Cryptographic Key Management for Backup Systems

Dan Boneh and Ric hard Lipton

Princeton Univer sity

Motivation

➢ **`REMOVE`** command misleads users into thinking file is completely erased.

➢ Files are still available on backup tape!

➢ Users may wish to completely erase certain files:

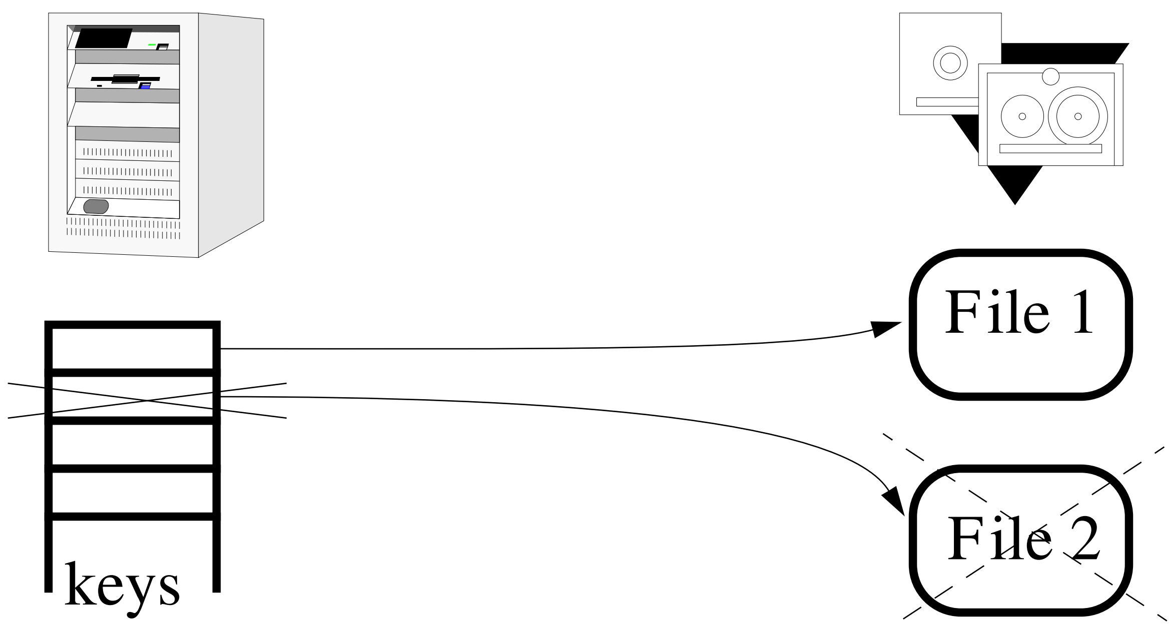
Personal mail messages,
Financial data,
Government documents.

➢ Can a file be erased without accessing the tape?

Solution

Encrypt file during backup.

Forgetting the encryption key *deletes* file from tape.



Crypto used to “forget” information.

Problem: key management.

Key Management

- All keys are stored in a `key-file`.
- `key-file` is protected (can only be accessed by root).
- `key-file` is bac ked up to tape encrypted with a `master-key`.
- `master-key` is stored on a separate system (e .g. off site).

File is completely removed only once `master-key` is erased.

key-file structure

Record structure:

file-name	
key-life	# keys
key(0) (current-key)	
key(1)	
⋮	
key(n)	
date-current-key-issued	

keys

During backup file is encrypted using current-key.

If `age-key>key-life` generate new key and delete oldest key.

Examples

1. Disk crash:
   First restore `key-file` from latest backup tape. Then restore file-system.

2. **`key-life`** $= \infty$, **`#keys`** $= 1$
   Any backup copy may be restored.
   Can remove ***all*** backup copies at will.

3. **`key-life`** $= 0$, **`#keys`** $= 1$
   Only most recent backup is available.

4. **`key-life`** $= 1$ month, **`#keys`** $= 12$
   One year of backup copies is available.
   May revoke at granularity of one month.

Master key management

➢ *`Master-key`* is used to recover *`key-file`* in case of disk crash.

➢ New *`master-key`* is generated at every bac kup (**`key-life`** = 0).

➢ T most recent *`master-key`*s are sa ved.

(**`#keys`** = T)

As a result file is completely removed after T da ys.

Master-key storage

Option I:

Sys-admin stores T recent master -keys on separate floppy disk.

Sys-admin ensures destruction of oldest `master-key.`

Option II:

Internet server .
MKSS - Master K ey Storage Server .

Internet server

1. MKSS generates new (public ,private) key pair daily . Same public key is sent to all sites .

2. During bac kup site writes to tape:

$$R = E(master-key, K_{public})$$

3. MKSS destroys private-key after T da ys.

To restore `key-file` site sends R to MKSS and receives `master-key` bac k.

Summary

➢Properties of the system:

1. Easy to install.
2. Little slow down in backup procedure.
3. Simple user interface.

➢Simpler than SFS (secure file system).

➢Block cipher should be "extra" secure.

➢Use secret-sharing to increase reliability of MKSS.