Forensic Analysis of the Resilient File System (ReFS) Version 3.4

DFRWS EU 2020

Paul Prade, Tobias Groß, Andreas Dewald

June 3rd-5th, 2020

IT Security Infrastructures Lab
Department of Computer Science
Friedrich-Alexander University Erlangen-Nuremberg (FAU)

Motivation

REFS IS A MODERN FILESYSTEM THAT IS DEVELOPED BY MICROSOFT

NOT OFFICIALLY DOCUMENTED

TRANSPARENCY IN FORENSIC PROCESSES AND TOOLS IS IMPORTANT

MORE IMPORTANT IF PROPRIETARY SOFTWARE/DATA GETS INVESTIGATED

Our Contributions

- We analyzed the internal structures and mechanics of ReFS v3.4
- We extended The Sleuth Kit (TSK) to support ReFS
- We propose strategies for recovering deleted files
- We implemented a page carver

Outline

- Background
- Important ReFS data structures
  - Showcase A: list content of root directory
  - Showcase B: list metadata of file
  - Showcase C: get content data of file
- Data Recovery
  - Deleted Files and Folders
  - Page carving
- Evaluation
- Summary

Background

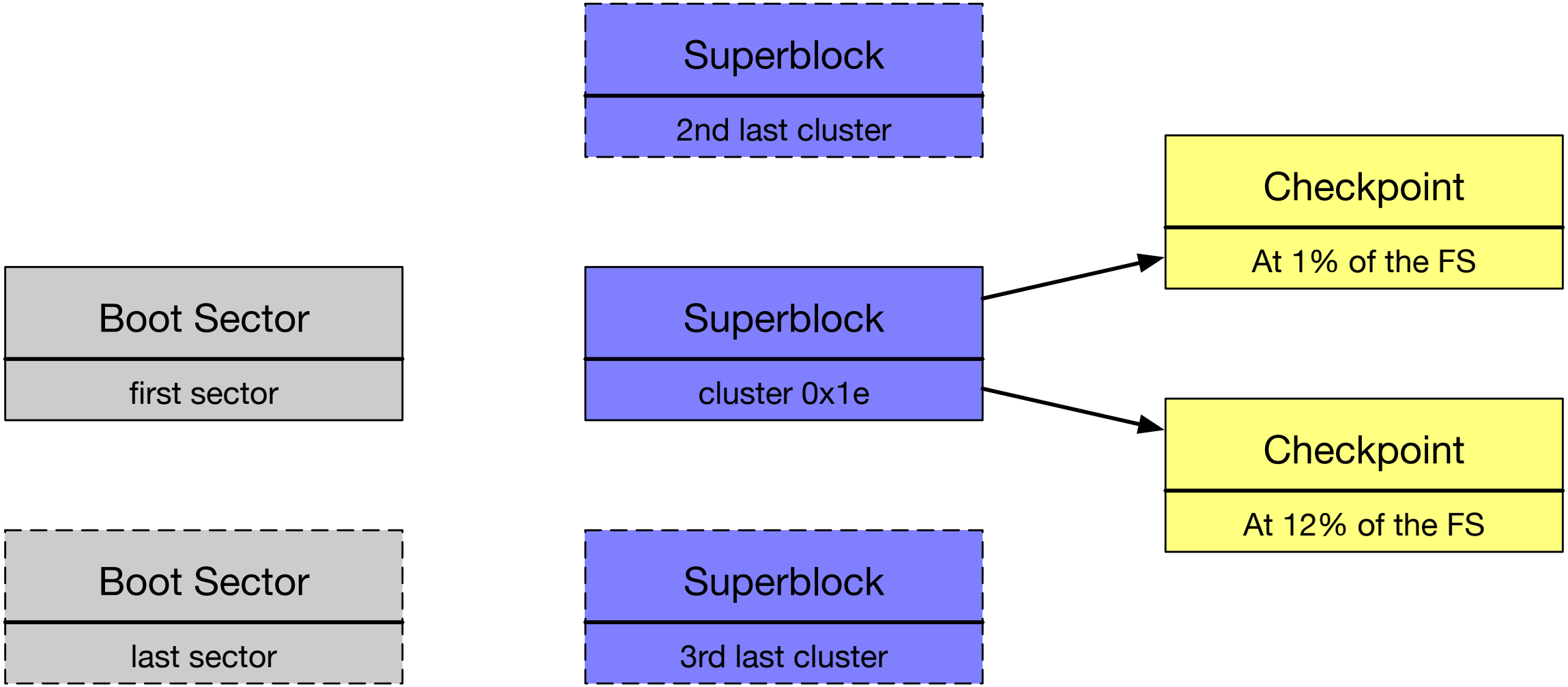
Background

- ReFS internal data is predominantly organized in tables / key value stores
- Content data is stored in data units called clusters (e.g. 4 KiB)
- Organizational file system data is stored in data units called pages (1-4 clusters)
- The copy on write mechanism operates on pages
- Tables are implemented as B+ tree, its nodes are complete pages

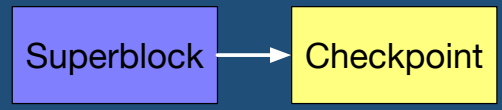
Important ReFS Data Structures

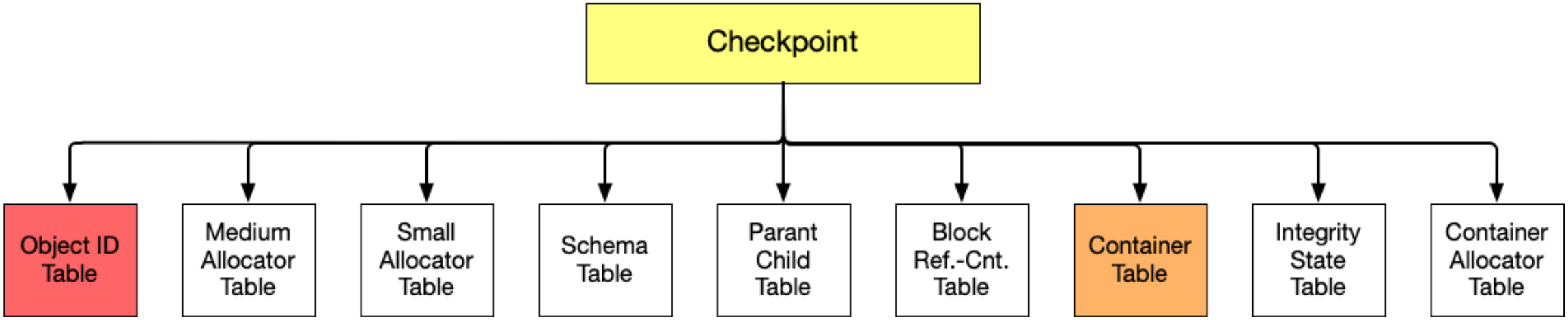
Showcase A: List Content of root Directory

Bootstrapping Filesystem Interpretation

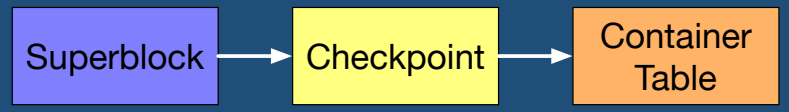


ReFS Root Tables

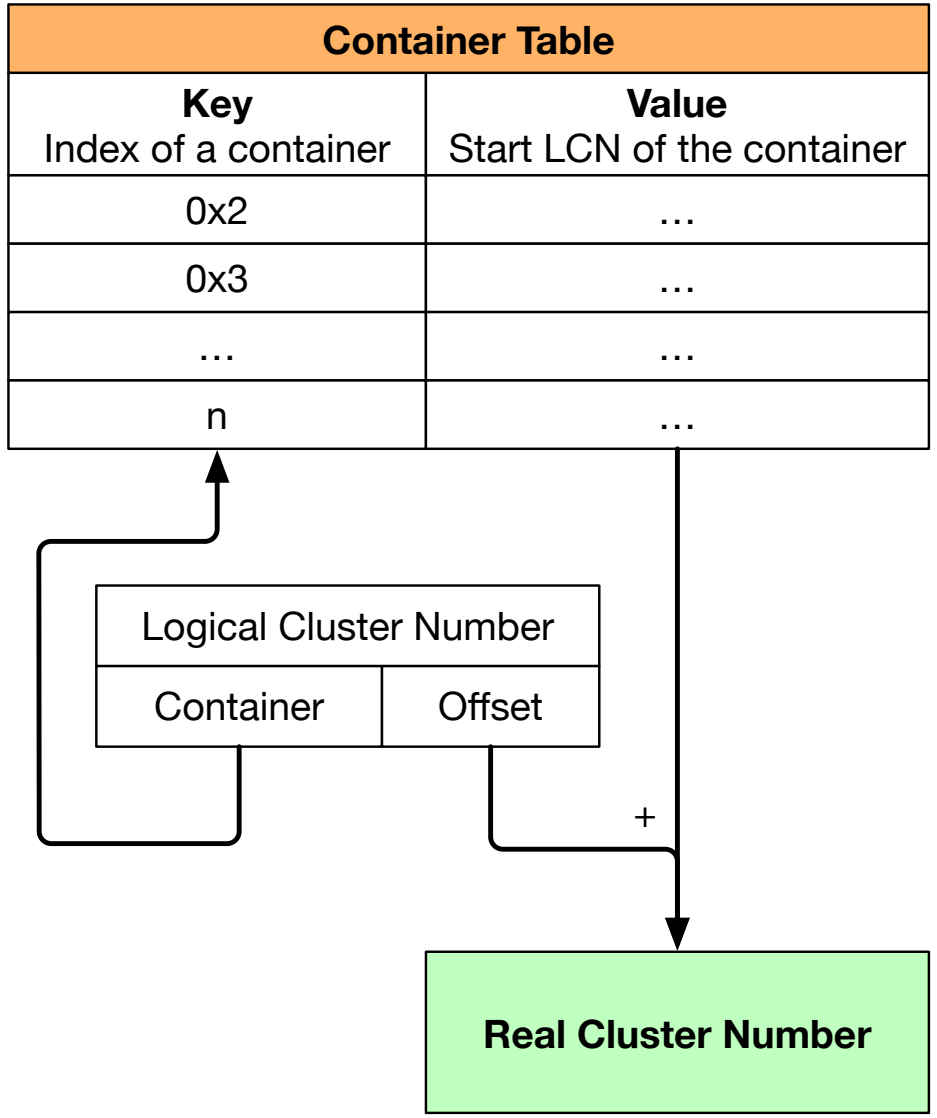




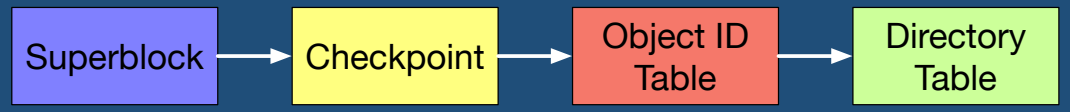
Container Table – Address Translation



- ReFS volume is separated into multiple bands
- Statistics about the usage of data and access times are collected
- Bands can be reorganized/swapped
- Container Table holds address translation data
- Superblock, Checkpoints, Container Table and Container Table Allocator Table use real addresses



Object ID Table – List root Directory



- Root directory address: 0x600
- Lookup Directory Table in Object ID Table
- Translate reference address to real cluster number
- Different entry types in Directory Table:
  - directory descriptor
  - file entry
  - directory link entry
  - ID2 entry

Object ID Table

Key	Table Reference
…	…
0x600	Reference + Checksum
0x701	Reference + Checksum
…	…

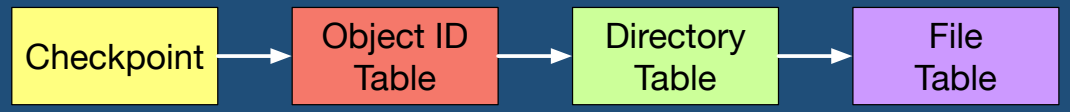
Directory Table (0x600)

Key	Table Reference
Dir Desc.	Metadata, ID: 0x0
ID: 0x1	f.txt
f.txt	Metadata, ID: 0x1
a/	Directory Ref: 0x701

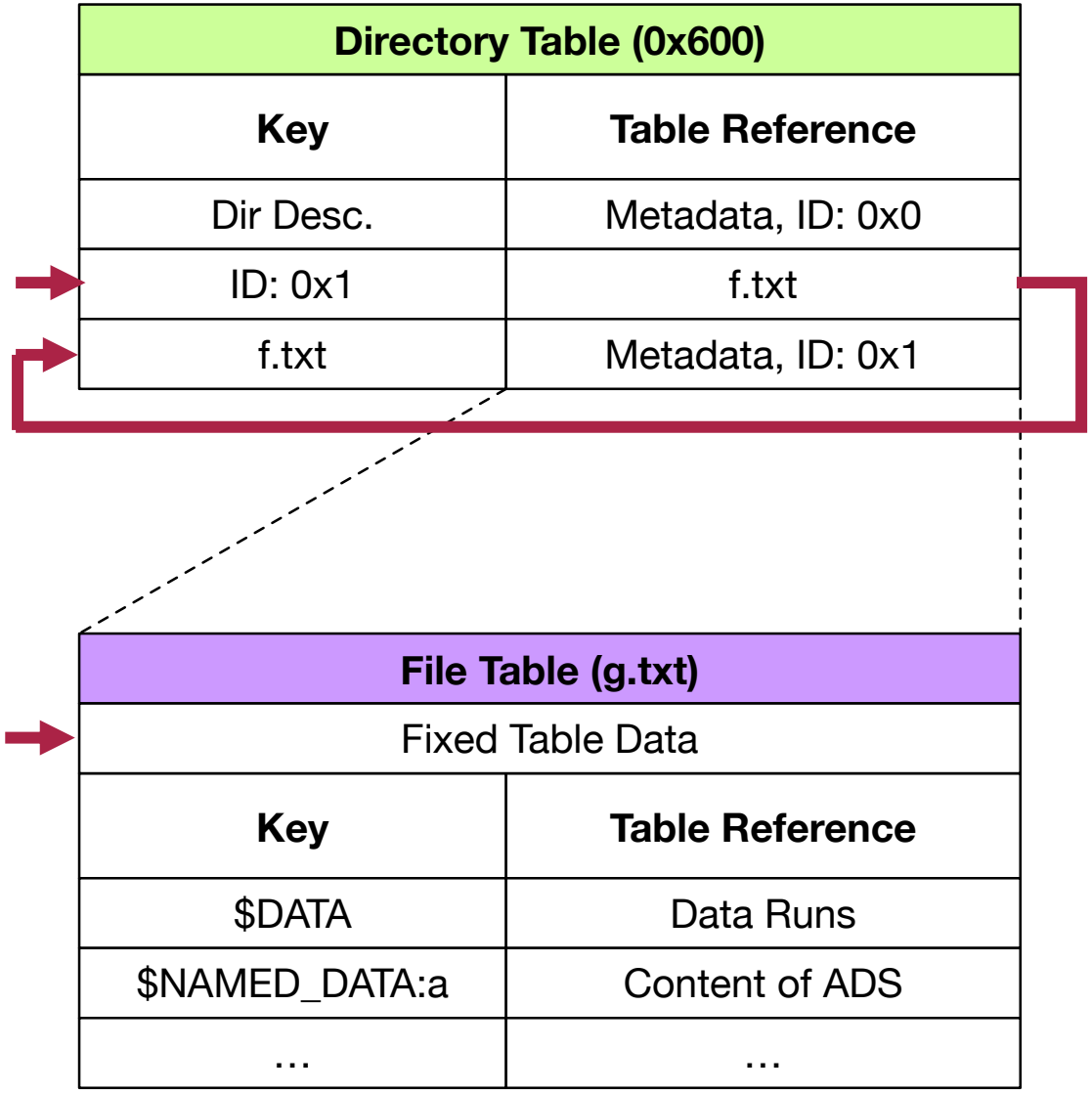
Important ReFS Data Structures

Showcase B: List Metadata of File

File Table – Home for the File Metadata



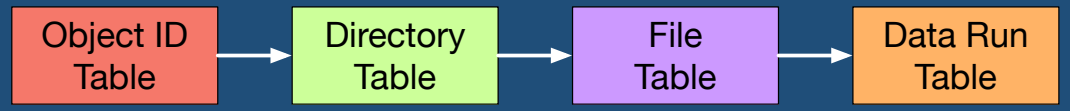
- Files can be accessed by metadata address or path, e.g. ./f.txt or 0x600|0x1
- File Table is embedded in Directory Table
- NTFS `$STANDARD_INFORMATION` is part of fixed table data. Contains e.g.:
  - MAC Timestamps
  - Size
  - Security Descriptor ID



Important ReFS Data Structures

Showcase C: Get Content Data of File

Data Run Table – Content of a File



- Data Run Table is embedded in File Table
- Stores rows of cluster runs which map virtual cluster numbers to logical cluster numbers
- Stores checksums of referenced clusters optionally

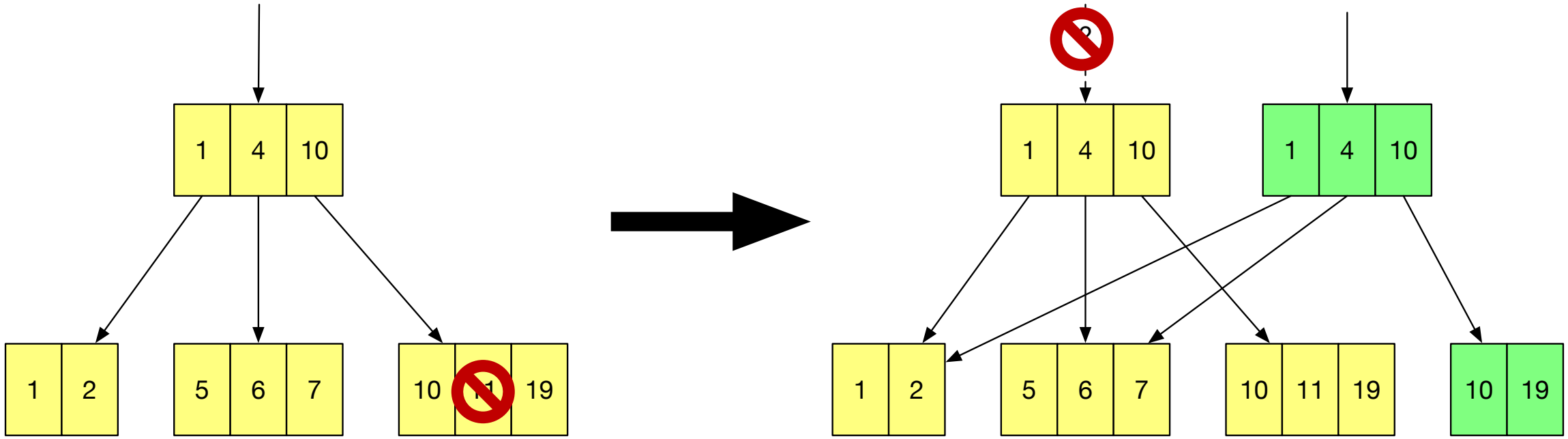
File Table (g.txt)	
Fixed Table Data	
Key	**Table Reference**
$NAMED_DATA:a	Content of ADS
$DATA	Data Runs

Data Run Table	
VCN | LCN | Length	**Checksums**
0 | 123 | 100	…
100 | 540 | 24	…
124 | 820 | 30	…
…	…

Data Recovery

Page Carving

ReFS Internal Data Organisation



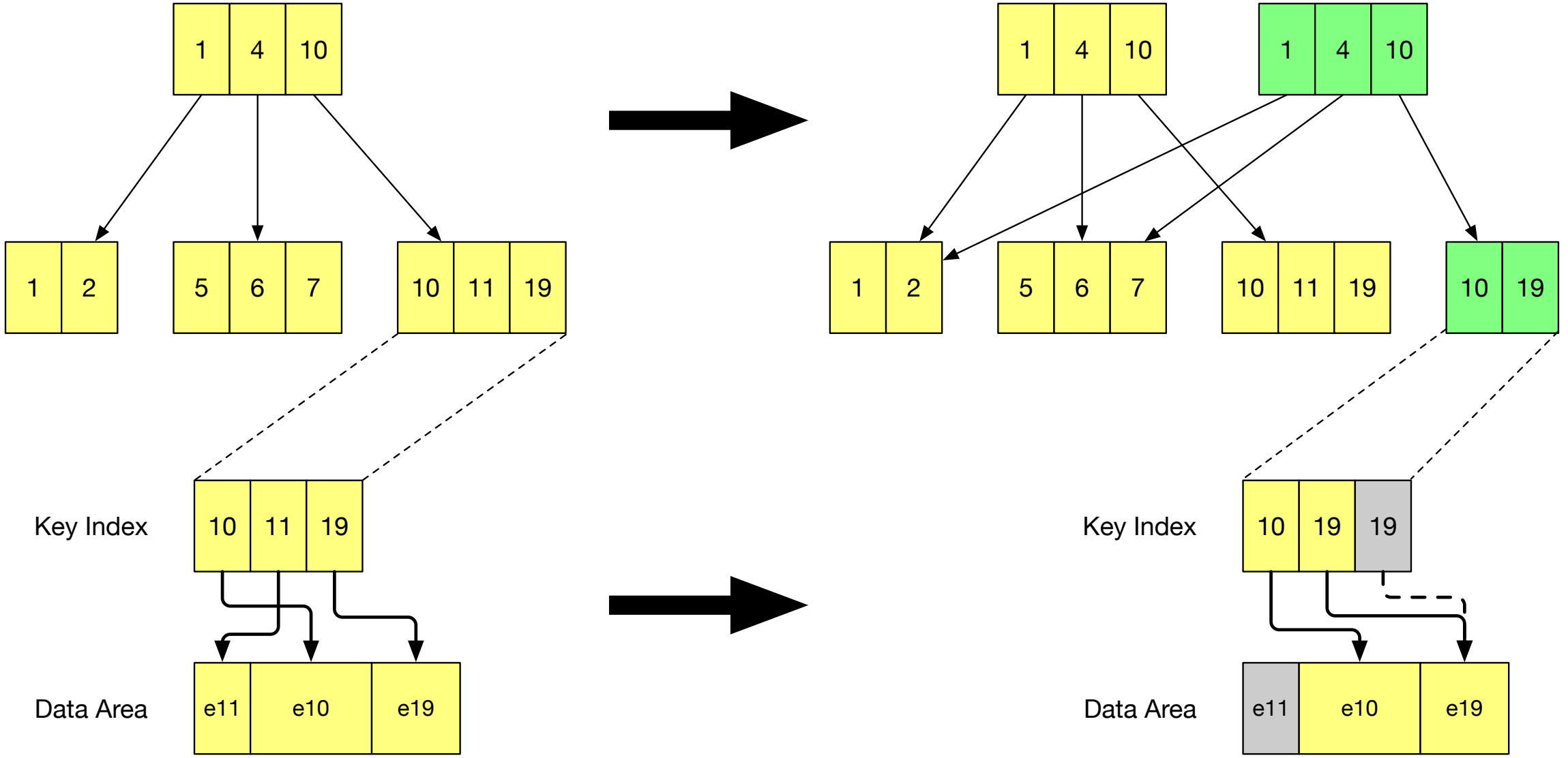
Page Carving

- Some nodes are not referenced by any accessible page anymore
- Scan every potential page of the disk for a page 80 bytes page header
- Table identifier allows to to dedicate found pages to specific tables

Data Recovery

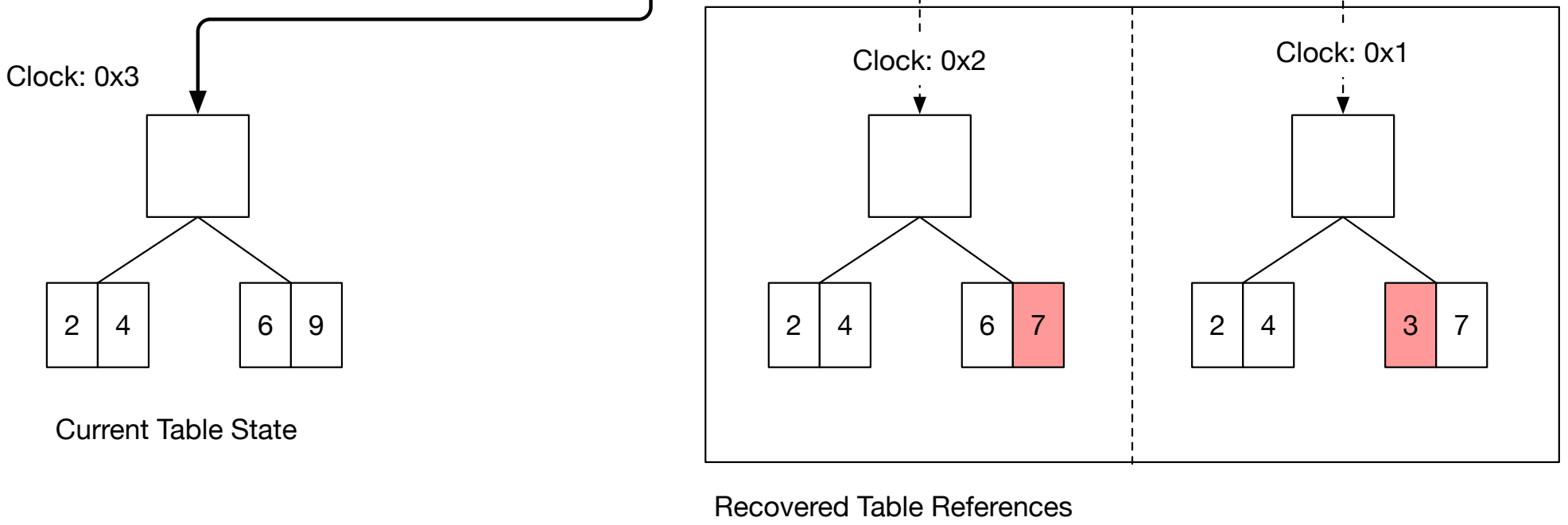
Deleted Files and Folders

Recover Deleted Files and Folders – Recover Deleted Entries in Tables



Recover older Filesystem States – Recover Entries in Object ID Table

Object ID Table	
Key	**Table Reference**
0x600	r1
0x600	r2
0x600	r3
…	…



Evaluation

Evaluation Data Set

- Generation of 8 test images with different total size and cluster size
- On each image we performed 1000 random actions:
  - add, delete, move, change, copy file
  - add, delete folder
- Documented outcome after every action in a TSK body file
- Logged only outcome for modified file/folder

Evaluation The Sleuth Kit Extension

Configuration	Interpreted entries	
	Directories	Files
2 GiB 4 KiB	47/46 (102%)	112/110 (101%)
2 GiB 64 KiB	44/43 (102%)	83/81 (102%)
5 GiB 4 KiB	45/44 (102%)	123/121 (101%)
5 GiB 64 KiB	20/19 (105%)	33/31 (106%)
40 GiB 4 KiB	25/24 (104%)	78/76 (102%)
40 GiB 64 KiB	56/55 (101 %)	143/141 (101%)
100 GiB 4 KiB	35/34 (102%)	64/62 (103%)
100 GiB 64 KiB	53/52 (101%)	110/108 (101%)

Output of the TSK extension, compared to the final state of the file system

Configuration	Restored entries	
	Directories	Files
2 GiB 4 KiB	15/154 (9%)	132/613 (21%)
2 GiB 64 KiB	19/159 (11%)	125/590 (21%)
5 GiB 4 KiB	14/144 (9%)	159/606 (26%)
5 GiB 64 KiB	28/159 (17%)	75/585 (12%)
40 GiB 4 KiB	12/129 (9%)	97/615 (15%)
40 GiB 64 KiB	11/159 (6%)	175/616 (28%)
100 GiB 4 KiB	10/147 (6%)	113/592 (19%)
100 GiB 64 KiB	13/152 (8%)	142/593 (23%)

Allocated and recovered files from TSK extension, compared to action log

Evaluation Page Carver

Configuration	Restored entries	
	Directories	Files
2 GiB 4 KiB	15/154 (9%)	139/613 (22%)
2 GiB 64 KiB	21/159 (13%)	133/590 (22%)
5 GiB 4 KiB	16/144 (11%)	169/606 (27%)
5 GiB 64 KiB	34/159 (21%)	88/585 (15%)
40 GiB 4 KiB	16/129 (12%)	102/615 (16%)
40 GiB 64 KiB	11/159 (6%)	183/616 (29%)
100 GiB 4 KiB	10/147 (6%)	124/592 (20%)
100 GiB 64 KiB	13/152 (8%)	150/593 (25%)

Allocated and recovered files from the carver compared to action log

Duration	Save Op.	Recovered States	Recoverable Pages
10 min	5	3 (2 valid)	6 / 15
30 min	15	3 (3 valid)	5 / 31
60 min	30	3 (3 valid)	5 / 47
120 min	60	3 (3 valid)	5 / 86
240 min	120	3 (3 valid)	4 / 141

Results of the experiment to analyze the recoverability of COW copies

Summary

- We investigated the internal structures of the Resiliant File System (ReFS)
- The implemented The Sleuth Kit (TSK) extension reports the same data as the official ReFS driver for the current state of a filesystem
- It can also reconstruct older states to some extent
- The implemented page carver can reconstruct more older states than the TSK extension

Thank you!