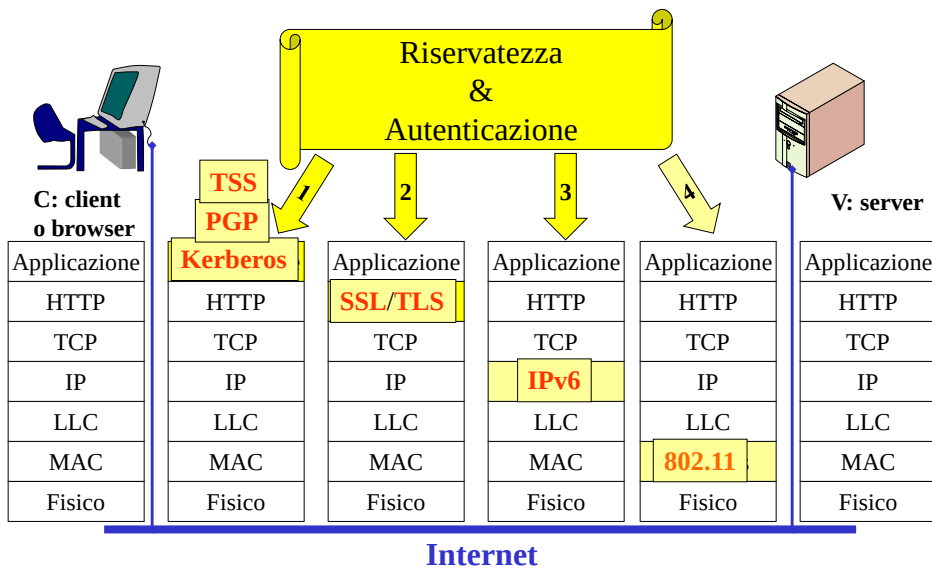


Applicazioni distribuite



Sicurezza a vari livelli

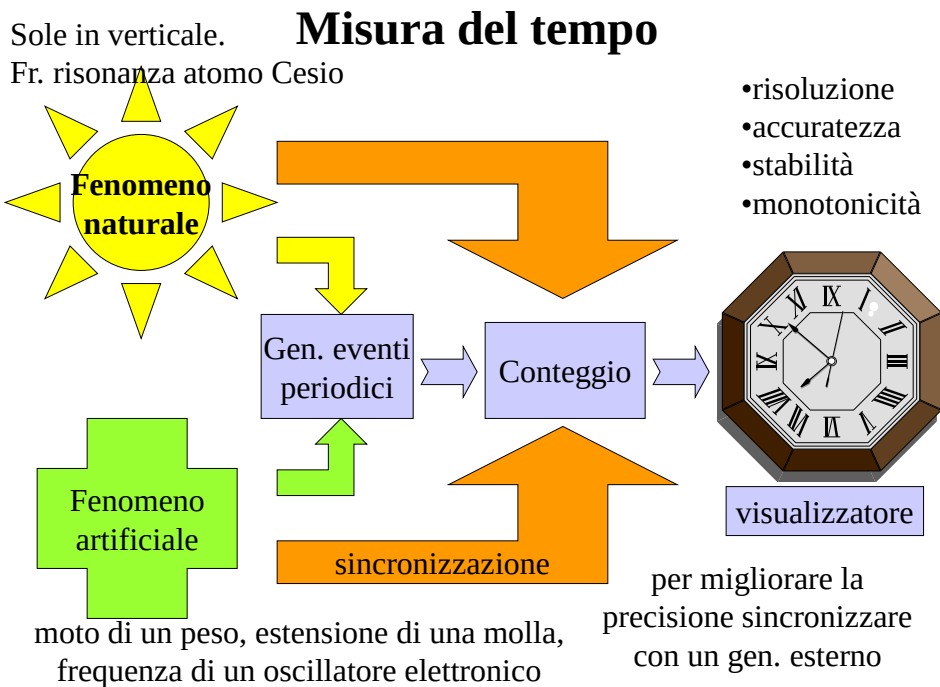
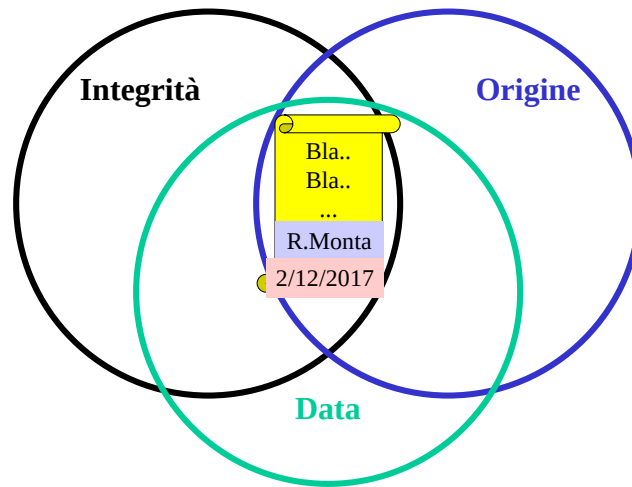


Servizi per le applicazioni

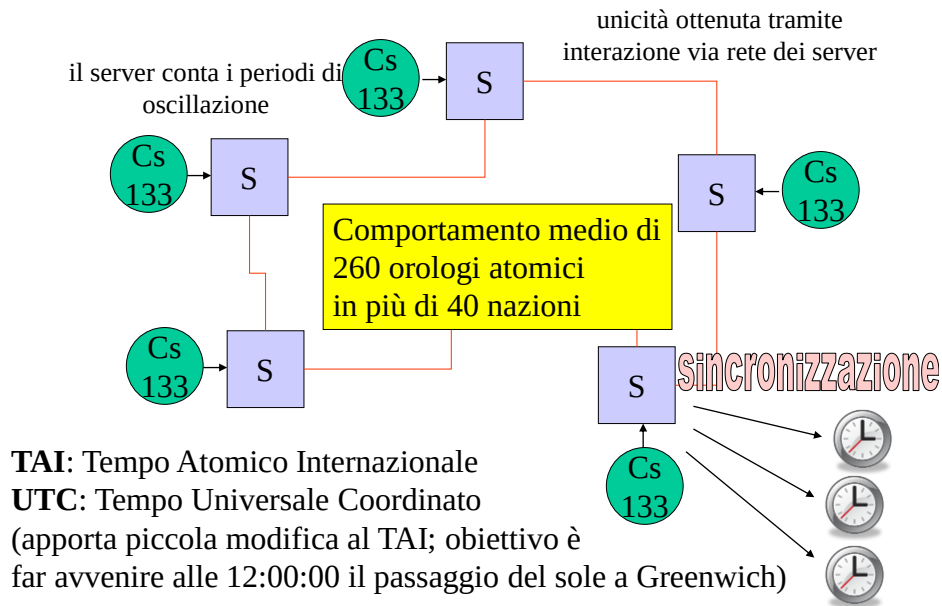
- TSS
- Kerberos
- PGP

Marche temporali sicure

Il documento informatico sicuro



Riferimenti internazionali: TAI e UTC



Marche temporali

- Messaggistica elettronica
- File system, Database
- Applicazioni distribuite
- Certificato di chiave pubblica
- Revoca e CRL

nessun controllo
 monotonicità (per ordinamento eventi)
 sincronismo (per coordinamento)
 anno/mese/giorno di inizio/fine
 data/ora con grande precisione

Firma digitale & marca temporale
 collocazione temporale del documento
 validità della chiave di firma

TSS: Time Stamp Service

Marcatura temporale di un documento

- Il tempo della marca non deve essere falso
- Tramite la marca deve essere possibile individuare in modo sicuro un documento, un istante ed un autore
- La modifica anche di un solo bit di una marca deve poter essere rilevata
- Deve essere possibile farsi marcare documenti mantenendone riservato il contenuto
- Chiunque deve poter sia farsi marcare i suoi documenti, sia verificare la marcatura dei documenti di chiunque altro

Ente fidato

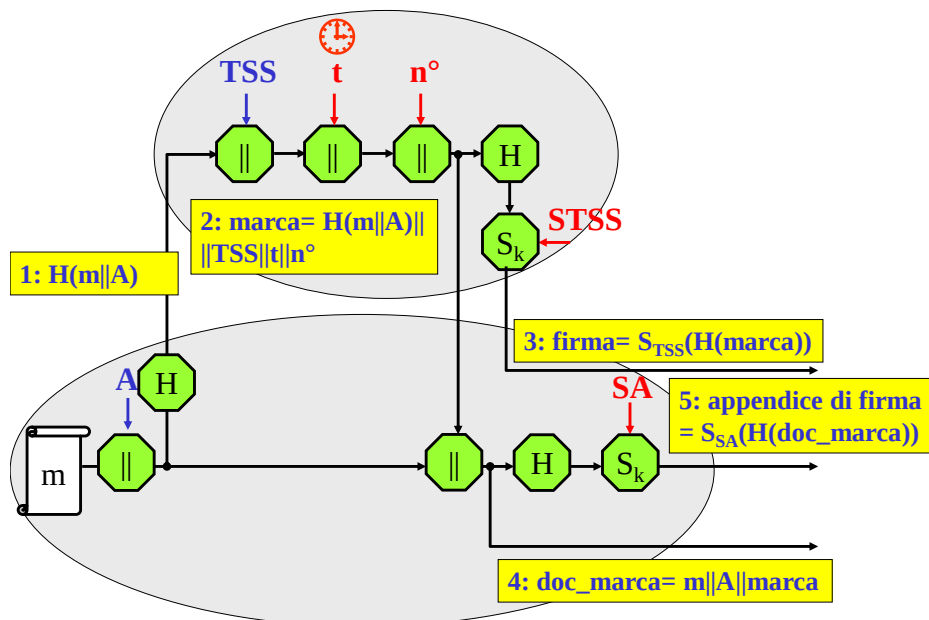
Firma digitale

Hash sicura

Servizio pubblico

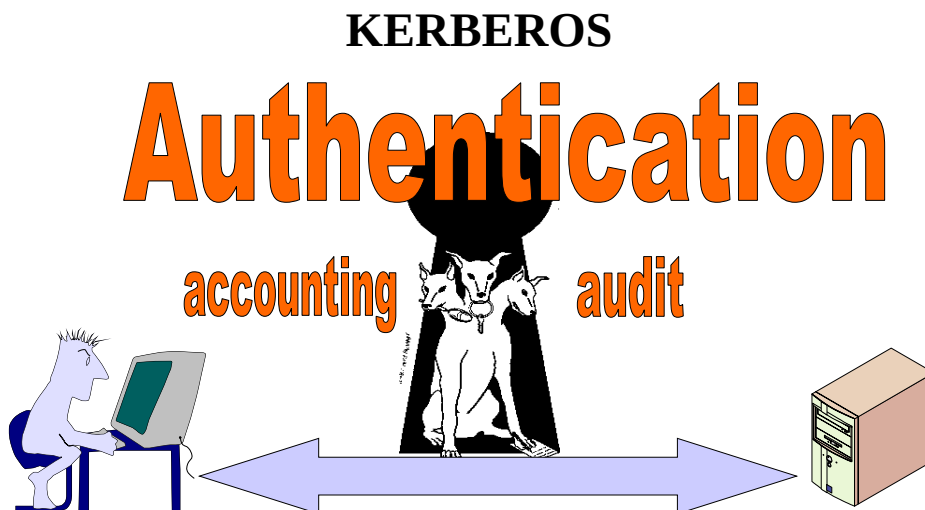
Chiave di verifica sicura

Marcatura e firma di un documento



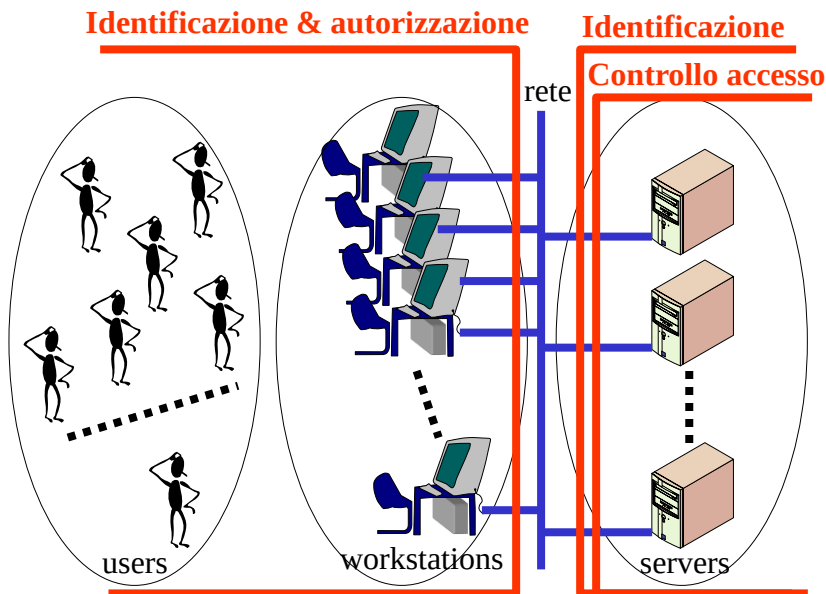
Problemi

- PKIX: PKI&TSS
 - efficienza
 - vita della chiave/vita dei documenti
 - fidatezza
-
- Nuovi standard?
 - notarizzazione
 - tempo relativo



In Greek mythology, a many headed dog, the guardian of the entrance of Hades

Single sign-on: una sola psw per accedere a molti servizi



KERBEROS

- Provides a centralized authentication server to authenticate users to servers and servers to users.
- Relies on conventional encryption, making no use of public-key encryption
- Two versions: version 4 and 5 (RFC 1510)

Three threats exist:

- User pretend to be another user.
- User alter the network address of a workstation.
- User eavesdrop on exchanges and use a replay attack.

Semplice dialogo di autenticazione

1. C -> AS: $ID_c || P_c || ID_v$
 2. AS -> C: Ticket AD_c per evitare intercettazione del ticket al passo 2 e riutilizzo al passo 3 (al passo 3 l'indirizzo del client deve essere uguale a quello originario)
 3. C -> V: $ID_c ||$ Ticket
- Ticket = $E_{kv}[ID_c, AD_c, || ID_v]$

Problemi:

Trasmissione della password

Ad ogni connessione a un server (anche lo stesso)
l'utente deve rifare il protocollo

Dialogo di autenticazione più sicuro

Per ogni sessione di login:

- C -> AS: $ID_c || ID_{tgs}$ Si evita invio della pwd!
- AS -> C: $E_{kc}[Ticket_{tgs}]$ E_{kc} deriva dalla pwd

$$Ticket_{tgs} = E_{ktgs}[ID_c || AD_c || ID_{tgs} || TS_1 || Lifetime_1]$$

La cifratura impedisce la costruzione di informazioni falsificate

Perchè TS_1 e $Lifetime_1$? Non si vuole che Eve catturi e riutilizzi il ticket (ad es. aspetta che il logout dell'utente, falsifica indirizzo di rete; timestamp e lifetime limitano possibilità di riutilizzo)

Dialogo di autenticazione più sicuro

Per ogni sessione di login:

- C → AS: $ID_c || ID_{tgs}$
- AS → C: $E_{kc}[Ticket_{tgs}]$

Si evita invio della pwd!

E_{kc} deriva dalla pwd

$$Ticket_{tgs} = E_{ktgs}[ID_c || AD_c || ID_{tgs} || TS_1 || Lifetime_1]$$

La cifratura impedisce la costruzione di informazioni falsificate

Dialogo di autenticazione più sicuro

Per ogni tipo di servizio:

1. C → TGS: $ID_c || ID_v || Ticket_{tgs}$
2. TGS → C: $Ticket_v$

$$Ticket_v = E_{kv}[ID_c || AD_c || ID_v || TS_2 || Lifetime_2]$$

Per ogni sessione di servizio:

C → V: $ID_c || Ticket_v$

Dialogo di autenticazione più sicuro

Per ogni tipo di servizio:

1. C → TGS: $ID_C || ID_V || Ticket_{tgs}$
2. TGS → C: $Ticket_V$

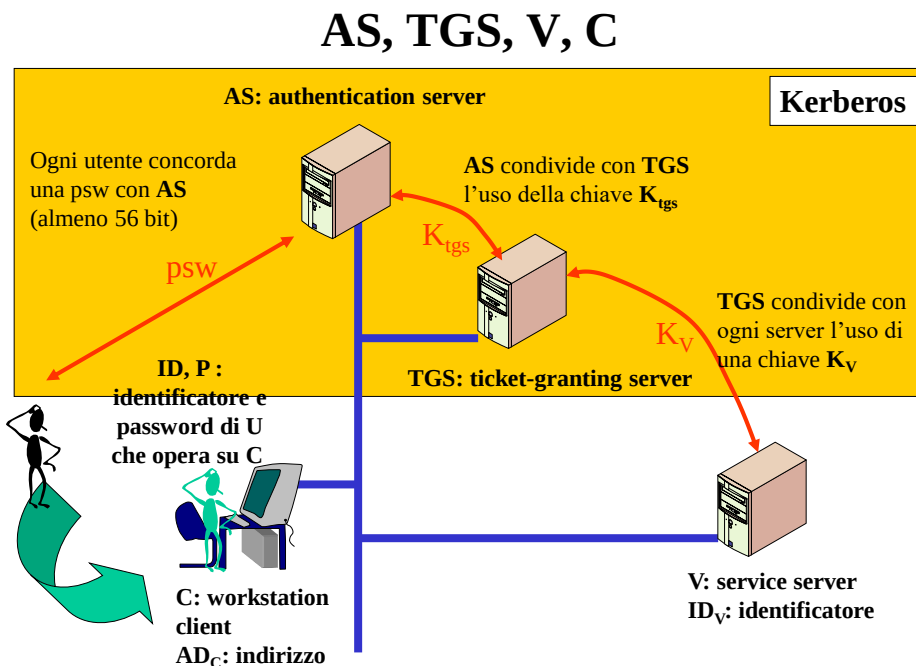
$$Ticket_V = E_{K_V}[ID_C || AD_C || ID_V || TS_2 || Lifetime_2]$$

Per ogni sessione di servizio:

C → V: $ID_C || Ticket_V$

Problemi:

- Durata del ticket (se troppo breve, l'utente deve re-inserire la pwd, se troppo lungo problema di intercettazione e riutilizzo)
- E se servisse anche l'autenticazione da parte dei server?
- Occorre che il TGS dimostri che la persona che utilizza il ticket è quella persona per cui è stato emesso



Il dialogo tra C, AS, TGS e V

1: All'inizio della sessione di lavoro sulla stazione **C**,
l'utente dichiara la sua identità ad **AS**

2: **AS** fornisce il permesso d'accesso a **TGS**
e lo sfida ad usarlo

ticket TGS, ΔT

3: **C** risponde alla sfida, richiedendo anche l'accesso al server **V**

4: **TGS** fornisce a **C** il permesso d'accesso a **V**
e lo sfida ad usarlo

ticket V, ΔT

5: **C** si qualifica a **V**

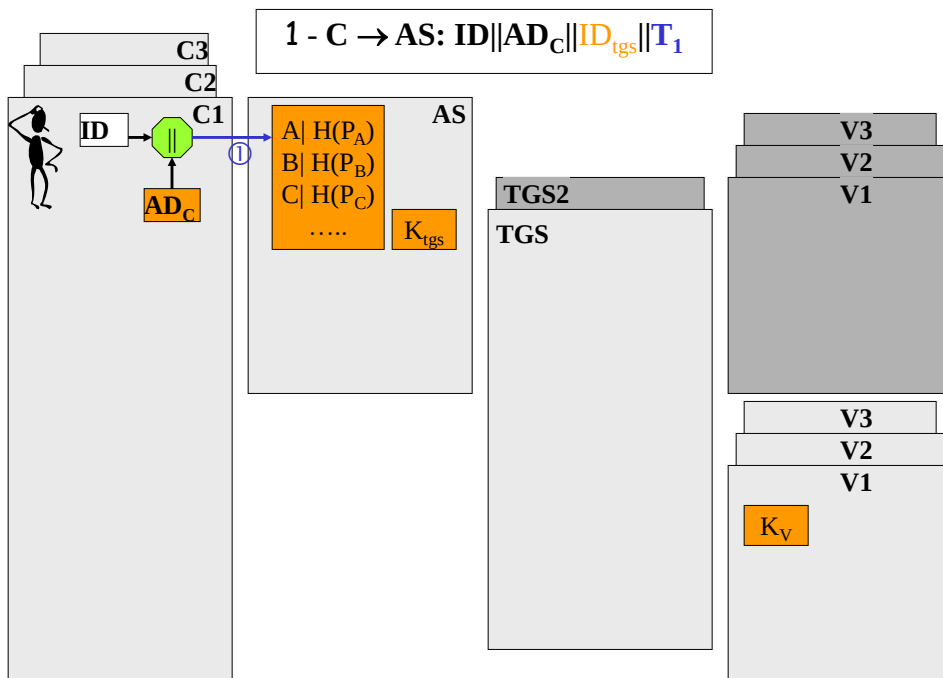
6: **V** si fa identificare da **C**

Svolgimento del servizio

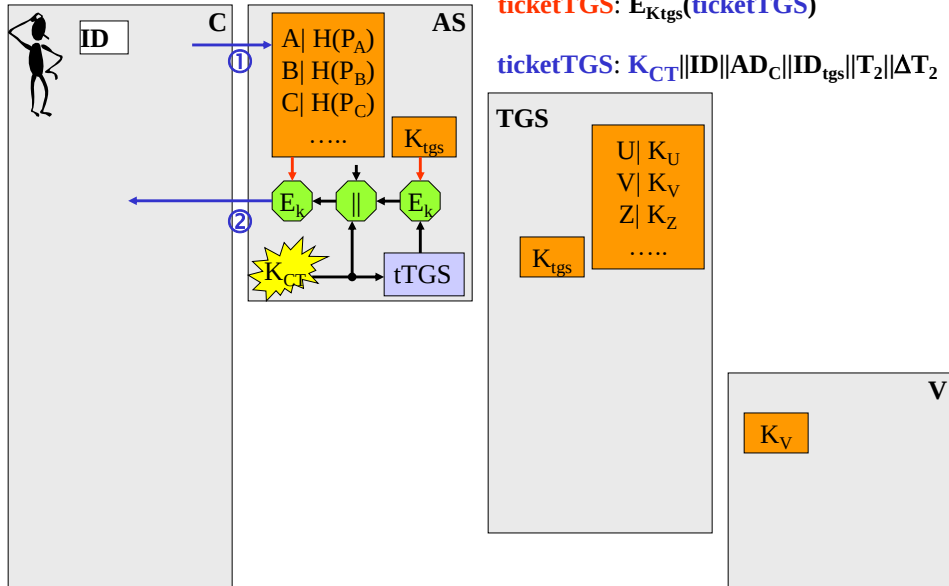
Successiva esigenza di servizio da V: goto 5

Accesso ad un nuovo server: goto 3

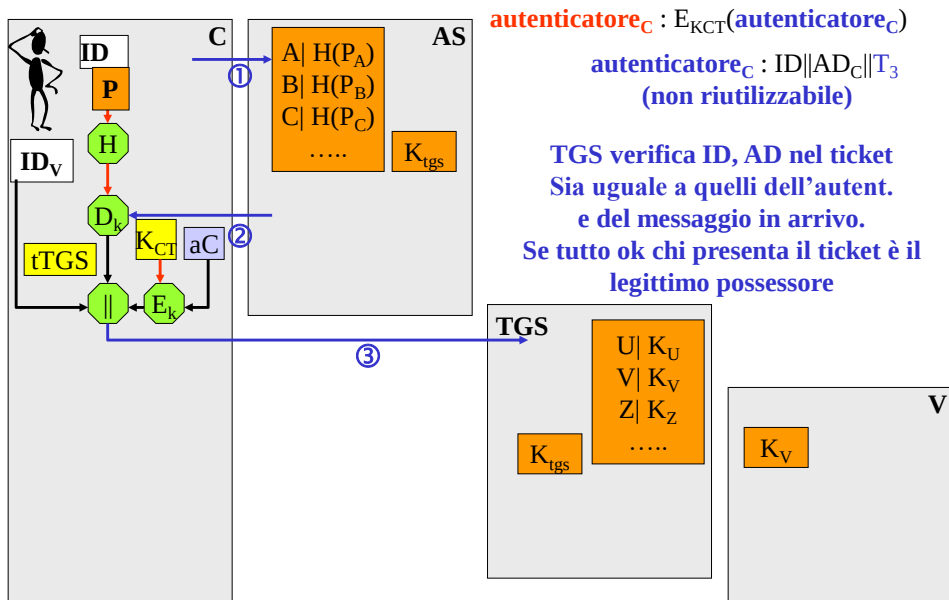
N: Fine della sessione o tempo scaduto



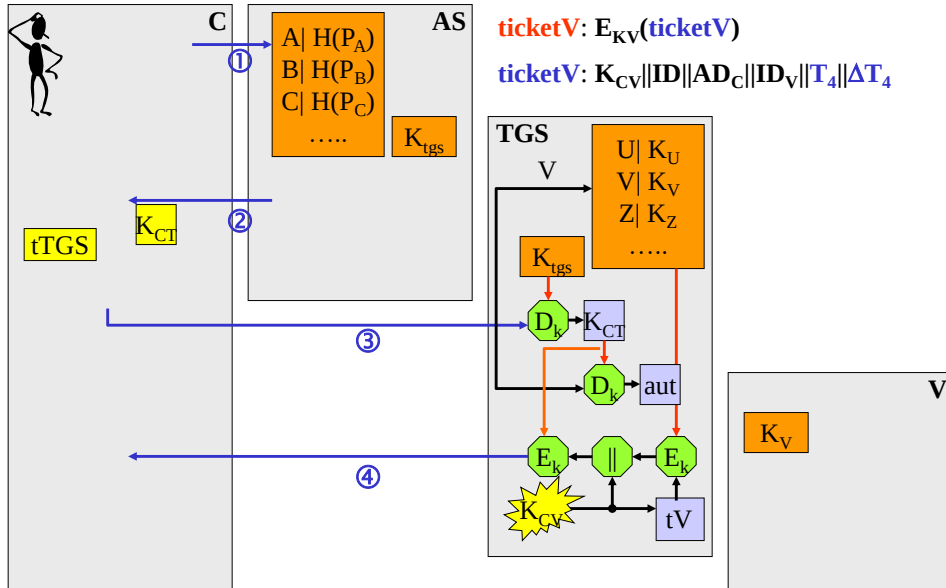
2 - AS $\rightarrow$ C: $E_{PSW}(K_{CT}||ID_{tgs}||T_2||\Delta T_2||\text{ticketTGS})$



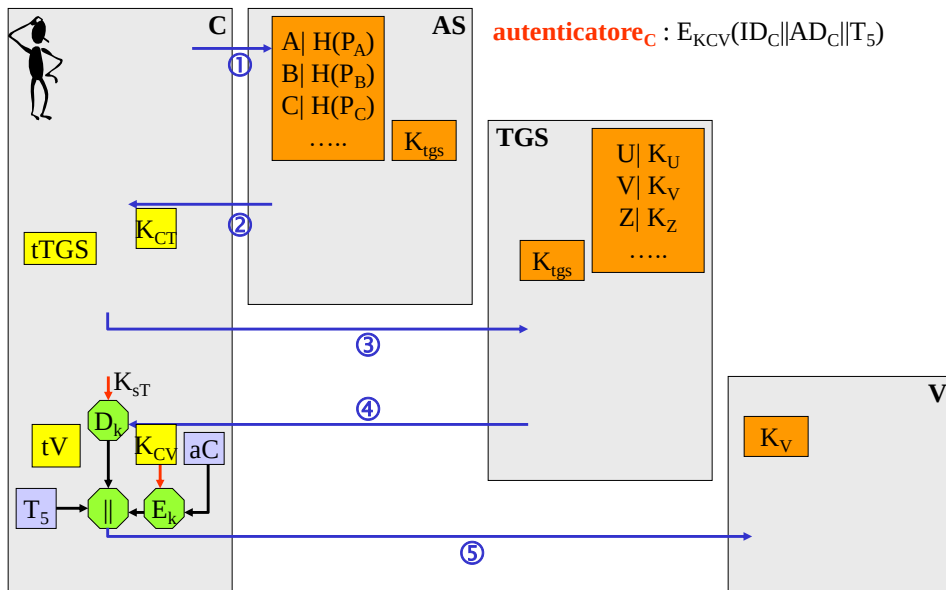
3 - C $\rightarrow$ TGS: $ID_V||\text{ticketTGS}||\text{autenticatore}_C$

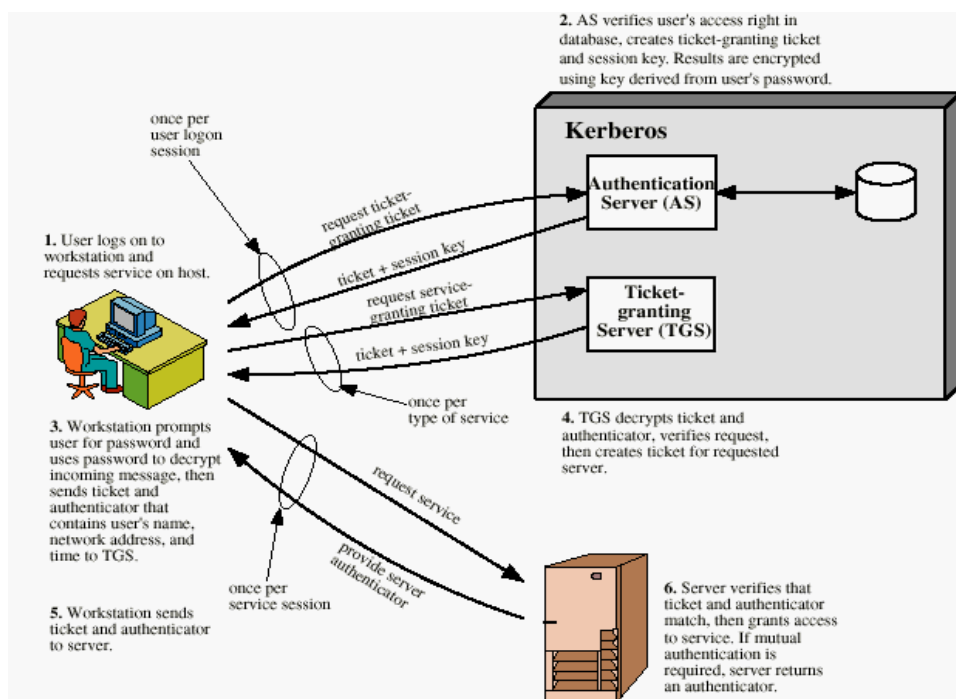
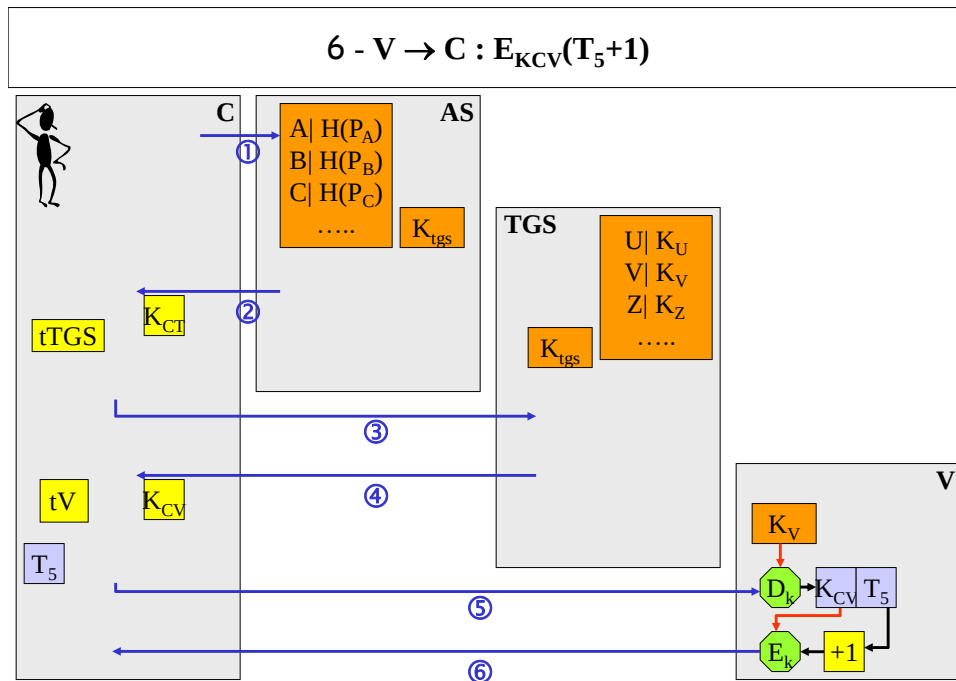


4 - TGS → C : $E_{K_{CT}}(K_{CV} || ID_V || T_4 || \text{ticket}_V)$

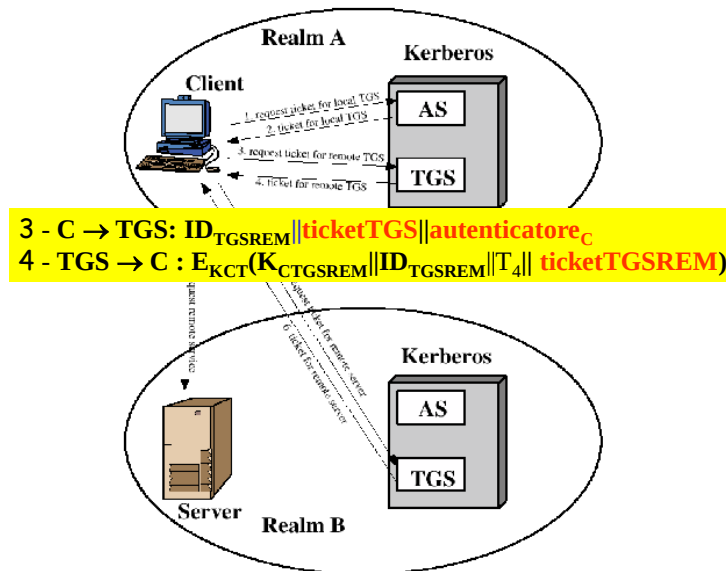


5 - C → V: **ticket_V** || **autenticator_C**





Request for Service in Another Realm



Kerberos v4

1. Dipendenza dal DES
2. Dipendenza dal protocollo IP per gli indirizzi
3. Durata del ticket (max 8 bit con unità di 5 minuti =>
 $2^8 \times 5 \rightarrow 1280$ minuti)
4. Autenticazione tra realm differenti (N^2)

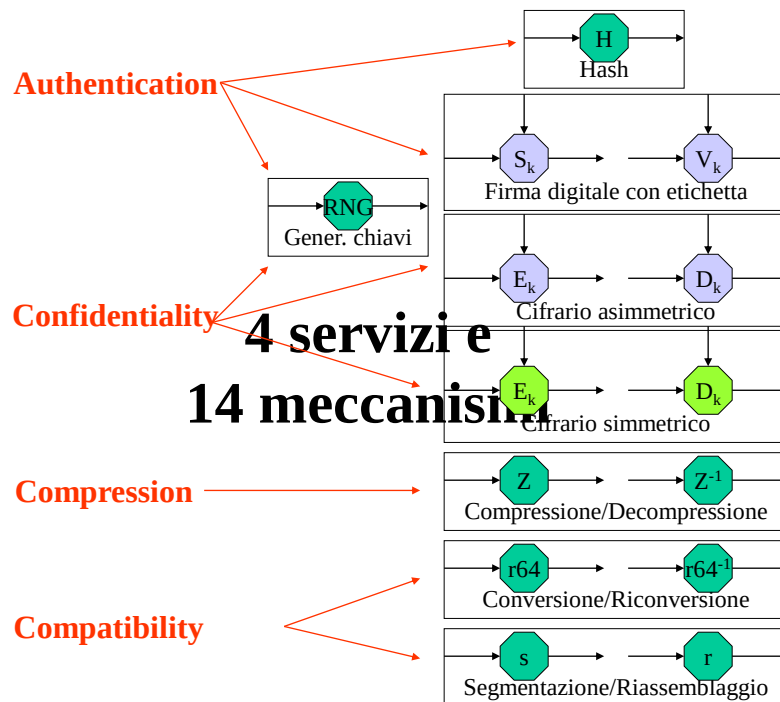
Pretty Good Privacy

- Philip R. Zimmerman is the creator of PGP (1993).

PGP provides
a **confidentiality** and **authentication** service
that can be used for
electronic mail and **file storage** applications.

Confidentiality: the act of keeping something private and secret from all but those who are authorized to see it.

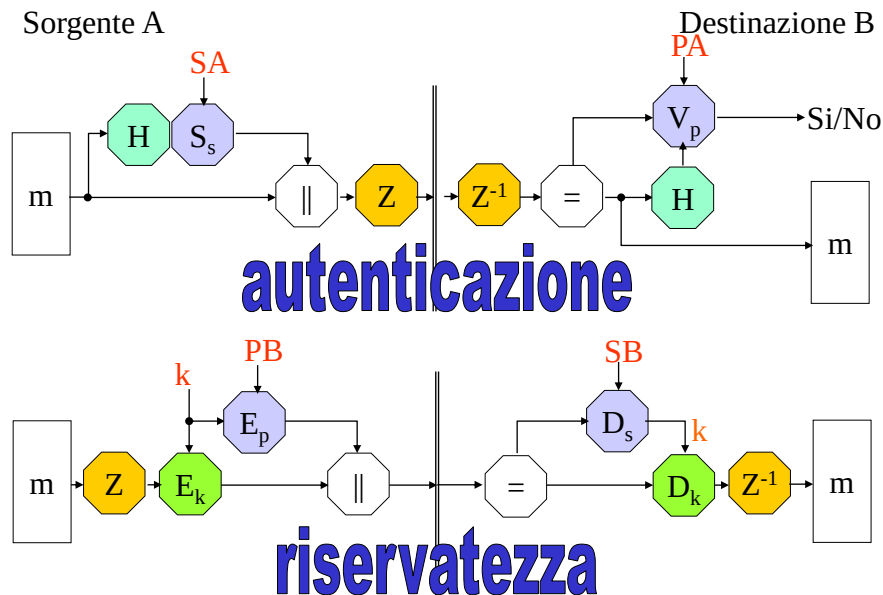
Authentication: to prove genuine by corroboration of the identity of an entity.



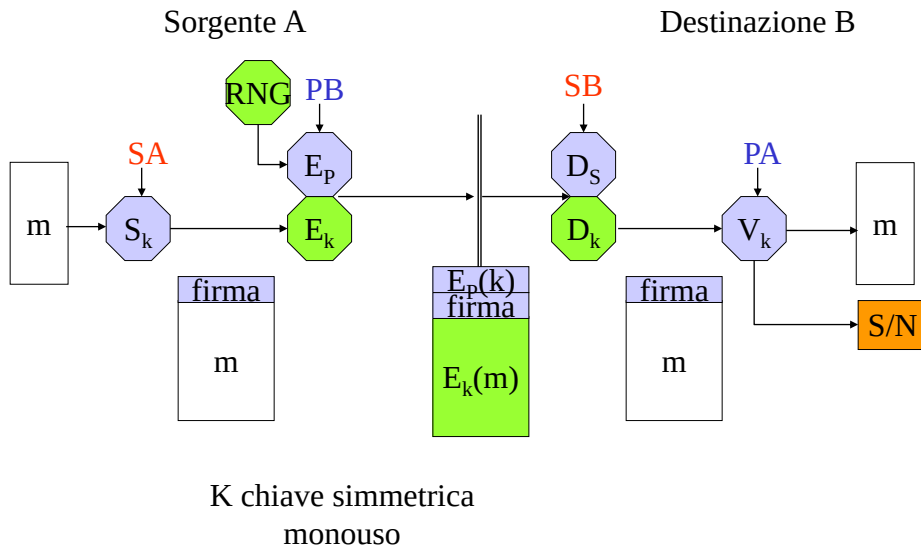
Summary of PGP Services

Function	Algorithm Used
Digital Signature	DSS/SHA or RSA/SHA
Message Encryption	CAST or IDEA or three-key triple DES with Diffie-Hellman or RSA
Compression	ZIP
E-mail Compatibility	Radix-64 conversion
Segmentation	

PGP: autenticazione o riservatezza



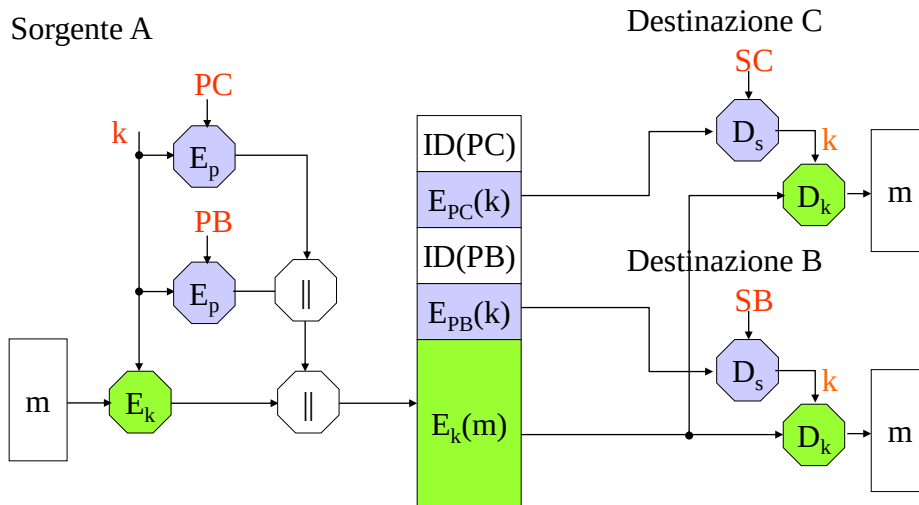
PGP: autenticazione & riservatezza



PGP: note

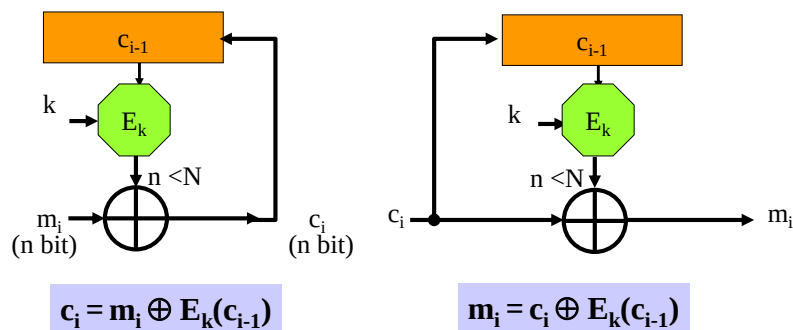
1. Firma con appendice; firme allegate o separate dal messaggio (vantaggi ad es. firme multiple, verifica di virus nel messaggio se file eseguibile)
2. Compressione per motivi di risparmio di trasmissione e memorizzazione.
 1. Nel caso della firma: compressione dopo la firma => se così non fosse ogni volta che verifico un messaggio dovrei ricomprimerlo oppure memorizzare m , compressione di m e firma; inoltre l'algoritmo di compressione non è deterministico, quindi quale è stato usato?
 2. Nel caso della cifratura: compressione prima della cifratura, elimino ridondanze eventuali nel messaggio

Messaggio riservato per più destinatari



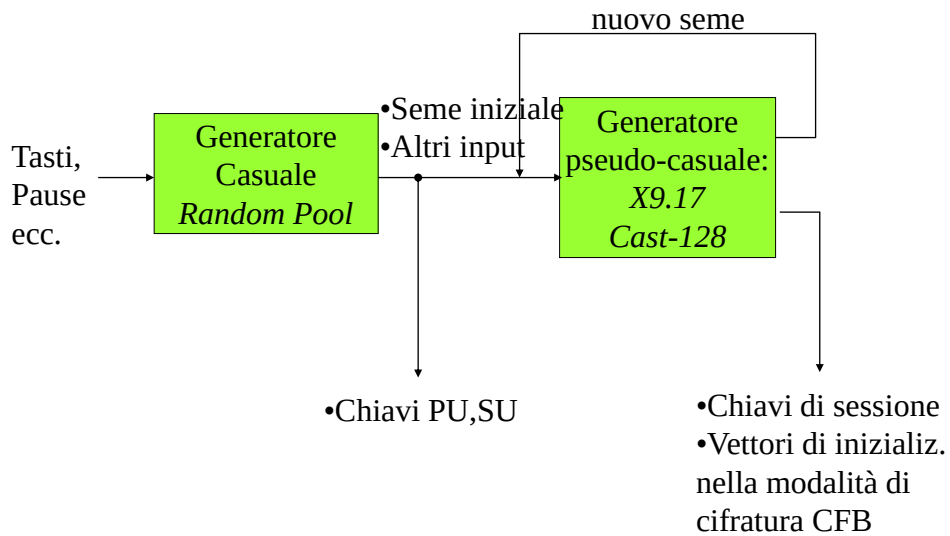
Modalità di cifratura del PGP

CFB (Chipher Feedback)



$N = 64/128 \text{ bit}$
 $n = 64 \text{ bit}$

RNG e PRNG

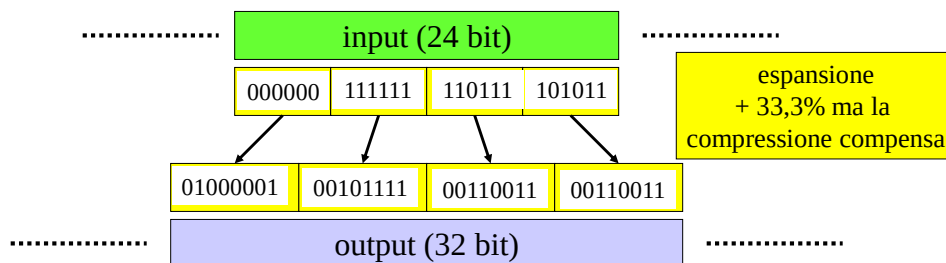


radix-64

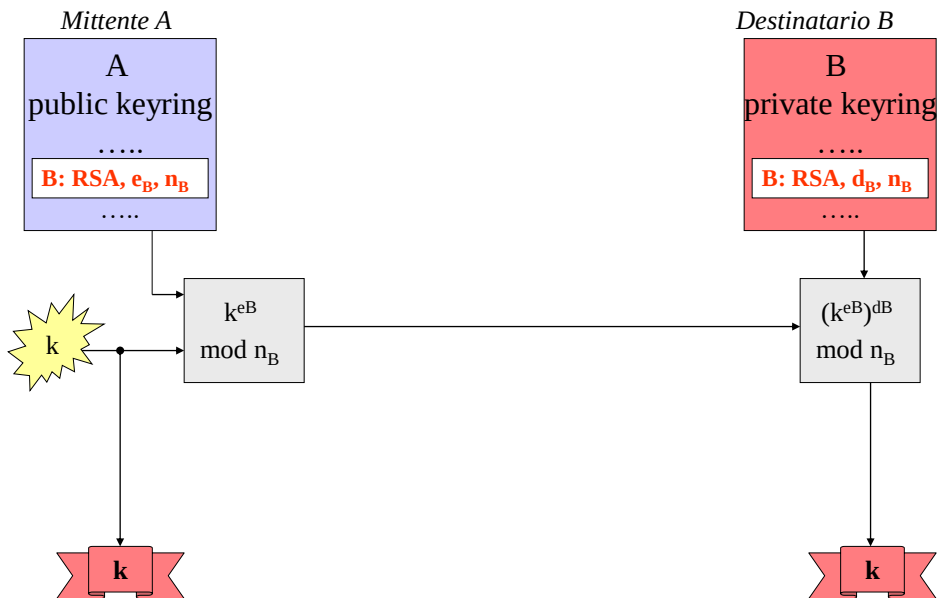
	000	001	010	011	100	101	110	111
000	A	B	C	D	E	F	G	H
001	I	J	K	L	M	N	O	P
010	Q	R	S	T	U	V	W	X
011	Y	Z	a	b	c	d	e	f
100	g	h	i	j	k	l	m	n
101	o	p	q	r	s	t	u	v
110	w	x	y	z	0	1	2	3
111	4	5	6	7	8	9	+	/

Alcuni sist. di posta elettr. permettono di usare solo blocchi di testo ASCII => conversione radix-64 (flusso di 8 bit binari in flusso di caratteri stampabili)

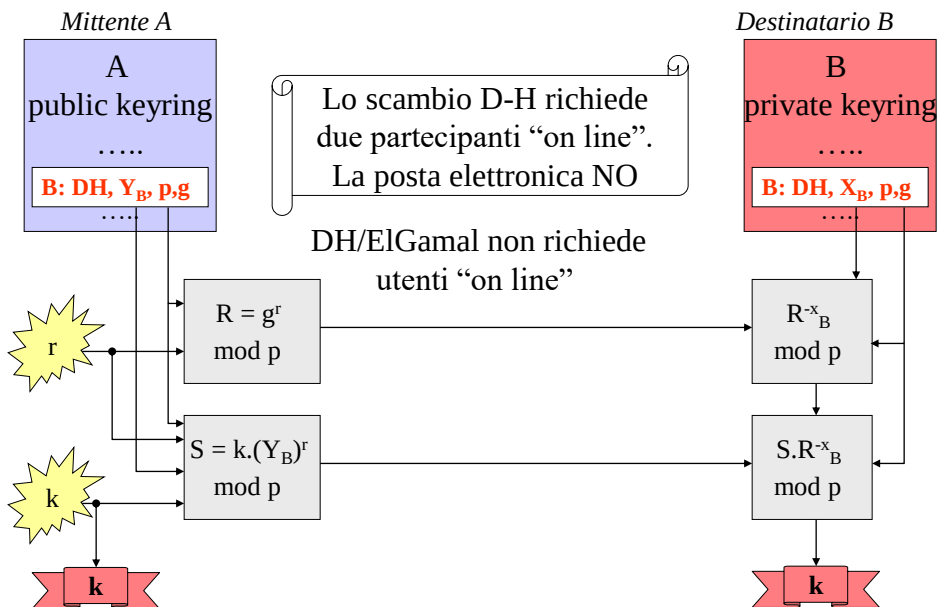
Blocchi di tre ottetti binari trasformati in 4 caratteri



Chiave di messaggio con RSA



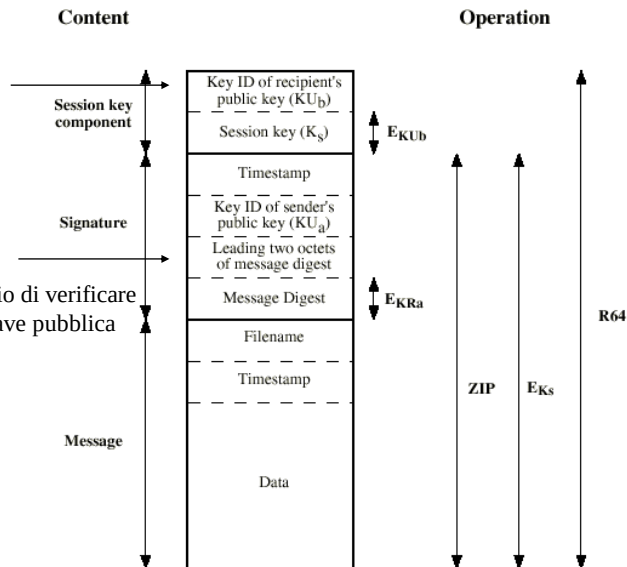
Chiave di messaggio con DH/ElGamal



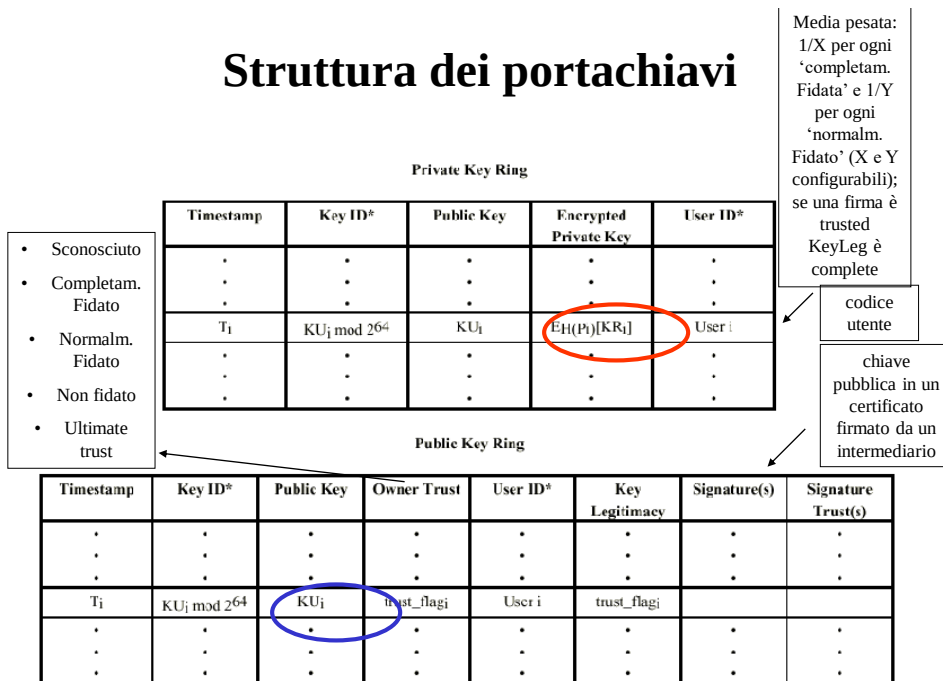
Format of PGP Message

PGP prevede che ogni utente possa avere più coppie di chiavi. KeyID per risparmiare spazio
Come creare un KeyID?
(64 bit meno sign.
 $KU_b \bmod 264$)

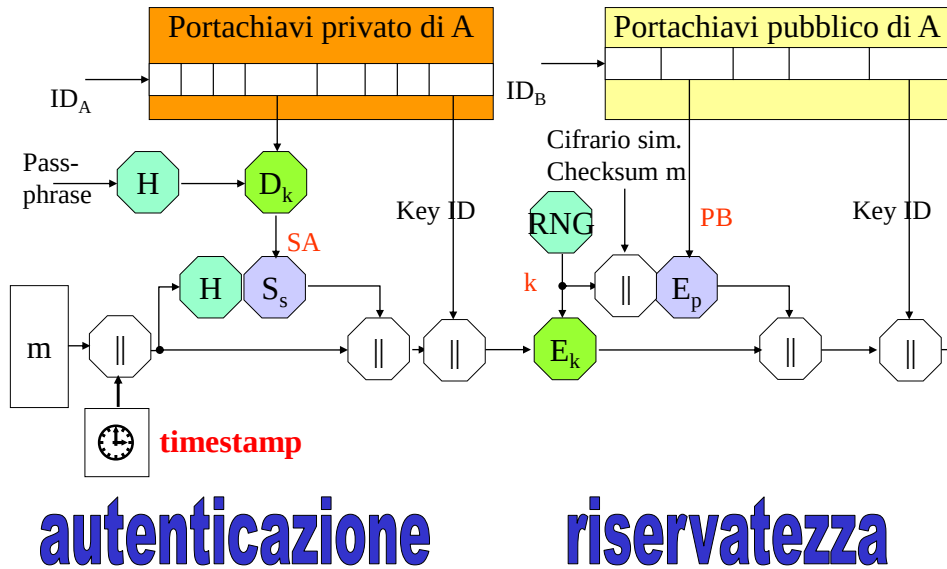
Per permettere al destinatario di verificare se è stata utilizzata la chiave pubblica corretta



Struttura dei portachiavi



Mittente: portachiavi privato e pubblico



Destinatario: portachiavi privato e pubblico

