Studio e realizzazione di strumenti di crittografia per sistemi cloud-storage

Progetto realizzato: pubcFS

Presentata da: Miro Mannino

Relatore: Renzo Davoli

Sessione II
A.A. 2010-11

Sicurezza del cloud computing

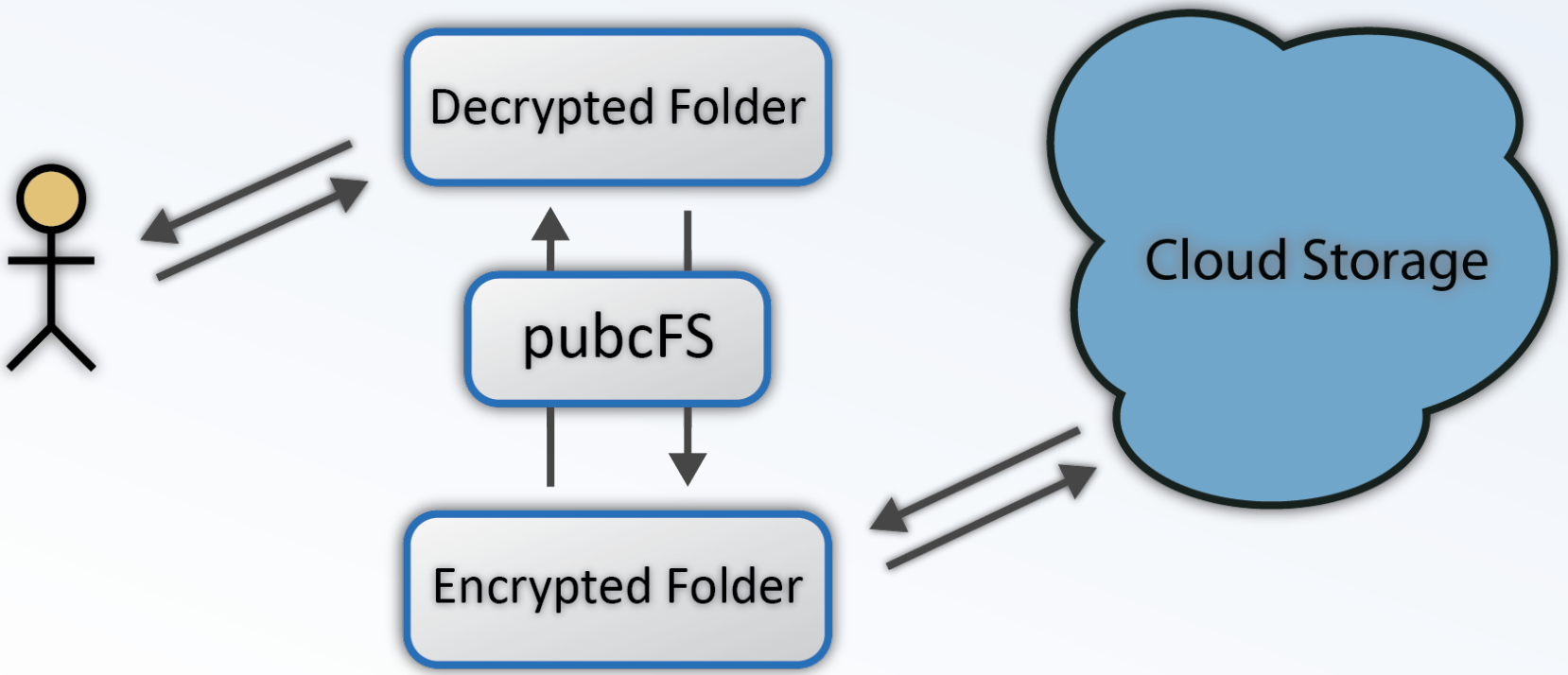
Richard Stallman affermò per primo che il cloud computing è pericoloso.

Affidamento dei propri dati a terze parti:

- La legge applicabile è quella del paese dove risiedono i dati.
- Stati con legislazioni troppo permissive non proteggono da abusi.
- In alcuni stati: libero accesso alle autorità senza alcuna autorizzazione. (Patriot Act)
- Protezione fisica dei dati.

pubcFS

- File system virtuale crittografico on-the-fly.
- Implementato utilizzando *FUSE*, scritto in C.
- Realizzato con l'intento di tutelarsi da servizi di cloud-storage come Dropbox.

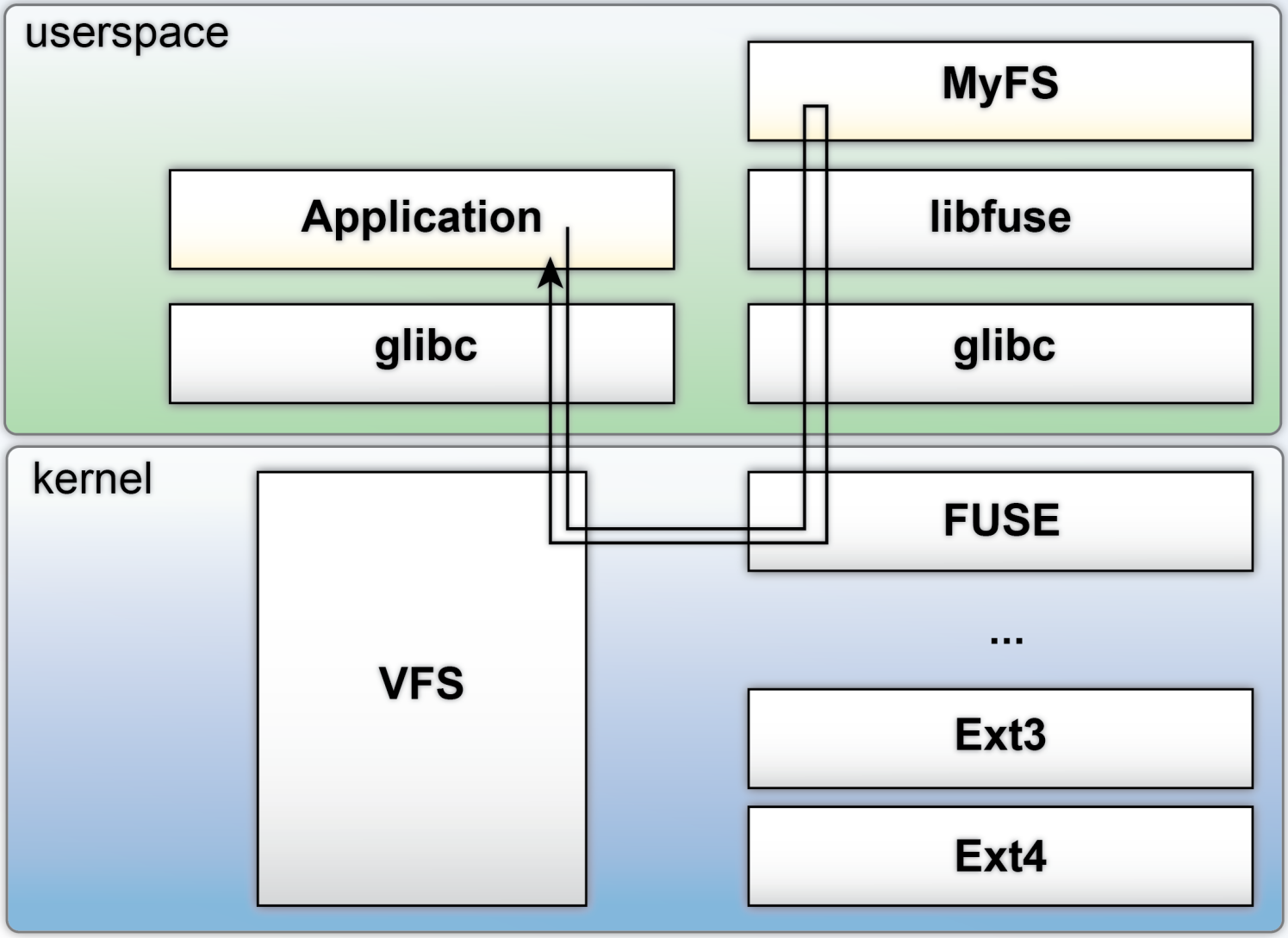


FUSE - Filesystem in Userspace

Permette l'implementazione di un *file system* completo in *userspace*.

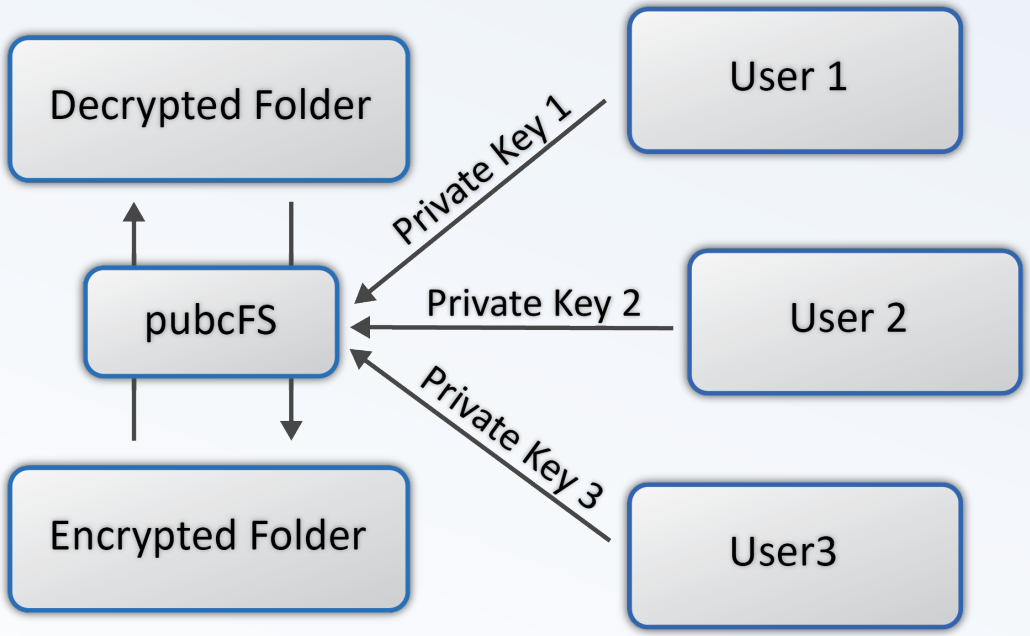
- Riscrittura delle *system call.*
- Utilizzo delle *system call* già presenti.
- Semplice installazione: nessuna *patch* o ricompilazione del *kernel.*
- Utilizzabile dagli utenti senza privilegi.

FUSE


userspace
MyFS
Application
libfuse
glibc
glibc
kernel
FUSE
VFS
...
Ext3
Ext4

pubcFS

- Simile ad *EncFS, CryptoFS ed altri.*
- Più utenti che condividono la stessa cartella non devono concordare una password.
- Per decifrare la cartella l'utente deve fornire la propria chiave privata.



pubcFS - Gestione utenti

- Un utente che ha accesso alla cartella può essere abilitato alla decifratura.
- Le informazioni di ogni utente sono memorizzate nella cartella reale tramite dei file.
- In questo modo è lo stesso servizio di cloud-storage a sincronizzare lo stato attuale degli utenti.
- Permessi sui file e gerarchia fra gli utenti gestibili tramite il servizio di cloud-storage.

pubcFS - Algoritmi crittografici

Algoritmo a chiave simmetrica per i contenuti

- *AES* in *CFB mode*
- Chiave simmetrica creata in maniera random all'inizializzazione.
- Chiave memorizzata in modo diverso per ogni utente utilizzando la loro chiave pubblica.
- Viene utilizzato *RSA* per la cifratura e decifratura della chiave simmetrica.
- Lettura delle chiavi pubbliche e private in formato *PEM.*

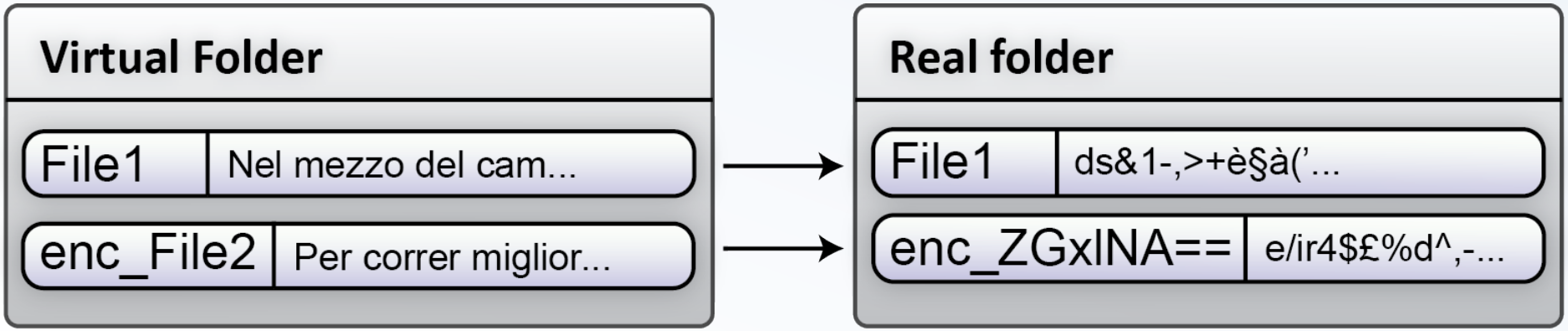
pubcFS - Read & Write

Suddivisione file in blocchi

- La scrittura di un solo byte modifica tutto il contenuto del blocco in cui appartiene.
- Cambiamenti locali ad un file modificano localmente il file, favorevole in fase di sincronizzazione per servizi come Dropbox.
- Dimensione blocchi configurabile per gestire il trade-off fra sicurezza e velocità.

pubcFS - Percorsi cifrati

- I *filename* vengono opzionalmente cifrati.
- pubcFS cifra il *filename* dei soli file e cartelle che hanno un prefisso “enc_”.
- Cifratura del *filename* eseguita con *AES.*
- Risultato della cifratura trasformato con *base64url* per essere memorizzabile come f*ilename*.



pubcFS - Concorrenza

Compatibilità con *UMfuse:*

- Più istanze del *file system* eseguite in concorrenza.
- Tutto il contesto di *pubcFS* viene mantenuto nella `user_data` di *FUSE.*

FUSE e multithread:

- FUSE esegue le operazioni in *multithread.*
- *thread local storage* per la memorizzazione di parte del contesto di *pubcFS*.
- Vita del contesto dipendente dal thread.

Conclusioni - Sviluppi futuri

- L'introduzione della crittografia asimmetrica necessita la creazione di un'interfaccia più semplice per utenti inesperti.
- Gestione più complessa della cifratura utilizzando altri suffissi sui *filename*.
- Configurazione degli algoritmi crittografici utilizzabili.
- Possibilità di utilizzare un algoritmo qualunque personalizzato dall'utente.