gateway.

44) Supponiamo di dover firmare dei documenti che hanno esattamente le dimensioni delle stringhe prodotte dalla funzione di hash adottata nella firma. Come è possibile modificare il processo di firma in modo da renderlo più veloce?

Si potrebbe velocizzare il processo di firma firmando solamente una parte del documento piuttosto che l'intero. Poiché comunque per parti di messaggio m e m' sarà computazionalmente difficile riprodurre H(m)=H(m') da parte di un intruso.