Forensic Analysis of Database Tampering

Kyriacos Pavlou and Richard T. Snodgrass

Computer Science Department
The University of Arizona
The problem: How to systematically perform forensic analysis on a compromised database.

- Recent federal laws (HIPAA, Sarbanes-Oxley Act etc.) and incidents of corporate collusion mandate *audit log security*.

- Snodgrass et al. [VLDB04] showed how to detect database tampering. Approach: *Hash* using a cryptographically strong hash function, *notarize* data manipulated by transactions and periodically *validate*.

- Forensic analysis to ascertain:
  - *When* the intrusion transpired
  - *What* data was altered
  - *Who* the intruder is
  - *Why* has this transpired
Outline

• Tamper Detection

• Forensic Analysis
  – The corruption diagram
  – Types of corruption events

• Forensic Algorithms
  – Three algorithms
  – Forensic strength

• Future Work
Tamper Detection

Two phases:

- **Normal Processing**
- **Validation**

The validation result is a single bit.
The Corruption Diagram

When

Actual time

NE: Notarization Event

VE: Validation Event

CE: Corruption Event

When

Where

I

notarization interval

validation interval

commit time

link

commit time

link

commit time

CE

validation interval

NE

VE

= TRUE

NE

I

VE

= TRUE

NE

NE

NE

NE

NE

NE

NE
• If a corruption is detected, the *forensic analyzer* springs into action.

• The analyzer tries to ascertain a *corruption region*: the bounds on the uncertainty of the “where” and “when” of the corruption.
Monochromatic Algorithm

When

Forensic analysis begins

Where

time of corruption \((t_c)\)

Corruption Region: captures the uncertainty as to the position of CE

\[\begin{align*}
VE_1 &= TRUE \\
VE_2 &= FALSE \\
NE_0 := & \text{place of corruption (commit time)} \\
NE_1, NE_2, NE_3, NE_4, NE_5, NE_6
\end{align*}\]
Monochromatic Algorithm

- Central insight: data can be rehashed by validator and checked.

- Corruption region bounds: $I_V \times I_N$
  - Area is solely dependent on the two intervals.

- Cannot handle CE$s$ involving timestamp corruption.
The RGB Forensic Algorithm

When

$\mathbf{I_V} = 4$ days

$\mathbf{I_N} = 2$ days

Postdating CE

$t_p$: postdating time

Forensic analysis begins

Postdating CE:

I\textunderscore V = postdating

I\textunderscore N = postdating

Notarization of Red & Green

Notarization of Blue & Green

Notarization of Red

$\mathbf{VE_1 = TRUE}$

$\mathbf{VE_2 = TRUE}$

$\mathbf{VE_3 = TRUE}$

$\mathbf{VE_4 = FALSE}$

Where

$\mathbf{NE_0}$

$\mathbf{NE_1}$

$\mathbf{NE_2}$

$\mathbf{NE_3}$

$\mathbf{NE_4}$

$\mathbf{NE_5}$

$\mathbf{NE_6}$

$\mathbf{NE_7}$

$\mathbf{NE_8}$
The RGB Forensic Algorithm

• Introduction of RGB partial hash chains:
  – Allows the bounding of both $t_l$ and $t_p$
  – Incurs extra NS cost

• Each of two corruption regions bounds: $I_V \times I_N$

• We would like to reduce the area of the corruption regions.
The Polychromatic Algorithm

$I_V = 4$ days
$I_N = 2$ days
Desired = 1 day

Backdating CE

Uncertainty can be arbitrarily shrunk via a logarithmic number of red and blue hash chains.

$t_b$: backdating time

$V_{E_4} = \text{FALSE}$

$V_{E_3} = \text{TRUE}$

$V_{E_2} = \text{TRUE}$

$V_{E_1} = \text{TRUE}$

Notarization of 2 Blues & 1 Green

Notarization of 2 Reds

Notarization of 2 Reds

Forensic analysis begins
Forensic Strength

Components:
- Work of forensic analysis
- Region-area of $CE$
- Width of postdating / backdating uncertainty

Inverse Forensic Strength:

$$IFS(D, I_N, V) = (\text{NumNotarizes}(D, I_N, V) + \text{ForensicAnalysis}(D, I_N, V))$$

$$\cdot \text{RegionArea}(I_N, V) \cdot \text{UncertaintyWidth}(D, I_N)$$

where

$V = I_V / I_N$ is the validation factor and
$D$ is the number of days before first validation failure.

- **Monochromatic**: $O( V \cdot D^2 \cdot I_N )$
- **RGB**: $O( V \cdot D \cdot I_N^2 )$  
  We assume that $D >> I_N$.
- **Polychromatic**: $O( ( V + \log I_N ) \cdot D )$
Future Work

• Develop a stronger lower bound for this problem.

• Accommodate multi-locus and complex CEs.

• Differentiate postdating and backdating CEs.

• Implement forensic analysis in validator.

• Consider interaction between transaction-time storage manager and underlying WORM storage.
Summary

• We have presented a means of performing forensic analysis.

• We have introduced a graphical representation to visualize CEs, termed the corruption diagram.

• We have designed three forensic algorithms.
  – Monochromatic
  – RGB
  – Polychromatic