Security Research Group,
Computer Science Department,
Friedrich-Alexander-University
Erlangen-Nuremberg, Germany

Introducing Anti-Forensics to SQLite Corpora and Tool Testing

Sven Schmitt

IMF 2018 - Hamburg, Germany

Contents

Motivation
Research question

Research question

- how can we transparently test and validate analysis tools for SQLite?
- how do SQLite analysis tools deal with databases containing anti-forensic manipulations?

Our answer

- serious evaluation requires publicly available data sets
- we extend a corpus of SQLite databases by anti-forensic aspects
- corpus will allow for comparing/reproducing test results
- scenarios will allow for improving analysis algorithms and tools

Motivation
On the importance of SQLite

SQLite DBMS very dominant

- one of the widestly spread database systems in the world [1]
- present on an overwhelming number of devices
- popular storage engine for personal data
- examples include: contacts, call lists, browser histories, messenger apps, etc.)

SQLite analysis is essential

- of high value in forensic investigations
- various tools exist for (forensic) analysis
- tools claim to rigorously analyze underlying database files
- appropriate evidence and reproducible test results are missing

Motivation
About the SQLite Forensic Corpus

What is the SQLite Forensic Corpus?

- set of SQLite database files freely available for tool testing [2]
- released by Nemetz et al. at DFRWS EU 2018
- comprising 77 databases grouped into 5 categories
- offering specific test cases, corner cases and pitfalls
- all databases conform to the SQLite file format specification

Evaluation by Nemetz et al.

- selection of SQLite analysis tools tested against the corpus
- results showed strenghts and weaknesses of tools
- none of the tools did handle all of the cases properly
- corpus proved to be helpful for improving algorithms/tools

Motivation
Contributions

Elaborating SQLite file format

- elaborating characteristics of the SQLite file format [3]
- manipulating databases to potentially mitigate forensic analysis

Extending the SQLite Forensic Corpus

- anti-forensic scenarios as extension to the corpus
- extension comprises 64 databases in 4 categories
- databases deliberately no longer conform to the file format
- we donate the extension into the public domain
  https://faui1-files.cs.fau.de/public/sqlite-forensic-corpus/

Evaluating robustness of SQLite analysis tools

- revealing room for improvement for all of the tools
- deriving claims regarding forensic analysis tools in general

Anti-Forensics
Definition

Anti-forensics definition by Harris [4]

- "any attempts to compromise the *availability* or *usefulness* of evidence to the forensics process"

Compromising the **availability**

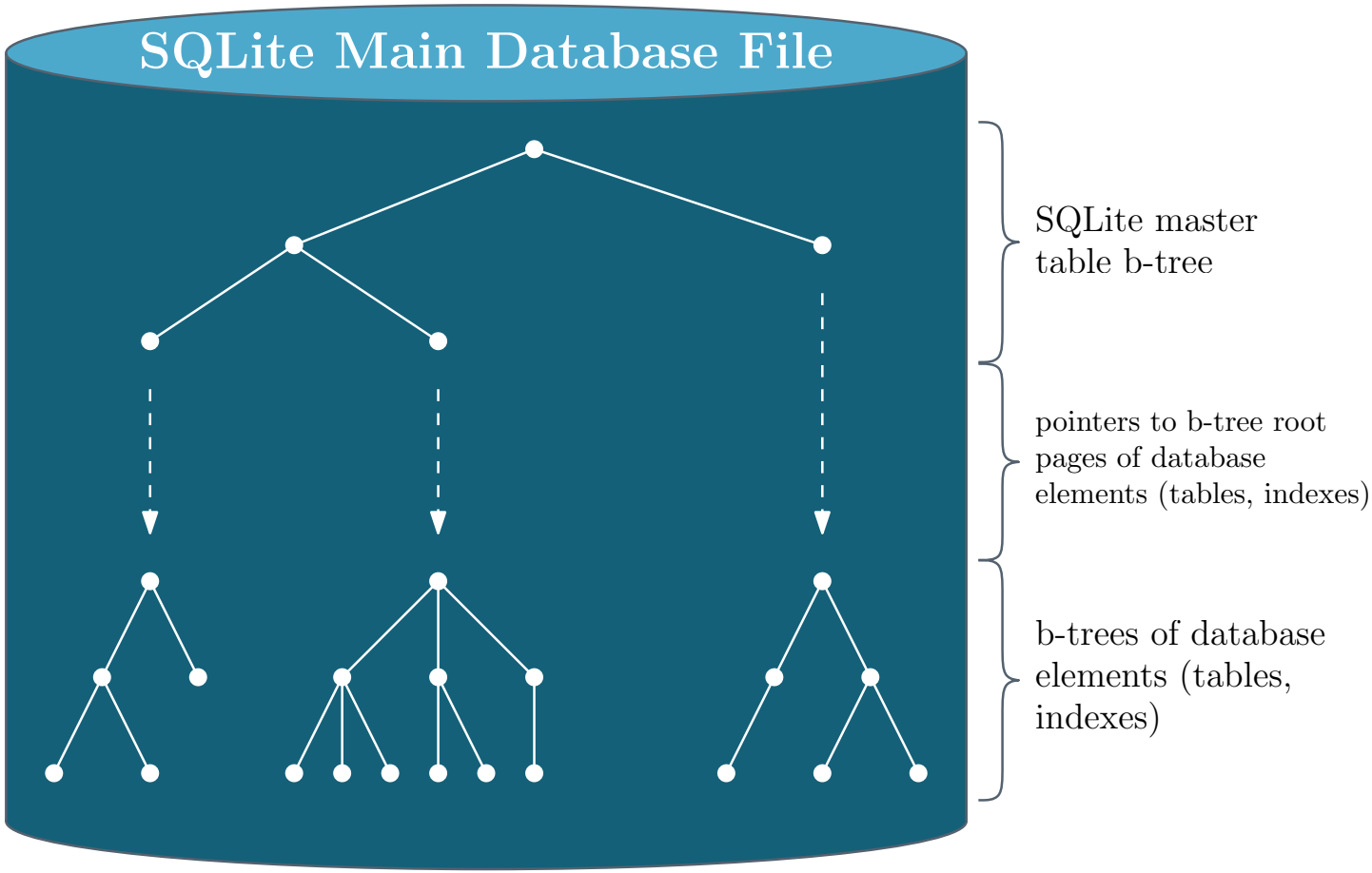
- preventing evidence from existing
- hiding existing evidence or
- manipulating evidence to be out of reach of the investigator

Compromising the **usefulness**

- obliterating the evidence itself or
- destroying its integrity

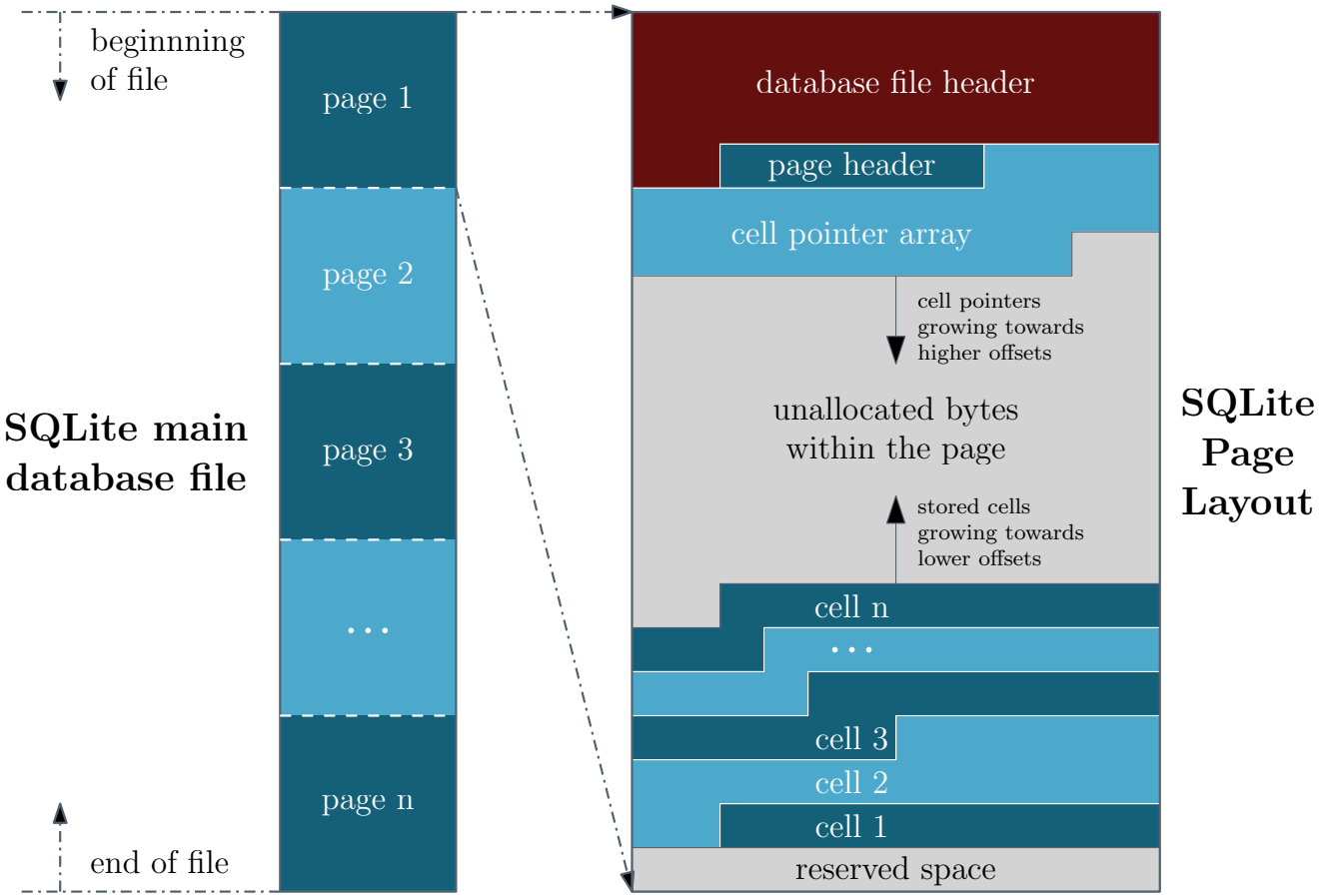
SQLite File Format Basics

B-tree Database Elements



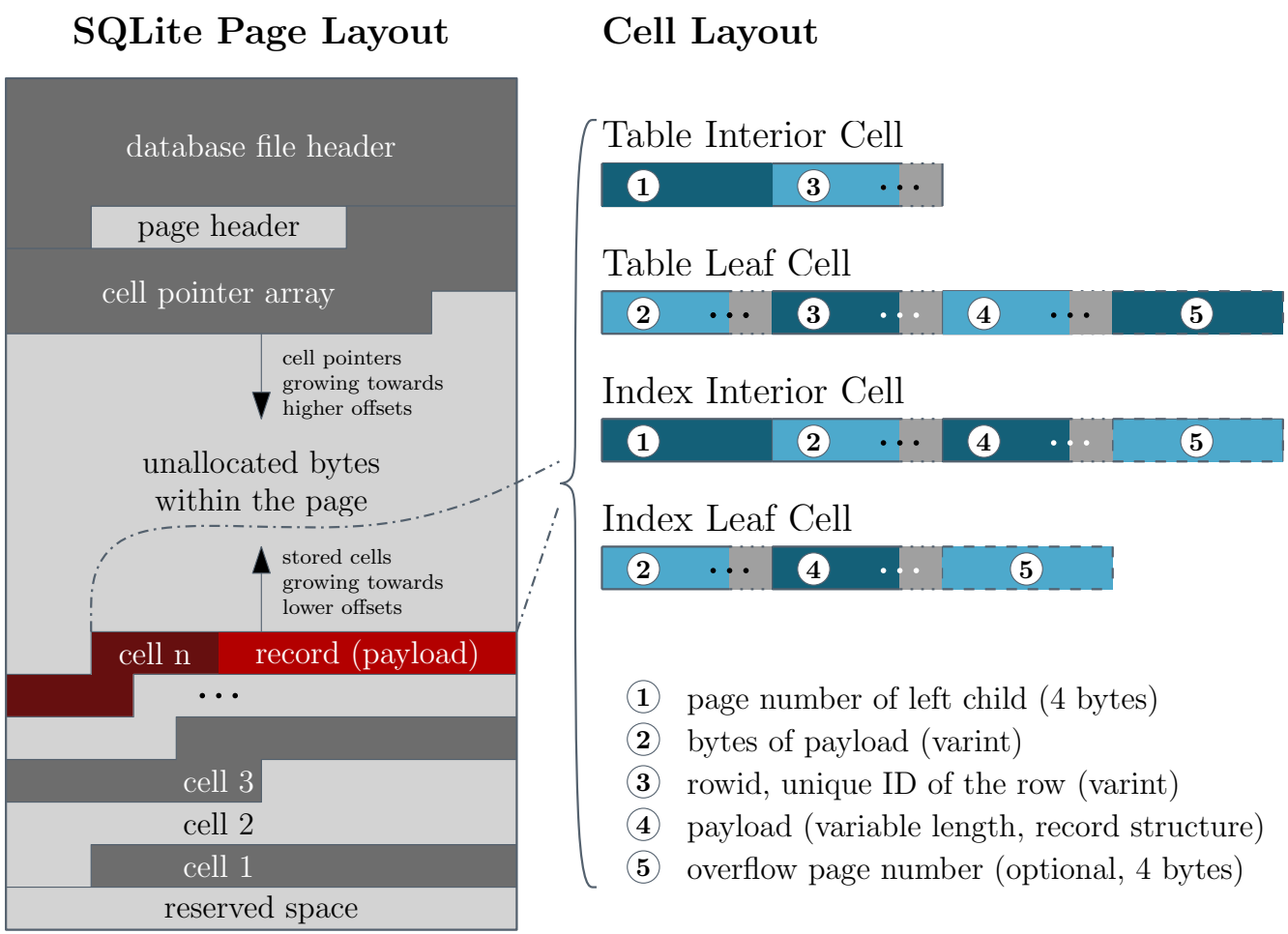
SQLite File Format Basics

B-tree Serialization and Page Layout



SQLite File Format Basics

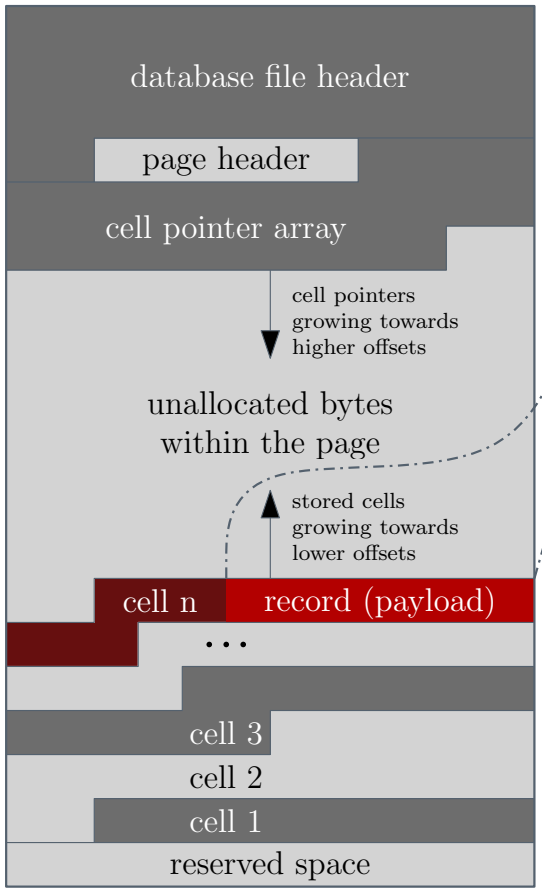
Cell Types and Layout



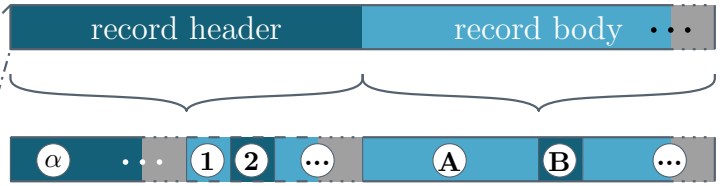
SQLite File Format Basics

Structure of a Record (Entry/Row)

SQLite Page Layout



Record Structure



- (α) record header length (varint)
- (1) type and length of content A (varint)
- (2) type and length of content B (varint)
- (…) type and length of further contents (varint)
- (A) content of first column in the row
- (B) content of second column in the row
- (…) contents of further columns in the row

SQLite File Format Basics

Serial Type Encoding [3]

Serial Type	Content Size	Data Type or Meaning
0	0	NULL.
1	1	big-endian 8-bit twos-complement int
2	2	big-endian 16-bit twos-complement int
3	3	big-endian 24-bit twos-complement int
4	4	big-endian 32-bit twos-complement int
5	6	big-endian 48-bit twos-complement int
6	8	big-endian 64-bit twos-complement int
7	8	big-endian IEEE 754-2008 64-bit float
8	0	integer of value 0 (version 3.3.0, 2006)
9	0	integer of value 1 (version 3.3.0, 2006)
10,11	-	*Reserved for internal use*
$N \geq 12$ (even)	$(N - 12)/2$	Binary large object (BLOB)
$N \geq 13$ (odd)	$(N - 13)/2$	String (text-encoding applied)

Anti-Forensic Corpus Extension
Categories & Folders

Category	Folder	Manipulated structures
Page & cell & pointers	11	Root page pointers
	12	Left child page pointers
	13	Overflow page chains
	14	Cell pointer array values
Cell metadata & contents	15	Cell metadata & serial types
	16	Zero-terminated contents
Unallocated areas	17	Freeblock structures
	18	Freelist trunks
Database size	19	Size of database files

Anti-Forensic Corpus Extension

Example scenarios in the corpus

Database	Description of manipulations
11-05	Pointer to rootpage points to page in other table
12-01	All left child pointers point to own page
13-03	Overflow page chains of records contain loops
14-05	Cell pointers point to an offset within a following page
15-02	Payload size is decreased (early end of cell)
16-02	First character of text field set to zero (string termination)
17-03	Inserted a loop into the chain of freeblocks
18-05	Pointer to next freelist trunk page points outside of DB
19-04	Some pages are removed from end of DB

Anti-Forensic Corpus Extension
Accompanying Metadata

Metadata accompanying the corpus

- besides the manipulated anti-forensic DBs
- four additional files for each scenarios

1. SQL statements used to create the initial database files including statements: CREATE, INSERT, DELETE, PRAGMA
2. XML files describing logical contents of DBs, allows for automating tests and result comparisons
3. text file describing manipulations performed on DBs
4. binary file storing replaced bytes (allows for reverting)

Anti-Forensic Corpus Extension

Accompanying Metadata


```
PRAGMA page_size=4096;
PRAGMA page_size;
PRAGMA encoding="UTF-8";
PRAGMA encoding;
PRAGMA secure_delete=0;

CREATE TABLE 'users' (
    'id' INT UNSIGNED NOT NULL,
    'name' TEXT NOT NULL,
    'surname' TEXT NULL,
    'plz' INTEGER NULL
);

INSERT INTO 'users'
    (id, name, surname, plz)
    VALUES
    (20001, 'Friedrich', 'Schwarz', '55569'),
    (20002, 'Bianca', 'Günther', '26835'),
    (20003, 'Julius', 'Dietrich', '33604'),
    (20004, 'Martina', 'Maier', '94496'),
    (20005, 'Thomas', 'Schmidt', '3246'),
    (20006, 'Josephine', 'Voigt', '25821'),
    (20007, 'Julian', 'Dietrich', '21376'),
    (20008, 'Finja', 'Friedrich', '56598'),
    (20009, 'Georg', 'Hahn', '29640'),
    (20010, 'Luca', 'Herrmann', '71554');
```

Anti-Forensic Corpus Extension

Accompanying Metadata


```
<?xml version='1.0' encoding='UTF-8'?>
<database>
    <meta>
        <tables>2</tables>
        <indices>0</indices>
        <views>0</views>
        <triggers>0</triggers>
        <entries>20</entries>
    </meta>
    <description>
        <dc:title>SQLite Forensic Corpus, Version
              2.0</dc:title>
        <dc:subject>Category: Page and cell
             pointers</dc:subject>
        <dc:description>Manipulated Root Page
             Pointers</dc:description>
        <dc:identifier>11-01.db</dc:identifier>
        [...]
    </description>
    <element>
        <meta>
            <type>table</type>
            <name>users</name>
            <reference>users</reference>
            <deleted>False</deleted>
            <columns>4</columns>
            <rowsTotal>10</rowsTotal>
            <rowsAlive>10</rowsAlive>
            <rowsDeleted>0</rowsDeleted>
        </meta>
        <sql>
            <columnDefinition>
                <meta>
                    <name>id</name>
                    <attribute>INT UNSIGNED NOT
                         NULL</attribute>
                </meta>
                <columnTypeName>INTEGER</
                     columnTypeName>
            </columnDefinition>
            <columnDefinition>
            [...]
            </columnDefinition>
            [...]
            <tableConstraint>
                <statement>PRIMARY KEY(id)</
                     statement>
                <flags>
                    <isPrimaryKey>True</
                         isPrimaryKey>
                </flags>
            </tableConstraint>
        </sql>
        <entries>
            <row>
                <column>
                    <name>id</name>
                    <content>20001</content>
                </column>
                [...]
            </row>
            [...]
        </entries>
    </element>
    [...]
</database>
```

Anti-Forensic Corpus Extension
Accompanying Metadata


```
# Welcome to sqlite_antifor. #
## Database filename is: 16/16-01.db ##
# Filename of manipulated database: 16/16-01_antifor.db #
## Size of database page: 4096 ##
## Database has 3 pages. ##
## Created recovery file: 16/16-01.db_recovery ##
# Selected option --decrease-payload-size. #
## Page #2 has 20 cell(s). ##
## Changed bytes of payload from 27 to 4 in cell 6 of page 2. ##
## Changed bytes of payload from 24 to 20 in cell 14 of page 2. ##
## Changed bytes of payload from 24 to 13 in cell 9 of page 2. ##
## Changed bytes of payload from 22 to 2 in cell 11 of page 2. ##
## Changed bytes of payload from 18 to 11 in cell 1 of page 2. ##
# Changed total number of bytes of payload in 5 cells. #
# Exited sqlite_antifor. #
```

Evaluation
Testing tools against the corpus

About analysis tools

- analysis of SQLite supported by a bunch of tools
- commercial or open source tools exist
- tools may extract logical data, deleted contents or both

Restriction to a few tools

- we selected 5 tools from different categories as mentioned above

Chosen analysis tools

- SQLite3 (open source, written in C)
- Undark (open source, written in C)
- SQLiteDoctor (proprietary software)
- Stellar Phoenix Repair for SQLite (prorietary software)
- SQLite Database Recovery (proprietary software)

Chosen analysis tools

- SQLite3 (open source, written in C)
  “command shell officially released as part of SQLite DBMS”
- Undark (open source, written in C)
  “SQLite deleted and corrupted data recovery”
- SQLiteDoctor (proprietary software)
  “repair and restore currupted DBs”
- Stellar Phoenix Repair for SQLite (prorietary software)
  “recover corrupted DBs, easily recovers all deleted records”
- SQLite Database Recovery (proprietary software)
  “repair and export corrupt SQLite files”, no deleted records

Evaluation
Testing tools against the corpus

How did we test?

- basically manually, running tools, inspecting results
- when analyzing 64 database files, “a lot can happen”
- categories abstracting from detailed results

Result categories

✓ → all elements correctly processed
✓* → some elements correctly, some wrongly processed (errors)
✗ → no element correctly processed
● → execution did not return (e.g. endless loop)
✚ → extraction failed due to crash (e.g. segfault)
– → all elements correctly extracted
upper rows → overall result of the execution
lower rows → results partially extracted

Anti-Forensic Corpus Extension

Accompanying Metadata

File: *.db	SQLite3	Undark	SQLite-Doctor	Phoenix Repair	DB Recovery
11-01	✓	✓*	✓*	✓*	✓
11-02	✓	✓*	✓*	✗	✓
11-03	15/25	✓*	5/10*	✗	15/25
11-04	✗	✓*	✗	✗	✗
11-05	✗	✓*	✗	✗	✗
12-01	✗	✓*	✓*	✚	●
12-02	158/700	✓*	✓*	✚	●
12-03	✗	✓*	5/10*	✚	●
12-04	✗	✓*	5/10*	✚	●
12-05	✗	✓*	✓*	85/700*	85/700
12-06	311/700	✓*	✓*	396/700*	396/700
13-01	7/10 3	✗ 3	2/5* 3	✗ 10	✚
13-02	7/10 3	2/10* 1	2/5* 3	✗ 10	✚
13-03	✓ –	2/10* 1	✓* –	✗ 10	✚
13-04	9/10 1	✚	4/5* 1	✗ 10	✚
13-05	✗ ✗	✚	3/5 ✗	✗ 10	✗ 3
13-06	✗ ✗	✚	✓* –	✗ 10	✗ 5
13-07	8/10 2	2/10* ✗	3/5* 2	✗ 10	✗ 9
13-08	1/10 ✗	2/10* ✗	✓* –	✗ 10	✗ 9

File: *.db	SQLite3	Undark	SQLite-Doctor	Phoenix Repair	DB Recovery
15-09	✗ ✗	19/20* ✗	✓* –	19/20* 1	19/20 1
15-10	✗ ✗	19/20* ✗	✓* –	19/20* 1	19/20 ✗
15-11	✗ ✗	10/20* 1	✓* –	10/20* 10	10/20 10
15-12	✗ ✗	✗ ✗	✗ 1	✚ 11	✗ 20
15-13	✓ –	✗ ✗	✓* –	✗ 5	✗ ✗
16-01	✗	✓*	✗	✗	✓
16-02	✗	✓*	✗	✗	✓
17-01	10/30	●	5/10*	✓*	✓
17-02	✗	●	✗	✓*	✓
17-03	10/30	●	5/10*	✓*	✓
17-04	✗	●	✗	✓*	✓
17-05	10/30	●	5/10*	✓*	✓
17-06	✓	1/30*	✓*	✓*	✓
17-07	✓	1/30*	✓*	✓*	✓
17-08	✓	1/30*	✓*	✓*	✓
17-09	✓	1/30*	✓*	✓*	✓
17-10	10/30	1/30*	5/10*	✓*	✓
17-11	✗	✚	✗	✓*	✓
17-12	10/30	1/30*	5/10*	✓*	✓
17-13	✗	1/10*	✗	✓*	✓

Anti-Forensic Corpus Extension

Accompanying Metadata

File: *.db	SQLite3	Undark	SQLite-Doctor	Phoenix Repair	DB Recovery
14-01	✗	✓*	✗	✗	✗
14-02	18/20	✓*	✓*	18/20*	18/20
14-03	✗	✓*	✗	✗	✗
14-04	5/20	✓*	✓*	18/20*	18/20
14-05	✗	✓*	✗	✗	✗
14-06	16/20	✓*	✓*	✚	18/20
14-07	✗	✓*	✗	✗	✗
14-08	18/20	✓*	✓*	18/20*	18/20
15-01	1/20 ✗	15/20* ✗	✓* –	✓* –	✓ –
15-02	✗ ✗	✗ ✗	✗ ✗	✓* –	✓ –
15-03	1/20 ✗	15/20* ✗	✓* –	✓* –	✓ –
15-04	✗ ✗	✗ ✗	✗ ✗	✓* –	✓ –
15-05	✗ ✗	15/20* ✗	✓* –	15/20* 5	15/20 5
15-06	✗ ✗	✗ ✗	✗ ✗	✗* 20	✗ 20
15-07	✗ ✗	15/20* ✗	✓* –	15/20* 5	15/20 5
15-08	✗ ✗	✗ ✗	✗ ✗	✗* 20	✗ 20

File: *.db	SQLite3	Undark	SQLite-Doctor	Phoenix Repair	DB Recovery
18-01	✓	✓*	✓*	✗	✓
18-02	✓	8/10*	✓*	✗	✓
18-03	✓	8/10*	✓*	✗	✓
18-04	✓	8/10*	✓*	✗	✓
18-05	✓	8/10*	✓*	✗	✓
19-01	✓	✓*	✓*	✓*	✓
19-02	✓	✓*	5/10*	✓*	✓
19-03	✗	✓*	✗	✓*	✓
19-04	✗	✓*	✗	✓*	✓

✓ → all elements correctly processed
* → some failures occurred
✗ → no element correctly processed
● → execution did not return (e.g. endless loop)
✚ → extraction failed due to crash (e.g. segfault)
– → all elements correctly extracted
upper rows → overall result of the execution
lower rows → results partially extracted

Conclusion
Summary

Corpus and test results

- extended the SQLite Forensic Corpus
- added 64 databases with anti-forensic manipulations
- tested (forensic) tools are sensitive to such tweaks

Desirable requirements for forensic tools

- tools shall be hardened against anti-forensic input
  - unexpected input
  - malformed data
- if a tools fails, it shall exit gracefully
- whenever analyses omit (parts of) evidence, tools shall clearly state this

Conclusion
Outlook

Participate!

- you are invited to read the paper
- make use of the corpus
  test against it when releasing new tools or algorithms
- contribute and help extend the corpus
- help keeping the corpus up to date

Bibliography
(please refer to paper for full references)

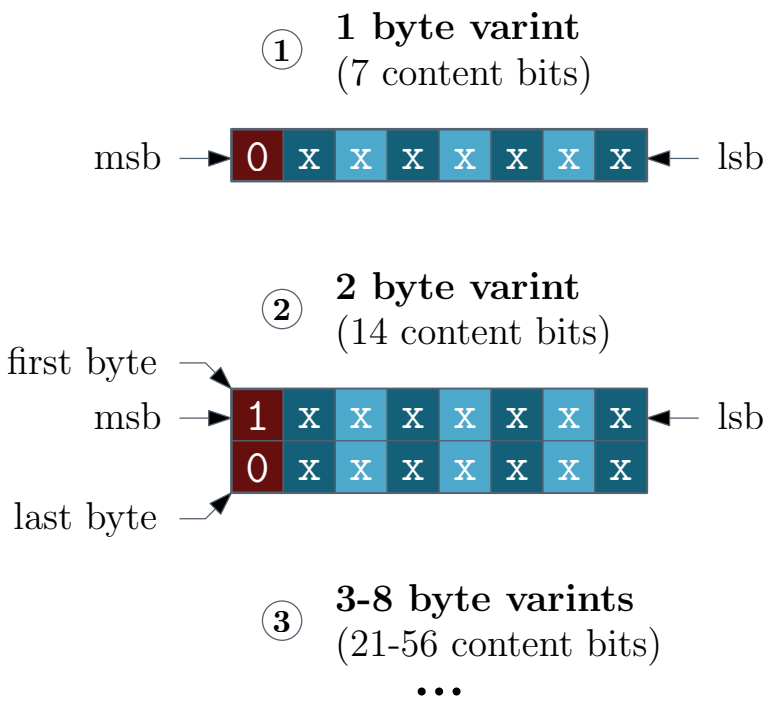
1. SQLite Online Documentation, "Most Widely Deployed SQL Database Engine", 2018.
2. Nemetz et al., "A Standardized Corpus for SQLite Database Forensics", Digital Investigation, Volume 24, S121 - S130, 2018, https://doi.org/10.1016/j.diin.2018.01.015.
3. SQLite Online Documentation, "File Format For SQLite Databases," 2018. [Online]. Available: http://www.sqlite.org/fileformat2.html
4. R. Harris, "Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem", Digital Investigation, vol. 3, pp. 44 – 49, 2006

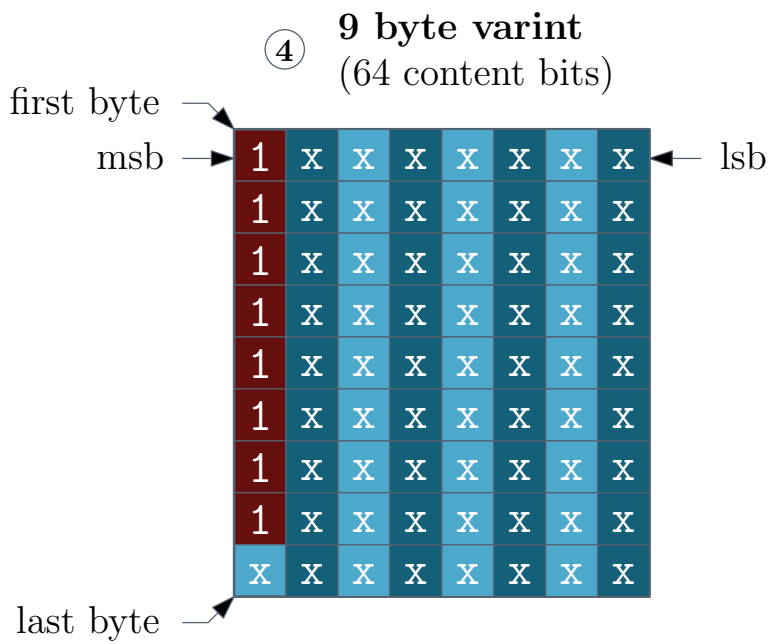
Thank you!

Questions?

Corpus: https://faui1-files.cs.fau.de/public/sqlite-forensic-corpus/

SQLite File Format Basics
Variable Integer Encoding





content bits: $x = \{0, 1\}$
high order bit indicator: last byte in the varint
high order bit indicator: another byte in the varint

Taxonomy on corpus sensitivity, Garfinkel et al. [2]

- **test data**: specifically created, no sensitive data, distribute freely
- **sampled data**: extracted out of larger source, difficult to redistribute (privacy? copyright?)
- **realistic data**: encountered in real life, distribution may be possible (copyright?)
- **real and restricted data**: real data, no public redistribution (privacy! copyright!)
- **real but unrestricted**: real data either publicly released (court cases) or publicly available (Flickr, Facebook)

Sensitivity of the SQLite Forensic Corpus

- **test data**: specifically created, no sensitive data, distribute freely
  privacy: ✓, copyright: ✓
- **sampled data**: extracted out of larger source, difficult to redistribute (privacy? copyright?)
  specifics and pitfalls: ✓, actual data: ✗
- **realistic data**: encountered in real life, distribution may be possible (copyright?)
  ✗
- **real and restricted data**: real data, no public redistribution (privacy! copyright!)
  ✗
- **real but unrestricted**: real data either publicly released (court cases) or publicly available (Flickr, Facebook)
  ✗

Forensic Corpora
Categorization

7 criteria by Garfinkel [3]

- **representative**: data encountered in (forensic) investigations
- **complex**: many sources, range of data, many human languages
- **heterogeneous**: range of computer systems and usage patterns
- **annotated**: ready to validate new tools and algorithms
- **available**: unclassified environment
- **distributed**: open file formats, tools for manipulation
- **maintained**: routinely augmented with new information

Forensic Corpora
Categorization

7 criteria of the SQLite Forensic Corpus

- **representative**: data encountered in (forensic) investigations
  DBs differ in settings, number of elements, contents
- **complex**: many sources, range of data, many human languages
  77 different, specifically crafted, potential pitfalls
- **heterogeneous**: range of computer systems and usage patterns
  file format independent of OS, HW architecture
- **annotated**: ready to validate new tools and algorithms
  metadata: SQL CREATE, descriptive XML, ground truth
- **available**: unclassified environment
  we donate the corpus into the public domain
- **distributed**: open file formats, tools for manipulation
  DBs are SQLite3, metadata is SQL/XML, all well-known
- **maintained**: routinely augmented with new information
  further extensions will be needed/helpful