Dipartimento Informatica e Applicazioni
"R. Capocelli"

Una breve introduzione ai file system cifrati

Luigi Catuogno
luicat@tcfs.unisa.it

Sommario

- Tecniche di protezione dei file
- Perché cifrare un file system?
- Principali servizi e caratteristiche di un file system cifrato

Casi di studio

- Il Cryptographic File System (CFS)
- Microsoft Encrypting File System (EFS)
- Transparent Cryptographic File System (TCFS)

Tecniche di protezione dei file

Cifratura manuale dei file

L'utente:

- Cifra i file "riservati" con apposite utility (es. il comando crypt di UNIX, PGP)
- Decifra i file protetti ogni volta che vuole accedervi
- Gestisce autonomamente strategie di protezione e chiavi di cifratura

Cifratura manuale dei file

Svantaggi:

- Durante il loro utilizzo i file non sono protetti
- L'utilizzo dei file cifrati e' piuttosto macchinoso
- L'utente deve provvedere alla protezione di tutti i file creati dalle applicazioni
- La gestione delle chiavi (una per file) e' completamente a carico dell'utente

Applicazioni che cifrano i dati

Le applicazioni:

- Gestiscono le operazioni di cifratura sui file che utilizzano/producono
- Assistono l'utente nella gestione dei file cifrati

Applicazioni che cifrano i dati

Svantaggi:

- L'aggiornamento puo' essere laborioso
- Lavorano soltanto sui "loro" file
- La gestione delle chiavi (una per applicazione) e' ancora a carico dell'utente

"user-level cryptography is cumbersome"

Matt Blaze

Cifratura a livello di sistema

- Trasparente all'utente ed alle applicazioni
- Maggiore robustezza

Cifratura a livello di sistema

- Protezione hardware del disco
- Dischi virtuali protetti
- File System cifrato

Cifratura a livello di sistema

Dischi con protezione Hardware:

Vantaggi:

- Buone prestazioni
- Affidabilita'
- Trasparenti alle applicazioni

Cifratura a livello di sistema

Dischi con protezione Hardware:

Svantaggi:

- Costo elevato
- Nessuna protezione dei backup
- Inadatti ad ambienti distribuiti

Cifratura a livello di sistema

Dischi virtuali protetti

Vantaggi:

- Costo contenuto
- Flessibilita'
- Trasparenti alle applicazioni

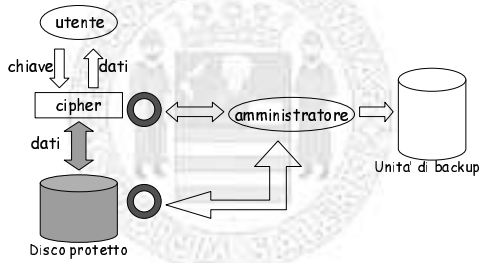
Cifratura a livello di sistema

Dischi virtuali protetti:

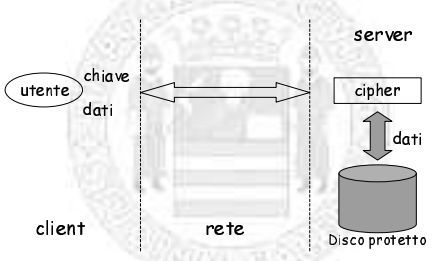
Svantaggi:

- Nessuna protezione dei backup
- Inadatti ad ambienti distribuiti

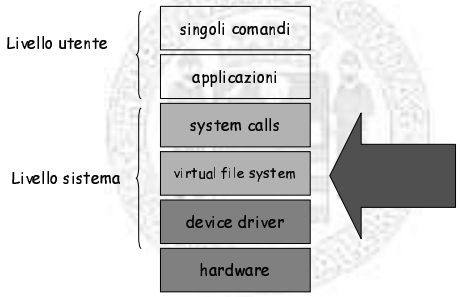
Il problema del backup



Il problema della condivisione



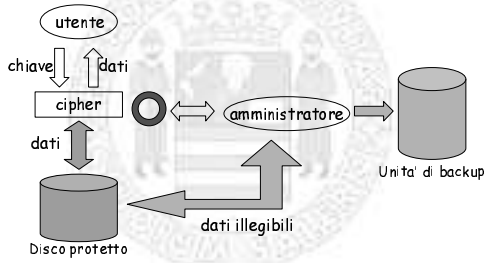
dove introdurre la cifratura?



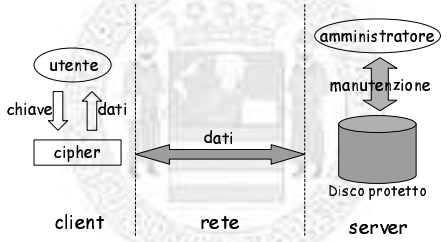
in questo modo:

- Non viene modificata l'organizzazione logica del disco
- La cifratura ha luogo sul contenuto dei file, a valle delle system call di read/write

Backup dei dati protetti



Protezione dei dischi condivisi



Perché cifrare un file system?

perché…

- Si possono proteggere i file senza l'intervento manuale dell'utente
- L'accesso ai file protetti non ne aumenta la vulnerabilita'
- Non e' necessario alcuna modifica alle applicazioni preesistenti
- Si mantiene la compatibilita' con le normali procedure di manutenzione dei dischi

Servizi e caratteristiche di un file system cifrato

Principali servizi di un F.S. cifrato

- Protezione di dati e *meta-dati*
- Interfaccia con l'utente/applicazione
- Condivisione dei file protetti
- Gestione delle chiavi
- Recupero delle chiavi
- Controllo sull'integrita' dei file

Alcuni criteri di valutazione di un file system cifrato

- Sicurezza
- Efficienza
- Trasparenza
- Portabilita'

Sicurezza

- Scelta dell'algoritmo di cifratura
- Scelta delle componenti fidate
- Rigore nell'implementazione

Efficienza

- Granularita' della protezione
  - Cosa proteggere, quando e quanto
- Semplicita' del modello
- Dipendenza dall'architettura/s.o.
  - Utilizzo di sistemi di cache per i dati
  - Impiego di risorse a basso livello

Casi di studio

Cryptographic File System (CFS)

Matt Blaze, AT&T Bell Labs, 1993

- Realizza una protezione dei dati orientata alle directory
- Gira in modalita' utente
- E' disponibile sulla maggior parte dei sistemi UNIX
- E' forse il piu' diffuso file system cifrato

Come funziona

- Crea directory cifrate sul disco
- Mappa le directory cifrate per chi ne fa richiesta (fornendo la chiave giusta) nel filesystem */crypt*
- Dopo l'immissione della chiave, il proprietario dei file vi accede in maniera del tutto naturale
- Quando l'utente ritira la chiave, la directory in chiaro scompare dal filesystem */crypt*
- Nella directory "originale" il contenuto ed il nome dei file resta sempre cifrato

Creazione di una directory cifrata

cmkdir /home/ciccio/privato
Key:(*inserire la pass-phrase*)
Again:

Accesso ad una directory cifrata

cattach /home/ciccio/privato formaggio
Key:(*inserire la pass-phrase*)
ls /crypt
formaggio
echo 'ciao'>/crypt/formaggio/saluti
ls /crypt/formaggio
saluti
cat /crypt/formaggio/saluti
ciao

Accesso alla directory sottostante

ls /home/ciccio/privato
dksdHG8sdjk
cat /home/ciccio/privato/dksdHG8sdjk
alksfhuih835lkl&*(35oihjs

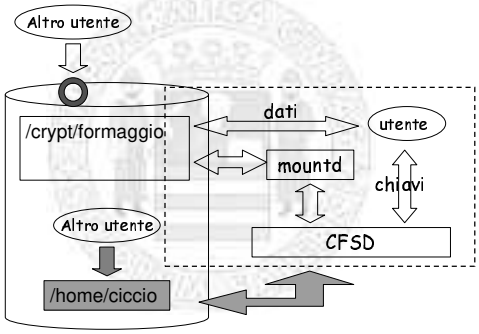
Termine della sessione

cdetach formaggio
ls /crypt
ls /home/ciccio/privato
dksdHG8sdjk

Implementazione

- CFS e' realizzato come un daemon che gira in modo utente ed "esporta" un filesystem NFS alla sua stessa macchina
- Nessuna modifica e' apportata al sistema operativo, alla struttura logica del disco ed al client NFS

Implementazione



CFS: sicurezza

- Utilizza l'algoritmo DES in modalita' ECB+OFB
- Nessuna informazione utile sui file e sulla chiave esce dalla zona fidata
- Nessun dato in chiaro circola al di fuori della zona fidata (neppure se il filesystem sottostante e' remoto)

CFS: Efficienza

- I dati protetti devono sempre transitare per le funzioni di rete
- Tutti i file presenti nella directory vengono indiscriminatamente cifrati
- Non si fa uso di risorse a basso livello
- Ogni accesso ad un file protetto e' in realta' un doppio accesso

CFS: trasparenza

- CFS e' trasparente alle applicazioni
- L'utente deve "ricordare" una pass-phrase per ogni directory cifrata
- Non e' possibile utilizzare normalmente il GID di un file protetto

CFS: inoltre

- Non e' prevista l'esplicita condivisione di file
- Non sono previsti meccanismi di recupero delle chiavi perse
- Non sono previsti meccanismi di controllo dell'integrita' dei file

Encrypting file system (EFS)

Microsoft corp., Redmond, 1998

- Cifra singoli file o directory
- Gira su MS Windows 2000
- E' strettamente collegato ad NTFS

Come funziona

- Crea cartelle protette sul disco, e cifra e decifra automaticamente i file in esse conenuti
- Crea singoli file protetti e vi accede automaticamente attraverso le funzioni di cifratura
- Le chiavi di cifratura possono essere gestite autonomamente dal sistema senza l'intervento dell'utente

in particolare

La protezione dei file e' indipendente da quella della directory in cui sono contenuti

infatti: l'operazione di spostamento di un file di una directory protetta in una directory non protetta...

non comporta la "sprotezione" del file

Protezione di file e directory

- L'utente, attraverso il pannello delle proprieta' attiva/disattiva la protezione
- EFS genera una chiave di cifratura
- EFS genera una coppia di chiavi pubblica/privata per l'utente
- EFS cifra con la chiave pubblica dell'utente la chiave di cifratura e la immagazzina nel file

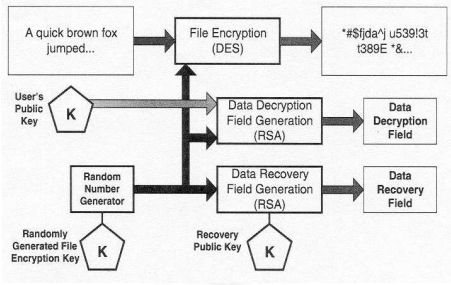
Protezione di file e directory/2

- EFS genera una coppia di chiavi pubblica/privata per l'agente di recovery (se non e' gia' stato fatto prima)
- EFS cifra con la chiave pubblica dell'agente la chiave di cifratura e la immagazzina nel file

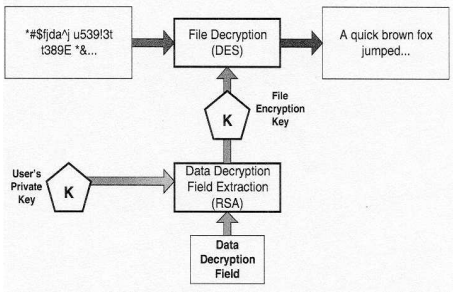
accesso ai file cifrati

- L'utente, accede in maniera naturale alle cartelle/file protetti
- EFS ottiene la chiave privata dell'utente estrae la chiave di cifratura dal file e la decifra
- EFS cifra/decifra i dati inviati/richiesti dall'utente con la chiave cosi' ottenuta

Scrittura in un file cifrato



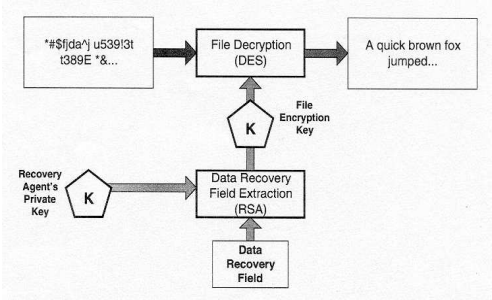
Lettura in un file cifrato



in caso di smarrimento della chiave

- L'agente di recovery decifra il file ottenendo la chiave da esso attraverso la sua chiave pubblica

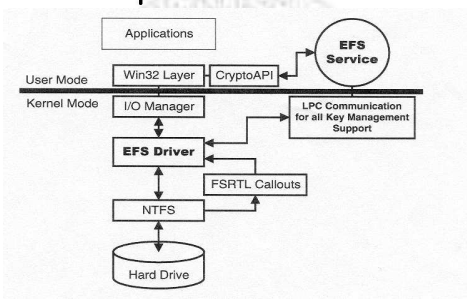
in caso di smarrimento della chiave



implementazione

- EFS driver: realizza le operazioni di key management, passa le informazioni necessarie all'accesso al F.S. cifrato alla FSRTL
- EFS FSRTL: gestisce le operazioni sul filesystem (read, write)
- EFS Service: realizza le operazioni di key management dal lato utente, gestisce le comunicazioni tra applicazioni ed EFS

implementazione



EFS: Sicurezza

- EFS utilizza l'algoritmo di cifratura DESX con chiavi da 128bit(U.S.) o da 40 bit (resto del mondo)
- La condivisione di file protetti via rete non e' protetta, ma richiede esplicitamente l'uso di un protocollo di rete sicuro esterno

EFS: Sicurezza/2

- La gestione delle chiavi (pub. e pri.) dell'utente non e' chiaramente definita
- Non e' prevista una esplicita protezione delle operazioni di recovery
- Non e' prevista una esplicita protezione dei backup

EFS: componenti fidati

- L'amministratore di sistema (recovery agent)
- La rete locale
- Le autorita' per la sicurezza di un dominio NT
- La workstation dell'utente (memoria, buffer, etc.)

EFS: Portabilita'

- EFS gira sotto Windows 2000 con il filesystem NTFS versione 5
- La copia di file cifrati su sistemi con versioni NTFS differenti (o con filesystem FAT) comporta la loro decifratura

EFS: inoltre

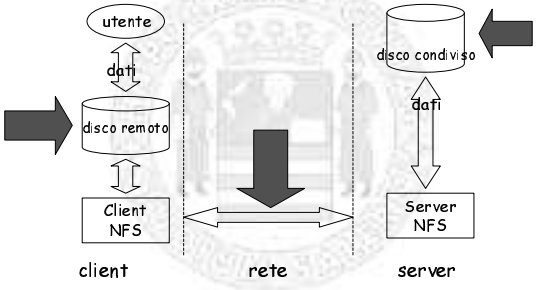
- EFS e' trasparente alle applicazioni
- L'utente deve fare attenzione a dove copia i file (pena inconsapevoli sprotezioni)
- L'utente potra' non immettere mai password di alcun genere, tutte le chiavi sono generate casualmente ed immagazzinate nel sistema

Transparent Cryptographic File System (TCFS)

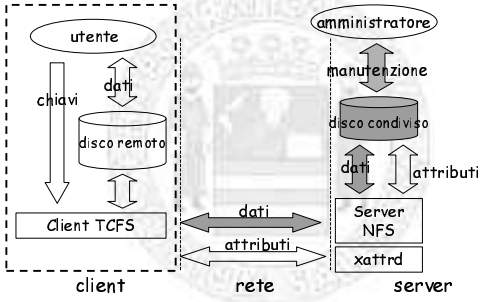
Universita' di Salerno, 1996

- Nasce per garantire la protezione dei dati in ambiente distribuito
- Cifra singoli file o directory
- E' realizzato come modulo dei kernel Linux (2.0.x)
- Fornisce la condivisione di file o directory tra gruppi di utenti

Network File System (NFS)



Architettura di TCFS



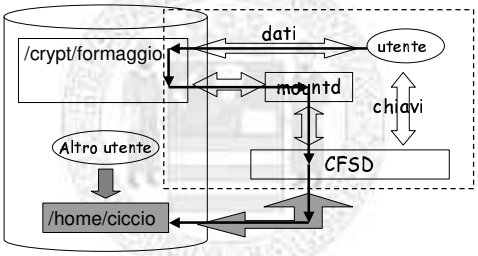
Architettura di TCFS: client

- TCFS client: riproduce tutti i servizi di un NFS, e' realizzato come modulo del kernel
- Utility di interfaccia

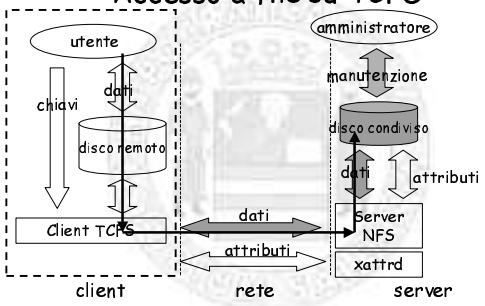
Architettura di TCFS: server

- NFS server standard
- xattrd: daemon che gestisce gli attributi di protezione di file e directory, gira in modo utente

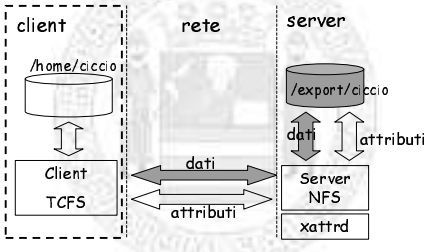
Accesso a file su CFS



Accesso a file su TCFS



Predisposizione di un filesystem TCFS



Gestione degli account

tcfsadduser ciccio
user 'ciccio' was successfully created

tcfsrmuser ciccio

Creazione della chiave

tcfsgenkey
password:*(si inserisce la password di sistema)*
please press 10 keys:*(dieci tasti a caso)*

Inizio di una sessione

tcfsputkey
password: *(si inserisce la password di sistema)*

Creazione di una directory cifrata

mkdir /home/ciccio/privato
chattr +x /home/ciccio/privato

Creazione di un file cifrato

mkdir /home/ciccio/privato2
cd /home/ciccio/privato2
cp /etc/passwd ./prova
chattr +x prova

Accesso ad una directory cifrata

cd /home/ciccio/privato
echo 'ciao'>saluti
cat saluti
ciao

Termine della sessione

tcfsrmkey

cd /home/ciccio/privato

permission denied

cd /home/ciccio/privato2

cat prova

permission denied

Accesso alla directory sottostante

ls /export/ciccio/privato

dksdHG8sdjk

cat /export/ciccio/privato/dksdHG8sdjk

alksfhuih835lkl&*(35oihjs

ls /export/ciccio/privato2

prova

cat /export/ciccio/privato2/prova

dfgsd88dsfP{KSF}KJMN$#

Condivisione di file cifrati

E' possibile effettuare la condivisione di file tra gruppi di utenti secondo lo schema a soglia (n,k)

In un gruppo di n utenti, per consentire la fruibilita' dei file condivisi, almeno k di esse deve fornire a TCFS la sua porzione

Creazione di account di gruppo

tcfsaddgroup -g gruppo

il sistema chiede all'utente il numero di componenti del gruppo, il valore della soglia, e le uid degli utenti

group 'gruppo' was successfully created

tcfsrmgroup -g gruppo

Inizio di una sessione di gruppo

tcfsputkey -g gruppo

password: *(si inserisce la password di sistema)*

La chiave di cifratura per il gruppo indicato, sara' attiva solo quando un numero di componenti pari al valore della soglia, ha effettuato l'operazione

Creazione di un file condiviso

mkdir /home/ciccio/privato2

cd /home/ciccio/privato2

cp /etc/passwd ./prova

chattr +g prova

termine di una sessione di gruppo

tcfsrmkey -g gruppo

La chiave di cifratura per il gruppo indicato, sara' disattivata solo quando il numero di componenti che hanno fornito la loro porzione di chiave, risultera' inferiore al valore della soglia

TCFS: sicurezza

- Utilizza DES, 3DES, IDEA, RC5, il motore di cifratura e' modulare
- Il motore di cifratura e' utilizzato secondo lo schema CBC

TCFS: sicurezza/2

- Nessun nessun dato/meta-dato circola in chiaro fuori dalla zona fidata
- Le chiavi di cifratura non escono dalla zona fidata
- Non e' possibile, sul server, accedere al contenuto dei file cifrati

TCFS: trasparenza

- L'utente deve ricordare solo la sua password di sistema
- Non occorrono modifiche agli applicativi
- Compatibilita' con le operazioni di manutenzione del disco (fsck, backup, etc.)

TCFS: efficienza

- Scalabilita' della cifratura (possono coesistere sul filesystem file protetti e non)
- I tempi di accesso(al netto della cifratura) sono confrontabili con quelli di NFS

TCFS: portabilita'

- TCFS gira sui sistemi Linux (kernel 2.0.x), necessita di rimaneggiamenti ad ogni upgrade del kernel
- Puo' fungere da server, qualsiasi macchina server NFS, su cui sia compilato il daemon xattrd (che gira in modo utente)

TCFS: inoltre

- E' stato realizzato un prototipo di TCFS per i sistemi operativi NetBSD ed OpenBSD
- E' allo studio un meccanismo di verifica dell'integrita' dei file
- Non sono previsti meccanismi di key-recovery