UsnJrnl Parsing for File System History

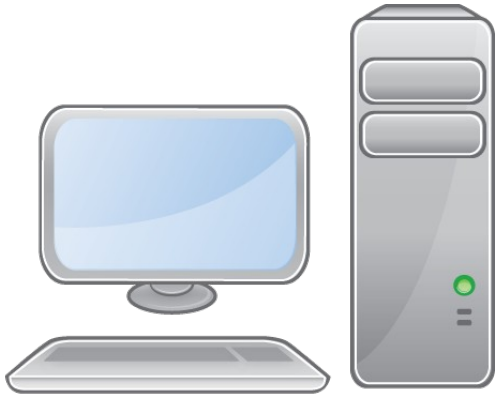
Students:
- Jeroen van Prooijen
- Frank Uijtewaal

Fox-IT:
- Yonne de Bruijn

Research question

How can the artefacts found in the UsnJrnl be effectively used in forensic research?

UsnJrnl?



Windows

Uses

NTFS

Contains metadata files like

UsnJrnl = Update sequence number Journal

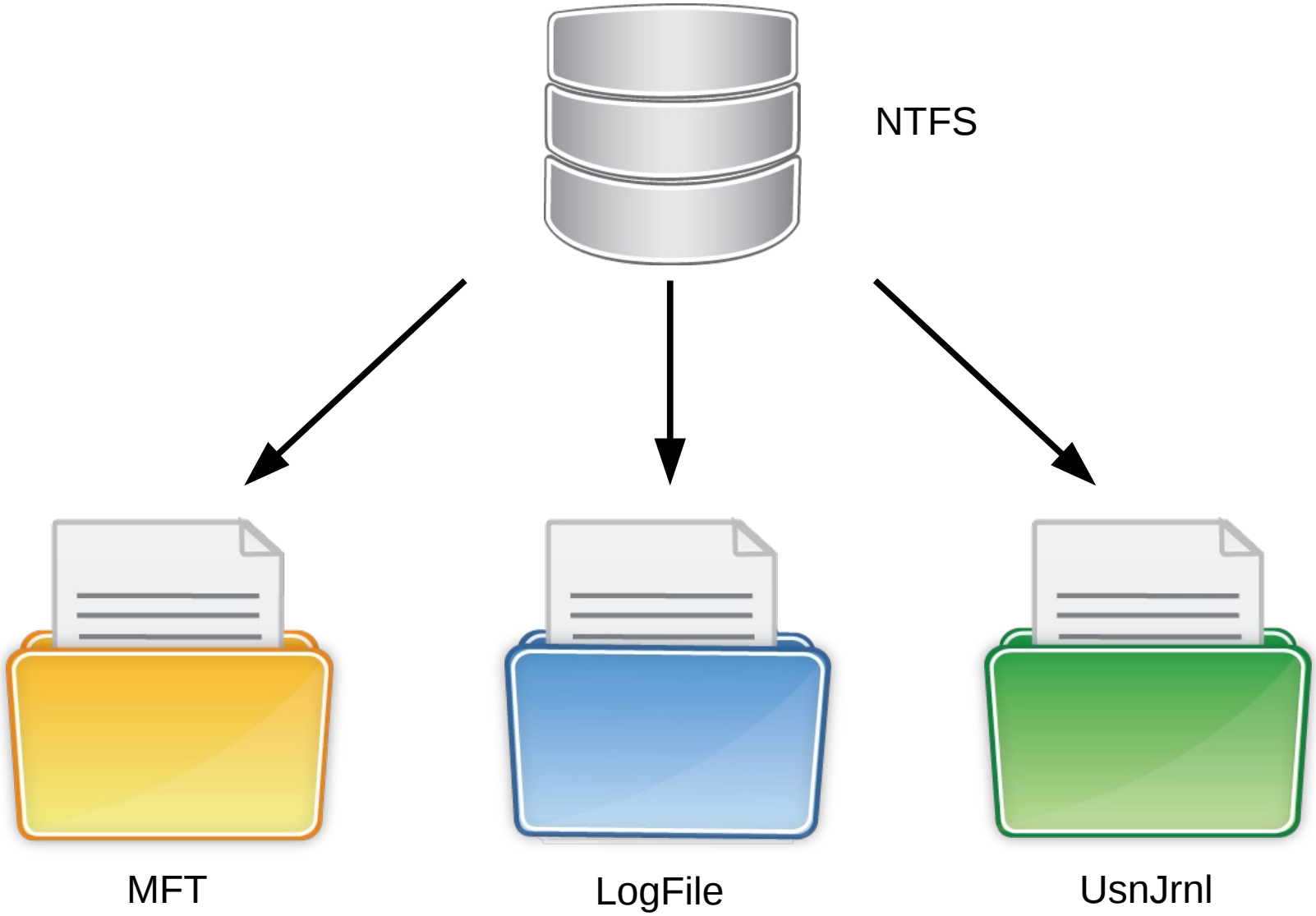
Why research the UsnJrnl?

Relatively young: since Windows Vista

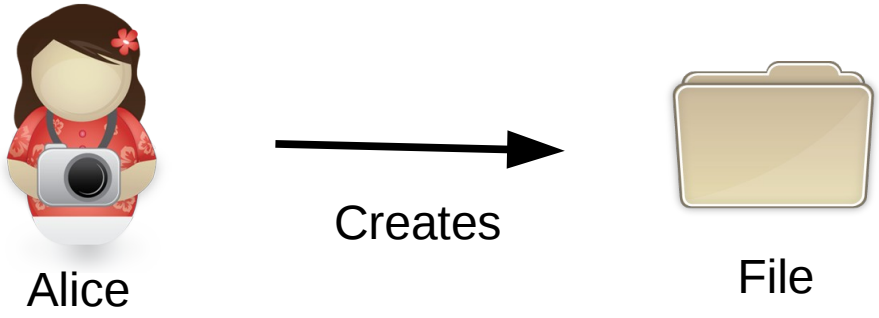
Often contains lots of historic data

Can be linked to other artefacts

The three files of interest

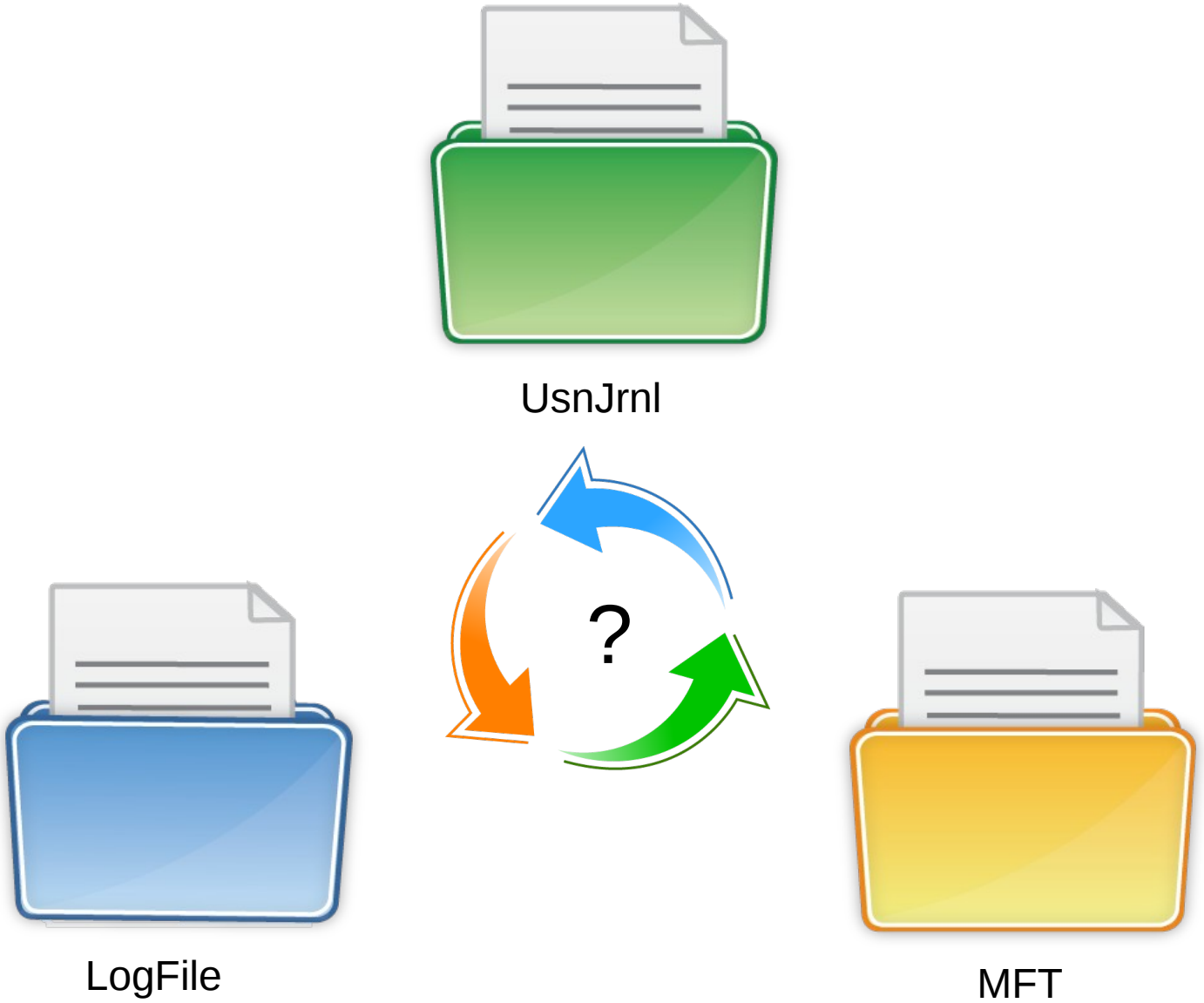


Context: effect of creating a file

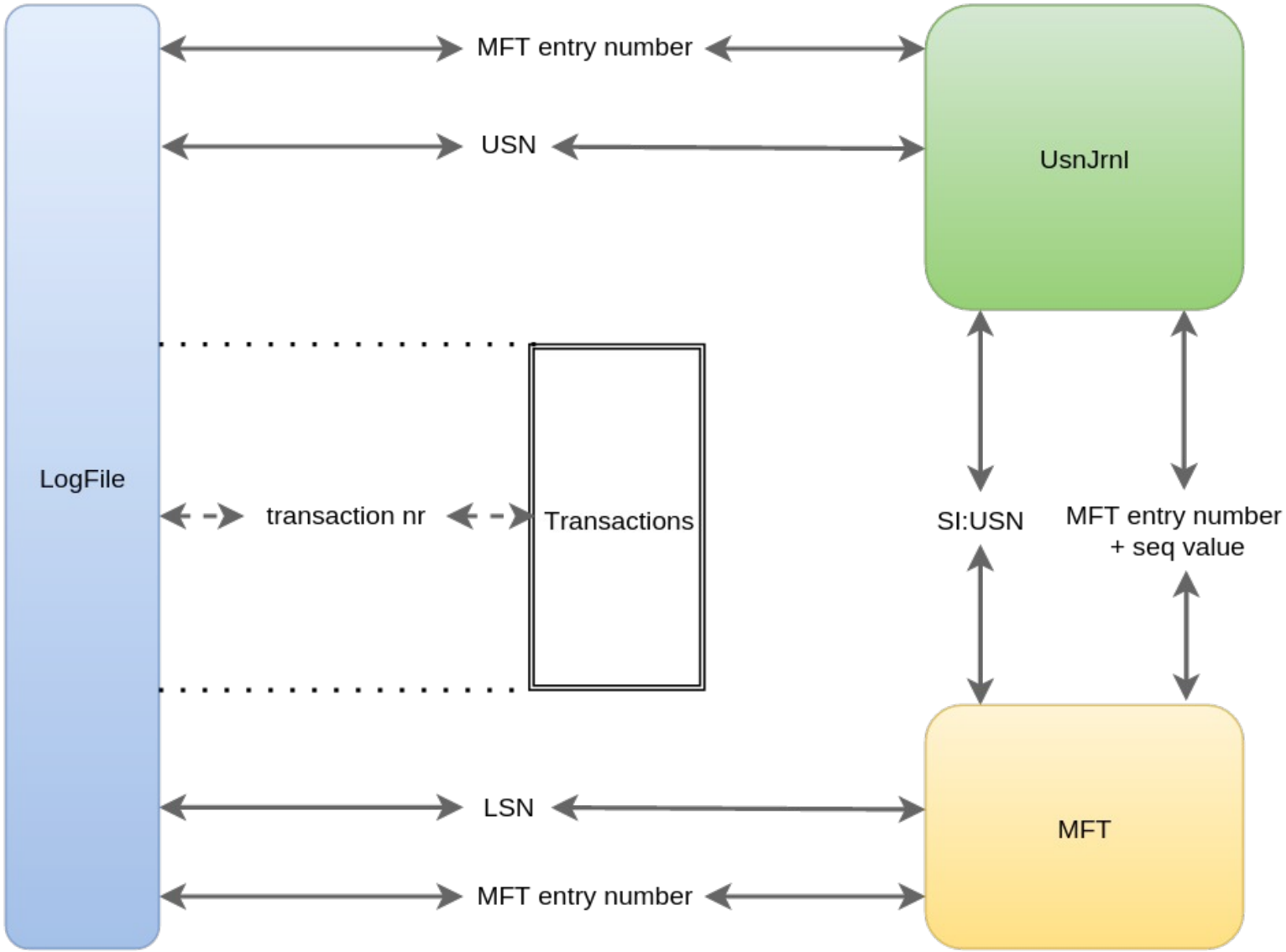


MFT LogFile UsnJrnl

How do they come together?



Model


LogFile
MFT entry number
USN
UsnJrnl
transaction nr
Transactions
SI:USN
MFT entry number
+ seq value
LSN
MFT entry number
MFT

MFT - overview

Master File Table

Keeps track of all files on NTFS

Only stores information on non-deleted files

MFT - structure

No header

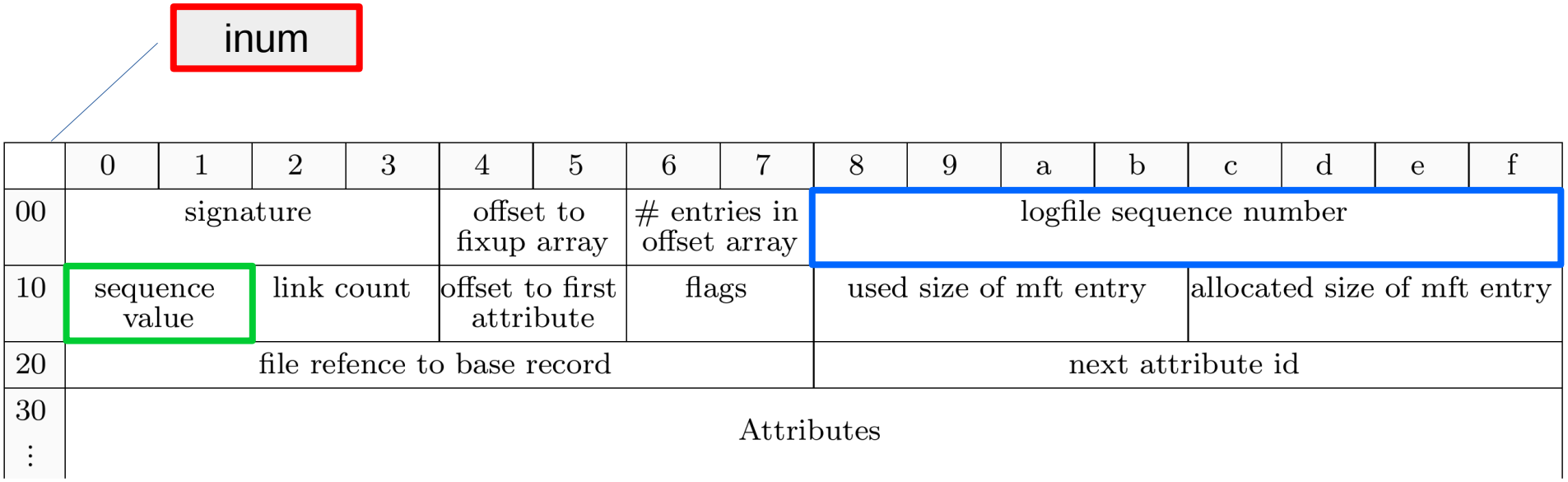
Consists of lots of *MFT entries*

MFT entries describe files/directories

A set of default entries:

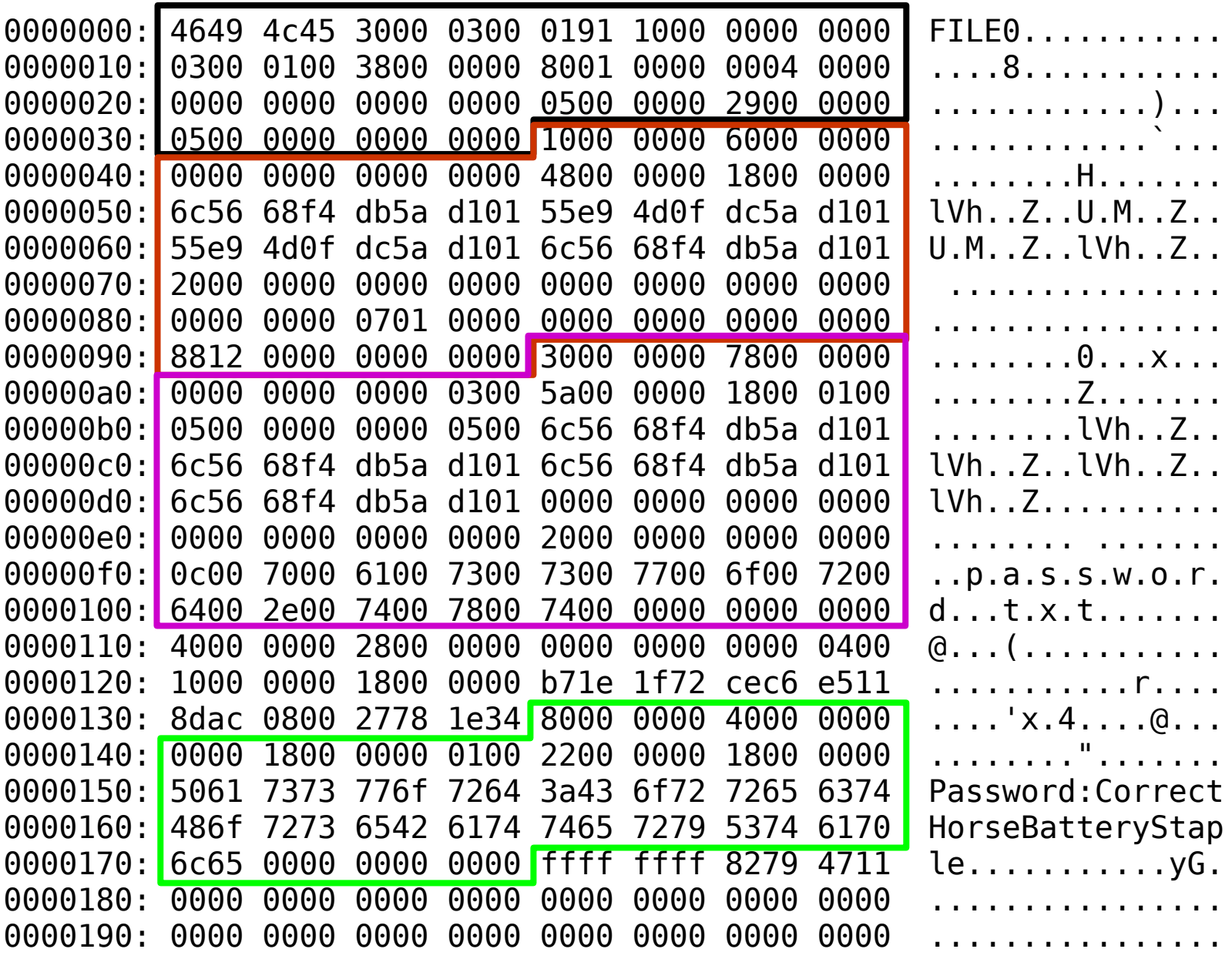
- 0: $MFT
- 1: $MFTMirr
- 2: $Logfile
- etc

MFT entry - structure



Attributes:

- Standard Information
- File Name


0000000: 4649 4c45 3000 0300 0191 1000 0000 0000 FILE0.....
0000010: 0300 0100 3800 0000 8001 0000 0004 0000 ....8.....
0000020: 0000 0000 0000 0000 0500 0000 2900 0000 ....)...
0000030: 0500 0000 0000 0000 1000 0000 6000 0000 ....`...
0000040: 0000 0000 0000 0000 4800 0000 1800 0000 ....H.....
0000050: 6c56 68f4 db5a d101 55e9 4d0f dc5a d101 lVh..Z..U.M..Z..
0000060: 55e9 4d0f dc5a d101 6c56 68f4 db5a d101 U.M..Z..lVh..Z..
0000070: 2000 0000 0000 0000 0000 0000 0000 0000 ....
0000080: 0000 0000 0701 0000 0000 0000 0000 0000 ....
0000090: 8812 0000 0000 0000 3000 0000 7800 0000 ....0...x...
00000a0: 0000 0000 0000 0300 5a00 0000 1800 0100 ....Z.....
00000b0: 0500 0000 0000 0500 6c56 68f4 db5a d101 ....lVh..Z..
00000c0: 6c56 68f4 db5a d101 6c56 68f4 db5a d101 lVh..Z..lVh..Z..
00000d0: 6c56 68f4 db5a d101 0000 0000 0000 0000 lVh..Z..........
00000e0: 0000 0000 0000 0000 2000 0000 0000 0000 ....
00000f0: 0c00 7000 6100 7300 7300 7700 6f00 7200 ..p.a.s.s.w.o.r.
0000100: 6400 2e00 7400 7800 7400 0000 0000 0000 d...t.x.t.....
0000110: 4000 0000 2800 0000 0000 0000 0000 0400 @...(...........
0000120: 1000 0000 1800 0000 b71e 1f72 cec6 e511 ....r....
0000130: 8dac 0800 2778 1e34 8000 0000 4000 0000 ....'x.4....@...
0000140: 0000 1800 0000 0100 2200 0000 1800 0000 ....".....
0000150: 5061 7373 776f 7264 3a43 6f72 7265 6374 Password:Correct
0000160: 486f 7273 6542 6174 7465 7279 5374 6170 HorseBatteryStap
0000170: 6c65 0000 0000 0000 ffff ffff 8279 4711 le..........yG.
0000180: 0000 0000 0000 0000 0000 0000 0000 0000 ....
0000190: 0000 0000 0000 0000 0000 0000 0000 0000 ....

LogFile - overview

Meant to guarantee file system recovery

- in case of a system failure

Contains lots of detailed historic data

Circular

LogFile - structure

The logfile consists of record pages

Every page has the following header structure

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	magic number				update seq array offset		update seq array		last lsn / offset to next page							
10	flags				page count		page position		next record offset		reserved 1					
20	last end lsn								fixup value		fixup aray					
30	(fixup array continuation)										(reserved 2)					

Pages contain so-called “LSN records”

LogFile LSN record structure

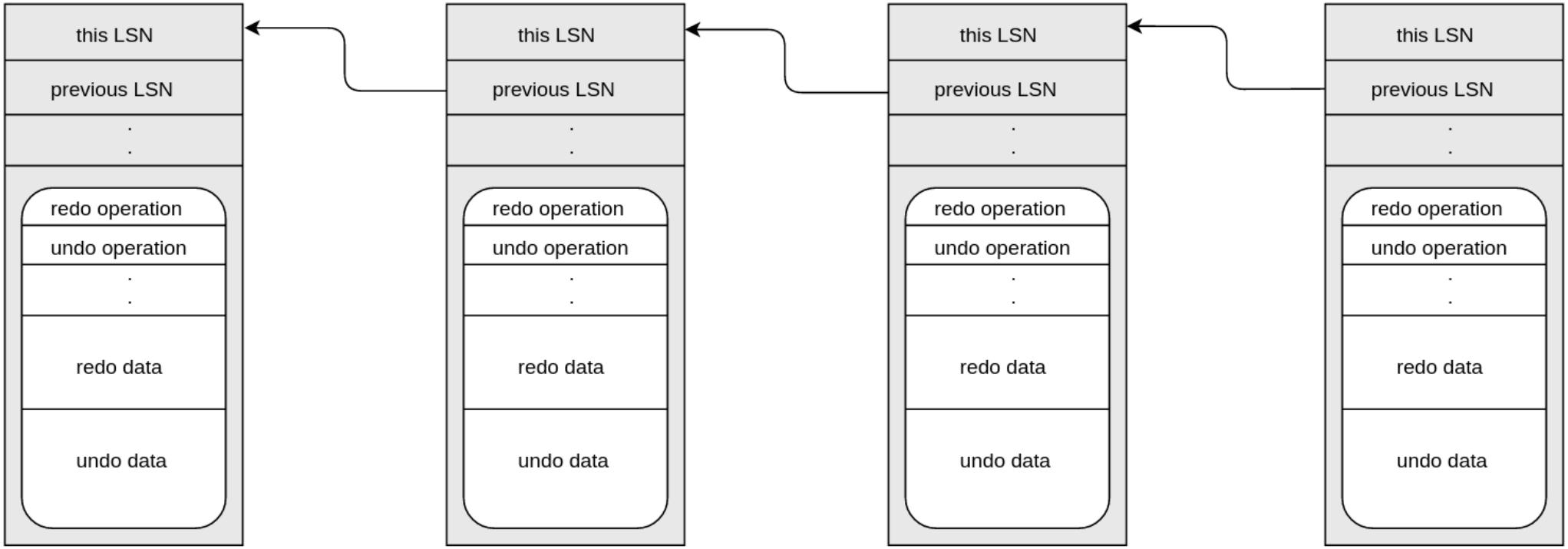
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	this lsn								previous lsn							
10	undo next lsn								data length				seq number		client index	
20	record type				transaction id				flag		reserved					
30	redo operation		undo operation		redo offset		redo length		undo offset		undo length		target attribute		lcn's to follow	
40	record offset		attribute offset		mft cluster index		alignment / reserved1		target vcn				alignment / reserved2			
50	target lcn				alignment / reserved3				redo and/or undo data							
60 ⋮	(continuation redo and/or undo data)															

Contains redo and undo data

Says something about a single change

LogFile LSN transactions

- LSN records are part of a transaction
- A transaction is an atomic unit



UsnJrnl - overview

Also called the “change journal”

Very concisely states what changed

Goes relatively far back in time

Timestamps

UsnJrnl - structure

No header

Consists of lots of *USN records*

Oldest clusters may be deallocated

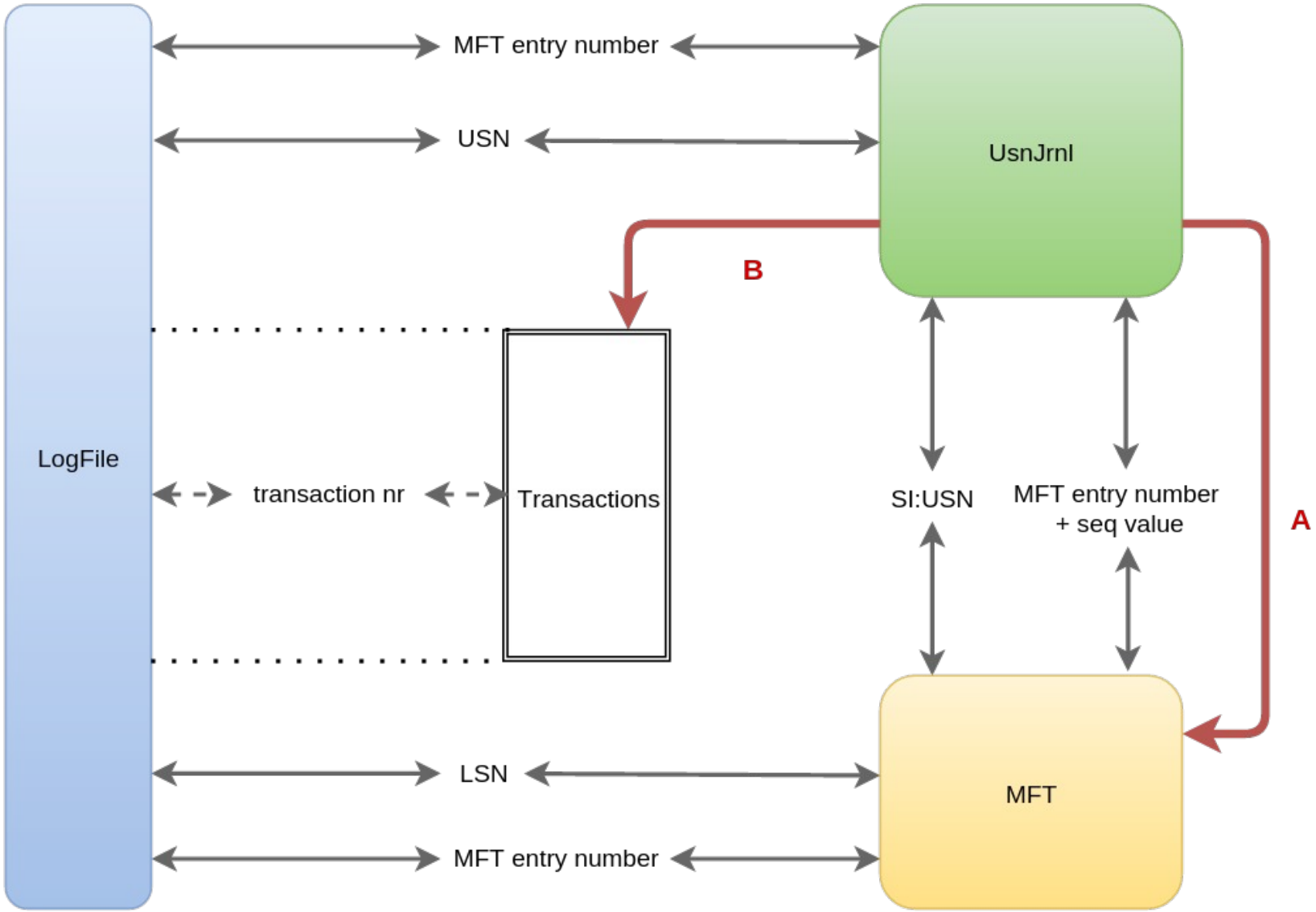
USN record - structure

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	record length				major version		minor version		file reference number							
10	parent file reference number								usn							
20	timestamp								reason				source info			
30	security info				file attributes				file length		file name offset		name			
40 ⋮	(name continuation)															

file reference number contains:

- MFT entry number
- MFT sequence value

Model


MFT entry number
USN
UsnJrnl
B
LogFile
transaction nr
Transactions
SI:USN
MFT entry number
+ seq value
A
LSN
MFT
MFT entry number

Conclusion: Forensic value

- UsnJrnl usually goes further back in time
- UsnJrnl is more reliably parsed
- Enables timelining LogFile transactions
- Easier to find transactions by filename
- Easier to find what files were deleted

Proof of concept – test case

file name	MFT inum	sequence value	state
cat.jpg	36	1	in use
target.jpg	40	1	deleted
secret.txt	41	1	deleted & overwritten
password.txt	41	2	deleted

Proof of concept – result 1/3

```
##############################################################################
# Current MFT information                                          #############
##############################################################################
MFT entry number: 41
Sequence value  : 3
Currently in use: False -> Historic data in MFT entry, easy to extract
File name       : password.txt

SUMMARY:
```

seq	USN record list
1	[3064, 3168, 3272, 3376, 3456, 3536, 3616, 3696, 3776, 3856]
2	[3936, 4096, 4200, 4304, 4392, 4480, 4568, 4656, 4744, 4832]

Proof of concept – result 2/3

```
================================================================================
 MFT entry 41; Sequence 2
================================================================================

    USN      : 3936
    File name: New Text Document.txt
    Timestamp: 2016-01-29 21:28:11.527128
    Reason   : FILE_CREATE
```

$LogFile transaction number: 104

LSN	Redo operation	Undo operation
1083171	Set Bits in Nonresident Bitmap	Clear Bits in Nonresident Bitmap
1083183	No-Operation	Deallocate File Record Segment
1083195	Add Index Entry Allocation	Delete Index Entry Allocation
1083222	Initialize File Record Segment	No-Operation
1083273	Set New Attribute Sizes	Set New Attribute Sizes
1083292	Update Nonresident Value	No-Operation
1083316	Set New Attribute Sizes	Set New Attribute Sizes
1083335	Forget Transaction	Compensation Log Record

Proof of concept – result 3/3

```
USN      : 4832
File name: password.txt
Timestamp: 2016-01-29 21:29:12.795932
Reason   : FILE_DELETE|CLOSE
```

$LogFile transaction number: 38

LSN	Redo operation	Undo operation
1085650	Delete Index Entry Allocation	Add Index Entry Allocation
1085675	Delete Index Entry Root	Add Index Entry Root
1085697	Deallocate File Record Segment	Initialize File Record Segment
1085711	Clear Bits in Nonresident Bitmap	Set Bits in Nonresident Bitmap
1085723	Set New Attribute Sizes	Set New Attribute Sizes
1085742	Update Nonresident Value	No-Operation
1085764	Set New Attribute Sizes	Set New Attribute Sizes
1085783	Forget Transaction	Compensation Log Record