

AGGRESSION

ASSET : risorse da proteggere da eventuali minacce e attacchi

OWNER: colui che possiede le risorse e le vuole proteggere

AGENTI DI MINACCIA: interagiscono con le risorse e vogliono alterarle

nel caso delle banche l'OWNER sono banca e cliente, gli agenti di minaccia sono banditi criminali o attivisti.

RISCHIO : probabilità di avvenire situazione come "dipendente riceve post-it con la password e qualcuno le legge"

$$R_i = P_i \cdot I_i \rightarrow \begin{matrix} \text{IN PATTI:} \\ \text{per esempio quantità di rubati} \end{matrix}$$

CONTROMISURE DI PROTEZIONE

PREVENZIONE

cercare di evitare breccie di sicurezza

CONFIDENZIALITÀ: alcune informazioni devono essere lette da solo un gruppo ristretto di persone. Per questo esistono i controlli di accesso per evitare lettura non autorizzata

PRIVACY: è la confidenzialità per gli individui, mentre la confidenzialità delle aziende è detta SECRECY

ANONIMITÀ: mantenere l'identità di un individuo privata

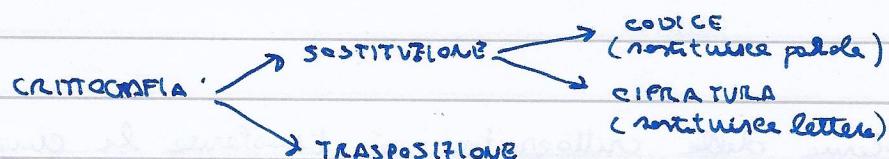
SECURITY POLICY: ci dice chi può accedere ai nostri dati

CRITTOGRAFIA

CRITTOLOGIA: studio delle scritture segrete

STEGANOGRAFIA: tecnologie di nascondere messaggi in altri messaggi

CRITTOGRAFIA: scienza delle scritture segrete

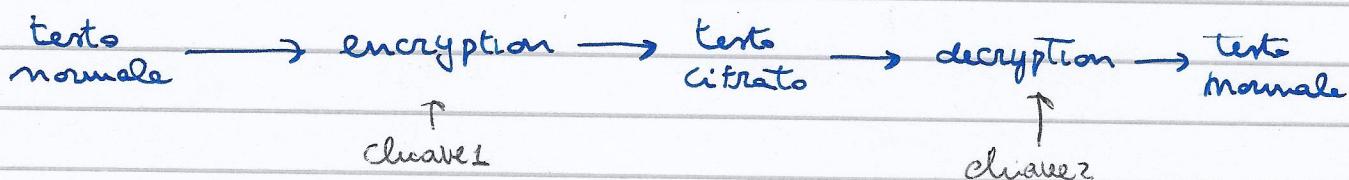


Un esempio di steganografia consiste nel nascondere un messaggio nei bit meno significativi di un'immagine

CANALE di comunicazione sicuro

- > confidenzialità: l'informazione rimane segreta
- > integrità: l'informazione non viene alterata durante la trasmissione
- > autenticazione: i principali sanno con chi stanno parlando

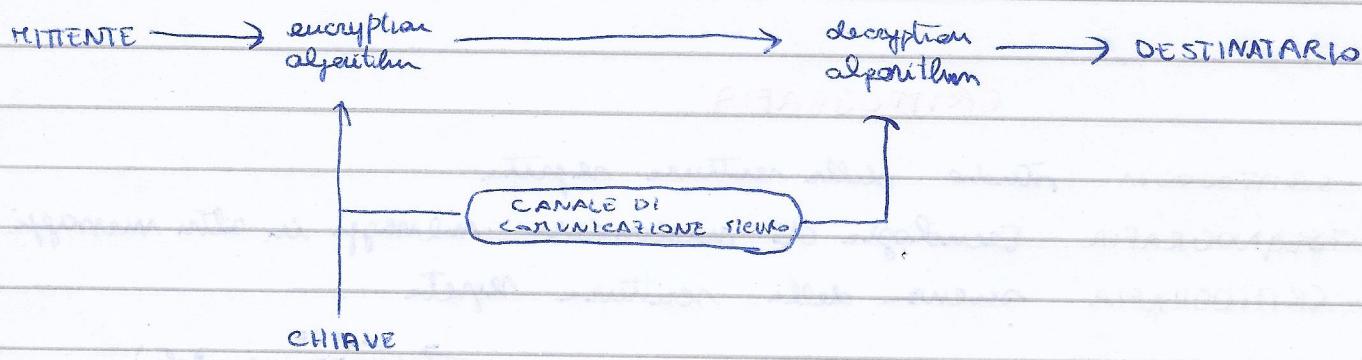
SCHEMA CRIPTOGRAFIA



La sicurezza dipende dal tipo di chiave

ALGORITMO SIMMETRICO: CHIAVE₁ = CHIAVE₂

ALGORITMO ASIMMETRICO: chiavi diverse



Il problema delle crittografie è trasferire le chiavi al destinatario.

Se un attaccante intercette la comunicazione della chiave è un problema

TIPI DI SICUREZZA

SICUREZZA INCENDIZIONALE

Il sistema è sicuro anche se l'attaccante ha potere di calcolo infinito. Utilizzando la TEORIA DELL'INFORMAZIONE creiamo un testo cifrato che non contiene abbastanza informazioni per ricevere il testo normale.

SICUREZZA CONDIZIONALE

Il sistema non può uscire facendo molti tentativi ma le potenze di calcolo reale sono finite e limitate. Il livello di sicurezza si riduce con le teorie della complessità.

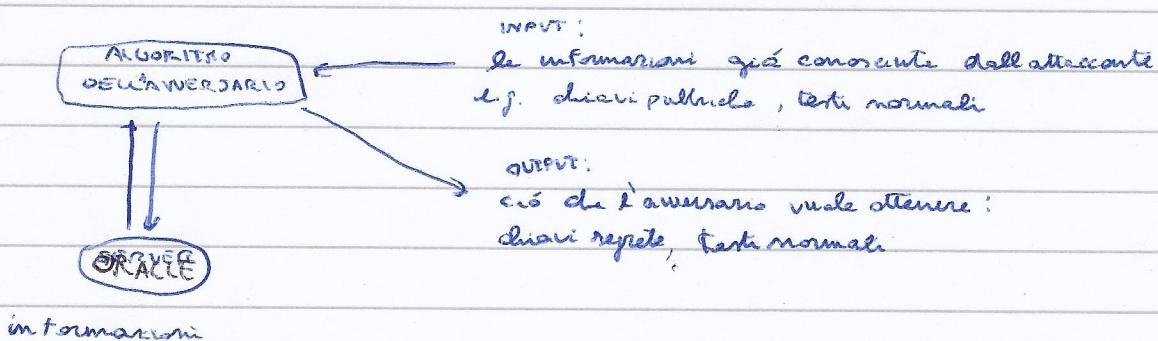
CRPTOANALISTI

È la scienza che permette di ottenere il testo normale del testo cifrato senza conoscere le chiavi.

METODI UTILIZZATI Attacco Criptanalitico, Attacco Forza Bruta.

ATTACCO CRIPTANALITICO

L'attaccante deve conoscere l'algoritmo di crittazione usato.



~~16 - TIPI DI ATTACCO~~

conoscendo solo il testo cifrato

$$\boxed{C_1 - E_K(M_1) \mid C_2 - E_K(M_2) \mid C_3 - E_K(M_3) \mid \dots \mid C_m - E_K(M_m)}$$

bisogna dedurre

$$\boxed{M_1 \mid M_2 \mid M_3 \mid \dots} \quad \text{testo normale}$$

oppure dedurre un algoritmo che permetta di calcolare M_{m+1} da $C_{m+1} = E_K(M_{m+1})$

conoscendo solo il testo normale:

$$\boxed{M_1 \cdot C_1 = E_K(M_1) \mid M_2 \cdot C_2 = E_K(M_2) \mid \dots}$$

bisogna dedurre una chiave inoltre oppure dedurre l'algoritmo

di calcolo M_{m+1} da $C_{m+1} = E_K(M_{m+1})$

DEFINIZIONE DI SICUREZZA

- specifica un attacco (tipo di attacco)
- specifica di cosa ha bisogno l'attaccante per vincere
- il sistema è sicuro se un attacco avviene con bassa probabilità

IS - MATEMATICA

A: alfabeto, insieme finito di simboli

~~Ma~~ A^* insieme di possibili messaggi

M: ^{messaggio} testo normale (plain text) è un ~~ad~~ elementi di A^*

C: insieme dei possibili testi cifrati. I testi cifrati possono avere alfabeto diverso dal testo normale.

K: insieme delle possibili chiavi

insieme
chiavi

insieme
messaggi
criptati

ogni chiave è appartenente a K determina una funzione biuniva: $M \rightarrow C$

E_k è della funzione di encryption (o funzione di trasformazione)

ogni chiave d appartenente a K determina una funzione biuniva: $C \rightarrow M$

D_d è la funzione di decryption

ENCRYPTION: applicare E_k

DECRYPTION: applicare D_d

SCHEMA DI ENCRYPTION aka CIPHER aka CIFRARIO

Un cifrario ^{contiene} è ~~so~~ un insieme di funzioni $E_k: e \in K$ (encryption) e un insieme di funzioni $D_d: d \in K$ (decryption)

NB a ogni $e \in K$ corrisponde una unica $d \in K$ (decryption è una funzione inversa)

$$D_d = E_e^{-1}$$

NB $D_d(E_e(m)) = m$

testo
decifrato
cioè
normale

~~decifrato~~ $E_e(m) =$ testo
cifrato

NB chiave e et chiave d formano una coppia di chiavi (e, d)

Per costruire il CIFRARIO dobbiamo fissare

- insieme dei messaggi M
- insieme dei messaggi cifrati C
- insieme delle chiavi K
- funzioni encryption $E_k: e \in K$
- funzioni decryption $D_d: d \in K$

ESEMPIO CIFRARIO

$M = \{m_1, m_2, m_3\}$ ci sono $m! = 3! = 6$ corrispondenze biunivoci da $M \rightarrow C$.

Lo stesso delle chiavi $K = \{1, 2, 3, 4, 5, 6\}$

specifico queste trasformazioni:

CHIAVE $k=1$

E_1	E_2	E_3	E_4	E_5	E_6
$m_1 \rightarrow c_1$	$m_1 \rightarrow c_1$	$m_1 \rightarrow c_2$			
$m_2 \rightarrow c_2$	$m_1 \rightarrow c_3$	$m_2 \rightarrow c_1$			scatena
$m_3 \rightarrow c_3$	$m_3 \rightarrow c_2$	$m_3 \rightarrow c_3$			

(chiave $k=2$)

(chiave $k=3$)

Supponiamo che Roberto e Alice decidano di usare la funzione di encryption E_3 , allora per cifrare m_1 , Alice lo trasforma in c_2

$$E_3(m_1) = c_2$$

Adesso Roberto riceve messaggio c_2 e deve decifrare quindi fa funzione decryption

$$D_3(c_2) = m_1$$

ENCRYPTION CON CHIAVE SIMMETRICA

il cifrario è i \leftrightarrow a chiavi simmetriche se per ogni coppia (e, d)

è computazionalmente facile determinare d conoscendo solo e (^{harder})

^{SPESO}
~~impratica~~ $e = d$

CIFRARIO A BLOCCI

divide il messaggio normale in blocchi di lunghezza t e li encrypta uno alla volta

CIFRARIO A SOVRIMENTO

cifrario a blocchi in cui i blocchi non devono avere lo stesso lunghezza (CESARE)

CODICI

operano su parole di lunghezza variabile

esempio codice

parola codice

the	1701
secret	5603
mischiefs	4008
that	3790
I	2879
set	0524

→ 5603 2879 0524 ...

secret I set ...

TECNICHE DI SOSTITUZIONE

CIFRARI CON SOSTITUZIONE SEMPLICE

Cifrario di Cesare: ogni carattere del messaggio normale viene sostituito col carattere in tre posizioni avanti

KHOOR ZRUOG → HELLO WORLD

Cifrario ROT13: ogni carattere del messaggio viene sostituito col carattere 13 posizioni più avanti

Cifrario Alfanumerico: sostituendo i caratteri lettere con caratteri numerici

CRYPTANALISI: per decifrare un messaggio di Cesare, n'è
basta la forza bruta finché non riceve un messaggio sensato.

P H H W P H O I W H U

oggv of chvgt

mttu nf bgufs

meet me after

:

:

L'attaccante assume che il testo cifrato si derivi da una sostituzione.
 K è l'insieme delle permutazioni sull'alfabeto A .
ad ogni $\epsilon \in K$ associamo una funzione di trasformazione E_ϵ (encryption)

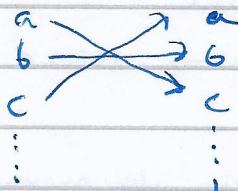
GENERALIZZAZIONE

un cifrario affine è un tipo di CIFRARIO A SOSTITUZIONE MONOALFABETICA

ENCRYPTION $e(m) = (a \cdot m + b) \text{ mod } (\frac{\text{dimensione}}{\text{alfabeto}})$ chiave = (a, b)

DECRIPTION $D(c) = a^{-1} (c - b) \text{ mod } (\frac{\text{dimensione}}{\text{alfabeto}})$

ogni lettera è mappata con un'altra lettera tramite funzione biunivoca



La dimensione dello spazio delle chiavi possibili (K) ha grandi dimensioni: $m!$ e.g. l'alfabeto inglese $26!$

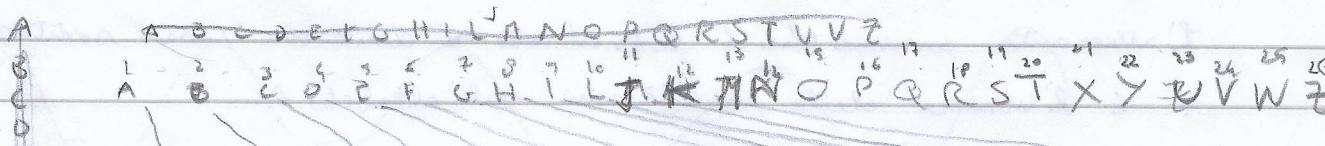
Le funzioni di cifratura deve essere iniettive, in questo modo Bob può decifrare in modo univoco il testo cifrato. Le chiavi (a, b) deve essere scelte in modo che la funzione risulti iniettive

esempio

chiave: $(2, 0)$ quindi $\text{funt.} = e_K(x) = 2x \text{ mod } 26$
 a, b

per decifrare il messaggio usiamo la funzione inversa

$$2x = 0 \text{ mod } 26$$



IP(R) $e_K("N") = 2 \cdot 14 \text{ mod } 26 = 28 \text{ mod } 26 = \frac{28}{2} = \frac{26}{1} = 2$

CIFRARIO DI VIGENÉR

M: DEFEND THE EAST WALL OF THE CASTLE
K: FORTIFICATION FORTIFICATION FO

	A	B	C	D	E	F	
A	B	A	B	C	D	E	F
C	D	B	C	D	E	F	G
E	F	C	C	D	E	F	G
G	H	D	D	E	F	G	H
I	J	E	F	G	H	I	J
K	L	F	F	G	H	I	J

CIFRARIO DI VERNAN

Per la cifratura viene creata una chiave casuale lunga come il testo in chiaro.

Dopo la creazione delle chiavi, il testo in chiaro e le chiavi vengono sommati come nel cifrario di Vigenére. Non si sommano gli ordinali delle lettere, ma i codici ASCII.

Questo cifrario non si us. perché le chiavi dovrebbero essere grande come il file da proteggere.

CIFARIO XOR

C (messaggio, chiave) = messaggio \oplus chiave = messaggio cifrato

D (messaggio cifrato, chiave) = messaggio cifrato \oplus chiave = messaggio

Se la chiave è più corta del messaggio, le chiavi vengono duplicate come nel cifrario di Vigenere.

m = "CIAO" =

0110 001

0110 111 +

K =

0111 1000

0111 1000 =

cifrato =

0001 1001

0001 0111 +
0111 1000 =
0110 1111

Riottenendo il testo in chiaro

cifrato =

0001 1001

0001 0111 +

K =

0111 1000

0111 1000 =

0110 0001

0110 1111

CRITTOGRAFIA SIMMETRICA

CIFRARI A FLOW vs CIFRARI A BLOCCI

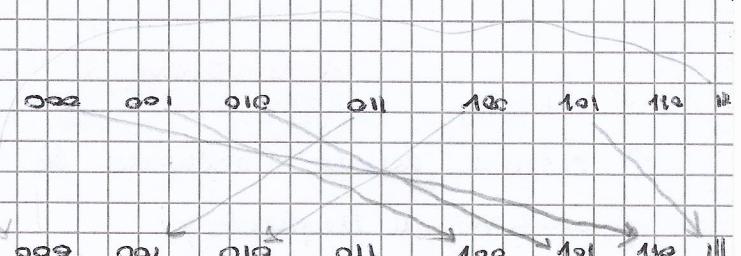
- I cifrari a blocchi dividono il messaggio in blocchi segnati egnuno dei quali viene criptato / decriptato
- I cifrari a flusso dividono il messaggio in singoli bit o byte durante la criptazione / decriptazione

CIFRARIO A BLOCCI IDEALE

se contiene blocchi da m bit allora

- > Ci sono una tabella di 2^m righe
- > Le chiavi ha dimensione $m \cdot 2^m$
- > ~~ci sono~~ $2^m!$ funzioni di encryption possibili per questo cifrario

PLAINTEXT	CIPHERTEXT
000	110
001	100
010	101
011	001
100	010
101	111
110	011
111	000



E' una delle $2^{3!}$ possibili funzioni di encryption (funzioni che trasformano)

STRUTTURA DI FEISTEL

Il cifrario ha una chiave di k bit
e la dimensione dei blocchi pari a m bit

c'è sono 2^m funzioni da ~~la~~ encryption (trasformazione) possibili

La maggior parte dei cifrari simmetrici sono basati su queste idee.

CIFRARI A SOSTITUZIONE-PERMUTAZIONE DI SHANNON

Sono la base dei cifrari a blocchi moderni.

I) cifrari SP si basano su due operazioni di crittografia

SOSTITUZIONE (S-Box)

confonde i bit in input

PERMUTAZIONE (P-Box)

distende i bit attraverso gli input del S-Box

CONFUSIONE E DIFFUSIONE

1) cifrari devono oscurare le proprietà statistiche del messaggio
(andare più frequenti)

DIFFUSIONE: rompe le strutture statistiche del messaggio

CONFUSIONE: rende la relazione tra testo crittato e la chiave più complesse possibili

APPROFONDIMENTO FEISTEL

È un cifrario a blocchi
~~e per bit~~

Il processo avviene in vari "round"

implementa il cifrario di SHANNON con P-Box e S-Box

Cifratura e decifratura sono operazioni identiche

F è la funzione dei paraggi

k_0, k_1, \dots, k_m le rotellature

left right

(i) Dividi i dati in ingresso in due parti uguali, (L_0, R_0)

(ii) Per ogni round $i=1, 2, \dots, m$ calcola

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, k_{i-1})$$

↓
funzione del Round

rotellatura
delle serrature

Alla fine si ottiene il testo critto (L_m, R_m)

In decifratura si ottiene così

$$R_{i-1} = L_i$$

$$L_{i-1} = R_i \oplus f(L_i, k_i)$$

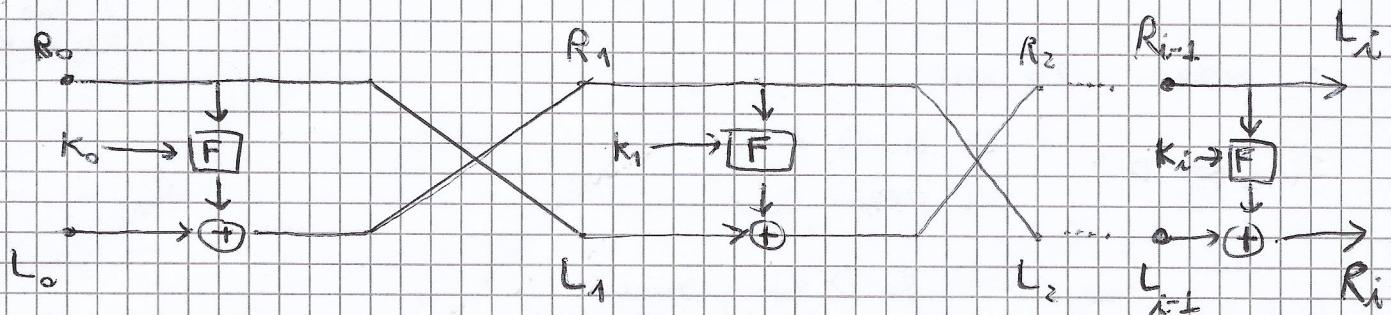
PROPRIETÀ DI \oplus

$$x \oplus 0 = x$$

$$x \oplus x = 0$$

$$x \oplus y = y \oplus x$$

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z$$



RETE DI FEISTEL

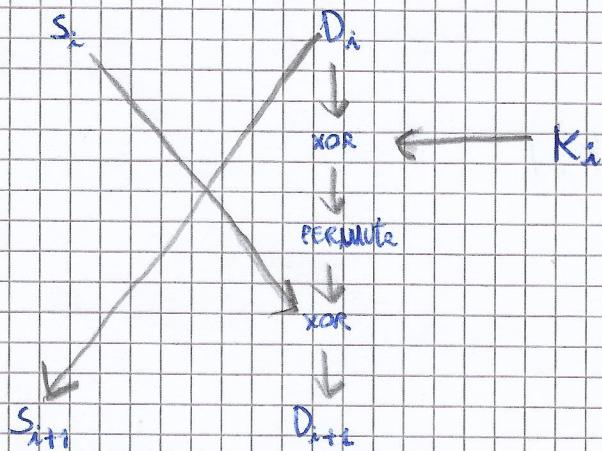
La rete di Feistel (1973) è un cifrario che mescola traspositioni e sostituzioni a livello di bit, non a livello di carattere del messaggio.

Non si usa da solo ma come componente di cifrari complessi (DES)

	C	H	I	...
<u>MESSAGGIO</u>	01000011	01101000	01101001	...
<u>CHIAVE</u>	0100.1110			

Algoritmo prevede di dei blocchi in cui si divide in due parti uguali il blocco da cifrare e scambiarli.

Una dei due blocchi viene anche sottoposto a permutazioni e sostituzioni quando lo sottoscrive



esempio

0100.0011 primo blocco da cifrare

$$S = 0100 \quad D = 0011$$

$S_{i+1} \leftarrow D$
D diventa la S
del prossimo ciclo

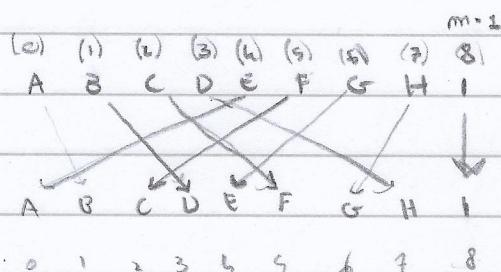
$D_{i+1} \leftarrow$
La D del prossimo
ciclo si ottiene con
un insieme di trasposizioni
e sostituzioni

ADDITIONE +
XOR TRA
sostituzione K_i e D_i

CIFRARIO AFFINE ESERCIZIO ESAME

E : carattere x → carattere $E(x)$

$$E(x) = (ax + b) \bmod m$$



$$E((2x+1) \bmod 9)$$

$$E(E) = 9 \bmod 9 = 0$$

$$E(F) = 11 \bmod 9 = 2$$

$$E(G) = 13 \bmod 9 = 4$$

$$\begin{array}{r} 13 \\ 9 \\ \hline 4 \end{array}$$

$$\begin{array}{r} 17 \\ 9 \\ \hline 8 \end{array}$$

$$E(17) = 17 \bmod 9 = 8$$

- > (a, b) è la chiave della crittografia
- > a ed m sono primi fra loro

DIMOSTRA ➔ L'ALGORITMO DECRIPTATIVO È INVIA

$$D(E(x)) = a^{-1} (E(x) - b) \bmod m$$

DATI: a^{-1} è un inverso moltiplicativo di a

$$\text{quindi } a \cdot a^{-1} = 1 \bmod m$$

$$\begin{aligned} & a^{-1} (E(x) - b) \bmod m = \\ & = a^{-1} ((ax + b) \bmod m - b) \bmod m = \end{aligned}$$

CIFRARIO A PERMUTAZIONE

In generale i cifrari prevedono una sostituzione in cui i caratteri del testo in chiaro sono sostituiti con caratteri diversi.

Invece un cifrario a permutazione prevede di modificare il testo in chiaro cambiando le posizioni dei caratteri.

$$\text{cioè } C = (Z_2)^m$$

ma K numero di tutte le permutazioni $\{1, \dots, m\}$

per una chiave K definiamo le funzioni di cifratura come:

$$e_K(x_1, \dots, x_m) = (x_{K[1]}, x_{K[2]}, \dots, x_{K[m]})$$

funzione decriptura:

$$d_K(y_1, y_2, \dots, y_m) = (y_{K^{-1}[1]}, y_{K^{-1}[2]}, \dots, y_{K^{-1}[m]})$$

esempio

$m = \text{"ATTACCO ALL ALBA X"}$

chiave = $K = (2, 3, 1)$ \Rightarrow quindi $K(1) = 2$

$$K(2) = 3$$

$$K(3) = 1$$

La chiave reale decide:

ordinare il testo in chiaro in gruppi di tre caratteri

ogni gruppo la prima lettera occuperà il secondo posto

le seconde lettere occuperà il terzo posto

la terza lettera occuperà il quinto posto

ATT ACC OAL ---

TAT CAC LOA ---

ci sono $m!$ possibili chiavi!! pari al numero di
possibili permutazioni del messaggio

date le permutazioni k e il messaggio $\{1, \dots, m\}$
possiamo definire la matrice di permutazioni $K_k = (k_{i,j})$
di dimensioni $m \times m$

$$k_{i,j} = \begin{cases} 1 & \text{se } j = k(i) \\ 0 & \text{altrimenti} \end{cases}$$

CIFRARLO A FLUSSO

Le caratteristiche dei cifrari a blocchi è il fatto di cifrare gli elementi del testo in chiavi mediante le stesse chiavi k

$$y = y_1 y_2 = e_k(x_1) e_k(x_2) \dots$$

Un approccio alternativo consiste nell'usare i cifrari a flusso
l'idea è generare una sequenza detta "flusso di chiavi"

$$z = z_1 z_2 \dots$$

e utilizzarla per estrarre le stampa

$$x = x_1 x_2 \dots$$

QUINDI

$$y = y_1 y_2 = e_{z_1}(x_1) e_{z_2}(x_2) \dots$$

dove

$$z_i = f_i(k, x_1, x_2, \dots, x_{i-1})$$

l'elemento z_i della sequenza è utilizzato per estrarre x_i , con $y_i = e_{z_i}(x_i)$

PER DECIFRARE $y_1 y_2 \dots$

bisogna calcolare $z_1, x_1, z_2, y_2 \dots$

ESEMPIO

$$y_i = (x_i + z_i) \bmod 26 \quad \begin{matrix} \text{FUNZIONE} \\ \text{CIFRATURA} \end{matrix}$$

$$x_i = (y_i - z_i) \bmod 26 \quad \begin{matrix} \text{FUNZIONE} \\ \text{DECIFRATURA} \end{matrix}$$

$$e_{z_i}(x_i) = (x_i + z_i) \bmod 2$$

Se detuiamo chiavi

e decifratute in termini binari
 $k = z_2$

$$d_{z_i}(y_i) = (y_i - z_i) \bmod 2$$

Osserviamo in questo modo l'addizione modulo 2 corrisponde alle XOR

$$z_{i+m} = \sum_{j=0}^{m-1} (c_j z_{i+j}) \bmod 2$$

dove $c_0, \dots, c_{m-1} \in \mathbb{Z}_2$
sono costanti predeterminate

In modo per generare le seguenti "flessi di chiavi" a partire da
valori k_1, \dots, k_m e scrivere valori per z_1, \dots, z_m come
 $= k_i$ per ogni i da 1 a m ; e nel calcolare i valori successivi
base ad una relazione di ricorrenza lineare grado m

esta ricorrenza è lineare perché z_{i+m} è funzione lineare dei termini precedenti
che grado m perché ciascun termine dipende dagli m precedenti

Esempio 4

Allora $m = 4$ e il "flesso di chiavi" è così generato

$$z_{i+4} = (z_i + z_{i+1}) \bmod 2 \quad \text{base } 111$$

Esempio consideriamo i valori iniziali: 1, 0, 0, 0

CIFRARIO DES

È un cifrario simmetrico a blocchi quindi il messaggio viene diviso in blocchi di m bit e i blocchi vengono crittati indipendentemente l'uno dall'altro la chiave è lunga 56 bit

STORIA:

DES pubblicato nel 1977

DES viene ridotto nel 1998

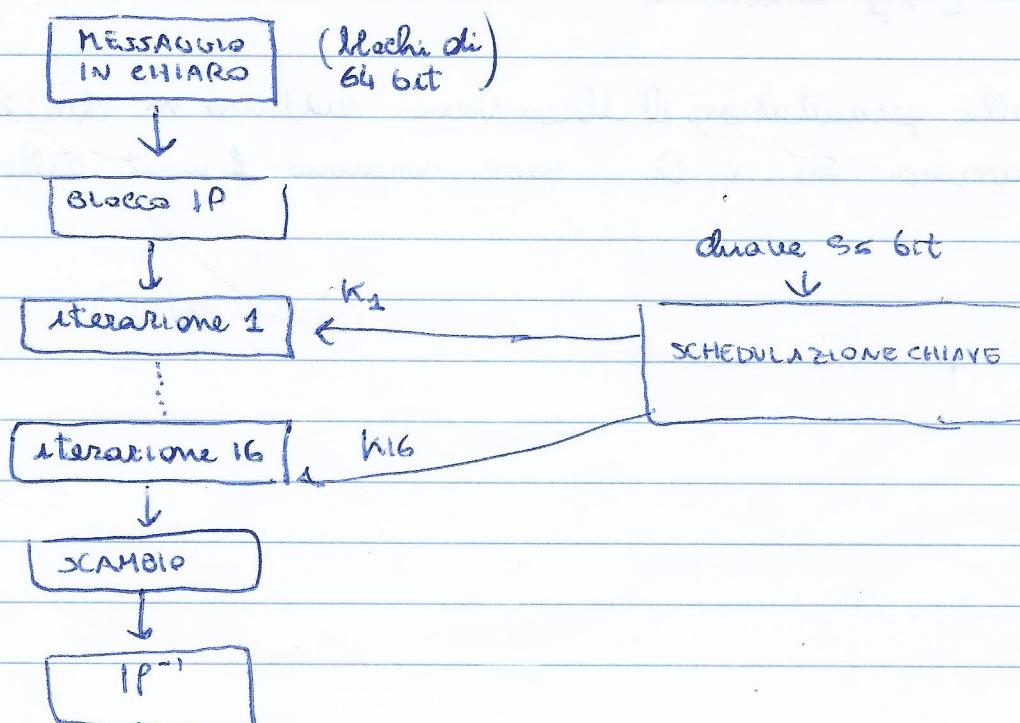
CARATTERISTICHE DEL DES

DIFFUSIONE: alterare la statistica del testo per evitare analisi di frequenze

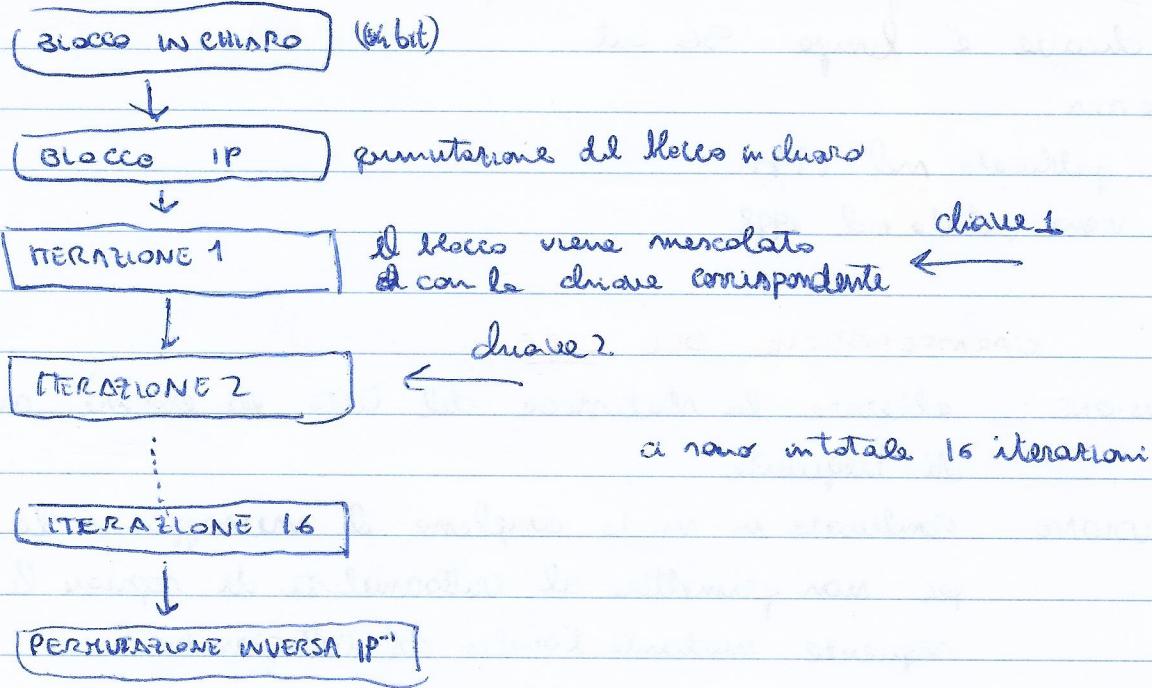
CONFUSIONE: Combinare in modo complesso il messaggio e le chiavi per non permettere al crittoanalista di ripetere le due sequenze mediante l'analisi del cattogramma

questi due proprietà sono realizzate attraverso una serie di permutazioni e combinazioni del messaggio con le chiavi

STRUTTURA DEL DES



- Un blocco di testo in chiaro (64 bit) viene permesso dal blocco IP
Il risultato di queste permutazioni andrà in input alla prima iterazione



PERMUTAZIONE INIZIALE BLOCCO IP

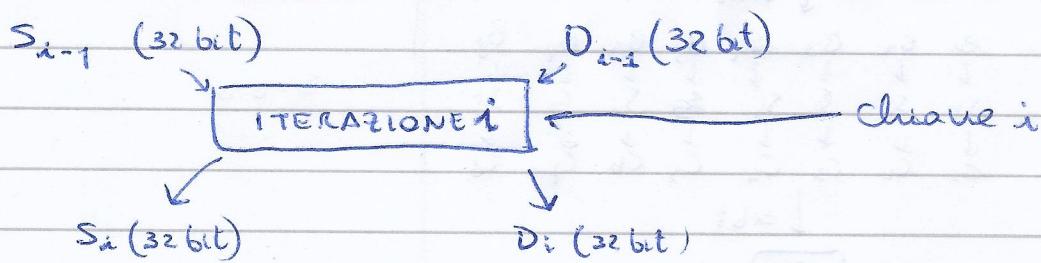
- I bit del blocco vengono scambiati in base alle matrici di permutazione IP : ovvero in posizione [1,1] andrà il bit in posizione 58 del testo in chiaro , ~~il secondo bit~~
- in posizione [1,2] andrà il 59 e così via

Al termine delle permutazioni il blocco viene suddiviso in due sottoblocki da 32 bit chiamati S_0 e D_0 , questi saranno l'input della prima iterazione

IP: permutazione iniziale

(STAMPARE)

ITERAZIONE 1

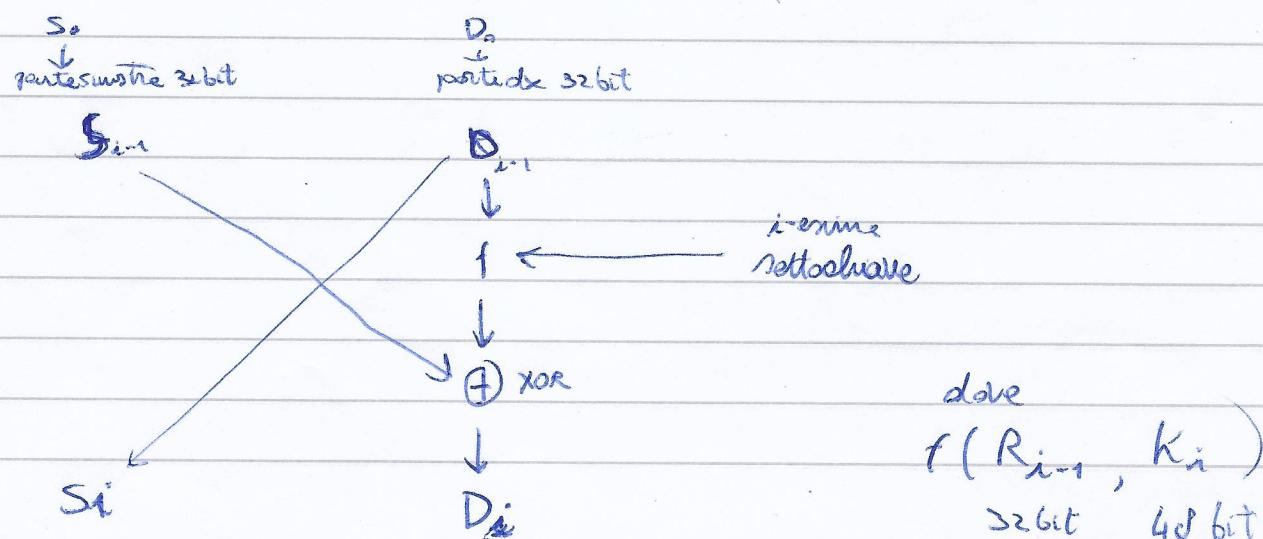


Ogni iterazione ha la seguente struttura:

INPUT: due blocchi S_{i-1} e D_{i-1} da 32 bit, una chiave K_i da 48 bit?

ELABORAZIONE: combinazione di S_{i-1} e D_{i-1} con K_i

OUTPUT: due nuovi blocchi S_i e D_i da 32 bit



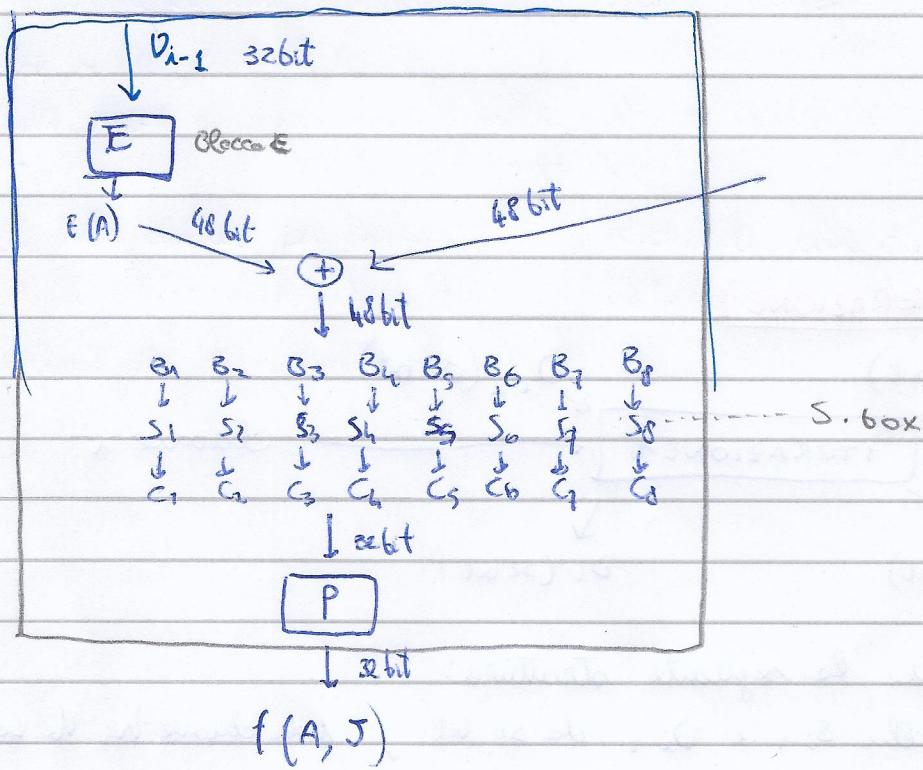
ELABORAZIONE DELL'ITERAZIONE

- Il nuovo blocco S_i è il vecchio blocco D_{i-1}
- Il nuovo blocco R_i è ottenuto mettendo in XOR il vecchio blocco S_{i-1} con l'output della funzione f

↓

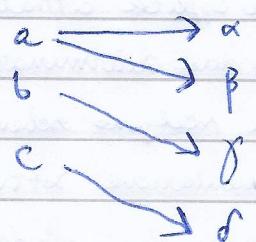
La funzione f combine R_{i-1} con le chiavi K_i

FUNZIONE f



CIFRARI A SOSTITUZIONE OMOFONA

questo tipo di cifrari serve a rendere difficile l'analisi delle frequenze. In questi cifrari le lettere dell'alfabeto sono mappate a uno o più simboli (^{dell'alf.} ~~carattere~~) le lettere più ricorrenti vengono mappate con più simboli;



In modo da sudistinguere le frequenze

CIFRARI A SOSTITUZIONE POLLALFABETICA

LINK ENCRYPTION VS END-TO-END ENCRYPTION

Si cura solo i nodi intermedi e non i utenti / destinatari

Messaggio esposto nello host mittente/destinato | Messaggio cifrato nello host mittente/destinatario

Messaggio esposto nei modi intermedi

Messaggio cifrato nei modi intermedi
Header ~~vuolte~~ vuolli

Ruolo dell'utente

trasparente all'utente

Utente applica crittura (whatsapp)

host mantiene servizi di crittura

Utente deve determinare l'algoritmo

Un servizio uguale per tutti gli utenti

Utente sceglie schema di crittografia

Può essere volte dell'hardware

Implementazione Software

O altri tutti i messaggi oppure non cifrano nulla

Utente sceglie se cifrare e cosa cifrare

IMPLEMENTAZIONE

Richiede una chiave per coppie

Richiede una chiave per coppia di

Comporti de host - modo intermedi

utenti

e una chiave per coppie modo int - modo int

Cifratura a chiave pubblica

La cifratura a chiave pubblica nasce nel 1975 per cercare di risolvere il PROBLEMA DELLA DISTRIBUZIONE DELLE CHIAVI e il PROBLEMA DELLE FIRME

Siamo

$E_2 : e \in K$ e $D_2 : e \in K$ un cifrario

condiviso la coppia (E_2, D_2)

dato il messaggio cifrato $c \in C$

CIFRATURA NORMALE

Per funzionare:

nella cifratura e nella decifratura si usa lo stesso algoritmo con le stesse chiavi.

Il mittente e il ricevente devono condividere l'algoritmo e la chiave

Per essere sicuri

> Le chiavi deve essere segrete

> Diventare impossibile o troppo lungo decifrare il messaggio cifrato

> Conoscere l'algoritmo usato e il testo cifrato dev'essere insufficiente per determinare la chiave

CIFRATURA A CHIAVE PUBBLICA

Per funzionare

- lo stesso algoritmo è usato nella cifratura e nella decifratura, ma con chiavi diverse

- Il mittente e il ricevente devono avere una delle coppi chiavi che formano le coppie usate dall'algoritmo.

Per essere sicuri

- Una delle due chiavi deve rimanere segreta

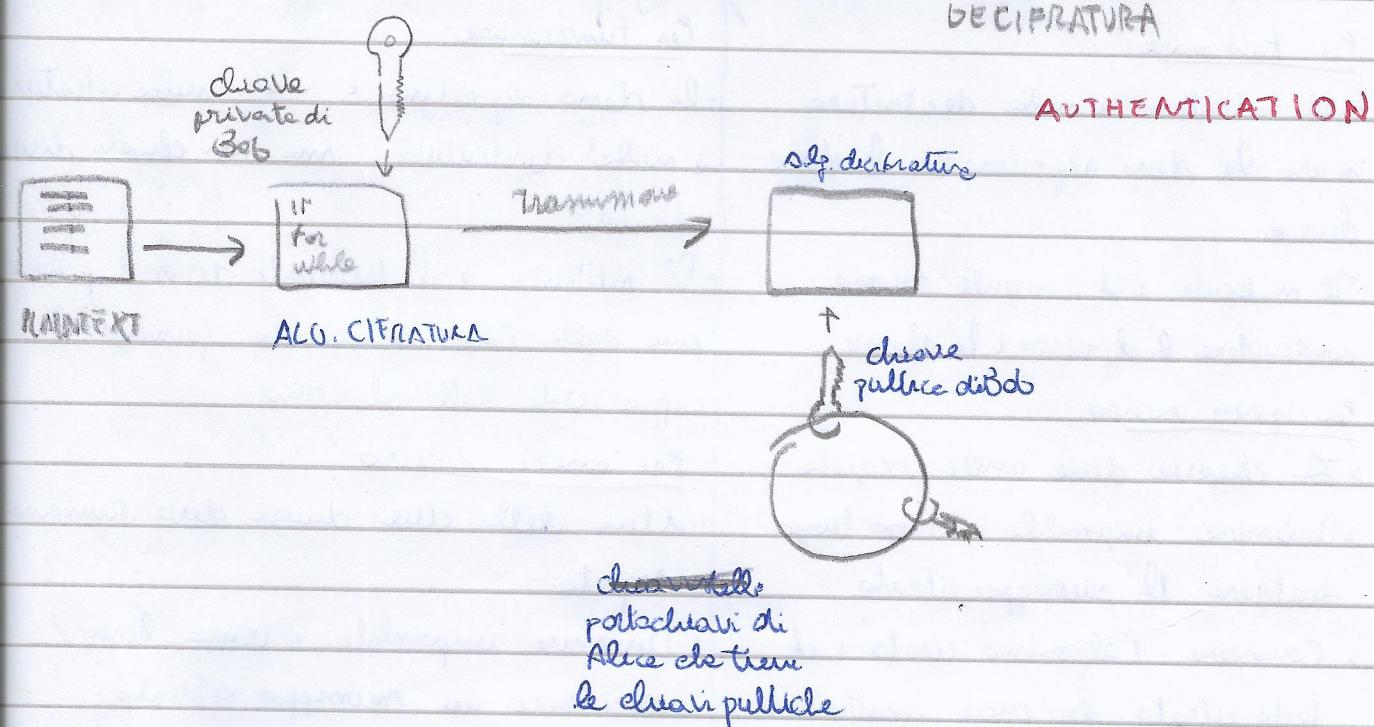
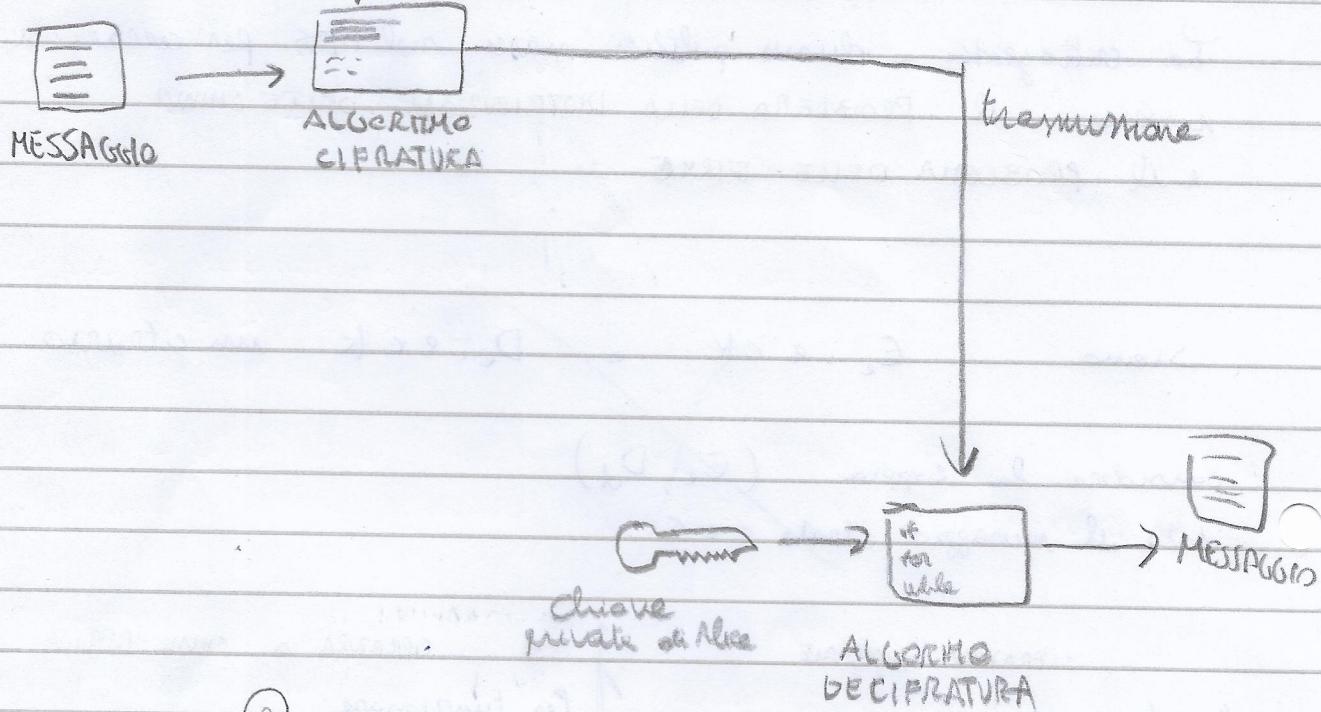
- Diventare impossibile o troppo lungo decifrare un messaggio cifrato

- Conoscere l'algoritmo, il messaggio cifrato e una delle due chiavi non deve bastare per trovare l'altra chiave

~~SECRET~~

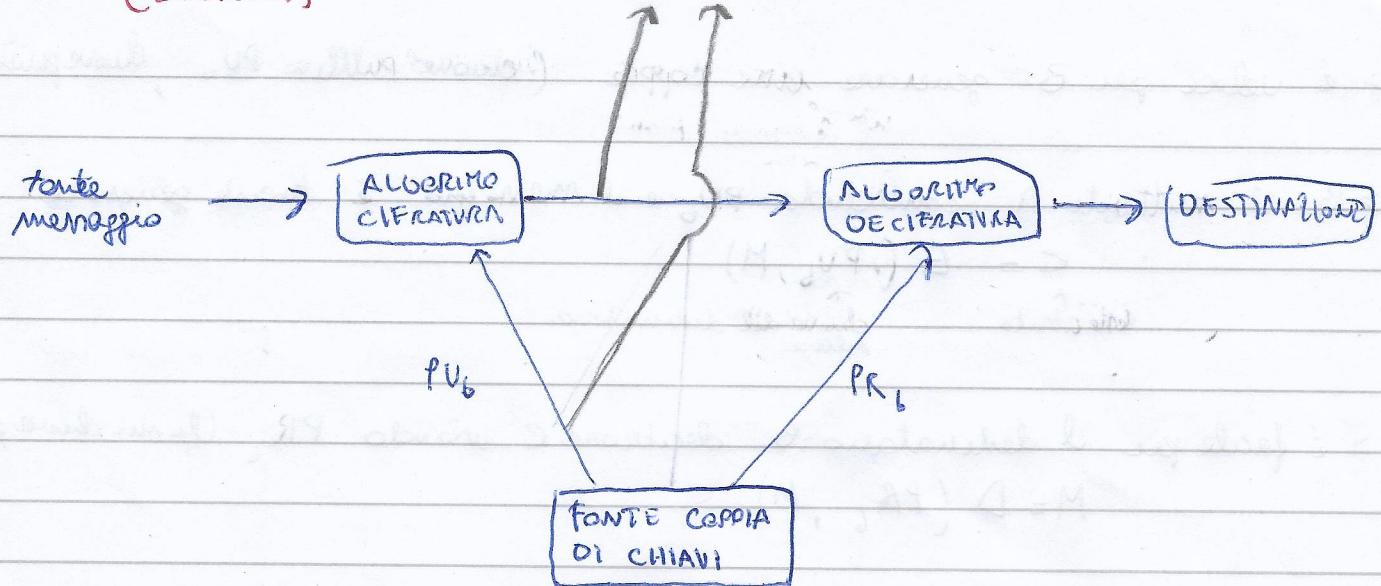
chiavistello di Bob
che tiene le chiavi
pubbliche
chiave pubblica di Ted
chiave pubblica di Mike
chiave pubblica di Alice

CIFRATURA



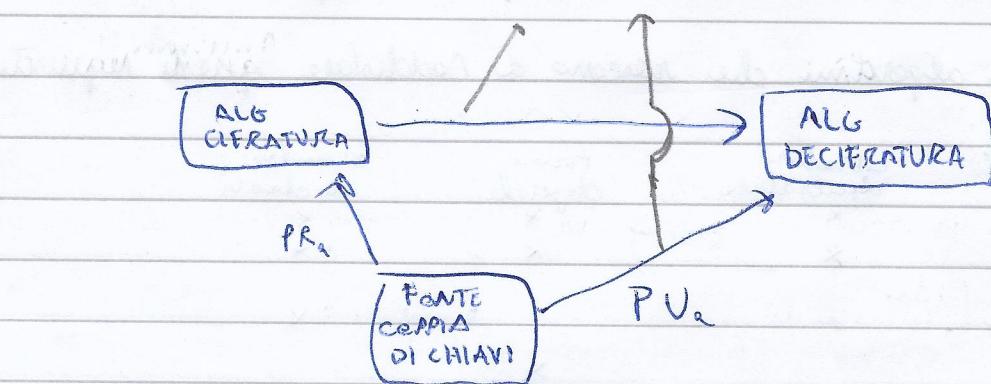
SEGRETITÀ
(SECRET)

CRIPTOANALISI



AUTENTICAZIONE

CRIPTOANALISI



ESERCIZI PER CRITTOGRAFIA A CHIAVE PUBBLICA

→ è veloce per B generare una coppia (chiave pubblica PUs , chiave privata PR_b)

> Per il multietate A, conoscendo PV_f e il messaggio, è facile generare:

> è facile per il destinatario B destruire C usando PR, (le mie chiavi private)

$$M = D(P_{R_t}, M)$$

Per l'attaccante è difficile, sapendo P_{U_f} , trovare P_{R_b}

Per l'attaccante è difficile, sapendo PV_0 e $c = E(PV_0, M)$ trovare M

Sono pochi gli algoritmi che riuscono a soddisfare questi requisiti

	Authoritarian dictatorship	Forms of rule	Stability of regime
RSA	x	dictator	x
RSA	x	dictator	x
Curve Elliptic	x	x	x
Diffie-Hellman			x
DSS		x	

12 FUNZIONI A UNA VARIABILE

$f : X \rightarrow Y$ questa è una funzione a una via re f^{-1} facile da calcolare per ogni $x \in X$ e f^{-1} è difficile da calcolare

example

$$\begin{aligned} p &= 3 \\ q &= 5 \\ M &= p \cdot q = 15 \end{aligned}$$

~~2624695722~~

p = 48611

$$q = 53993$$

$$M = 79 = 262 \text{ } 465 \text{ } 37 \text{ } 23$$

$$f: X \rightarrow N \quad \text{definite come} \quad f(x) = x^3 \bmod m$$

Esempio

$$f(2489991) = 1981394214 \quad (\text{calcolare } f \text{ è facile})$$

date yem trave x tale che $x^3 \equiv y \pmod{m}$

Celadole f^{-1} é difficile

FUNZIONE BOTOLA

$$f_k : X \rightarrow Y$$

tale che:

$$y = f_k(x) \quad \text{tacile conoscendo } x \in k$$

$$x = f_k^{-1}(y) \quad \text{tacile conoscendo } k \in y$$

$$x = f_k^{-1}(y) \quad \text{non difficile conoscendone } y$$

CRPTOANALISI A CHIAVE PUBBLICA

- > Forza Bruta
- > calcolare le chiavi private a partire dalla chiave pubblica
- > ATTACCO MESSAGGIO PROBABILE

Messaggio Criptato
di n bit

(decrittazione)

ATTACCANTE:

$$c = E(PU_A, m)$$

calcola $y_i = E(PU_A, x_i)$ per tutti i possibili
plaintext x_i

itera da $i=1$ a $\lambda = 2^{32}$

entro che non appena $y_i = c$ ragionando
che $y_i = x_i$ (il messaggio è messo)

TEORIA DEI NUMERI

per

fattorizzare un numero m abbiamo scriverlo come prodotto
di altri numeri: $m = a \cdot b \cdot c$

Moltiplicare numeri è facile, fattorizzare un numero è più difficile
NUMERI PIÙ LUNGHII DI 1024 bit sono impossibili da fattorizzare

La FATTORIZZAZIONE IN NUMERI PRIMI riguarda scrivere un numero come prodotto di numeri primi

$$a = \prod_{p \in P} p^{a_p} = 2^{a_2} \cdot 3^{a_3} \cdot 5^{a_5} \cdot 7^{a_7} \cdot 11^{a_{11}} \dots$$

due numeri sono PRIMI FRA LORO se non hanno fattori (divisori) comuni oltre a 1

ESEMPIO

8 e 15 sono primi fra loro.

Fattori di 8: 1, 2, 4, 8

Fattori di 15: 1, 3, 5, 15

permette trovare il MASSIMO COMUNE DIVISORE dobbiamo guardare i fattori primi comuni e prendere quelli con potenze minime

$$18 = 2^1 \cdot 3^2$$

$$300 = 2^2 \cdot 3^1 \cdot 5^2$$

$$\gcd = \text{MCD}(18, 300) = 2^1 \cdot 3^1 \cdot 5^0$$

ARITMETICA MODULARE

Per ogni $a \geq m$, esiste unico q e resto r tali che $a = q \cdot m + r$

Se $a \bmod m = b \bmod m$ allora $a \equiv b \pmod{m}$

CONGRUENTI IN NODULO m : $a \equiv_m b$

$$\begin{aligned} & \text{es:} \\ & 7 \bmod 5 = 12 \bmod 5 \\ & \Rightarrow 7 \equiv_5 12 \end{aligned}$$

PROPRIETÀ DEI CONGRUENTI IN NODULO m

i) $a \circ b = (a \bmod m) \circ (b \bmod m)$ per $\circ \in \{\cdot, +\}$

ii) $(a \circ b) \bmod m = [(a \bmod m) \circ (b \bmod m)] \bmod m$

iii) se $a \circ b \equiv_m a \circ c$ e $a \bmod m$ sono primi fra loro

allora $b \equiv_m c$

esempi

i)

$$30 \bmod 4 =$$

$$(5 \cdot 6) \bmod 4 =$$

$$\left((5 \bmod 4) \cdot (6 \bmod 4) \right) \bmod 4 =$$

$$(1 \cdot 2) \bmod 4 = 2$$

ii)

$$a \cdot b =_m a \cdot c$$

$$b =_m c$$

$$ii) 8 \cdot 4 =_3 8 \cdot 1 \Rightarrow 4 =_3 1$$

a e m sono primi fra loro; 8 e 3 sono primi fra loro

FUNZIONE DI EULERO TORIENTE

RESIDUI sono quei numeri che sono primi tra loro con m.

rendui formano l'insieme ridotto dei residui di m

esempio

$$m = 10$$

insieme completo dei residui: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

insieme ridotto dei residui: $\{1, 3, 7, 9\}$ tutti tranne 2 e 5 e 6 e 8

La funzione toriente di Euler denta il numero di rendui nell'insieme ridotto dei rendui

funzione toriente di p: $\phi(p) = p - 1$ se p è un numero primo

$$\phi(p \cdot q) = \phi(p) \cdot \phi(q) = (p-1)(q-1)$$

se p e q sono numeri primi

TEOREMA DI EULERO

$\phi(m)$
 $a \equiv_m 1$ per ogni $a \in m$ tale che $\text{MCD}(a, m) = 1$

e se $a \in m$ primo tra loro

esempio

$\forall a \in 3 \quad \text{allora} \quad a \equiv_m 1 \Rightarrow 3^{\phi(10)} \equiv_{10} 1$
 $\Rightarrow 3^4 \equiv_{10} 1 \Rightarrow 81 \equiv_{10} 1$

ALGORITMI RSA

creati nel 1978

- > La sicurezza si basa sulla difficoltà di fattorizzare grandi numeri
- > LE CHIAVI sono coppie di grandi numeri primi (più di 100 cifre)

STRUTTURA

• Se m un numero conosciuto da mittente e destinatario

dividiamo il messaggio in blocchi da $\lceil \log_2 m \rceil$ bit

messaggio $010101010111 \quad \forall m=4, \log_2 4=2 \Rightarrow \underline{01} \underline{01} \underline{01} \underline{01} \underline{11}$

OGNI BLOCCO rappresenta un numero M tale che $M < m$

CIFRATURA e DECIFRATURA

$$C = M^e \bmod m$$

$$M = C^d \bmod m = (M^e)^d \bmod m = M^{ed} \bmod m$$

con "realisti" valori di d e e

RSA ha un algoritmo di extrazione a chiave pubblica con:

$$\text{CHIAVE PUBBLICA} = (e, n)$$

$$\text{CHIAVE PRIVATA} = (d, n)$$

REQUISITI

- > è possibile trovare dei valori e, d, n per ogni $M \in \mathbb{N}$ con la condizione che $M^{ed} \mod n = M$ per ogni $M \in \mathbb{N}$
- > è facile calcolare $M^e \mod n$ e $C^d \mod n$ per ogni $M \in \mathbb{N}$
- > è difficile computazionalmente calcolare d , conoscendo e e n .

CORRETTEZZA DELL'RSA

x e y sono inversi moltiplicativi mod r se $xy \mod r = 1$

Se d e e sono moltiplicativi inversi mod $\phi(n)$,

cioè $ed \mod \phi(n) = 1$

allora $M^{ed} \mod n = M$ per ogni M

LEMMA

Siamo p e q numeri primi tali che $n = p \cdot q$

Se p e q sono moltiplicativi inversi mod $\phi(n)$:

$$p \cdot e \mod \phi(n) = 1 \Rightarrow M^{ed} \equiv_p M \quad \text{e} \quad M^{ed} \equiv_q M$$

Cioè: $(M^{ed} - M)$ è multiplo di p e di q

33 - ALGORITMI RSA ESEMPI

GENERARE COPPIA DI CHIAVI (pubbliche, private)

- genera due numeri primi molto grandi p e q
- calcola $m = p \cdot q$ e $\phi = (p-1)(q-1)$
- scegli uno e con $1 < e < \phi$ e con e primo fra loro
- calcola d tale che $e \cdot d \bmod m = 1$
- PUBBLICA (e, m) e tieni PRIVATA (d, m) e scrivete p e q

CIFRATURA con PR(d, m) da P.U. (e, m)

- divide il messaggio M in blocchi M_1, M_2, M_3, \dots con $M_i < m$
- calcola $C_i = M_i^e \bmod m$

DECIFRATURA con PR (d, m)

- calcola $M_i = C_i^d \bmod m$
↓
blocchetto cifrato

ESEMPIO

- genera due numeri primi grandi $p = 47$ $q = 71$
- calcola $m = p \cdot q = 3337$ e $\phi = (p-1)(q-1) = 46 \cdot 70 = 3220$
- scegli uno e coprimo di ϕ e minore di ϕ : $e = 79$
- calcola d tale che $e \cdot d \bmod m = 1 \Rightarrow 79 \cdot d \bmod m = 1$
 $\Rightarrow d = 1019$

PUBBLICA $(e, m) = (79, 3337)$ e tieni PRIVATA $(d, m) = (1019, 3337)$

CIFRATURA

divide il messaggio in blocchi e.g. 688 232 687 966 668

$$M_1 = 688^79 \bmod m = 688^{79} \bmod 3337$$

$$M_2 = 232^{79} \bmod 3337$$

ARITMETICA MODULARE

RELAZIONE DI CONGRUENZA

L'aritmetica modulare si basa sul concetto di congruenza modulo m .
dati tre numeri interi a, b, m diciamo che a e b sono congruenti modulo m se la differenza $(a-b)$ è multiplo di m

$$a \equiv b \pmod{m}$$

Ese: $38 \equiv 14 \pmod{12}$

perché $a-b = 38-14 = 24$ è multiplo di 12

INFATI $a \pmod{m} = b \pmod{m}$

$$38 \pmod{12} = 14 \pmod{12}$$

$$\begin{array}{r} 38 \mid 12 \\ \underline{3} \quad | \\ 2 \end{array} = \begin{array}{r} 14 \mid 12 \\ \underline{12} \quad | \\ 2 \end{array}$$

partendo da 00:00, dopo
38 ore sono le 2:00

partendo da 00:00, dopo
14 ore sono le 2:00

PROPRIETÀ SIMMETRICA:

$$\text{se } a \equiv b \pmod{m} \text{ allora } b \equiv a \pmod{m}$$

dim: se m divide $(a-b)$ allora m divide anche $(b-a)$

INVARIANZA PER ADDIZIONE

aggiungendo o sottraendo delle stesse quantità alle numeri congruenti modulo m , i numeri ottenuti sono ancora congruenti modulo m

$$\text{se } a \equiv b \pmod{m} \text{ allora } (a+c) \equiv (b+c) \pmod{m}$$

INVARIANZA PER MOLTIPLICAZIONE

$$a \equiv b \pmod{m} \Leftrightarrow (a \cdot c) \equiv (b \cdot c) \pmod{m}$$

INVARIANZA PER POTENZA

$$a \equiv b \pmod{m} \Leftrightarrow a^c \equiv b^c \pmod{m}$$

INVERSO MOLTIPLICATIVO

$$a^{-1} \equiv b \pmod{m} \Leftrightarrow a \cdot b \equiv 1 \pmod{n}$$

ARITMETICA DELLE CONGRUENZE MODOLO M

394.0001

Le proprietà sopra elencate ci fanno capire che le congruenze modulo m sono una relazione di equivalenza. Inoltre le classi congruenti mod m determinano un insieme chiamato:

$$a = k \cdot m + \text{resto}$$

ovvero

$$a - \text{resto} = k \cdot m \quad (\text{matematico})$$

poniamo dividere l'insieme \mathbb{N} in m classi (sottosettimoni)

secondo la relazione di equivalenza:

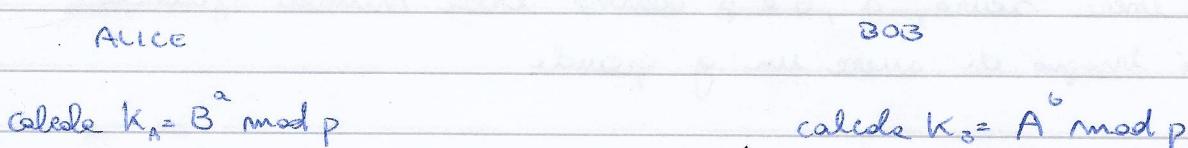
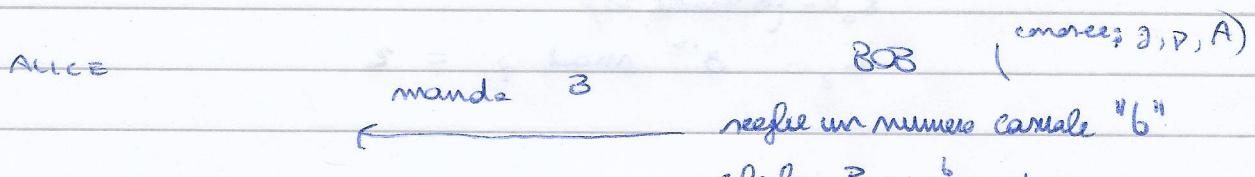
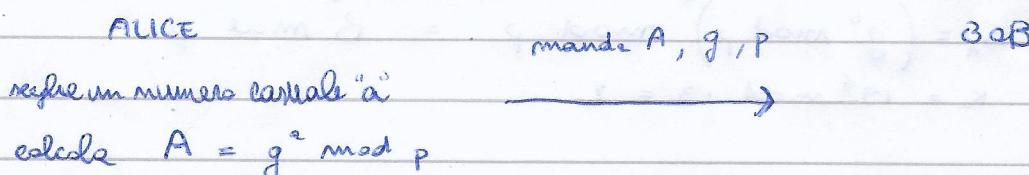
SCAMBIO DI DIFFIE HELLMAN

Lo scambio di Diffie Hellman è un protocollo critografico che consente a due entità di stabilire una CHIAVE CONDIVISA SEGRETA utilizzando una comunicazione incinta, senza la necessità che le due entità si siano neanche incontrate in precedenza. La CHIAVE così ottenuta può essere usata per comunicare con un cifrario simmetrico.

DESCRIPTION

Il generatore modulio m , o semplicemente generatore è un intero g le cui potenze modulo m sono i coprimi di m

$$g^x \bmod m = \begin{array}{l} \text{numero} \\ \text{cercano} \\ \text{di } m \end{array}$$



Per aritmética modular, $k_1 = k_2$ jache $B^2 \text{ mod } p \equiv A^6 \text{ mod } p$

esempio

Alice e Bob si accordano di usare un numero primo $p=23$ e la base $g=5$

$$p = 23 \quad g = 5$$

Alice sceglie un numero segreto $a=6$, calcola $A = g^a \bmod p$

$$A = 5^6 \bmod 23 = 8$$

Alice manda a Bob A

$$\text{Alice} \xrightarrow{A=8} \text{Bob}$$

Bob sceglie l'intero segreto $b=15$ e calcola $B = g^b \bmod p$

$$B = 5^{15} \bmod 23 = 19$$

Bob manda ad Alice B

$$\text{Alice} \xrightarrow{B=19} \text{Bob}$$

Alice calcola $K_A = (g^b \bmod p)^a \bmod p = B^a \bmod p$

$$K_A = 19^6 \bmod 23 = 2$$

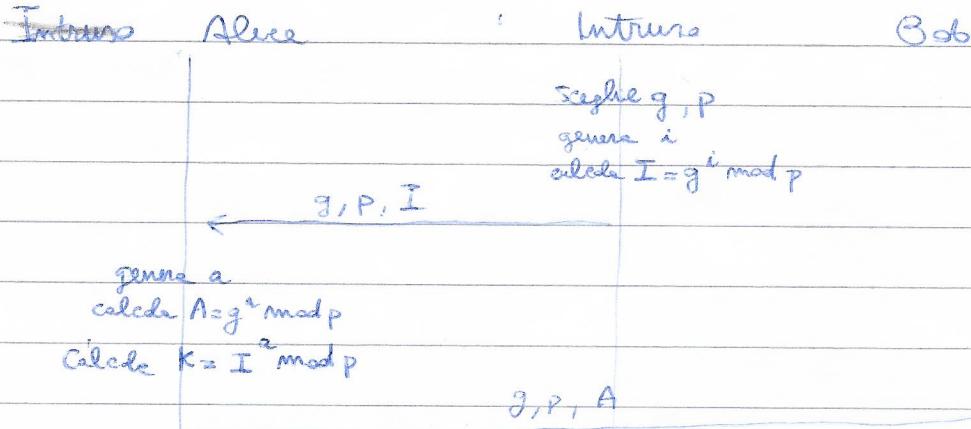
Bob calcola $K_B = (g^a \bmod p)^b \bmod p = A^b \bmod p =$

$$K_B = (5^6 \bmod 23)^{15}$$

$$K_B = 8^{15} \bmod 23 = 2$$

per Φ essere sicuri, a, b e p devono essere numeri grandi
non ci è bisogno di avere un g grande

ATTACCO A DIFFIE HELLMAN



ESEMPIO CRITTOGRAFIA

esame - 2 - 25 Giugno 2003

Si consideri il seguente algoritmo crittografico basato sull'idea di sostituire ogni lettera dell'alfabeto m con $(a \cdot m + b) \bmod 26$ dove la chiave è data da $K = \{a, b\}$ con a e b interi positivi nell'intervallo $[0, 25]$

Si dimostri, mediante un esempio, che se a e 26 non sono coprimi allora lo schema non è utilizzabile

prendiamo la chiave $K = \begin{smallmatrix} a & b \\ 2 & 0 \end{smallmatrix}$, 2 non è coprime di 26

$$m_1 = 0 \Rightarrow C(0) = (2 \cdot 0 + 0) \bmod 26 = 0 \bmod 26 = 0$$

$$m_2 = 13 \Rightarrow C(13) = (2 \cdot 13 + 0) \bmod 26 = 26 \bmod 26 = 0$$

La lettera ad indice 0 viene cifrata con lo stesso simbolo delle lettere ad indice 13. Quindi il ^{la funzione} ~~scrittura~~ non è invertibile

esame 6 giugno 2006 - ex 1

ALESSANDRO ARMANI

Si utilizzzi la procedura di crittura per trasposizione.

come chiavi si utilizzi la permutazione 2 3 4 1

$$K(1) = 2$$

$$K(2) = 3$$

$$K(3) = 4$$

$$K(4) = 1$$

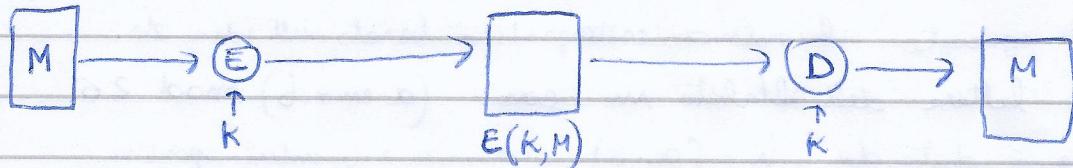
A L E S S A N D R O - A

L E S A A N D S O - A R

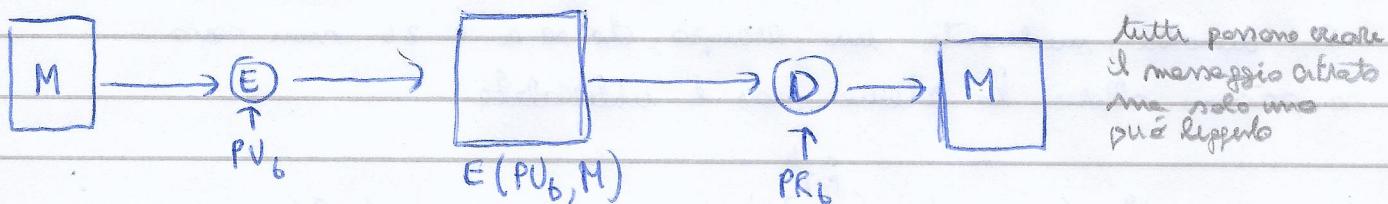
05/06/2006 252

Si indichino le proprietà di messaggio ammurate dai seguenti schemi crittografici

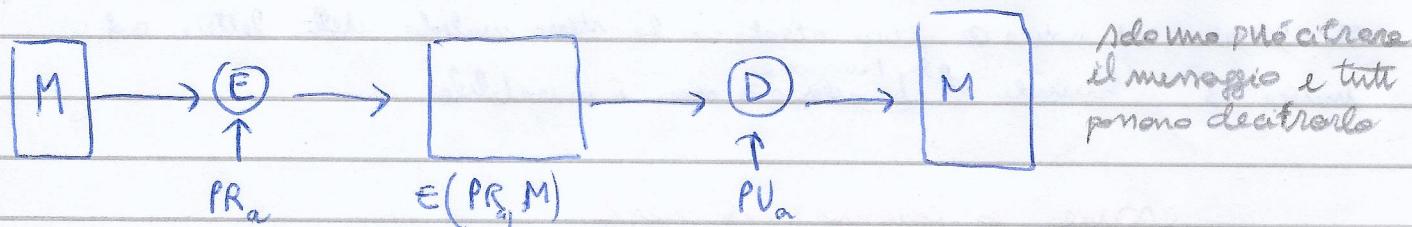
mittente A



CRITTOGRAFIA SINTETICA : CONFIDENZIALITÀ ed AUTENTICAZIONE

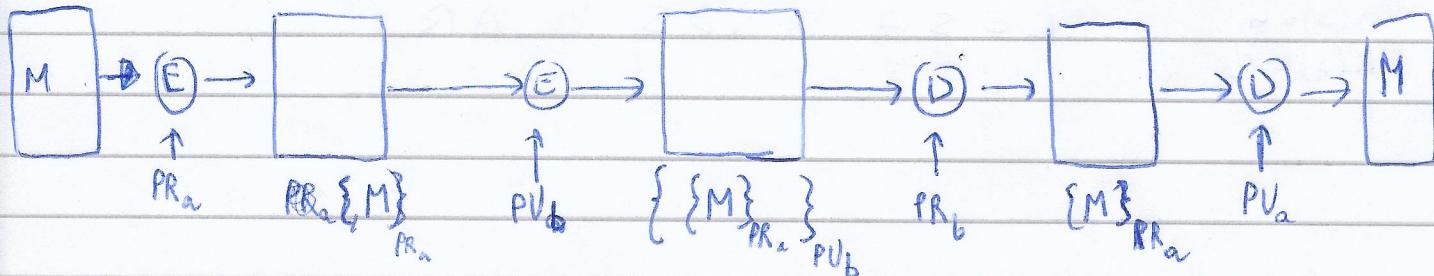


CRITTOGRAFIA ASIMMETRICA : CONFIDENZIALITÀ



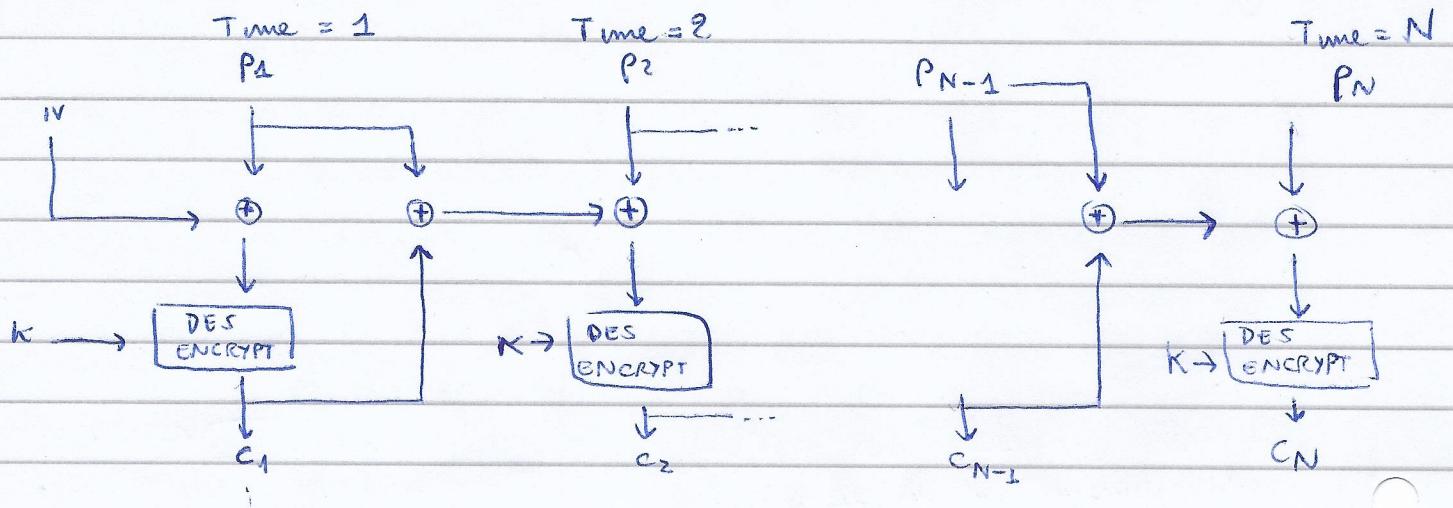
CRITTOGRAFIA ASIMMETRICA: Autenticazione e Non-Ripudio

chiunque può decifrare il messaggio, ma quel messaggio è sicuramente di Alice se noi riusciamo a decifrarlo.

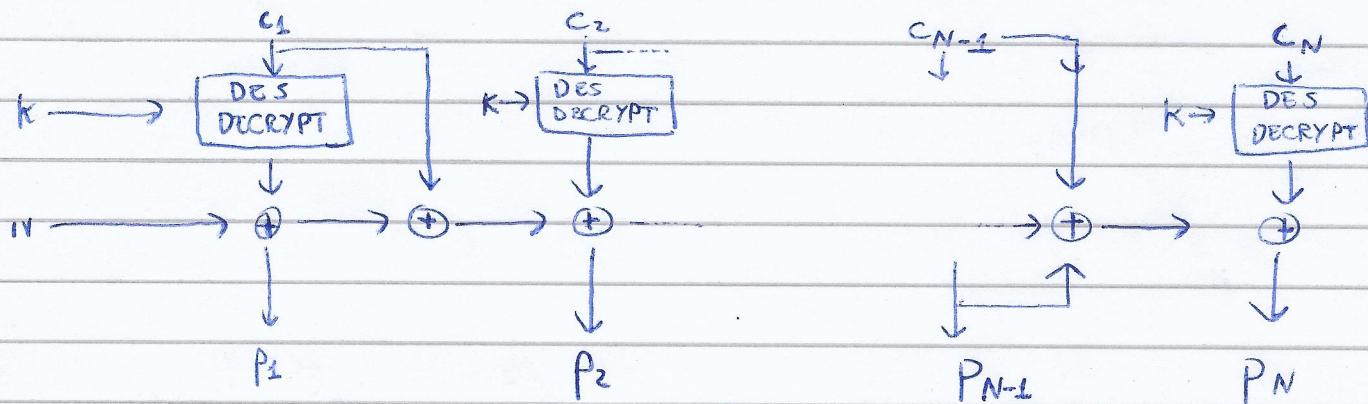


CRITTOGRAFIA CHIAVE PUBBLICA: Confidentialità, autenticazione e Non-Ripudio

Il seguente schema crittografico per cifratura a blocchi è usato in Kerberos v4.



DISEGNA LO SCHEMA CRIPTOGRAFICO DA USARE PER LA DECIFRATURA



Esercizio 1 Diffie Hellman 10 LUGLIO 2009

Si consideri lo schema di Diffie Hellman dove $p=11$ e $g=2$

Se Alice ha le chiavi pubbliche $A=3$, qual è la chiave privata di Alice?

$$A = g^a \pmod{p} \quad \text{trava a}$$

$$3 = 2^a \pmod{11}$$

impotenza di due

$$\begin{array}{r} 16 \\ 11 \longdiv{1} \\ \hline 5 \end{array}$$

$$\begin{array}{r} 32 \\ 22 \longdiv{1} \\ \hline 10 \end{array}$$

$$\begin{array}{r} 64 \\ 55 \longdiv{5} \\ \hline 9 \end{array}$$

$$2^6 \pmod{11} = 9$$

quindi chiave privata $a=6$

Se Bob ha chiavi pubbliche $B=3$, qual è la chiave segreta k ?

$$\begin{aligned} k &= (g^b \pmod{p})^a \pmod{p} = 3^a \pmod{11} = 3^6 \pmod{11} = \\ &= 729 \pmod{11} = 3 \end{aligned}$$

Esercizio 2 10 LUGLIO 2009

Un utente cerca di scegliere una password con lunghezza minima di 1 carattere e massima di 8 caratteri. È possibile testare 1000 password al secondo. L'amministratore di sistema vuole disabilitare le password non appese a altre le parole del 10% che vengono scelte password composte da almeno 2 cifre:

prob corrente

$$\frac{1}{10^8}$$

prob finale

$$\frac{1}{10}$$

dove ora ogni tentativo 10⁷ password

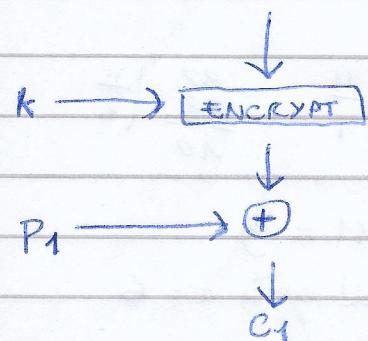
$$s = v \cdot t \Rightarrow t = \frac{10^7 \text{ password}}{10^5 \text{ password/secondo}} = 100 \text{ secondi}$$

053 10 LUGLIO 2009

Dato questo schema crittografico per cifrature a blocchi.

COUNTER è una sequenza di bit arbitraria delle stesse lunghezza dei blocchi b. L'operazione di somma è XOR

COUNTER



c₁



$$P_1 \leftarrow \begin{matrix} + \\ \end{matrix}$$



$$K \rightarrow \begin{matrix} \text{ENCRYPT} \\ \uparrow \end{matrix}$$

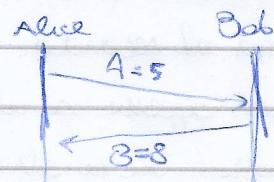
COUNTER

05 20 06 2006

$$p = 11, g = 2$$

$$A = g^a \pmod{p} \Rightarrow 5 = 2^a \pmod{11}$$

$$B = g^b \pmod{p} \Rightarrow 8 = 2^b \pmod{11}$$



$$\begin{array}{r} 2^0 \ 2^1 \ 2^2 \ 2^3 \ 2^4 \ 2^5 \\ \cancel{\cancel{2}} \cancel{\cancel{2}} \cancel{\cancel{2}} \cancel{\cancel{2}} \cancel{\cancel{2}} \end{array} \begin{array}{c} 16 \\ | \\ 11 \\ \hline 5 \end{array} \quad \begin{array}{c} 32 \\ | \\ 22 \\ \hline 10 \\ | \\ 2 \end{array} \quad \begin{array}{c} 64 \\ | \\ 55 \\ \hline 9 \\ | \\ 5 \end{array} \quad \begin{array}{c} 128 \\ | \\ 121 \\ \hline 7 \\ | \\ 11 \end{array}$$

$$2^0 = a = 4$$

$$\begin{array}{r} 2^0 \ 2^1 \ 2^2 \ 2^3 \ 2^4 \ 2^5 \\ \cancel{\cancel{2}} \cancel{\cancel{2}} \cancel{\cancel{2}} \cancel{\cancel{2}} \cancel{\cancel{2}} \end{array} \rightarrow \begin{array}{c} 8 \\ | \\ 11 \\ | \\ 10 \\ | \\ 2 \end{array}$$

$$\begin{array}{c} 8 \\ | \\ 10 \\ | \\ 8 \\ | \\ 2 \end{array} \quad b = 3 \quad \rightarrow b = 8$$

$$K = B^a \pmod{p} = 8^4 \pmod{11} = 32768 \pmod{11} =$$

$$K = A^b \pmod{p} = 5^3 \pmod{11} =$$

$$K = A^b \pmod{p} = 5^3 \pmod{11} = 125 \pmod{11}$$

$$\begin{array}{c} 125 \\ | \\ 121 \\ | \\ 4 \\ | \\ 11 \end{array}$$

BREVITTA 214 6-5-2006

$$p=7$$

$$q=3$$

$$e=5$$

$$M=4$$

$$\text{chiavi pubbliche } (e, n) = (5, n) = (5, 21)$$

$$\text{chiavi private } (d, n) = (d, 21)$$

$$n = p \cdot q = 7 \cdot 3 = 21$$

$$C_0 = M_0^e \bmod n = 4^5 \bmod 21 = 256 \bmod 21 = 16$$

$$\text{trova } d \text{ tale che } e \cdot d \bmod n = 1 \Rightarrow 5d \bmod 21 = 1$$

$$\begin{array}{r} 29 \\ 21 \mid 1 \\ \hline 4 \end{array} \quad \begin{array}{r} 30 \\ 21 \mid 1 \\ \hline 9 \end{array}$$

le chiavi di decrittazione sono date da (d, n)

$$\text{dove } d = e^{-1} \bmod \phi(n)$$

$$d = e^{-1} \bmod \phi(n) = \frac{1}{5} \bmod \phi(21)$$

$$\phi(21) = \#\{1, 2, 4, 5, 8, 11, 13, 17, 19, 20, 10, 12\} = 12 \quad 21 = 7 \cdot 3$$

$$d = 5^{-1} \bmod 12$$

$$\begin{array}{r} 50 \\ 12 \mid 10 \\ \hline 50 \end{array} \quad \text{divide per 10} \rightarrow d=5$$

3° PDF

AUTENTICAZIONE e FIRMA DIGITALE

3 INTEGRITÀ DEL MESSAGGIO

- proteggere l'integrità
- confermare l'identità del mittente

4 FUNZIONI DI AUTENTICAZIONE

Qualsiasi messaggio di autenticazione o meccanismo di firma digitale si affida a una FUNZIONE DI AUTENTICAZIONE per generare un AVVINCATORE, cioè il valore usato per autenticare il messaggio

FUNZIONI DI AUTENTICAZIONE

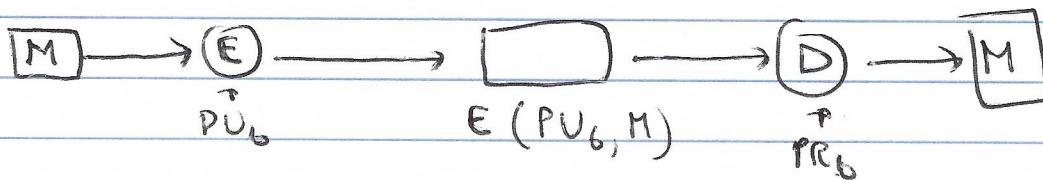
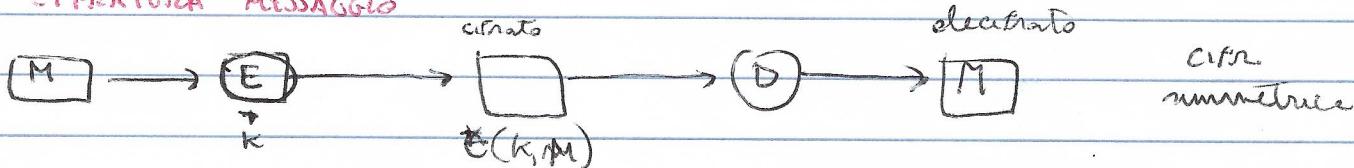
- CRIPTURA DEL MESSAGGIO INTERO
- CODICE DI AUTENTICAZIONE DEL MESSAGGIO :

Una funzione del messaggio è una chiave segreta che produce un valore di lunghezza fissa da usare come autenticatore

- FUNZIONE DI HASH CRIMICORAFICA

Una funzione mappa un messaggio di qualsiasi lunghezza in un valore hash di lunghezza fissa, da usare come autenticatore

CRIPTURA MESSAGGIO



7 CONTROLLO degli ERRORI nelle cifrature del messaggio

- > Servono dei metodi automatici per determinare se il testo cifrato in arrivo si decide a un testo non placcato intellegibile.
- > Una soluzione comune nel dare al messaggio normale una struttura che è facilmente riconoscibile, MA NON può essere replicata serve ricorrere alle funzioni di cifratura.

MESSAGE AUTHENTICATION CODE (MAC)

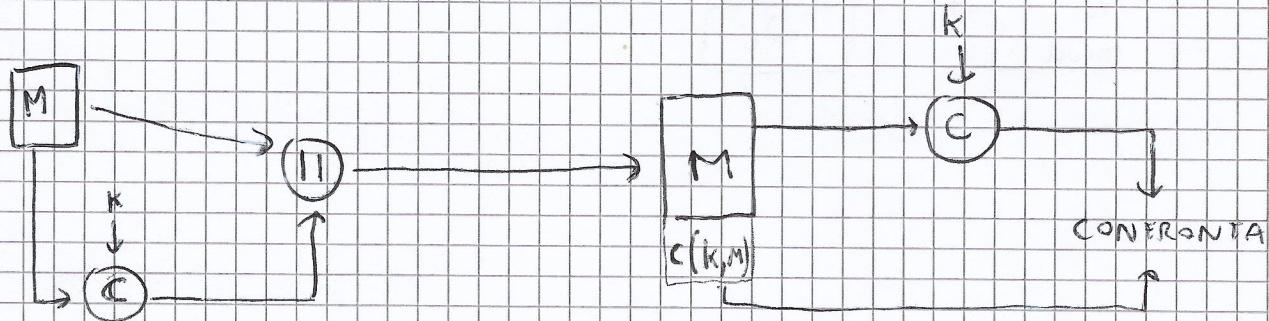
un MAC è un piccolo blocco di dati usato per garantire l'autenticazione e l'integrità di un messaggio digitale, generato con le crittografie simmetriche.

- Un algoritmo MAC prende in input una chiave segreta e un messaggio da autenticare di lunghezza arbitraria
- L'algoritmo MAC restituisce un codice di AUTENTICAZIONE MAC

INPUT: M, K

OUTPUT: $C(M, k)$ codice MAC

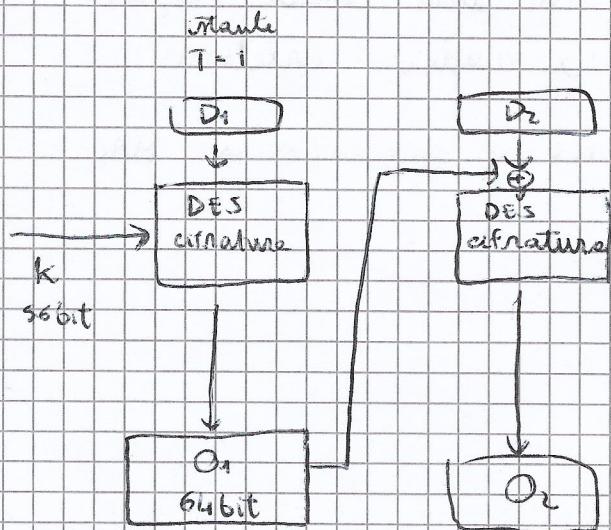
ESEMPIO AUTENTICAZIONE



ALGORITMO AUTENTICAZIONE DATI (DAA)

- È basato sull'algoritmo di crittografia DES
- È il più popolare algoritmo MAC
- Il messaggio input viene diviso in blocchi da 64 bit

$$M = D_1 \ D_2 \ D_3 \ \dots \ D_N$$



14 FUNZIONI DI HASH

CRITTOGRAFICHE

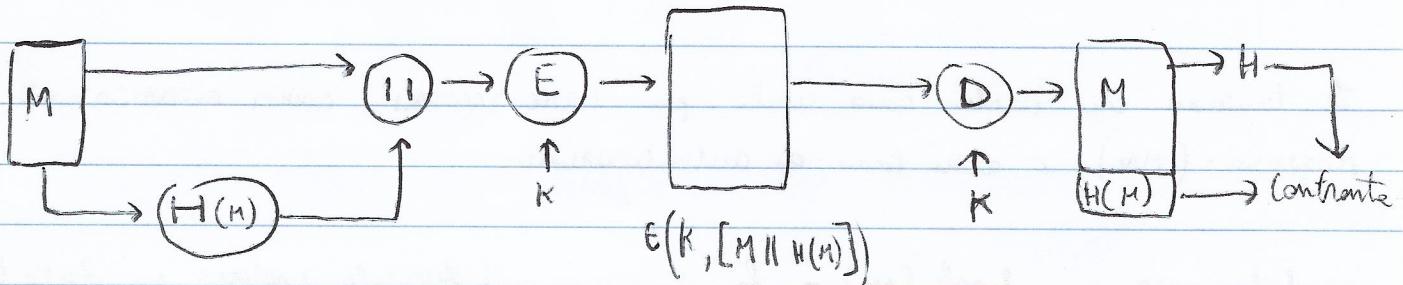
INPUT delle funzione: messaggio M di dimensione variabile

OUTPUT delle funzione: messaggio ombra $H(M)$ di dimensione fisica

$H(M)$ è detto codice hash

(valore di hash)

SCHEMA



Una funzione critografica di hash è una speciale funzione di hash
che dispone di proprietà che le rendono adatte all'uso nelle critografie

Una funzione di hash prende in input messaggio di DIM. arbitraria
e restituisce una stringa binaria di dimensione fisica (valore di hash)
(hash digest)

Tale funzione di hash è progettata per essere inidirizzabile:
molto difficile da invertire ; dato l'output, l'unico modo per ricreare
l'input è provare tutti i possibili input per vedere se corrispondono
all'output analizzato.

PROPRIETÀ FONDAMENTALI DI CHIAVI

F. HASH CRITTOGRAFICHE

- la funzione deve identificare univocamente il messaggio.
Non è possibile che due messaggi differenti producano lo stesso valore di Hash (output)
- deve essere deterministica: un messaggio produce sempre lo stesso Hash
- deve essere veloce calcolare un hash di qualsiasi messaggio
- deve essere computationalmente impossibile generare un messaggio partendo dal suo valore di Hash

Le funzioni di Hash sono usate per FIRME DIGITALI, CODICI AUTENTICAZIONE MESSAGGI (MAC) e altre forme di autenticazione.

$$\text{dato } m, \text{ hash}(m) = h \quad \text{è difficile trovare } m \text{ dato } h$$

Svantaggi

- Le funzioni crittografiche di Hash sono vulnerabili ad attacchi lunghezza-estensione
- date $\text{hash}(m)$ e $\text{length } \text{len}(m)$, scegliendo un appropriato m'
 - un attaccante può calcolare $\text{hash}(m || m')$

HASH DISPENSE

WEAK COLLISION RESISTANCE

Per qualsiasi valore x , è computazionalmente impossibile trovare x tale che $H(x) = y$

STRONG COLLISION RESISTANCE

È computazionalmente impossibile trovare qualsiasi coppia (x, z)

Tale che $H(x) = H(z)$

\uparrow \uparrow
 incognita incognita

ONE WAY PROPERTY

Dato un qualsiasi valore di hash y , è computazionalmente impossibile

trovare x tale che $H(x) = y$

\uparrow \uparrow
 incognita incognita

WEAK

STRONG COLLISION RESISTANCE

Dato un qualsiasi valore x , è infatti possibile trovare y

Tale che $H(x) = H(y)$

\uparrow \uparrow
 motivo incognita

APPLICAZIONI

- La proprietà "one way" viene usata nell'autenticazione

$A \rightarrow B : M \parallel H(M||S)$ dove S è segreto condiviso tra A e B

ATTACCO DEL COMPLEANNO

L'attacco del compleanno è ~~un~~ usato per le ~~cifras~~ struttura i principi matematici del paradosso del compleanno.

DATA UNA FUNZIONE f , LO SCOPO DELL'ATTACCO È TROVARE DUE NUMERI x_1 E x_2 TEL CHE $f(x_1) = f(x_2)$. LE COPIE (x_1, x_2) È Dette collisione.

IL METODO PER TROVARE UNA COLLISIONE È QUELLO DI TESTARE TUTTI I POSSIBILI INPUT TUTTO NON SI OTTIENE LO STESSO RISULTATO. A CAUSE DEL PARADOSSO DEL COMPLEANNO QUESTO ATTACCO PUÒ ESSERE MOLTO EFFICIENTE.

FIRMA DIGITALE

Le firme digitali è un metodo matematico teso a dimostrare l'autenticità di un messaggio inviato tra mittente e destinatario attraverso un canale non sicuro. La firma digitale garantisce che:

- il mittente è veramente chi dice di essere (AUTENTICAZIONE)
- il mittente non può negare di averlo inviato (NON RIPUDIO)
- il messaggio non sia stato alterato lungo il percorso (INTEGRITÀ)
(le firme non garantiscono confidentialità)

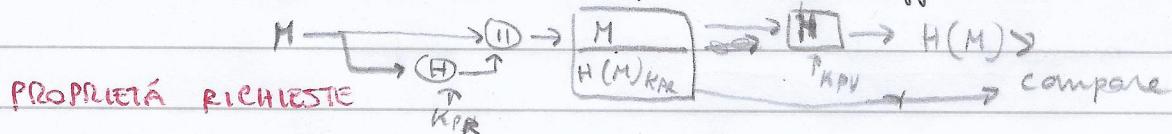
FIRMA DIGITALE CON CRIPTOGRAFIA A CHIAVE PUBBLICA

> un algoritmo genera casualmente le chiavi private da un insieme di valori possibili e restituisce una coppia di chiavi:

CHIAVE PRIVATA con cui si firma il documento

CHIAVE PUBBLICA con cui si verifica l'integrità

- > Un algoritmo di firma prende un ingresso MESSAGGIO e CHIAVE PRIVATA e calcola il CODICE HASH del messaggio e lo critta con la chiave privata producendo una Firma
- > Un algoritmo di verifica prende un ingresso MESSAGGIO e CHIAVE PUBBLICA e FIRMA accetta o rifiuta la firma che compare nel messaggio



• La firma deve essere verificata

Messaggio x Chiave Privata → FIRMA

facendo uso delle chiavi pubbliche

• Deve essere comp. difficile generare una firma valida senza avere la CHIAVE PRIVATA

Una FIRMA DIGITALE è UNA TRIPLOTA DI ALGORITMI:

Generator message verification
alg alg alg
(G, S, V)

generatore della chiave pubblica

G: genera chiave pubblica pk, chiave privata sk da partire dal valore in input^{1^m} dove m è il parametro di sicurezza

S: restituisci un tag prendendo un ingresso chiave privata e stringa x (MESSAGGIO)

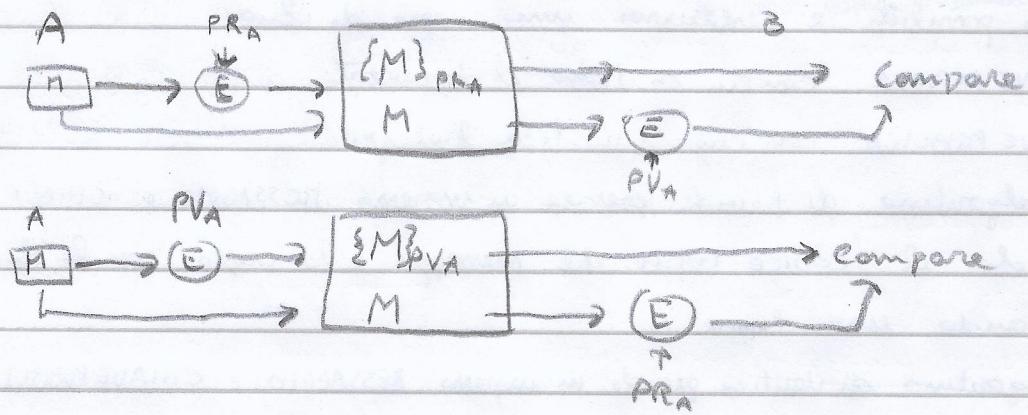
V: chiave pubblica x tag x stringa x → Bool

algoritmo valuta questi valori per capire se la firma è valida o meno

PIANO CON CRIPTOGRAFIA ASSIMETRICA

Ogni utente ha una chiave privata con cui può decifrare i messaggi che gli vengono inviati e formare i messaggi che invia agli altri utenti che hanno chiavi pubbliche che altri utenti utilizzano per cifrare i messaggi da inviare all'utente e poi decifrare le sue forme per stabilire l'autenticità.

Le chiavi vengono generate da un algoritmo e con le garanzie che le chiavi private non sono in grado di decifrare un testo cifrato con le chiavi pubbliche (e viceversa).



CERTIFICATO DIGITALE

Mentre le crittografie simmetriche ammettono un certificato digitale è un documento elettronico che attesta l'associazione univoca tra chiave pubblica e l'identità di un soggetto, che dichiara di utilizzare le chiavi pubbliche nell'ambito di procedure di cifratura assimmetrica e/o autenticazione tramite firma digitale.

Il certificato digitale contiene informazioni sulle chiavi, intorno all'identità del proprietario, e la firma digitale di un ente che ha verificato i contenuti del certificato. Se la firma è valida e il software che esamina il certificato si affida all'emittente, allora può utilizzare tali chiavi per comunicare in modo sicuro col soggetto del certificato.

Il certificato, fornito da un ente terzo fidato e riconosciuto come AUTORITÀ DI CERTIFICAZIONE (CA) e a sua volta autenticato per evitare la falsificazione sempre attraverso FIRMA DIGITALE, ovvero affidato con le chiavi private dell'associazione la quale fornisce poi la rispettiva chiave pubblica associata per verificarlo.

Documento con ID, e
chiavi pubbliche di Mario Rossi

Nome	Mario
Cognome	Rossi
Indirizzo	via Roma
CHIAVE PUBBLICA DI MARIO ROSSI	

chiavi private
del soggetto
certificatore

Certificato di Mario Rossi

Nome	
Cognome	
Indirizzo	
CHIAVE PUBBLICA DI MARIO ROSSI	
FIRMA DEL SOGGETTO CERTIFICANTE	

SCOPO

Scambierete le chiavi pubbliche in modo nuovo che gli utenti diversi
impraticabile quando il numero di utenti comincia a crescere
Lo scopo del certificato digitale è garantire che una chiave pubblica
non appartenga alle vere identità dei soggetti che la rivendono come propria

Questo è molto importante in crittografia asimmetrica, infatti ogni messaggio
cifrato con una data chiave pubblica può essere decifrato solo da chi possiede
le relative chiavi private.

SMART CARD

Una smartcard è un dispositivo hardware che possiede potenzialità di elaborazione e memorizzazione dati in grado di garantire elevati standard di sicurezza.

La smartcard è formata da un supporto in plastica nel quale è incastonato un microcircuit integrato connesso ad un'interfaccia di collegamento come un'antenna. Il microcircuito integrato fornisce funzionalità di calcolo e memorizzazione. L'antenna (o cattuttore) consente al microcircuito di dialogare con un terminale di lettura collegato ad un computer.

CRIPTOGRAFIA

La smartcard è microprocessore, grazie alle caratteristiche di protezione dei dati intrinseche del microcircuit, è un mezzo adeguato per proteggere le chiavi private utilizzando la crittografia e utilizzando la FIRMA DIGITALE per l'autenticazione.

CLASSIFICAZIONE

SMARTCARD A SOLA MEMORIA VS SMARTCARD CON MICROPROCESSORE

SMARTCARD COLLEGATE CON ANTENNA VS SMARTCARD COLLEGATE CON CAVO ATTACCA

SMARTCARD A SOLA MEMORIA

La smartcard a sola memoria offre unicamente la funzionalità di memorizzazione sicure dei dati. Il microcircuito integrato contiene una componente di memoria permanente nella quale si può leggere e scrivere attraverso un circuito pre-programmato. Il circuito legge comprende un meccanismo di protezione che salvaguarda l'accesso ai dati, basato sui dati permessi di accesso. Le smartcard a sola memoria vengono usate per carte prepagate, riacquette punti etc., e il meccanismo di memoria serve ad evitare l'aumento fraudolento del credito.

SMART CARD A MICROPROCESSOR

La Smartcard a microprocessore, grazie alle potenze di calcolo, è considerabile un piede calcolatore altamente affidabile, maneggevole e capace di elaborare e memorizzare informazioni salvaguardandone la sicurezza.

La smartcard contiene un sistema operativo che implementa le logiche operative delle smartcard.

L'ES si occupa delle gestione interne della memoria e fornisce varie funzioni gestite
lettura e scrittura in memoria, funzioni di programmazione dei settori di eccellenza
e funzioni critoografiche.

Una smartcard e' microprocedente le piu comandi: oltre a letture e scritture, le smartcard forniscono comandi di gestione dell'accesso alle memorie (es. comandi di verifica del PIN) e comandi di gestione del file system interno.

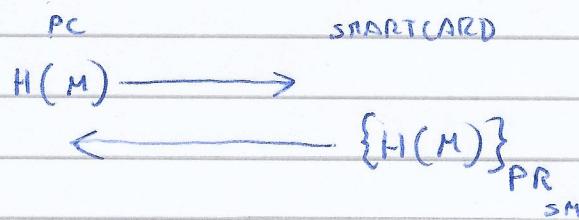
Teletone Mobile

Le SIM è un dispositivo di identificazione dell'utente. le SIM conserva ID dell'utente e generi le chiavi crittografiche usate per cifrare le trasmissioni digitali della voce.

CRITICOGRAFIA A CHIAVE PUBBLICA

Quando una smartcard comunica con un PC, può implementare uno schema di firme digitali. La smartcard contiene le chiavi private dell'utente ed è in grado di cifrare stringhe di testo (a causa delle ridotte potute).

Il PC calcola lo hash del documento da firmare e lo invia alla SMARTCARD.
La smartcard offre tale hash con le chiavi private memorizzate
ed invia al PC il risultato.



ESERCIZI CRITTOGRAFIA / HASH

esame 2 - 25 Giugno 2009

se K una chiave crittografica data

se $M = M_1 M_2 \dots M_m$ (con $M_i \geq 1$) un messaggio

i blocchi M_i non blocchi di ugual lunghezza

Algoritmo di funzione di Hash:

$$H(M_1) = E(K, M_1)$$

$$H(M_1 \dots M_i M_{i+1}) = E(K, H(M_1 \dots M_i) \oplus M_{i+1}) \text{ per } i=1 \dots m-1$$

$$H(M_1 \dots M_i M_{i+1}) = E(K, E(K, M_1 \dots M_i) \oplus M_{i+1}) \text{ per } i=1 \dots m-1$$

$$H(M_1 \dots M_i M_{i+1}) = \{ \{M_1 \dots M_i\}_K \oplus M_{i+1} \}_K \text{ per } i=1 \dots m-1$$

Si dimostri che lo schema non è sicuro mostrando che un dato messaggio $A_1 A_2$ ed un blocco arbitrario B_1 , TESI:
è possibile determinare un blocco B_2 tale che
 $H(B_1, B_2) = H(A_1, A_2)$ collusione debole

soluzione

$$H(A_1 A_2) = \{ \{A_1\}_K \oplus A_2 \}_K \quad e \quad H(B_1, B_2) = \{ \{B_1\}_K \oplus B_2 \}_K$$

Quindi è sufficiente trovare un blocco B_2 tale che

$$\{B_1\}_K \oplus B_2 = \{A_1\}_K \oplus A_2$$

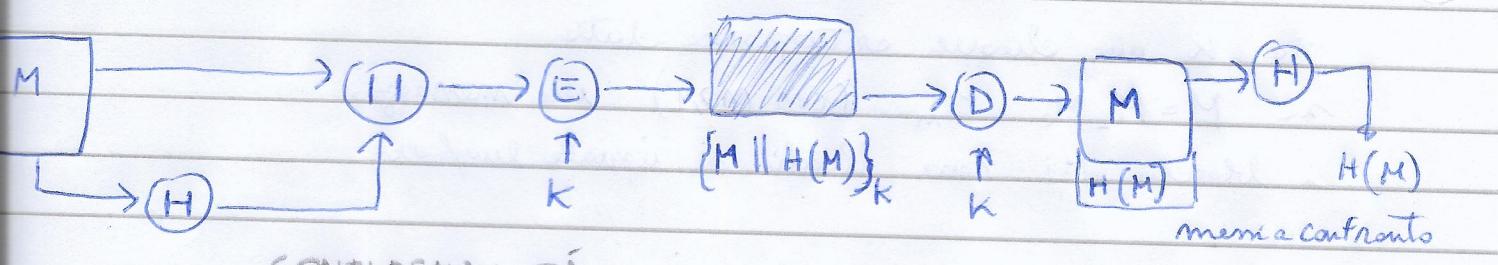
rimuovendo $\oplus \{B_1\}_K$ entrambi i lati

~~$$\{B_2\}_K \oplus \{B_1\}_K \oplus B_2 = \{B_1\}_K + \{A_1\}_K \oplus A_2$$~~

Quindi se $B_2 = \{B_1\}_K \oplus \{A_1\}_K \oplus A_2$

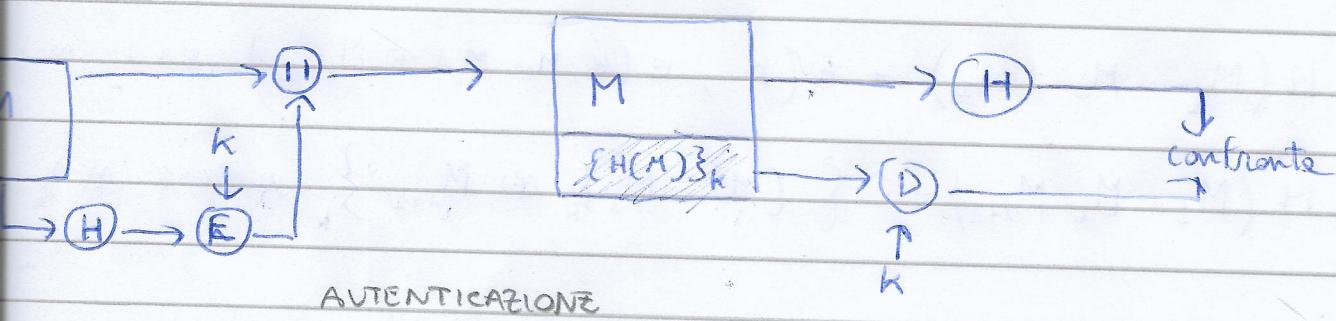
allora $H(B_1, B_2) = H(A_1, A_2)$

es 4 20-6-2006

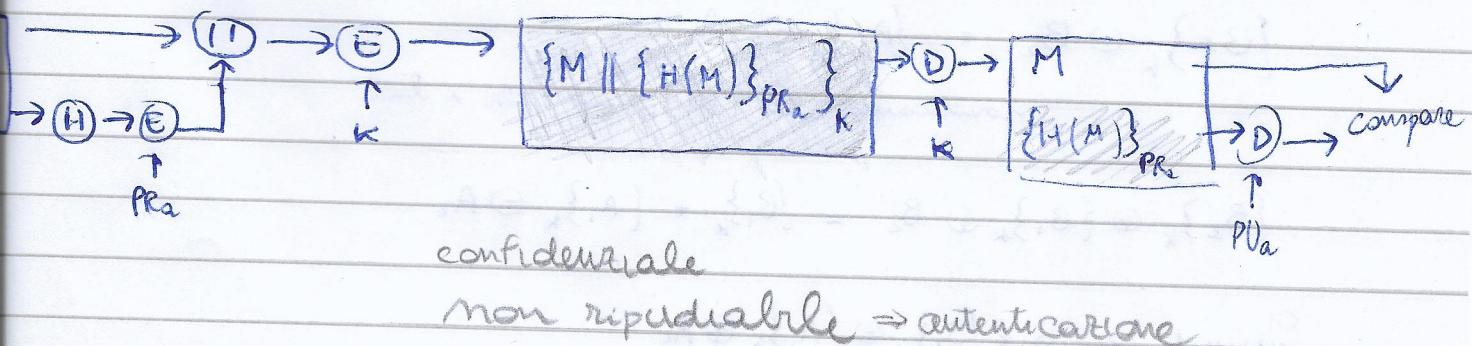
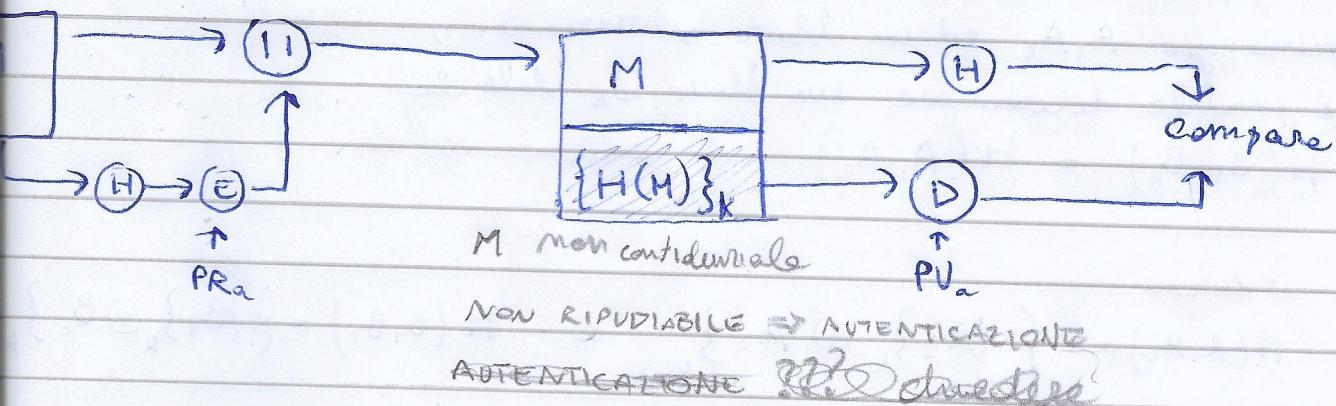


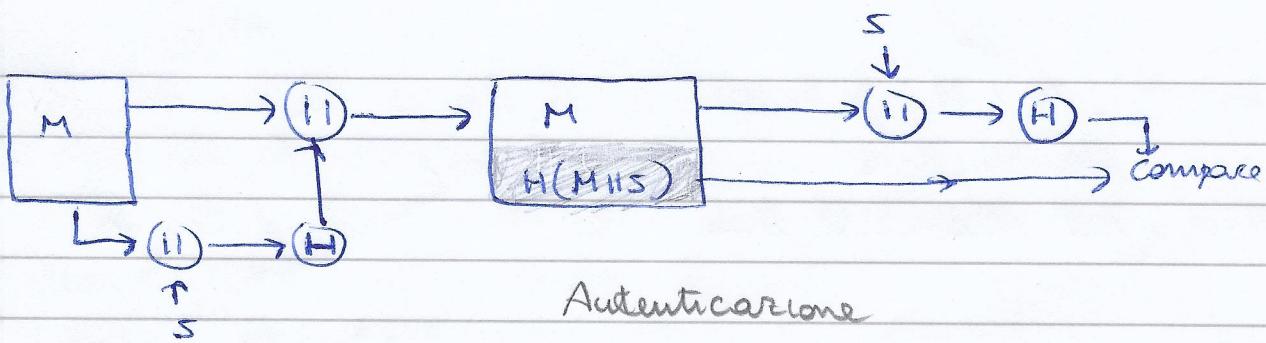
CONFIDENZIALITÀ

AUTENTICAZIONE



AUTENTICAZIONE





PROTOCOLLI DI SICUREZZA

- mettere in sicurezza reti di sensori
- sincronizzare due dispositivi rete filo
- controllo di accessi di una rete privata
- sistemi di pagamento online

alcuni esempi in cui
vengono utilizzati protocolli di sicurezza

Per risolvere si utilizzano protocolli come IPsec, SSH, SSL, Kerberos etc

Un protocollo è un insieme di regole (convenzioni) che determinano
lo scambio di messaggi tra due entità.

Un protocollo di sicurezza usa meccanismi critografici per addossare
requisiti di sicurezza.

FORMALISMO

Nomi : A per Alice, B per Bob

Chiavi : K e chiave inversa K^{-1}

Chiavi simmetriche : $\{M\}_{K_{AB}}$ dove K è la chiave comune di A e B

FIRMA : $\{M\}_{K^{-1}}$ per esempio se firmo con la chiave privata di A : $\{M\}_{K_A^{-1}}$

CIFRATURA : $\{M\}_K$ per esempio se cifro con la chiave pubblica di A : $\{M\}_{K_A}$

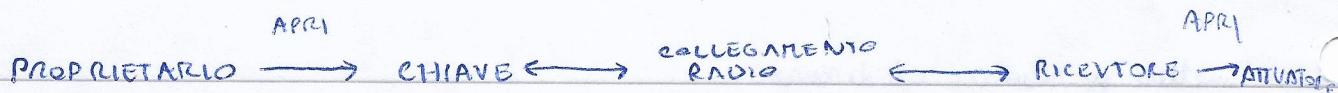
NONCE : numero casuale usato una volta : N_A

TIMESTAMP : T denota il tempo in cui una chiave reade

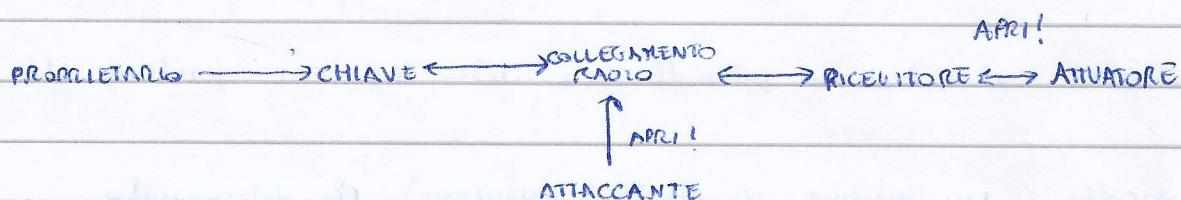
CONCATENAZIONE di MESSAGGI : $\{M_1, M_2\}$, $M_1 \parallel M_2$, $[M_1, M_2]$

ESEMPIO CHIAVE ELETTRONICA PER AUTO

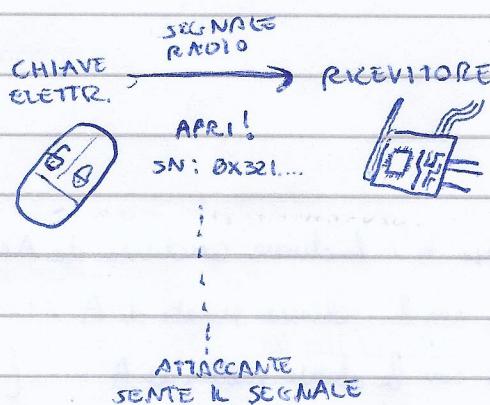
MANAGERS
SICURO



1° OBIETTIVO: Il ricevitore attiva l'attuatore solo se il proprietario ha premuto il pulsante delle chiavi.



Assumiamo che il codice numero chiave SN sia ^{un segreto} la chiave condensa tra CHIAVE ELETTR. e RICEVITORE.

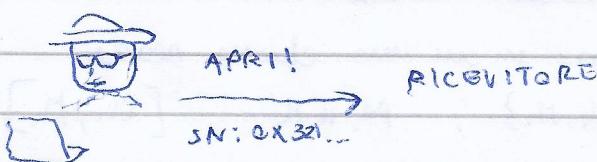


PESCARA IDEA:

L'attaccante può sentire il SN e utilizzarlo per aprire le macchine in futuro.

- MANCA AUTENTICAZIONE

- IL SN non è segreto

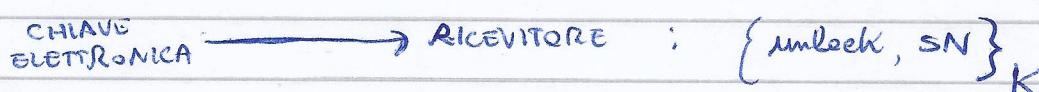


Il ricevitore non può controllare l'autenticità delle richieste

TERZA

IDEA: proteggere le richieste di SN

- La chiave Elettronica cifra la richiesta con la chiave condensa K e manda ciò al ricevitore.



PROBLEMA: L'attaccante può sentire il messaggio cifrato e replicarlo per aprire le macchine.

SPIEGAZIONE: abbiamo aumentato la sicurezza di SN ma l'attacco è ancora possibile

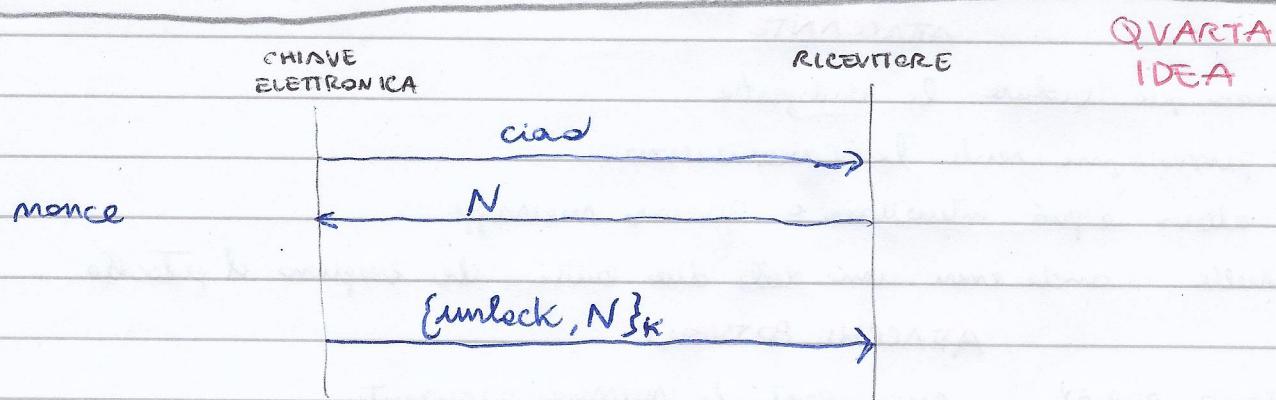
OBIETTIVO RIVISTO: Il ricevitore apre la macchina solo se il proprietario ha premuto DI RECENTE la chiave elettronica.

- Ora le chiavi elettroniche citron un timestamp con chiave condensa K e manda il risultato a RICEVITORE

CHIAVE ELETTRONICA → RICEVITORE : $\{ \text{unlock}, T \}_K$

L'ATTACCANTE non può più copiare - incollare il messaggio per aprire la macchina perché ~~dovendo~~ la richiesta di apertura è valida solo una volta.

Il SN non serve più.



Il numero nonce N viene usato per far sì che la richiesta non venga eseguita dall'attaccante infatti la richiesta vale solo una volta

NOTAZIONE

CH.EL. → RIC. : chao

RIC. → CH.EL : N

CH.EL → RIC. : $\{ \text{unlock}, N \}_K$

NOTAZIONE DI ALTRE IMPLEMENTAZIONI

$I \rightarrow R : \{ I, T_I, K \}_{KR}^{Timestamp}$

$R \rightarrow I : \{ R, I \}_{KR}$

multi

REGOLE

- Le due entità hanno le loro chiavi private e le chiavi pubbliche altrui
- Le entità possono generare e controllare timestamp eNonce e utilizzarli
- Le entità devono implementare il protocollo

OBIETTIVI

Autenticare i messaggi

ATTACCANTE

non può violare la crittografia

e' passivo ma non le commuiscione

è attivo e può intercettare e generare messaggi

potrebbe anche essere una delle due entità che eseguono il protocollo

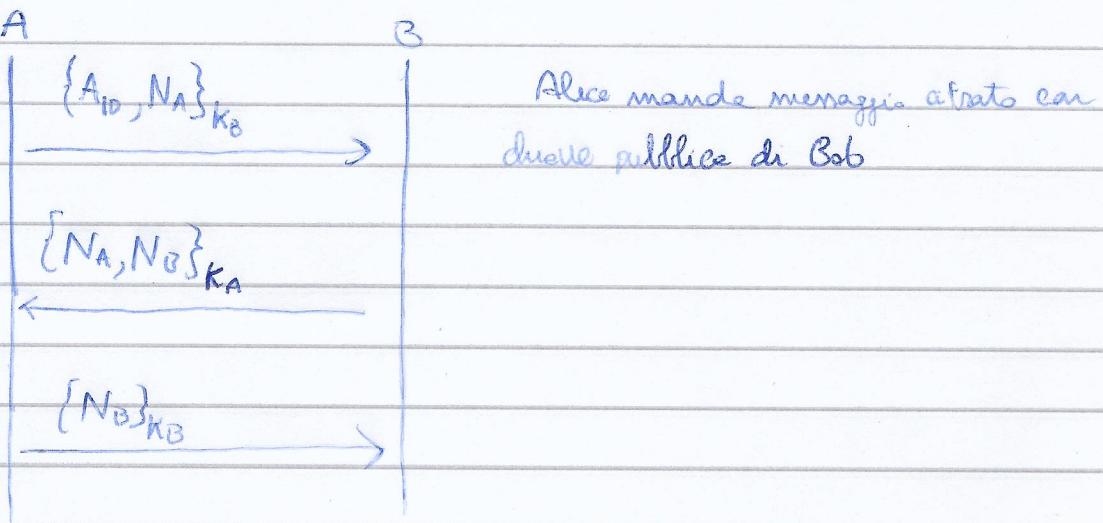
ATTACCHI POSSIBILI

> ATTACCO REPLAY: rinvia parti di messaggi precedenti

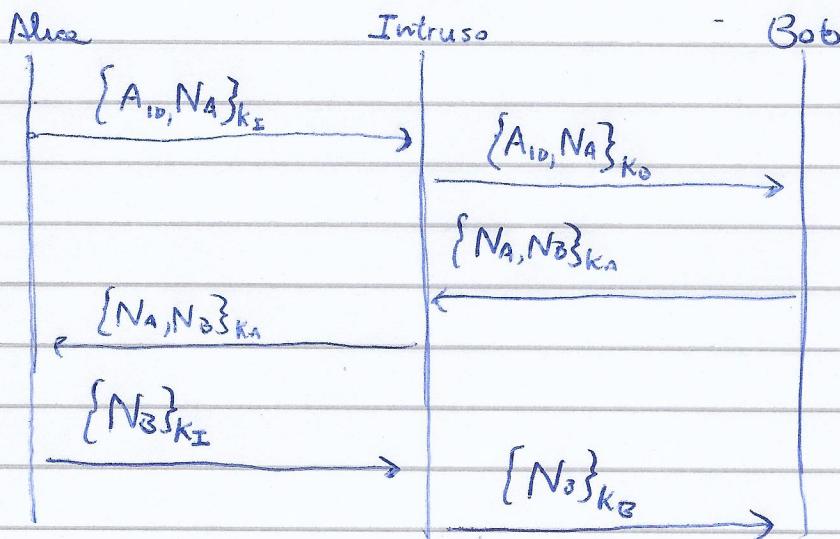
> ATTACCO VANO IN MEZZO: $A \leftrightarrow M \leftrightarrow B$

> ATTACCO RIFLESSIONE: rinvia informazione al mittente

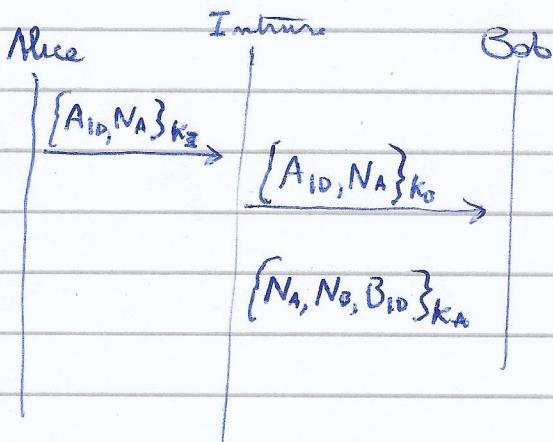
> ATTACCO DIFETTO DI TIPO: sostituisce il tipo nel messaggio; per esempio un suo nome al posto del nome



ATTACCO



MIGLIORAMENTO



PROTOCOLLO NSPK

(1970)

OBIETTIVO: autenticazione reciproca

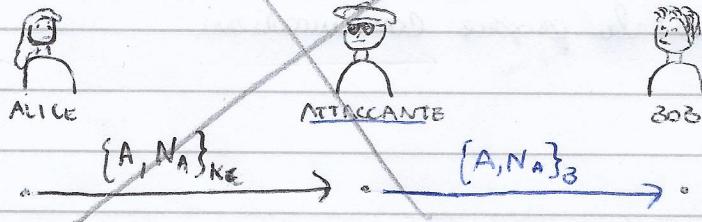
$$\begin{array}{l} A \rightarrow B : \{ A_{ID}; N_A \}_{K_B} \\ B \rightarrow A : \{ N_A; N_B \}_{K_A} \\ A \rightarrow B : \{ N_B \}_{K_B} \end{array}$$

^{B-A}
cioè non Alice, questo è il mio monaco N_A

cioè Alice non; Questo è il tuo monaco,
dato che lo potrò decifrare, e non per fare
Bob. Ecco il mio monaco N_B

cioè Bob sa che questo è il tuo monaco, dato che
lo potrò decifrare, non per forza Alice

ATTACCO ALL'NSPK



NEEDHAM-SHROEDER A CHIAVE PUBBLICA (INSICURO)

Alice e Bob sono due entità che comunicano in rete su un collegamento non sicuro

S è un server di cui Alice e Bob si fidano, e quindi distribuisce le chiavi pubbliche

K_{PA} chiave pubblica di Alice, K_{SA} chiave segreta di Alice

K_{PB} e K_{SB} chiavi di Bob

K_{PS} e K_{SS} chiavi del server

Alice e Bob usano le chiavi pubbliche per cifrare e usano le chiavi private per decifrare

Il server usa le chiavi private K_{SS} per cifrare e usa le chiavi pubbliche K_{PS} per decifrare così facendo il server fornisce le proprie comunicazioni

NEEDAM SHROEDER CHIAVE SEGRETA

Alice (A) e Bob (B) sono due entità che comunicano su un canale non sicuro
 S è un server di cui Alice e Bob si fidano
 K_{AS} è una chiave simmetrica nota esclusivamente a A e S
 K_{BS} è una chiave simmetrica nota esclusivamente a B e S
 K_{AB} è una chiave simmetrica di sessione generata da S
 N_A e N_B sono dei monaci, ovvero numeri casuali da usare una volta

Descrizione

$A \rightarrow S : A_{10}, B_{10}, N_A$

A pedisce un messaggio al server con la mia identità, identità di Bob e un monaco N_A

Il server S genera la chiave di sessione K_{AB} ed invia ad Alice:

- la chiave di sessione K_{AB} appena generata
- le coppie (K_{AB}, A_{10}) cifrato col la chiave K_{BS}
- Il monaco N_A ancora col il messaggio è franco
- B_{10} rimette ad Alice di sapere con chi A sta condividendo la chiave

Tutto questo è cifrato con la chiave K_{AS} , nota solo ad Alice e al Server

$S \rightarrow A : \{ N_A, K_{AB}, B_{10}, [K_{AB}, A]_{K_{BS}} \}_{K_{AS}}$

Alice comunica a Bob le chiavi di sessione e l'identificatore di A, il tutto cifrato con la chiave K_{BS} .

$A \rightarrow B : \{ K_{AB}, A \}_{K_{BS}}$

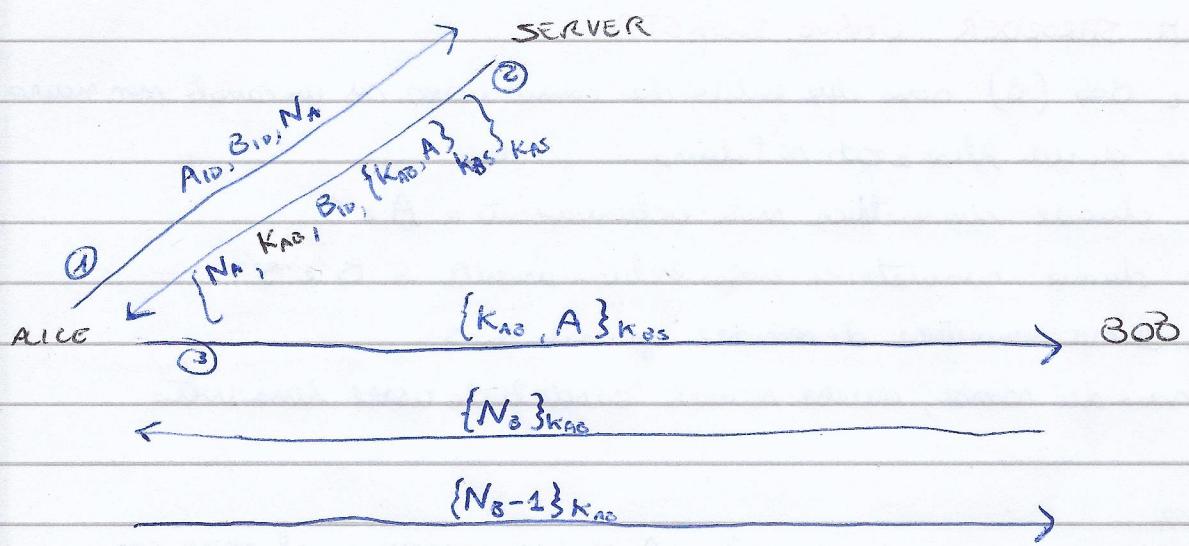
B può decifrare il messaggio e il fatto che me lo ha cifrato da un'entità fidata, il server, lo rende autentico.

Bob risponde ad Alice con un monaco cifrato con la chiave di sessione K_{AB}

$B \rightarrow A : \{ N_B \}_{K_{AB}}$

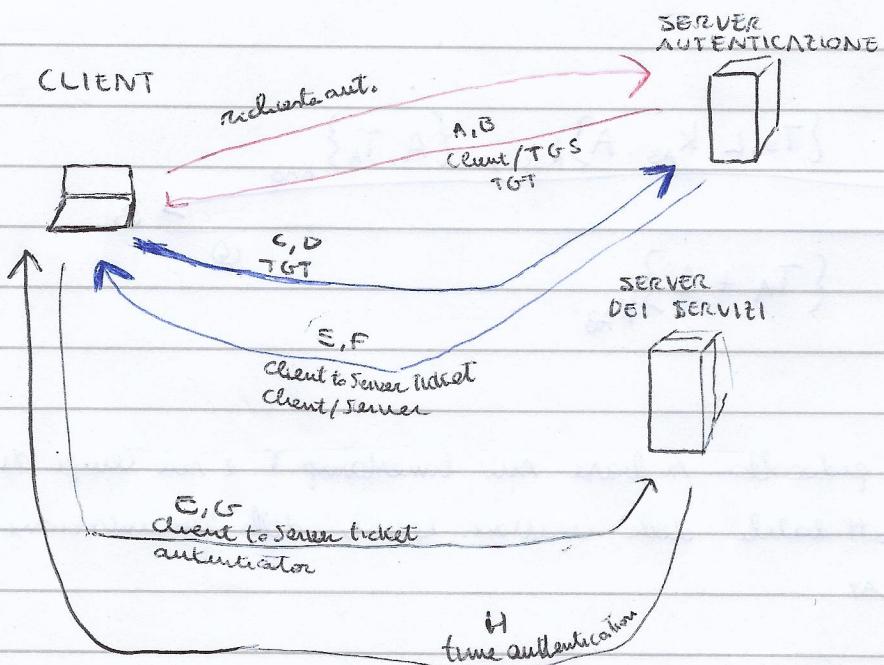
Alice decifra il monaco di Bob, lo modifica, lo cifra nuovamente, e lo rispedisce indietro provando così che è ancora attiva e in possesso delle chiavi

$A \rightarrow B : \{ N_B - 1 \}_{K_{AB}}$



PROTOCOLLO KERBEROS

Kerberos è un protocollo per l'autenticazione tramite crittografia che permette a diversi terminali di comunicare su una rete sicura provando le proprie identità e cifrando i dati.



Key Distribution Server (KDC)

Authentication Server (AS)

Ticket Granting Server (TGS)

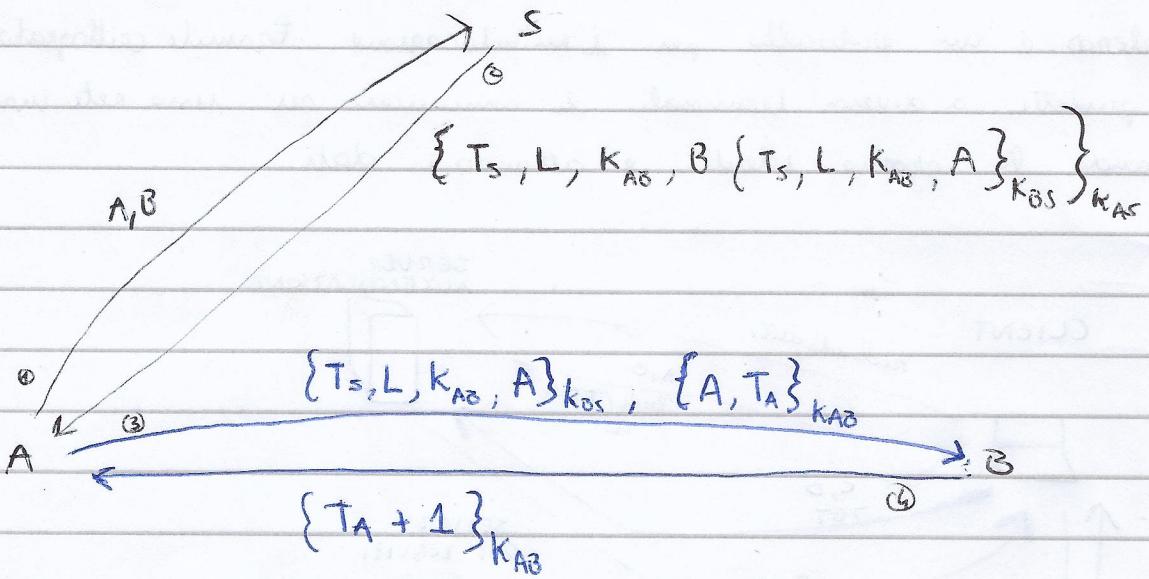
si divide in due parti

da un database delle chiavi segrete

i ticket servono per provare l'identità degli utenti

SS : Service Server

DESCRIZIONE



Le numerose del protocollo si basa sui timestamp T e sui tempi di vita L come indicatori affidabili della creazione recente delle comunicazioni per evitare attacchi REPLAY

OPERAZIONI DI KERBEROS

VIENTE : autenticazione di base

- un cliente inserisce username e password

CLIENT: AUTENTICAZIONE AS

1: il client manda un messaggio non crittato richiedendo i servizi per l'utente

2: l' AS controlla se il client è nel suo database . Se lo è invia due messaggi al client

messaggio A: chiave di gestione client-TGS crittata usando la chiave privata dell'utente

messaggio B: Ticket-Granting TICKET che include l'identificativo del client,

l'indirizzo di rete, il tempo di validità del ticket e le chiavi di sessione client-TGS

Il ticket-granting ticket è critto col la chiave segreta di TGS

3: Quando il client riceve il messaggio A e il messaggio B , decifra il messaggio A (chiave utente) ottenendo la CHIAVE DI SESSIONE client-TGS , queste chiavi e usate per le successive comunicazioni col TGS

Il client non può decifrare il messaggio B, che è stato crittato con la chiave segreta di TGS

Client: AUTENTICAZIONE TGS

1. Quando richiede dei servizi, il client invia i seguenti due messaggi a TGS

MESSAGGIO C: composto dal Ticket-Granting Ticket (mandatagli dall'AS
nel messaggio 3) e dell'identificativo del servizio richiesto

MESSAGGIO D: autenticatore (formato da Client ID e timestamp)
cifrato usando la CHIAVE di SESSIONE client-TGS

Ricevendo i messaggi C e D, il server TGS decifra il messaggio C con le proprie chiavi, dal Ticket-Granting Ticket, il server TGS estrae la chiave di sessione TGS. Adesso il server TGS va a decifrare il messaggio D con la chiave di sessione TGS.

A questo punto il server TGS invia i seguenti messaggi al client:

MESSAGGIO E: ticket client-server che include l'ID del client, indirizzo di rete del client, il periodo di validità e la chiave di sessione client-server.
Il ticket client-server è cifrato con le chiavi del server effettivo.

MESSAGGIO F: chiavi di sessione client-server cifrate usando la chiave di sessione client-TGS.

Client: AUTENTICAZIONE SS

Dopo aver ricevuto i messaggi E e F, il client può autenticarsi per il SS.

MESSAGGIO G:

MESSAGGIO H: un nuovo autenticatore, che include l'ID del client, l'indirizzo del client, il periodo di validità, la chiave di sessione client-server.
L'autenticatore è cifrato con la chiave di sessione client-server.

Il SS decifra il messaggio E (ticket client-server) con la sua chiave e manda il rispettivo messaggio al client.

MESSAGGIO I: il timestamp trovato nell'autenticatore, aumentato di uno e cifrato con la chiave di sessione client-server.

ESERCIZI PROTOCOLLI DI SICUREZZA

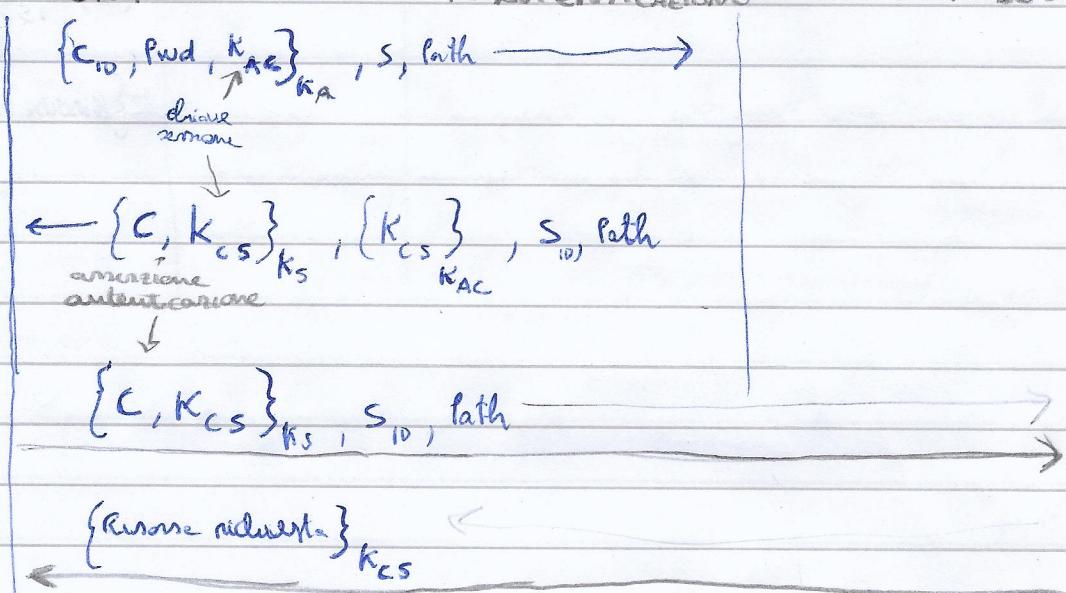
8 luglio 2008

C è il client e vuole accedere a risorse dettate S, identificate da Path. Perme di accedere alle risorse, C deve autenticarsi presso un server di Autenticazione (A) per poi farne richiamare un'autorizzazione di autenticazione, ovvero il messaggio $E(K_S, [C, K_{CS}])$, che poi presenterà al server del servizio S insieme alla richiesta delle risorse.

C: CLIENT

A: SERVER AUTENTICAZIONE

S: SERVER



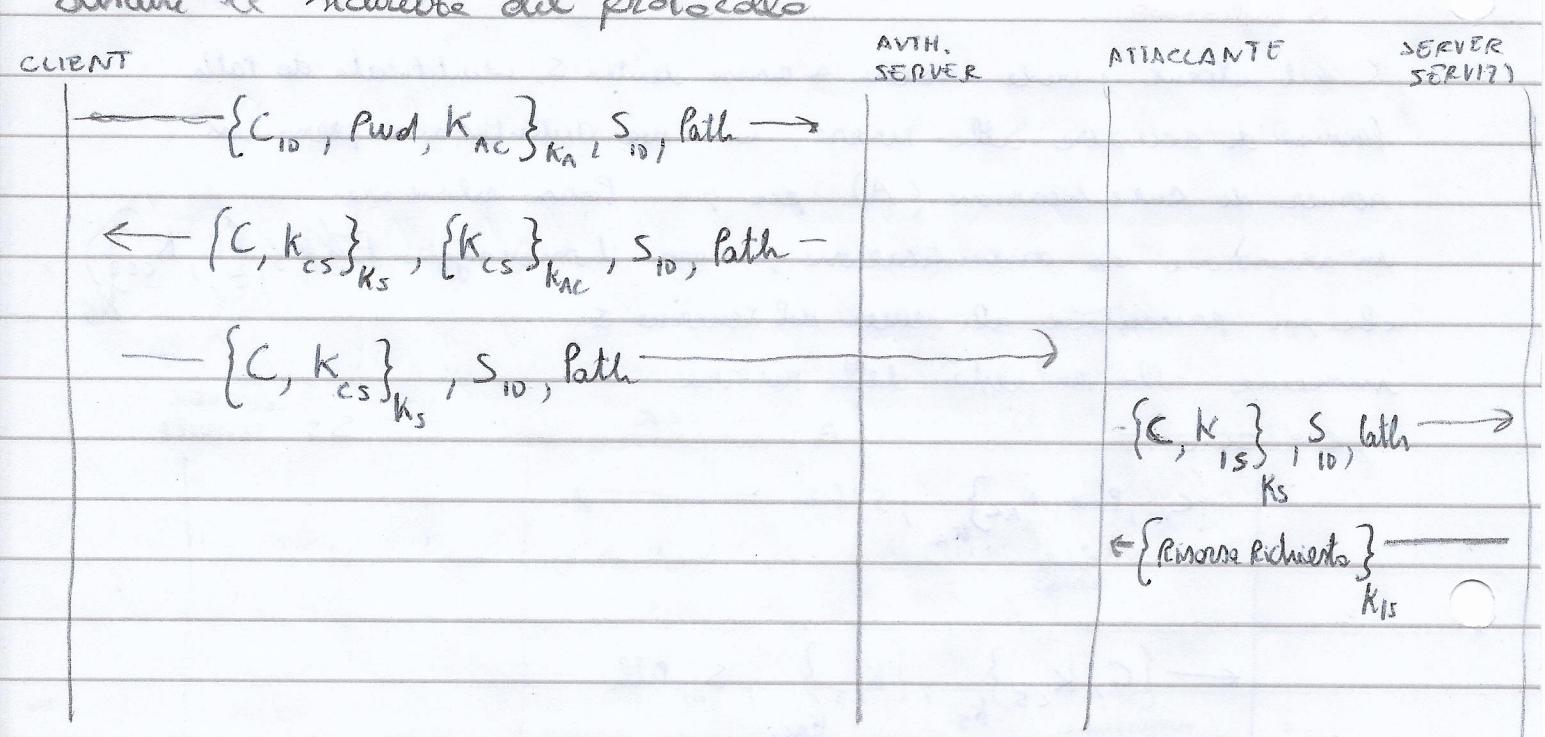
Quali sono le proprietà di sicurezza che darebbe garantire un protocollo di questo tipo?

Sia A che S devono autenticare C, ovvero se A e S completano la loro parte del protocollo, C deve aver inviato la propria parte del protocollo con gli stessi valori di S e Path.

C deve autenticare S

Le risorse registe identificate da Path deve rimanere segrete

avverte le incertezze del protocollo



Si consideri il seguente protocollo per l'autenticazione di due agenti in posesso di una chiave simmetrica condivisa k

$A \rightarrow B : N \oplus k$

$B \rightarrow A : N$

A genera un nonce, la somma XOR alle chiavi e manda a B

Quando B riceve il messaggio, fa $(N \oplus k) \oplus k$ ed ottiene il nonce.

Poi manda il nonce ad A.

INDICA LE PROPRIETÀ FONDAMENTALI DI UN PROTOCOLLO DI AUTENTICAZIONE DI DUE AGENTI

- A deve autenticare B (assicurarsi che ne veramente lui)
- le chiavi condivise deve rimanere segrete

LE PROPRIETÀ SONO SODDISFATTE?

Se l'attaccante intercette $(N \oplus k)$ e N può calcolare $K = (K \oplus N) \oplus N$

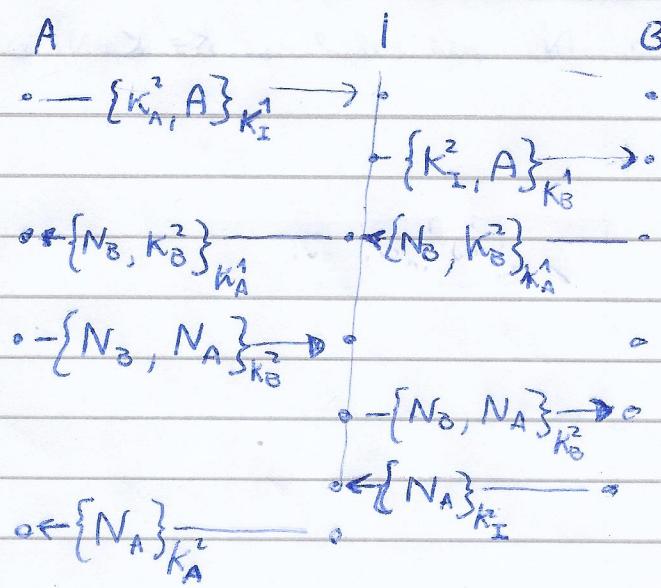
es 6 6-5-2006

- ⑥ $A \rightarrow B : \{K_A^2, A\}_{K_B^1}$ A manda confidencialmente la sua chiave pubblica e il suo ID a B
- ⑦ $B \rightarrow A : \{N_B, K_B^2\}_{K_A^1}$ B manda confidencialmente le sue chiavi pubbliche e un nonce N_B per autenticare A
- ⑧ $A \rightarrow B : \{N_B, N_A\}_{K_B^2}$ A invia il nonce N_B per autenticarsi con un nuovo nonce. Il messaggio è confidionale
- ⑨ $B \rightarrow A : \{N_A\}_{K_A^2}$ B si autentica scrivendo N_A in modo confiduale

Pareossalmente alle tue del protocollo A e B si sono autenticati.

A dovrebbe essere certo che K_B^1 è una chiave pubblica di B
e B dovrebbe essere certo che K_A^2 è una chiave pubblica di A.

ATTACCO



Il primo messaggio è lo stesso del protocollo,
 $\{K_A^2, A\}_{K_B^1}$ questo messaggio ammette che K_A^2 è una nuova chiave pubblica di A, ma B non ha modo di verificare l'autenticità di tale informazione.

La soluzione è cambiare il messaggio in $\{K_A^1, A\}_{inv(K_A^1)}$
 $inv(K_A^1)$ è la chiave privata corrispondente alla chiave pubblica K_A^1
questo messaggio è a tutti gli effetti un certificato emesso da A
(mentre la chiave pubblica K_A^1) in cui viene aperto da K_A^2 è una nuova chiave pubblica di A

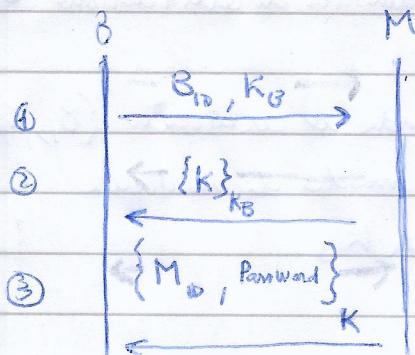
016 20.6.2006

$B \rightarrow M : B_{ID}, K_B$

Le istanze Base manda a Mobile le sue chiavi pubbliche

$M \rightarrow B : \{K\}_{K_B}$

$M \rightarrow B : \{M, P\}_K$



Le istanze Base manda a Mobile le sue chiavi pubbliche e da me ID.

Mobile manda a Base un messaggio crittato con chiave pubblica di Base. Solo Base potrà decifrarlo e il messaggio contiene una chiave da usare per crittografia simmetrica.

Base manda a Mobile un messaggio crittografato che contiene la Password

Se come B non è certo che K provenga da M

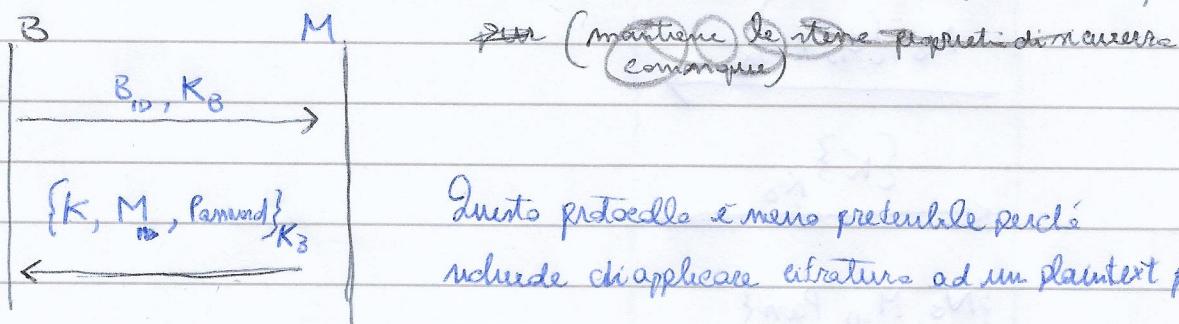
(cattivo messaggio crittato con chiavi pubbliche garantisce confidenzialità ma non autenticazione perché chiunque può generarlo)

Nel passo ③ Mobile manda a Base

un messaggio crittato con crittografia simmetrica

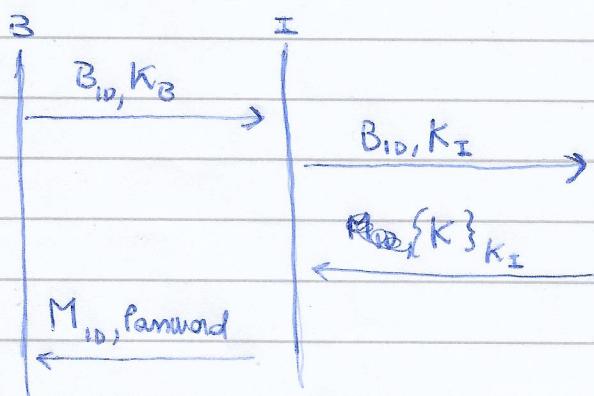
contenente M_{ID} e Password, questo messaggio garantisce anche autenticazione

Suggerire perché questo protocollo è potenzialmente meno prevedibile,



Questo protocollo è meno prevedibile perché include chi appurare cifratura ad un plaintext più lungo

ATTACCO POSSIBILE



Base manda le sue chiavi pubbliche a Mobile, ma I intercetta il messaggio.

I manda a Mobile le chiavi pubbliche di I, ma gli dice di essere B

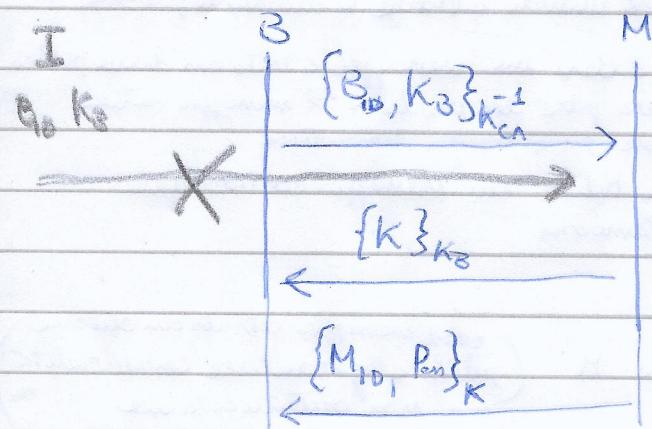
M manda a Base le sue chiavi simmetriche che viene erroneamente cifrate con le chiavi dell'utente invece che con le chiavi di B

L'utente è in grado di stabilire la Password di remoto

Migliore il protocollo sapendo che

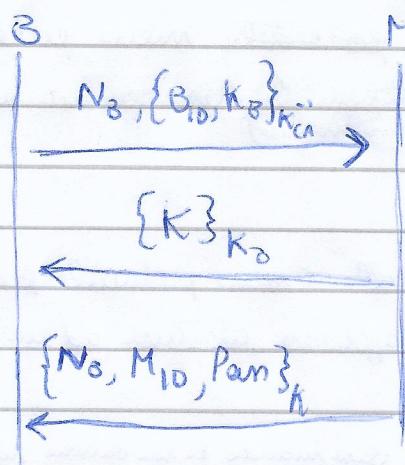
B possiede il certificato $\{B, K_B\}_{K_{CA}}^{-1}$ emesso da un'autorità di certificazione CA; il certificato è relativo alla propria chiave pubblica K_B

M possiede K_{CA} ovvero la chiave pubblica dell'autorità di certificazione CA



Se B manda il certificato $\{B_{ID}, K_B\}_{K_{CA}}^{-1}$ allora M ha modo di verificare l'autenticità di K_B .

Il protocollo è ancora sensibile a REPLY ATTACK



MECANISMO DI SICUREZZA

il meccanismo di sicurezza definisce le funzioni di basso livello (^{software}
^{hardware}) che implementano i controlli imposti dalla politica di sicurezza ed espressi
formalmente nel modello di sicurezza

ESEMPIO DI POLITICA DI SICUREZZA

Uno studente ha pieno accesso all'informazione da lui creata.
Gli studenti non hanno accesso all'informazione degli altri studenti a meno che non
 ricevano esplicitamente il permesso : gli studenti possono reperire ^{de} una pre-definita
 lista di file.

TIPI DI POLITICHE

DISCRETIONALI (DAC)

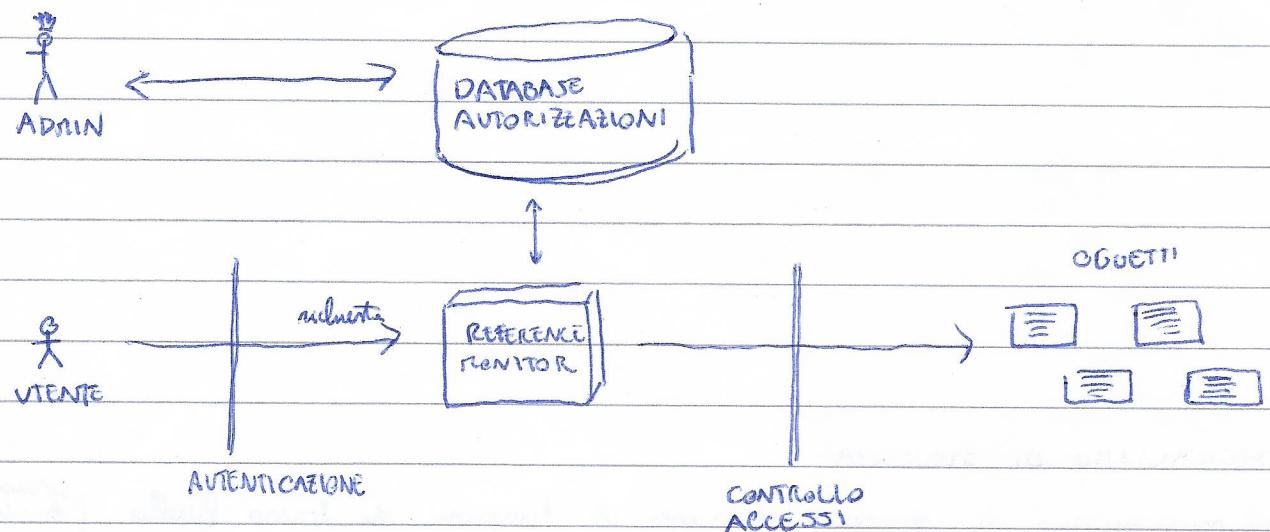
Varate su autorizzazione, le queste politiche di sicurezza controllano
l'accesso basato sull'identità del richiedente e l'accesso basato su
regole che descrivono cosa il richiedente possa fare.

OBLIGATORIE (MAC)

queste politiche controllano l'accesso basato su regole obbligatorie
determinate da

BASATE SUI RIVOLI

queste politiche controllano l'accesso dipendente dai ruoli che gli utenti
hanno all'interno del sistema e su ~~dipendente~~ l'accesso dipendente da



Ogni richiesta passa attraverso il "reference monitor" che deve restituire le permissio:

- TAMPER-PROOF: a prova di manipolazione, non è possibile alterarlo
- NON OLTREPASSABILE:
 - deve essere contenuto in una zona sicura del sistema software
 - deve avere piccolo

STATO DI PROTEZIONE

- > Uno stato di un sistema è l'insieme di valori corrente di tutta la memoria dei componenti del sistema (register, allogenzi, indirizzi...)
- > Proteggere lo STATO DI PROTEZIONE di un sistema protegge un insieme degli indirizzi

Esempi di Stati di Protezione

> FILE SYSTEM

una parte dello STATO DEL SISTEMA determina che sta

DISCRETIONARY ACCESS CONTROL (DAC)

Il DAC è un tipo di controllo di accesso definito come una forma per limitare l'accesso a contenuti appartenente a soggetti e/o gruppi. I controlli sono discretionali in quanto un soggetto con un determinato permesso d'accesso può trasmettere questi permessi a qualcun altro soggetto.

Un sistema complesso può presentare sia DAC che MAC (^{mandatory} _{access control})

In tal caso l'accesso alle risorse che gli utenti possono scambiarsi i DAC mentre il MAC è un secondo livello di controllo che pone vincoli al controllo DAC.

IMPLEMENTAZIONE DEL DAC CON PROPRIETARIO

Ogni oggetto ha un proprietario che controlla ~~assegna~~ il permesso di accesso a quell'oggetto.

I proprietari, grazie all'creazione del DAC, hanno il potere di creare decisioni sulle politiche e di designare caratteristiche di maniera.

esempio file system

u user proprietario del file

g group intenti che non membri del gruppo del file

s other altri intenti

IMPLEMENTAZIONE DEL DAC CON COMPETENZE

un esempio di implementazione del DAC con COMPETENZE sono i "livelli di capacità"

- i soggetti possono trasferire i loro accessi ad altri soggetti
- un soggetto non può accedere a una risorsa prima di aver ricevuto l'autorizzazione
- i soggetti generalmente non possono accedere a tutte le risorse

MATRICE DI CONTROLLO ACCESSI

La matrice degli accessi è un modello astratto di STATO DI PROTEZIONE che caratterizza i diritti di ogni soggetto verso ogni oggetto del sistema.

La matrice degli accessi è una tabella con una riga per i soggetti e una colonna per gli oggetti

O insieme degli oggetti

S insieme dei soggetti

P insieme dei privilegi (diritti)
(permessi)

$P(s, o)$

esempio

utente utente 1		utente utente 2	FILE 1	dispositivo
Ruolo 1	read, write, execute, delete	execute	read	write
Ruolo 2	read	read, write		

MODELLO HRU

il modello HRU è uno schema che specifica politiche di sicurezza e riguarda l'integrità dei diritti di accesso in un sistema.

CARATTERISTICHE

Sono disponibili delle procedure per modificare i diritti di accesso di un soggetto sull'oggetto o

DESCRIZIONE

Il modello HRU prevede un sistema di sicurezza con un insieme di permessi P e un insieme di comandi C.

Una "configurazione" del sistema è la Trippla (S, O, M) di cui uno STATO
insieme soggetti oggetti matrice accessi

La matrice M contiene una riga per ogni soggetto e una colonna per ogni oggetto

I comandi sono composti da operazioni primitive e possono avere anche come per le liste di pre-condizioni che stabiliscono che determinati permessi P sono presenti per determinate coppie (S, O)

Le indirizzi primitive possono modificare la MATRICE degli ACCESSI
aggiungendo e rimuovendo PERNESI, JOUETTI, OGGETTI

TRANSIZIONE GENERICA

comando è (x_1, \dots, x_k)

if riga₁ in $M(x_{s1}, o_{s1})$
then operazione,

ESEMPIO TRANSIZIONE

comando create_file (s, o)

create o

enter Own into $M(s, o)$

enter Read into $M(s, o)$

enter Write into $M(s, o)$

end

comando confer.until (s₁, s₂, o)

if Own ∈ $M(s_1, o)$

then enter Write into $M(s_2, o)$

end

OPERAZIONI PRIMITIVE

Operazione

condizioni

Nodus State

enter p into $M(s, o)$

$s \in S$

$s' = s$

$o \in O$

$o' = o$

$M'(s, o) = M(s, o) \cup p$
altri celle inalterate

Mandatory Access Control (MAC)

Il controllo di accesso vincolato (MAC) è un tipo di controllo di accesso alle risorse di un sistema in cui il sistema vincola le capacità di un soggetto di eseguire diverse operazioni su un oggetto **descrizione**.

un soggetto può essere un processo o un thread, un oggetto può essere un file, una cartella, porte TCP/UDP e altri.

Il MAC si basa sulle specifiche a priori di una serie di attributi di sicurezza per soggetti e oggetti.

Quando un soggetto tenta l'accesso a un oggetto il sistema avrà una regola di autorizzazione che esaminerà gli attributi di sicurezza di soggetti e oggetti e decide se l'accesso può essere ammesso o deve essere rifiutato.

Un MAC può essere adottato da un sistema operativo o da un database

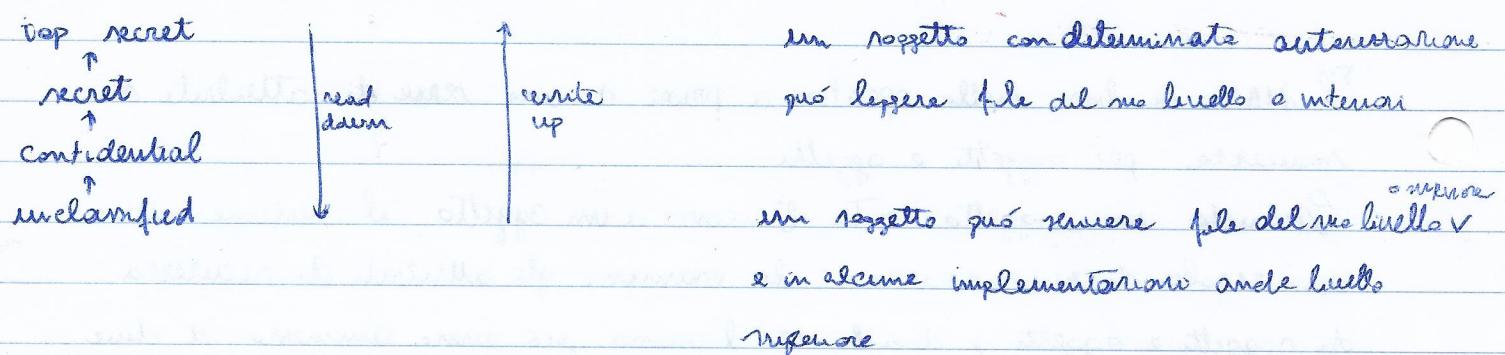
caratteristiche

- > Le decisioni sul controllo di accesso dipendono delle etichette di sicurezza che indicano il livello di sicurezza dell'oggetto
- > Non sempre i soggetti possono trasferire i loro privilegi di accesso
- > Quando è messo nel contesto multietate, ogni oggetto ha un livello di segretezza (top secret, etc) e un soggetto utente può solo accedere a oggetti di sicurezza uguale o inferiore

MODELLI MAC

- I modelli pensano avere politiche di sicurezza per la confidentialità (BLP)
 - per l'integrità (Biba, Clark Wilson)
- Alcuni modelli si applicano ad ambienti con politiche STATICHE (BLP)
altri considerano cambiamenti DINAMICI dei privilegi di accesso.
- Ci sono modelli INFORMATIVI (Clark Wilson) o formali (BLP, HRV)

MAC: ordine lineare



MAC: reticolo (LATTICE) dei livelli di sicurezza

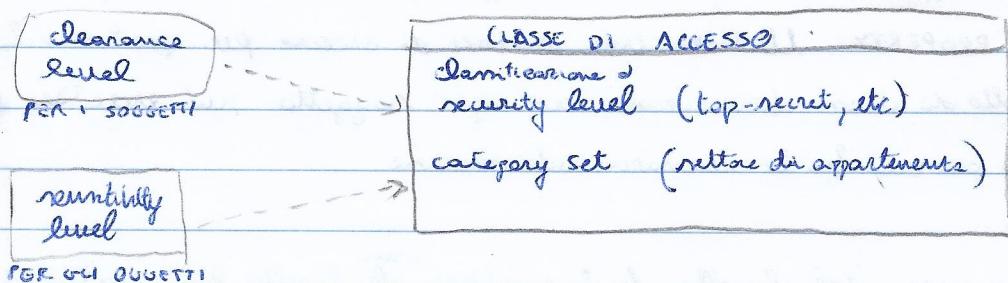
Date due oggetti allo stesso livello di sicurezza, qual è il minimo livello di sicurezza che deve avere un oggetto per leggerli entrambi? Qual è il ~~minimo~~ livello di sicurezza che un oggetto deve avere per scrivere su entrambi?

Si concentra su riservatezza dei dati e accesso a informazioni classificate

Modello Bell-Lapadula (BLP)

ai soggetti sono assegnati i "clearance level" (livelli autorizzazione) agli oggetti sono assegnati i "sensitivity level" (livelli sensibilità)

I due livelli sono generalmente chiamati "classe di accesso"



Un soggetto può accedere ai vari oggetti secondo i METODI DI ACCESSO

METODI DI ACCESSO: Read, Write, Append (^{aggiunta di contenuto senza lettura}), execute

STATO SICURO

Ogni transazione preserva la sua sicurezza partendosi da uno stato sicuro ad un altro stato sicuro. Il passaggio da uno stato all'altro è definito dalle funzioni di transizione.

Uno stato è definito sicuro se tutti gli oggetti e i soggetti sono conformi alle politiche di sicurezza.

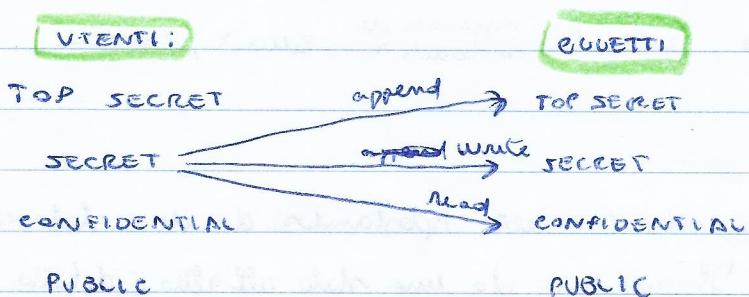
Per determinare se un modo di accesso è permesso, viene confrontato il "clearance level" del soggetto con la classificazione dell'oggetto.

Le scheme di accesso sono espressi con un refidol: ogni operazione è controllata da un reference monitor; l'operazione è concerne solo se il sistema andrebbe a trovarsi in uno stato sicuro dopo l'operazione.

BLP determina due regole MAC e una regola DAC

- > SIMPLE SECURITY PROPERTY: un soggetto può accedere a un oggetto solo se il
(read down) livello di maturità del soggetto è maggiore o uguale a quello dell'oggetto
- > STAR PROPERTY: se un soggetto ha un livello di maturità inferiore all'oggetto,
(append up) può accedere all'oggetto solo per operazioni di "append"
(write equal) se un soggetto ha un livello uguale all'oggetto, può accedervi solo per operazioni WRITE
(read down) se un soggetto ha un livello ^{SUPER}LOW dell'oggetto, può accedervi solo per operazioni READ
- > DISCRETIONARY SECURITY PROPERTY: utilizza una matrice di accesso per specificare il controllo di accessi discettionale, cioè ogni soggetto può eseguire gli stessi accessi per cui ha le necessarie autorizzazioni.

Il trasferimento di risorse dal livello high-sensitivity ^{a un} al livello low-sensitivity
è limitato dal concetto di fiducia



TRANSIZIONI

CREATE OBJECT: attiva un oggetto rendendolo accessibile

DELETE OBJECT: disattiva un oggetto attivo

GET ACCESS: un soggetto guadagna l'accesso a un oggetto

RELEASE ACCESS:

GIVE ACCESS: i diritti di accesso a un oggetto vengono ^{assegnati} ~~trasferiti~~ da un soggetto all'altro
l'operazione viene eseguita solo se rispetta le politiche di maturità

RESCIND ACCESS: revoca un accesso precedentemente garantito con "GIVE ACCESS"

CHANGE SUBJECT SECURITY LEVEL: cambia il "security level" di un soggetto. Il nuovo
livello deve essere inferiore

CHANGE OBJECT SECURITY LEVEL:

MODELLO BIBA

Creato da Biba nel 1975, è un sistema basato su transizioni di stato che descrive un insieme di regole di controllo accessi per mantenere l'integrità dei dati.

- I soggetti e gli oggetti sono raggruppati in base a livelli di integrità
- Il modello BIBA è fatto in modo che i soggetti non ponano corrompere gli oggetti di livello più alto del loro

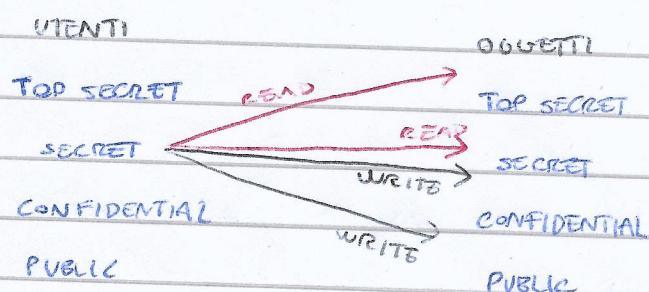
OBIETTIVI INTEGRITÀ DEI DATI

- prevenire modifiche dei dati da entità non autorizzate
- prevenire modifiche non autorizzate sui dati, da entità autorizzate
- mantenere le connivenze interne e esterne (i dati riflettono il mondo reale)

Il modello BIBA mira all'integrità piuttosto che alla confidenzialità dei dati.
Le caratteristiche del BIBA è READ UP, WRITE DOWN (^{compartono} di sé)

Nel modello BIBA, gli utenti possono solo CREARE CONTENUTO al loro LIVELLO DI INTEGRITÀ o inferiore.

gli utenti possono leggere contenuti del loro LIVELLO DI INTEGRITÀ o superiore



In generale può ricevere degli ordini a un sottoposto che spedire gli ordini a un maggiore. In questo modo gli ordini del generale rimangono intatti. Una recluta non può ricevere ordini per il suo sergente

REGOLE

Simple integrity property: NO READ DOWN

Star integrity property: NO WRITE UP

INVOCATION PROPERTY: un processo in basso non può richiedere accessi più alti del suo livello

RELAX NO READ DOWN: un oggetto può fare READ DOWN se allora il proprio livello di integrità

RELAX NO WRITE UP

MODELLO BREWER NASH (MURAGLIA CINESE)

Il modello Brewer Nash, anche detto Modello Muraglia Cinese, è stato creato per fornire controlli accesi che può cambiare dinamicamente. Lo scopo del modello è quello di mitigare i conflitti di interesse tra organizzazioni commerciali.

In questo modello, l'informazione non può scorrere tra soggetti e oggetti in modo da creare conflitti di interesse.

Questo modello è usato negli studi dei commercialisti; quando un commercialista accede ai dati del suo cliente, la ACME s.p.a., il commercialista non può accedere ai dati delle aziende concorrenti.

Questo modello usa il principio di volgimento dei dati in "dom di conflitto".

C: insieme delle aziende

S: insieme dei soggetti (commercialisti)

O: insieme degli oggetti, si riferiscono a una precisa azienda

I: plechi che si riferiscono alle stesse aziende sono raggruppati nel "company dataset"

cd: $O \rightarrow C$ sapendo l'oggetto, ti dice a che azienda si riferisce
company dataset

Le classi di conflitto di interesse indicano quali aziende sono in competizione

cic: $O \rightarrow P(C)$ sapendo l'oggetto, ti dice l'insieme delle aziende che non dovrebbero conoscere il contenuto dell'oggetto
conflict of interest classes

LA ETICHETTA DI SICUREZZA di un oggetto è la coppia $(cic(o), cd(o))$

Credit Lyman

company dataset:

tutti gli oggetti che si riferiscono a Credit Lyman

Deutsche Bank

classe di conflitto d'interesse:

$N(s, o)$ è restituita se s può accedere a o
nello stato di partita, $N(s, o) = \text{falso}$ per ogni $s \in o$

gli permessi di accesso cambieranno dinamicamente e devono essere
corrispondenti ad ogni transazione

REGOLA PER PREVENIRE IL CONFLITTO DI INTERESSI

Un soggetto s (commercialeste) può accedere a qualsiasi informazione
fornita da chi non ha mai acceduto a informazioni da un'altra azienda
che si trova nello stesso database di conflitto d'interessi

CONTROLLO ACCESSI BASATO SU RUOLI

RBAC è un approccio che ha lo scopo di mettere restrizioni agli accessi agli utenti non autorizzati.

RBAC è usato da aziende con molti dipendenti e può implementare regole DAC e MAC.

RBAC è un meccanismo di controllo accessi, basato su RUOLI e PRIVILEGI

- UTENTI
- RUOLI
- PERMESSI

DESCRIZIONE

In una azienda o organizzazione, i ruoli sono creati per varie funzioni lavorative. I permessi di effettuare determinate operazioni sono assegnati a ruoli specifici. I soggetti (dipendenti, dirigenti etc) vengono assegnati ruoli specifici. Un soggetto, attraverso il suo ruolo, può ottenere il permesso per svolgere una operazione.

SOGGETTO	PERMESSI dello SPAGHETTI
camerlengo	PULIRE UFFICIO
portiere	PULIRE BAGNO

PERMESSI DEL CAMERLENGHO
aprire cestinetti

La gestione degli utenti avviene tramite l'assegnamento dei ruoli all'utente

REGOLE

Assegnamento dei ruoli: Un soggetto può esercitare un permesso solo se ha un ruolo assegnato

Autorisazione dei ruoli: Un ruolo di un soggetto deve essere autorizzato

Autorisazione dei permessi: Un soggetto può esercitare un permesso solo se

il permesso è autorizzato per il ruolo attivo del soggetto
potranno essere assegnati anche altri ruoli, come uno gerarchico
dove ruoli ereditano i permessi

ESEMPIO

CONTROLLO ACCESSI

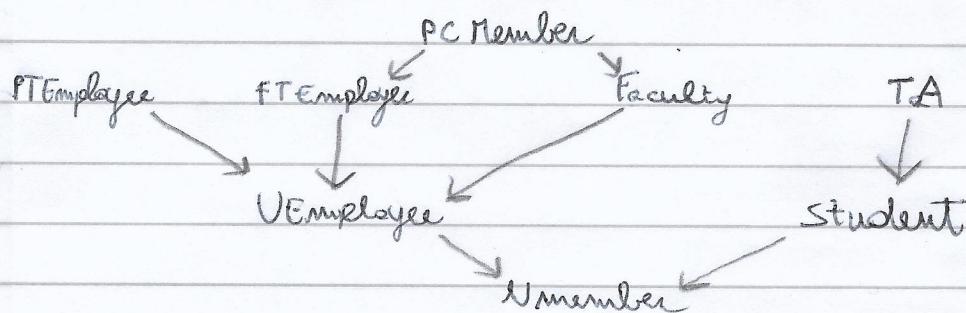
User Assignment

utente	ruolo
Alice	PC Member
Bob	Faculty
Charlie	Faculty
David	TA
David	student
Eve	UEmployee
Fred	student
Greg	UMember

Permission Assignment

ruolo	permesso
PC Member	Grant Tenure
Faculty	Assign Grades
TA	Assign HW Scores
UEmployee	Receive HBenefits
Student	Register for Courses
UMember	Use Gym

GERARCHIA dei ruoli:



Politica ARBAC:

com-assign1: UEmployee: {Student, TA} \Rightarrow PTEmployee

com-assign2: UEmployee: {UEmployee, Faculty} \Rightarrow Student

com-assign3: UEmployee: {Faculty} \Rightarrow TA

Input

condizione

output

CONTROLLO ACCESSI - esercizio 25 Giugno 2009

Alice possiede il file alice.bat

Bob può solo leggerlo e rernerlo, mentre Charlie può solo eseguirlo

Bob possiede bob.bat, Charlie può solo leggerlo, Alice può solo leggerlo e rernerlo

Charlie possiede charlie.bat, Alice può solo rernerlo, Bob può solo eseguirlo

ogni file può essere letto, scritto, eseguito dagli utenti che lo possiedono

	Alice.bat	Bob.bat	Charlie.bat
Alice	r w x	r w	w
Bob	r w	r w x	x
Charlie	x	r	r w x

si mire la matrice di controllo accessi da ottenere se

Charlie dà ad Alice il permesso di leggere Charlie.bat e Alice rernerse

a Bob il permesso di rernerre alice.bat

in = 6-5-2006

Si consideri il MAC di Bell La Padula e si indichino i permessi concessi ad un utente con security level (secret, {red, green, blue}) relativamente a documenti classificati nel seguente modo

1: (top secret, {red})

2: (secret, {red})

3: (secret, {red, black}) nessun diritto (categorie estranea)

4: (secret, {white}) nessun diritto. (categorie estranee)

5: (confidential, {red, blue, green}) lettura

6: (confidential, {white}) nessun diritto

7: (top secret, {red, green, blue, black}) scrittura

utente

(secret, {red, green, blue})

(r_2, c_2) domine (r_1, c_1) se e solo se $r_1 \leq r_2 \wedge c_1 \leq c_2$

es: $(\text{secret}, \{\text{red, green, blue}\})$ ob: $\text{top-secret}, \{\text{red}\}$
 $\text{secret} \leq \text{top-secret}$ $\text{S1} \wedge \{\text{red, green, blue}\} \subseteq \{\text{red}\}$ NO

NO READ UP: Un oggetto con security label x_5 può leggere un oggetto con security label x_6 solo se x_5 domine x_6 .

NO WRITE DOWN: Un oggetto con security label x_5 può scrivere su un oggetto con security label x_6
solo se x_5 domine x_6

$(\text{secret}, \{\text{red, green, blue}\})$ domine $(\text{top secret}, \{\text{red}\})$? NO
 $\text{top secret} \leq \text{secret}$ $\text{NO} \wedge \{\text{red}\} \subseteq \{\text{red, green, blue}\}$ S1

$(\text{top secret}, \{\text{red}\})$ domine $(\text{secret}, \{\text{red, green, blue}\})$ NO
 $\text{secret} \leq \text{top secret}$ $\text{S1} \wedge \{\text{red, green, blue}\} \subseteq \text{red}$ NO

$(\text{secret}, \{\text{red, green, blue}\})$ domine $(\text{secret}, \{\text{red}\})$?
 $\text{secret} \leq \text{secret}$ $\text{S1} \wedge \{\text{red}\} \subseteq \{\text{red, green, blue}\}$ S1

lettura
abilità

$(\text{secret}, \{\text{red}\})$ domine $(\text{secret}, \{\text{red, green, blue}\})$?
 $\text{secret} \leq \text{secret}$ $\text{S1} \wedge \{\text{red, green, blue}\} \subseteq \{\text{red}\}$ NO

$(\text{secret}, \{\text{red, green, blue}\})$ domine $(\text{top secret}, \{\text{red, green, blue, black}\})$?
 $\text{top secret} \leq \text{secret}$ $\text{NO} \wedge \{\text{red, green, blue, black}\} \subseteq \{\text{red, green, blue}\}$ NO

$(\text{top secret}, \{\text{red, green, blue, black}\})$ domine $(\text{secret}, \{\text{red, green, blue}\})$?
 $\text{secret} \leq \text{top secret}$ $\text{S1} \wedge \{\text{red, green, blue}\} \subseteq \{\text{red, green, blue, black}\}$ S1

scrittura abilità

Ex 8 6-5-2006

append
last modify

> si consideri il seguente insieme di diritti $\{R, W, X, A, L, M, Own\}$

Ulteriorando le sintassi HRU si scriva il comando per
revoke-all-rights (s_1, s_2, o) cd quale s_1 cancella tutti i diritti di s_2
per l'oggetto o

revoke all rights (s_1, s_2, o)

delete R, W, X, A, L, M, Own from M(s_2, o)

end