Festplattenverschlüsselung

TrueCrypt

Sebastian Berschneider
sebastian.berschneider@informatik.stud.uni-erlangen.de

16. Juni 2010

Motivation

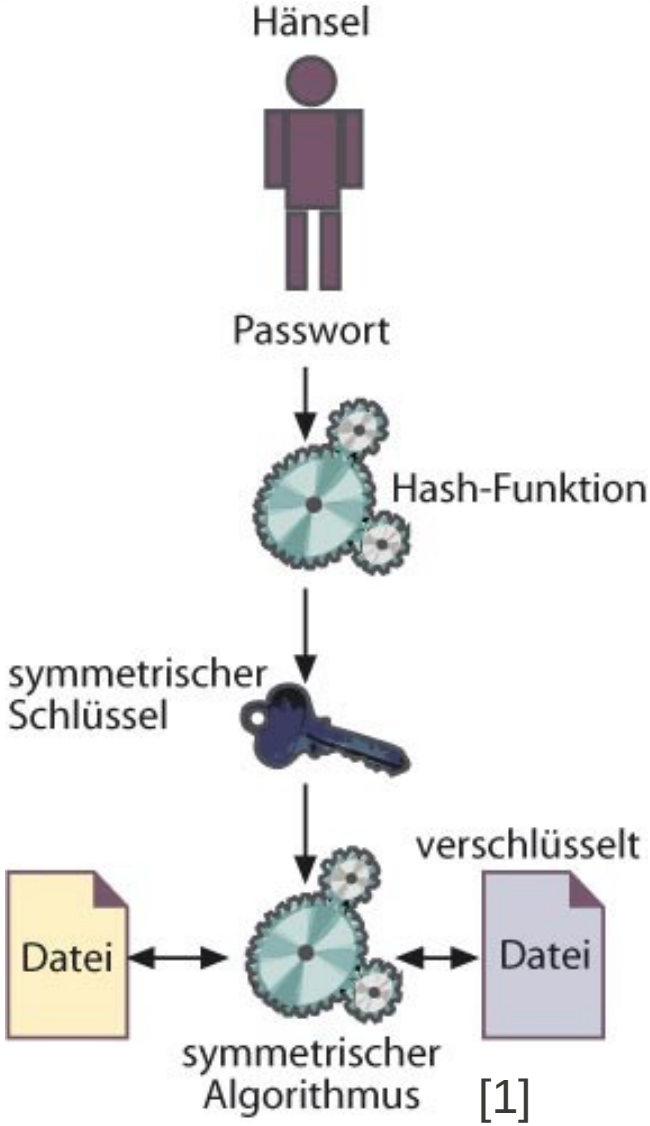
- Bei physischem Zugang des Angreifers können Schutzsysteme vom Betriebssystem ausgehebelt werden
  - Boot- oder Diagnose-CD-ROM hebelt Passwörter und Dateiberechtigungen aus
  - Sensible Daten liegen schutzlos auf der Platte
  - Selbst wenn sensible Daten verschlüsselt, können die Private-Keys (SSH, GPG,...) gelesen werden
- Selbst verschlüsselte Dateien können unverschlüsselt im Auslagerungsspeicher liegen

Inhaltsübersicht

- Motivation
- Grundlagen
  - Authentifizierung
  - Unterschiede zw. Partitions- und Dateiverschlüsselung
- TrueCrypt

Authentifizierung

- Passwort nicht selbst Festplattenschlüssel
  - Passwort mit Hash-Funktion umgewandelt
    → Symmetrischer Schlüssel
  - Verschlüsselung der Dateien durch symmetrischen Algorithmus



[1]

Authentifizierung

- Zertifikatsverfahren
  - Flexibler
  - Asymmetrische Verwaltung der Festplattenschlüssel
- *Public Key* dient der Verschlüsselung
  - Zum Dechiffrieren *private key* notwendig

Unterschiede zw. Partitions- und Dateiverschlüsselung

	Partitionsverschlüsselung (TrueCrypt)	Dateiverschlüsselung (eCryptFS)
Methode	Verschlüsselung der kompletten Partition	Verschlüsselung einzelner Dateien innerhalb eines Dateisystems
Speicherung verschlüsselter Daten	Allokieren eines Blocks aus dem vorhandenen Dateisystem;	Auf das gemountete Dateisystem; Ggf. Unverschlüsselte Daten auf gleicher Partition
Meta-Daten	verschlüsselt, daher nicht reproduzierbar	unverschlüsselt, Datei sichtbar im Dateisystem
Swap-Space	verschlüsselbar	Nicht verschlüsselbar

Inhaltsübersicht

- Motivation
- Grundlagen
- TrueCrypt
  - Methoden
  - Plausible Deniability
  - Header

Algorithmen

- für Verschlüsselung
  - AES
  - Twofish
  - Serpent
  - Kombinationen der oben genannten
- Hashing-Algorithmen (zum “Mischen”):
  - RIPEMD-160
  - SHA-512
  - Whirlpool

Verschlüsselungs-Möglichkeiten

- einzelne Partition oder Festplatte
- Container
- System-Verschlüsselung

Methoden

- Partitionen/Container werden als virtuelle Laufwerke eingebunden

- blockweise Ver-/Entschlüsselung
  - bei jedem Lesen/Schreiben muss ein Block des *Volumes* ent-/verschlüsselt werden

Pre-Boot-Authentication

- System-Verschlüsselung
- System-Partition wird komplett verschlüsselt
  - außer MBR und Bootloader
- spezieller Bootloader zum Entschlüsseln der Systempartition
- Swap-Partition verschlüsselt

Pre-Boot-Authentication

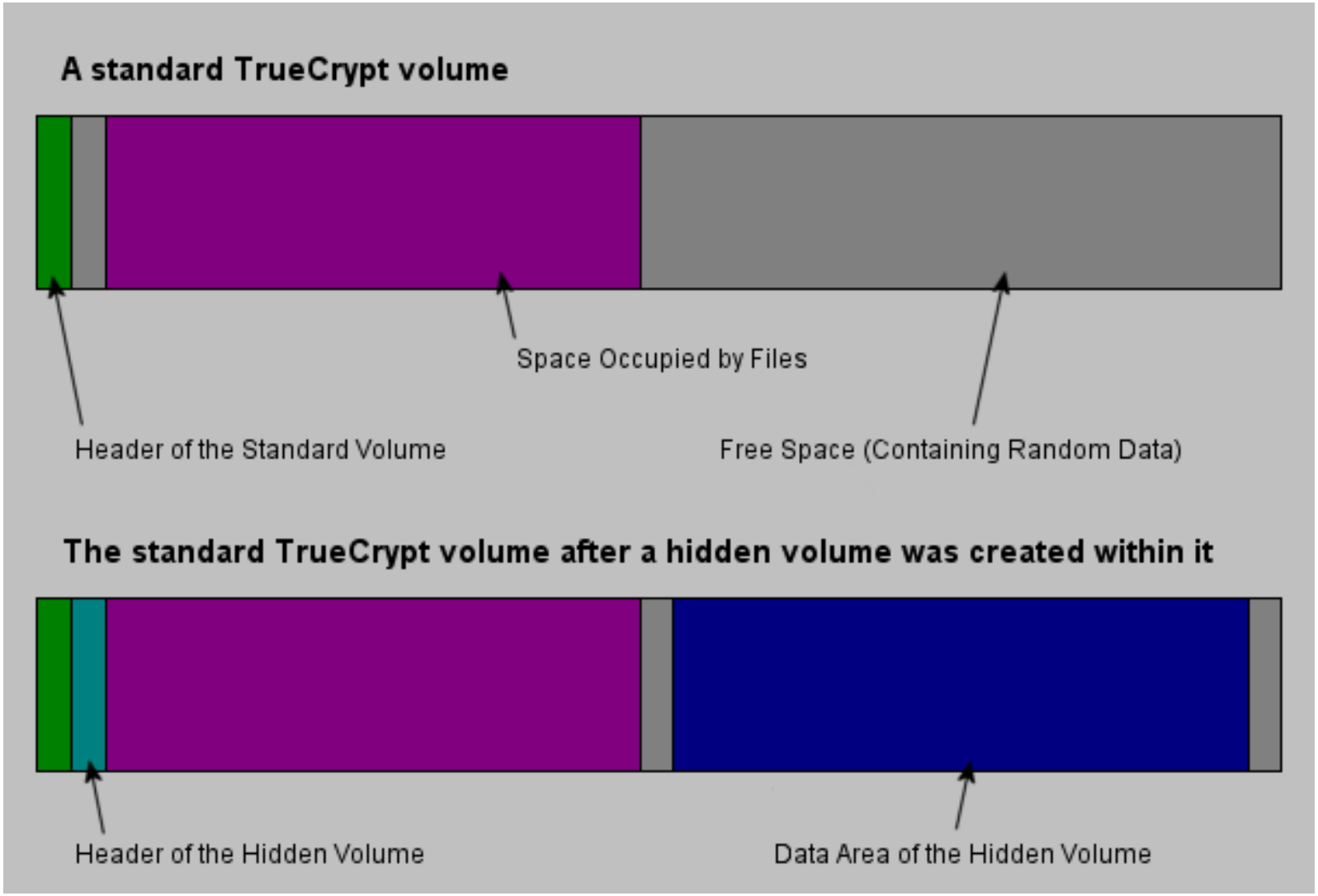
- System-Verschlüsselung nur unter Windows

- *Pre-Boot-Authentication* kann (momentan) durch ein Bootkit ausgehebelt werden
  - Frei verfügbar
  - Soll Entwickler anregen, diese Lücke zu schließen

Plausible Deniability

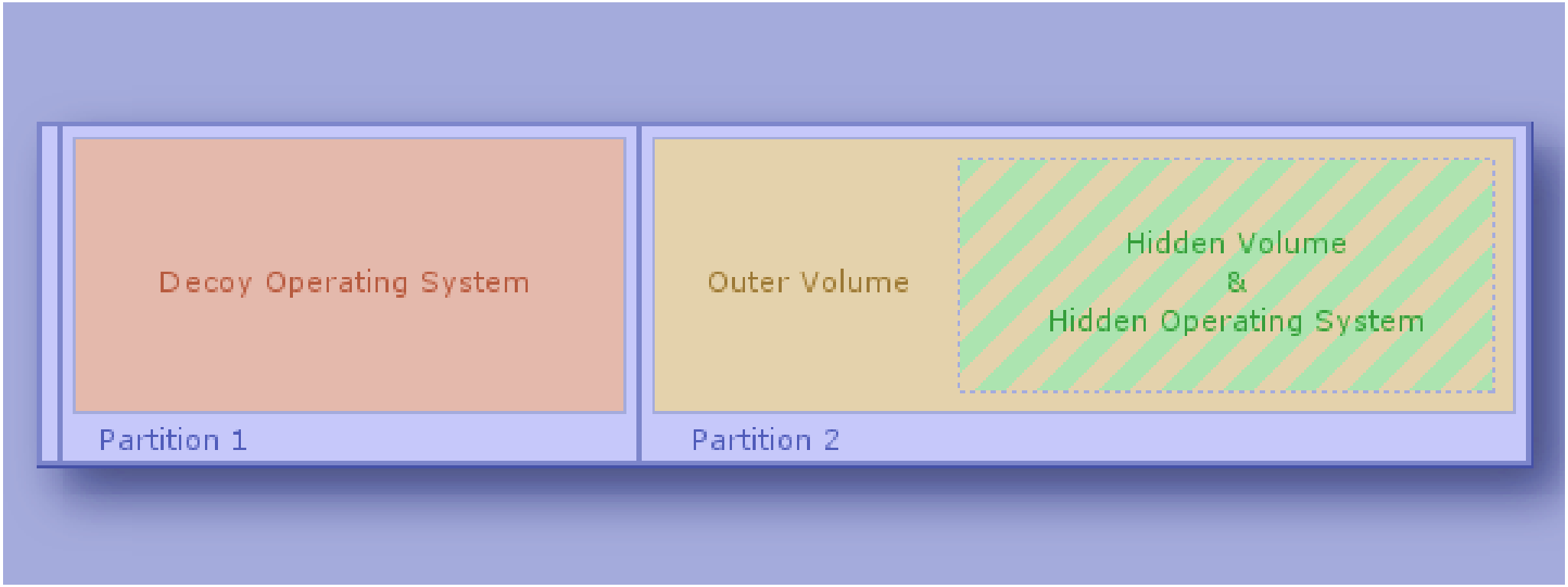
- *Hidden Volume*
- *Hidden Operating System*
- 2 verschiedene Passwörter
  - “echtes” Passwort bindet “echtes” (Hidden-)Volume ein
  - Pseudo-Passwort bindet Pseudo-Volume ein

Hidden Volume



[3]

Hidden Operating System



[3]

TrueCrypt-Header

- nicht eindeutig als TrueCrypt-Volume erkennbar
- werden mit Passwort/*Keyfiles* verschlüsselt
- *Header* enthalten eigentlichen *Master-Key*, der Daten verschlüsselt
  - dadurch Passwort-/Keyfile-Austausch möglich, ohne Volume neu zu verschlüsseln
  - Achtung: Master-Key wird NICHT neu erzeugt

Encryption Schema

Offset	Größe	Beschreibung
0	64	Salt
64	4	ASCII string “TRUE”
	...	...
72	4	CRC-32 Checksumme der (verschlüsselten) Bytes 256-511
...	...	...
252	4	CRC-32 Checksumme der (verschlüsselten) Bytes 64-251
256	Var.	Master Keys
...	...	...
65536	65536	Bereich für Hidden Volume Header
131072	Var.	Daten-Bereich
Size-131072	65536	Backup header
Size-65536	65536	Backup header for Hidden Volume

- Bytes 0-512 werden gelesen
- Bytes 65536–66047 werden gelesen
- Bytes 0-512 werden entschlüsselt (Trial and Error Prinzip)
  - Zusammenstellung von Passwort, Salt, Mischalgorithmus
  - Verschlüsselungs-Algo.
  - Operationsmodus
  - Schlüsselgröße(n)

Encryption Schema

Offset	Größe	Beschreibung
0	64	Salt
64	4	ASCII string “TRUE”
	...	...
72	4	CRC-32 Checksumme der (verschlüsselten) Bytes 256-511
...	...	...
252	4	CRC-32 Checksumme der (verschlüsselten) Bytes 64-251
256	Var.	Master Keys
...	...	...
65536	65536	Bereich für Hidden Volume Header
131072	Var.	Daten-Bereich
Size-131072	65536	Backup header
Size-65536	65536	Backup header for Hidden Volume

- Entschlüsselung erfolgreich, wenn #64 = “TRUE” und Checksummen mit gespeicherten Checksummen übereinstimmen
  - wenn nicht erfolgreich, entschlüssele Bytes 65536–66047
    - wenn wieder nicht erfolgreich, wird Mounting abgebrochen

Header-Backup

Offset	Größe	Beschreibung
0	64	Salt
64	4	ASCII string “TRUE”
	...	...
72	4	CRC-32 Checksumme der (verschlüsselten) Bytes 256-511
...	...	...
252	4	CRC-32 Checksumme der (verschlüsselten) Bytes 64-251
256	Var.	Master Keys
...	...	...
65536	65536	Bereich für Hidden Volume Header
131072	Var.	Daten-Bereich
Size-131072	65536	Backup header
Size-65536	65536	Backup header for Hidden Volume

- Header-Backup nicht Kopie des Headers
- Neben embedded auch external Header möglich
- defekter Header kann dadurch wiederhergestellt werden
- Bei external Header: Evtl. anderes Passwort

Zusammenfassung

- Verschlüsselung von Partition, Container, System (Pre-Boot-Authentication)
- Plausible Deniability (Hidden-Volumes, Hidden-Operating System)
- TrueCrypt-Volumes nicht erkennbar
- Passwörter problemlos austauschbar
- Header-Backups

Quellen

- [1] http://www.heise.de/ct/artikel/Datentresor-289846.html
- [2] http://www.invisco.de/files/VFS_Diagram.png
- [3] http://www.truecrypt.org
- [4] Max Lindner – Verschlüsselte Dateisysteme in Mehrbenutzer-Szenarien
- [5] http://pvs.informatik.uni-heidelberg.de/Teaching/DASY-07/seeliger.pdf
- [6] http://citp.princeton.edu/memory-content/memory_5.jpg
- [7] http://www.linux-magazin.de/Heft-Abo/Ausgaben/2007/03/Geheime-Geschaefte/(offset)/2
- http://ecryptfs.sourceforge.net/ecryptfs-faq.html
- http://sysphere.org/~anrxc/nsnd/nsnd-ecryptfs.html#sec-2.4

Quellen

- http://en.wikipedia.org/wiki/Comparison_of_disk_encryption_software
- http://pvs.informatik.uni-heidelberg.de/Teaching/DASY-07/nguyen.pdf
- Erez Zadok, Jason Nieh: FiST: A Language for Stackable File Systems
- Michael Austin Halcrow: eCryptfs: An Enterprise-class Cryptographic Filesystem for Linux
- http://berlin.ccc.de/~packet/cryptfs-eh02-workshop/cryptfs-eh02-workshop.pd
- http://koeln.ccc.de/archiv/drt//crypto/linux-disk.html#Cryptfs
- http://www.fsl.cs.sunysb.edu/docs/cryptfs/index.html
- c't Ausgabe 25/08: Lahmgesichert?