Motivation
Motivation

**Corporate collusion** has given rise to **regulations** for trustworthy long-term **data management**.

- Code of Federal Regulations of FDA: Clinical trials
Motivation

Corporate collusion has given rise to regulations for trustworthy long-term data management.

- Code of Federal Regulations of FDA: Clinical trials
- Sarbanes-Oxley Act: Financial transactions
- HIPAA – Health Insurance Portability and Accountability Act; Canada’s PIPEDA: Disclosure of medical information
Introduction
Introduction

• File systems & DB communities
Introduction

- File systems & DB communities
  - tamper detection / prevention mechanisms
Introduction

- File systems & DB communities
  - tamper detection / prevention mechanisms
- Audit log security & compliant records
  - Creation
  - Storage
  - Access
  - Maintenance / Retention
Introduction

- File systems & DB communities
  - tamper detection / prevention mechanisms
- Audit log security & compliant records
  - Creation
  - Storage
  - Access
  - Maintenance / Retention

Governed by laws & regulations
Spectrum of Approaches to Achieve Trustworthiness
Spectrum of Approaches to Achieve Trustworthiness

Information **Restriction**

*immutable* retained records
*access control*
Spectrum of Approaches to Achieve Trustworthiness

Information Restriction
- immutable
- retained records
- access control

Information Accountability
- transparent
- information
- set of rules
- easily determine appropriate use
“[Information] accountability must become a primary means through which society addresses appropriate use.” (Weitzner et al., CACM 2008)
Restriction vs Accountability
Restriction vs Accountability

• Home Security
  - Locked doors and windows (restriction)
  - Sweeping front yard, cameras (accountability)
Restriction vs Accountability

• Home Security
  - Locked doors and windows (restriction)
  - Sweeping front yard, cameras (accountability)

• Bank Security
  - The vault is unlocked during business hours.
  - Easy access
  - CCTV cameras everywhere
Information Accountability
Information Accountability

- Tried and tested idea
Information Accountability

• Tried and tested idea

• Example: Bullae, sigils, seals, etc
Information Accountability

- Tried and tested idea
- Example: Bullae, sigils, seals, etc
Information Accountability

- Tried and tested idea
- Example: Bullae, sigils, seals, etc
Information Accountability

- Tried and tested idea
- Example: Bullae, sigils, seals, etc
Information Accountability

- Tried and tested idea
- Example: Bullae, sigils, seals, etc
Information Accountability

- Tried and tested idea
- Example: Bullae, sigils, seals, etc
Information Accountability

- Tried and tested idea
- Example: Bullae, sigils, seals, etc
Information Accountability (2)
Information Accountability (2)

• Tried and tested idea
Information Accountability (2)

• Tried and tested idea

• Example: Fair Credit Reporting Act (1970)

No rules on the collection of data and analysis but on their use (credit report).
Information Accountability (2)

- Tried and tested idea
- Example: Fair Credit Reporting Act (1970)

No rules on the collection of data and analysis but on their use (credit report).

The consumers are allowed access to the data.
Information Accountability

- Tried and tested idea
- Example: **Fair Credit Reporting Act** (1970)

No rules on the collection of data and analysis but on their use (credit report).

The consumers are allowed access to the data.

Agencies using credit reports are accountable for their decisions.
Information Accountability (3)
Information Accountability

- Tried and tested idea
Information Accountability

- Tried and tested idea
- Example: Creative Commons Licensing
Information Accountability

- Tried and tested idea
- Example: Creative Commons Licensing

Do not attempt to prevent the lawful use of works they protect by using technology, but rather set forth rules regulating the use of the works.
Tamper-Indicating Seals for Nuclear Safeguarding
Tamper-Indicating Seals for Nuclear Safeguarding
Accountability \cap Databases

Security

Accountability
Accountability ∩ Databases
Related Work & Security Spectrum

- Methodology
- Restriction
- Accountability

Data Granularity
- Coarse
- Fine
Related Work & Security Spectrum

Methodology

accountability

restriction

coarse

Data Granularity

fine

Fossilization (Hsu & Ong)

IBM

EMC

NetApp
Related Work & Security Spectrum

Methodology vs. Data Granularity

Coarse restriction to fine granularity

- Investigative Data Mining (Mena)
- Fossilization (Hsu & Ong)
Related Work & Security Spectrum

- Forensic Server Project (Carvey & Kleiman)
- Investigative Data Mining (Mena)
- Fossilization (Hsu & Ong)
Related Work & Security Spectrum

- Forensic Server Project (Carvey & Kleiman)
- Investigative Data Mining (Mena)
- Fossilization (Hsu & Ong)
- Indexing Structures (Goodrich et al.)

Methodology vs. Data Granularity:
- Accountability vs. Restriction
- Coarse vs. Fine

Brands:
- IBM
- EMC
- NetApp
Related Work & Security Spectrum

Methodology

accountability

restriction

Data Granularity

course

fine

Forensic Server Project (Carvey & Kleiman)

Investigative Data Mining (Mena)

Fossilization (Hsu & Ong)

Indexing Structures (Goodrich et al.)

SarbOx Workflows (Agrawal et al.)
Related Work & Security Spectrum

Forensic Server Project (Carvey & Kleiman)
Investigative Data Mining (Mena)
Fossilization (Hsu & Ong)
Indexing Structures (Goodrich et al.)
SarbOx Workflows (Agrawal et al.)

Methodology
accountability
restriction

Data Granularity
coarse
fine

IBM
EMC
NetApp
Total Recall
Related Work & Security Spectrum

- Forensic Server Project (Carvey & Kleiman)
- Investigative Data Mining (Mena)
- Fossilization (Hsu & Ong)
- Indexing Structures (Goodrich et al.)
- SarbOx Workflows (Agrawal et al.)

Methodology

Accountability

Restriction

Data Granularity

Coarse

Fine

Recent Work

Total Recall
Info Accountability of Fine-Grained Data

- Fragile watermarking scheme for detecting malicious modifications of database relations [Guo, Li, Liu, and Jajodia 2006].

- Efficient audit-based compliance for relational data retention [Hasan, Winslet, and Mitra 2009].

- Tamper detection in audit logs [Snodgrass, Yao, and Collberg 2004].
Accountability \( \cap \) Databases \( \cap \) Time
Accountability ∩ Databases ∩ Time
Accountability $\cap$ Databases $\cap$ Time

- Security
- Accountability
- Time
- Databases

Watermarking
Accountability \cap Databases \cap Time
Accountability ∩ Databases ∩ Time

- Security
- Provenance
- Watermarking

Temporal Databases
Accountability ∩ Databases ∩ Time
Temporal concepts are found throughout this area of interest.
Outline

• Information Accountability

• Reference Architecture & Execution Phases

• Forensic Analysis

• Refinements

• Enterprise Considerations
Outline

- Information Accountability
- Reference Architecture & Execution Phases
- Forensic Analysis
- Refinements
- Enterprise Considerations
Approach
Approach

- **Continuous assurance** technology
  - provides technology-enabled auditing
  - produces audit results close to occurrence of relevant events
  - achieves meaningful operationalization of information accountability.
Approach

- **Continuous assurance** technology
  - provides technology-enabled **auditing**
  - produces *audit results close to occurrence* of relevant events
  - achieves meaningful **operationalization of information accountability**.

- **Cryptographic hashing** captures state of database as it evolves.
Reference Architecture Threat Model
Reference Architecture Threat Model

- **Trusted computing base (TCB)**
  - Correctly booted and running hardware, OS and DBMS
  - TCB runs correctly until intrusion

- **A trusted external digital notarization service (EDNS)**

- **The adversary could be**
  - Inside/outside intruders who gain full control of the whole TCB and logs
  - Malware such as virus, bugs, power surge

- **Regret Interval**: minimum time before someone can reverse the change
  - Determined by the specific application
Reference Architecture Threat Model

- **Trusted computing base (TCB)**
  - Correctly booted and running hardware, OS and DBMS
  - TCB runs correctly until intrusion

- **A trusted external digital notarization service (EDNS)**

- **The adversary could be**
  - Inside/outside intruders who gain full control of the whole TCB and logs
  - Malware such as virus, bugs, power surge

- **Regret Interval**: minimum time before someone can reverse the change
  - Determined by the specific application
Reference Architecture Threat Model

• **Trusted computing base (TCB)**
  - Correctly booted and running hardware, OS and DBMS
  - TCB runs correctly until intrusion

• **A trusted external digital notarization service (EDNS)**

• **The adversary could be**
  - Inside/outside intruders who gain full control of the whole TCB and logs
  - Malware such as virus, bugs, power surge

• **Regret Interval**: minimum time before someone can reverse the change
  - Determined by the specific application
Reference Architecture Threat Model

- Trusted computing base (TCB)
  - Correctly booted and running hardware, OS and DBMS
  - TCB runs correctly until intrusion

- A trusted external digital notarization service (EDNS)

- The adversary could be
  - Inside/outside intruders who gain full control of the whole TCB and logs
  - Malware such as virus, bugs, power surge

- Regret Interval: minimum time before someone can reverse the change
  - Determined by the specific application
Total Chain Computation Phase

- User Application
- DBMS
- Transaction Time Database
- Secure Site
  - Notarizer
  - Secure Master Database
  - Forensic Analysis Algorithms
- External Digital Notarization Service
  - CSO GUI
  - DBA GUI
  - CSI GUI
Total Chain Computation Phase

- User Application
- DBMS
- Transaction Time Database
- Secure Site
- Notarizer
- Secure Master Database
- Forensic Analysis Algorithms
- External Digital Notarization Service
- CSO GUI
- DBA GUI
- CSI GUI
Total Chain Computation Phase
Total Chain Computation Phase

- User Application
- DBMS
- Secure Site
- Transaction Time Database
- Secure Master Database
- Forensic Analysis Algorithms
- Notarizer
- External Digital Notarization Service

Transactions flow from User Application to DBMS to Secure Site. Secure Site includes Secure Master Database and Forensic Analysis Algorithms. Notarizer connects Secure Site to External Digital Notarization Service.
Total Chain Computation Phase

- User Application
- DBMS
- Transaction Time Database
- Secure Site
- Secure Master Database
- Forensic Analysis Algorithms
- Notarizer

Transactions:
- External Digital Notarization Service
- CSO GUI
- DBA GUI
- CSI GUI
Total Chain Computation Phase

User Application

DBMS

transactions

records

External Digital Notarization Service

External Digital Notarization Service

Transaction Time Database

Notarizer

Secure Site

Secure Master Database

Forensic Analysis Algorithms

CSO GUI

DBA GUI

CSI GUI
Total Chain Computation Phase

Transactions flow from the User Application to the DBMS. The DBMS records transactions and sends them to the Notarizer. The Notarizer generates a hash value for the transactions and sends it to the External Digital Notarization Service. The Secure Site contains the Secure Master Database, Forensic Analysis Algorithms, CSO GUI, DBA GUI, and CSI GUI.
Total Chain Computation Phase

- User Application
- DBMS
- Transaction Time Database
- Secure Site
- Notarizer
- Secure Master Database
- Forensic Analysis Algorithms
- CSO GUI
- DBA GUI
- CSI GUI
- External Digital Notarization Service

Transactions flow through the system, with records being handled by the DBMS, and transactions being tracked through the Secure Site. The Notarizer generates hash value and notary ID, which are sent to the External Digital Notarization Service.
Total Chain Computation Phase

- User Application
  - transactions
  - DBMS
    - records
    - Transaction Time Database
    - Secure Site
      - Secure Master Database
        - Forensic Analysis Algorithms
          - notary ID
          - hash value
      - Notarizer
        - notary ID
        - External Digital Notarization Service
          - CSO GUI
          - DBA GUI
          - CSI GUI
Tamper Detection and Forensic Analysis Phase

Validator

Secure Master Database

Forensic Analysis Algorithms

Secure Site

DBMS

Transaction Time Database

External Digital Notarization Service

CSO GUI

DBA GUI

CSI GUI
Tamper Detection and Forensic Analysis Phase

- Secure Site
  - Secure Master Database
  - Forensic Analysis Algorithms
  - Validator
- DBMS
- Transaction Time Database
- CSO GUI
- CSO GUI
- DBA GUI
- External Digital Notarization Service
Tamper Detection and Forensic Analysis Phase

- **DBMS**
- **Secure Master Database**
- **Forensic Analysis Algorithms**
- **Transaction Time Database**
- **Secure Site**
- **Validator**
- **External Digital Notarization Service**
  - CSO GUI
  - DBA GUI
  - CSI GUI
Tamper Detection and Forensic Analysis Phase

- **DBMS**
- **Secure Master Database**
- **Forensic Analysis Algorithms**
- **Secure Site**
- **Validator**
- **Transaction Time Database**

**External Digital Notarization Service**
- **CSO GUI**
- **DBA GUI**
- **CSI GUI**

Data flow:
- Data from DBMS
- Rehash from Secure Master Database

System components:
- DBMS
- Secure Master Database
- Forensic Analysis Algorithms
- Secure Site
- Validator

Graphic elements:
- Blue arrows indicate data flow and operation steps.
Tamper Detection and Forensic Analysis Phase

- DBMS
- Secure Master Database
- Forensic Analysis Algorithms
- Secure Site
- Validator
- Notary ID
- Rehash
- Data
- Transaction Time Database
- External Digital Notarization Service
- CSO GUI
- DBA GUI
- CSI GUI
Tamper Detection and Forensic Analysis Phase

DBMS

Secure Master Database

Forensic Analysis Algorithms

Validator

External Digital Notarization Service

CSO GUI

DBA GUI

CSI GUI

hash value + notary ID

rehash

data

Transaction Time Database

Secure Site
Tamper Detection and Forensic Analysis Phase

- **DBMS**: Database Management System
- **Secure Master Database**: Stores secure, tamper-proof data
- **Forensic Analysis Algorithms**: Analyzes data for tampering
- **Validator**: Validates the hash value and notary ID
- **notary ID**: Identification of the notary
- **hash value**: Hash of the data
- **result(s)**: Validation result

Flow:
- Data from DBMS to Secure Site
- Rehash from Secure Master Database to Validator
- Hash value + notary ID to External Digital Notarization Service
- Result(s) from External Digital Notarization Service to Validator
- Validator to Secure Site
- Secure Site to DBMS
Tamper Detection and Forensic Analysis Phase

**CSO GUI**

**CSI GUI**

**Algorithms**

**Forensic Analysis**

**Validator**

**Notarization Service**

**External Digital Notarization Service**

**Secure Site**

**DBMS (including Audit Log)**

**Database Transaction Time**

**Secure Master Database**

**Forensic Analysis Algorithms**

**Data**

**Rehash**

**Hash value + notary ID**

**Result(s)**
Tamper Detection and Forensic Analysis Phase

DBMS

Secure Site

Forensic Analysis Algorithms

Secure Master Database

Validator

External Digital Notarization Service

hash value + notary ID

result(s)

CSO GUI

DBA GUI

CSI GUI

Transaction Time Database

data

rehash

result(s)
Tamper Detection and Forensic Analysis Phase

- **DBMS**
- **Secure Site**
  - **Secure Master Database**
    - **Forensic Analysis Algorithms**
  - **Validator**
    - notary ID → result(s)
    - result(s) → hash value + notary ID

- **External Digital Notarization Service**
  - CSO GUI
  - DBA GUI
  - CSI GUI

- **Transaction Time Database**
- Data input from DBMS
- Hash value and notary ID are verified through the **Validator**.
Tamper Detection and Forensic Analysis Phase

- DBMS
  - Data
  - Transaction Time Database
- Secure Site
  - Secure Master Database
  - Forensic Analysis Algorithms
  - Hash value + notary ID
  - Result(s)
  - External Digital Notarization Service
- CSO GUI
- DBA GUI
- CSI GUI
- Corruption region(s)

Validator

- Notary ID
- Result(s)
Outline

• Information Accountability

• Reference Architecture & Execution Phases

• Forensic Analysis

• Refinements

• Enterprise Considerations
Tampering, Detection and Forensic Analysis
Tampering, Detection and Forensic Analysis

Transaction Processing

DBMS state (Legal)  Tampering
Tampering, Detection and Forensic Analysis

Transaction Processing

DBMS state (Legal) → Tampering → DBMS state (Illegal)

DBMS state (Legal) → ( Illegal)
Tampering, Detection and Forensic Analysis

Transaction Processing

DBMS state (Legal) → Tampering → DBMS state (Illegal)
Tampering, Detection and Forensic Analysis

Transaction Processing → DBMS state (Legal) → Tampering → DBMS state (Illegal) → Validation

$t_c$
Tampering, Detection and Forensic Analysis

Transaction Processing

DBMS state (Legal) → Tampering → DBMS state (Illegal) → Validation → Tamper Detection

$t_c$
Tampering, Detection and Forensic Analysis

Transaction Processing

DBMS state (Legal) → Tampering → DBMS state (Illegal) → Validation → Tamper Detection

$t_c$ $t_{FVF}$
Tampering, Detection and Forensic Analysis

Transaction Processing

DBMS state (Legal) → Tampering → DBMS state (Illegal) → Validation → Tamper Detection

$\text{tc}$

$\text{t}_{FVF}$

Forensic Analysis Algorithms
Tampering, Detection and Forensic Analysis

Transaction Processing

DBMS state (Legal) → Tampering → DBMS state (Illegal) → Validation → Tamper Detection

$t_c$  
$t_{FVF}$

Forensic Analysis Algorithms
Corruption Region(s)
The Corruption Diagram
The Corruption Diagram

When

Actual time

Where
The Corruption Diagram
The Corruption Diagram

When

Where

Commit time
The Corruption Diagram

When

Where
The Corruption Diagram

When

$NE_0$

$NE$: Notarization Event

Where
The Corruption Diagram

When

NE: Notarization Event

Where

NE₀

Schema notarization
The Corruption Diagram

When

NE: Notarization Event

Where

Schema notarization

$NE_0$
The Corruption Diagram

NE: Notarization Event

When

Where

Schema notarization

NE0

NE1
The Corruption Diagram

NE: Notarization
Event

NE₀

link

NE₁

Schema
notarization

When

Where
The Corruption Diagram

When

$NE_0$\hspace{1cm}NE_1$

$NE$: Notarization Event

Schema notarization

Where
The Corruption Diagram

When

Where

$NE_0$:

$NE_1$: Notarization
Event

$NE_2$:

Schema
notarization
The Corruption Diagram

NE: Notarization Event

When

Where

Schema notarization

NE₀

NE₁

NE₂

link
The Corruption Diagram

When

Where

NE: Notarization Event

Schema notarization

NE₀

NE₁

NE₂
The Corruption Diagram

When

NE: Notarization Event

Where

NE₀

NE₁

NE₂

NE₃

Schema notarization
The Corruption Diagram

When

NE: Notarization Event

VE: Validation Event

Schema notarization

Where
The Corruption Diagram

NE: Notarization Event

VE: Validation Event

VE_1 = TRUE

Schema notarization
The Corruption Diagram

NE: Notarization Event
VE: Validation Event

When

Where

NE₀

NE₁

NE₂

NE₃

NE₄

VE₁ = TRUE

Schema notarization
The Corruption Diagram

NE: Notarization Event
VE: Validation Event

When

Where

NE₀

NE₁

NE₂

NE₃

VE₁ = TRUE

NE₄

NE₅

Schema notarization
The Corruption Diagram

- **NE**: Notarization Event
- **VE**: Validation Event

Schema: notarization
When

NE: Notarization Event

VE: Validation Event

Schema notarization

The Corruption Diagram
The Corruption Diagram

When

NE: Notarization Event

VE: Validation Event

NE_0: Notarization

VE_1 = TRUE

NE_1

NE_2

NE_3

NE_4

NE_5

NE_6

VE_2 = TRUE

Where

notarization interval

I_N

Schema notarization
The Corruption Diagram

When

NE: Notarization Event
VE: Validation Event

Validation interval

NE0
NE1
NE2
NE3
NE4
NE5
NE6

VE1 = TRUE

VE2 = TRUE

Schema notarization

Notarization interval

I_N

I_V

Where
The Corruption Diagram

When

NE: Notarization Event

VE: Validation Event

CE: Corruption Event

Schema notarization

Where

NE_0

NE_1

NE_2

NE_3

NE_4

NE_5

NE_6

VE_1 = TRUE

CE

When

Where
The Corruption Diagram

When

NE: Notarization Event

VE: Validation Event

CE: Corruption Event

clock time \( (t_c) \)

Schema notarization

Where

NE\(_0\)

NE\(_1\)

NE\(_2\)

NE\(_3\)

NE\(_4\)

NE\(_5\)

NE\(_6\)

VE\(_1\) = TRUE

VE\(_2\)

When:

Where:

Notarization

Validation

Corruption

Event

Event

Event
The Corruption Diagram

When

NE: Notarization Event

VE: Validation Event

CE: Corruption Event

Clock time ($t_c$)

Commit time ($t_l$)

Where

Schema notarization

$NE_0$ to $NE_6$
Types of Corruption Events

CE: Corruption Event

When

clock time: $t_c$

locus time: $t_l$

Where

$NE_0$

$NE_1$

$NE_2$

$NE_3$

$NE_4$

$NE_5$

$NE_6$

$VE_1$

$VE_2$

$NE_0$
Types of Corruption Events

CE: Corruption Event

When

Data-only Corruption

Where

clock time: $t_c$

locus time: $t_l$
Types of Corruption Events

CE: Corruption Event

Data-only Corruption

CE: Corruption Event

clock time: \( t_c \)

locus time: \( t_l \)
Types of Corruption Events

CE: Corruption Event

clock time: $t_c$

locus time: $t_l$

When

Data-only Corruption

Retroactive

Where

VE

NE

$NE_0$, $NE_1$, $NE_2$, $NE_3$, $NE_4$, $NE_5$, $NE_6$, $VE_1$, $VE_2$
Types of Corruption Events

When

CE: Corruption Event

clock time: $t_c$

locus time: $t_l$

Data-only Corruption

Where

$NE_0$

$NE_1$ $NE_2$ $NE_3$ $NE_4$ $NE_5$ $NE_6$

$VE_1$

$VE_2$
Types of Corruption Events

- **CE**: Corruption Event
- **NE**: Node Event
- **VE**: Verifiable Event
- **clock time**: $t_c$
- **locus time**: $t_l$

Data-only Corruption

Introactive

Graph with nodes $NE_0, NE_1, NE_2, NE_3, NE_4, NE_5, NE_6, VE_1, VE_2$.
Types of Corruption Events

Timestamp Corruption

CE: Corruption Event

clock time: $t_c$

 locus time: $t_l$

$NE_0$

$NE_1$

$NE_2$

$NE_3$

$NE_4$

$NE_5$

$NE_6$

$VE_1$

$VE_2$

When

Where
Types of Corruption Events

Timestamp Corruption

CE: Corruption Event

clock time: $t_c$

locus time: $t_l$

When

Where
Types of Corruption Events

CE: Corruption Event

When

Timestamp Corruption

clock time: $t_c$

locus time: $t_l$

postdating time: $t_p$

Where

NE: Corruption Event

VE: External Event
Types of Corruption Events

CE: Corruption Event

Timestamp Corruption

When

clock time: $t_c$

Where

locus time: $t_l$
Types of Corruption Events

- CE: Corruption Event
- clock time: $t_c$
- locus time: $t_l$

Timestamp Corruption

When

Where
Types of Corruption Events

CE: Corruption Event

Timestamp Corruption

When

clock time: $t_c$

backdating time: $t_b$

locus time: $t_l$

Where

$NE_0$

$NE_1$

$NE_2$

$NE_3$

$NE_4$

$NE_5$

$NE_6$

$VE_1$

$VE_2$
Forensic Analysis
Forensic Analysis

- If a corruption is detected, then the forensic analysis phase begins.
Forensic Analysis

- If a corruption is detected, then the forensic analysis phase begins.

- A **forensic analysis algorithm** is run as directed by the Database Administrator.
Forensic Analysis

- If a corruption is detected, then the *forensic analysis* phase begins.

- A *forensic analysis algorithm* is run as directed by the Database Administrator.

- Attempt to ascertain a *corruption region*: the bounds on the uncertainty of the “where” and “when” of the corruption.
Detection Resolution

- **Temporal Detection Resolution** ($R_t$): the finest granularity of temporal bounds uncertainty of a CE.

- **Spatial Detection Resolution** ($R_s$): the finest granularity of spatial bounds uncertainty of a CE.
Monochromatic Algorithm

When

Where

$VE_1 = TRUE$

$NE_3$

$NE_2$

$NE_1$

$NE_0$
Monochromatic Algorithm

When

Where

\( \text{NE}_0 \)

\( \text{NE}_1 \)

\( \text{NE}_2 \)

\( \text{NE}_3 \)

\( \text{VE}_1 = \text{TRUE} \)

Total Chain
Monochromatic Algorithm

When

Where

Total Chain

$VE_1 = TRUE$

$NE_3$

$NE_2$

$NE_1$

$NE_0$
Monochromatic Algorithm

When

Where

Total Chain

VE_1 = TRUE

NE_2

NE_3

NE_4

NE_0
Monochromatic Algorithm

When

Where

Total Chain

$\text{CE}$

$NE_0$

$NE_1$

$NE_2$

$NE_3$

$NE_4$

$VE_1 = \text{TRUE}$
Monochromatic Algorithm

When

Where

CE

Total Chain

$NE_0$

$NE_1$

$NE_2$

$NE_3$

$NE_4$

$NE_5$

$VE_1 = \text{TRUE}$
Monochromatic Algorithm

When

Where

Total Chain

CE

NE₁

NE₂

NE₃

NE₄

NE₅

VE₁ = TRUE

NE₀
Monochromatic Algorithm

When

Where

$NE_0$

$NE_1$

$NE_2$

$NE_3$

$NE_4$

$NE_5$

$NE_6$

$VE_1 = \text{TRUE}$

Total Chain
Monochromatic Algorithm

When

Where

Total Chain

VE_2 = FALSE

VE_1 = TRUE

NE_0

NE_1

NE_2

NE_3

NE_4

NE_5

NE_6
Monochromatic Algorithm

Time of first validation failure ($t_{FVF}$)
Monochromatic Algorithm

When

Forensic analysis begins

<table>
<thead>
<tr>
<th>When</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>NE₀</td>
<td>CE</td>
</tr>
<tr>
<td>NE₁</td>
<td></td>
</tr>
<tr>
<td>NE₂</td>
<td></td>
</tr>
<tr>
<td>NE₃</td>
<td></td>
</tr>
<tr>
<td>NE₄</td>
<td></td>
</tr>
<tr>
<td>NE₅</td>
<td></td>
</tr>
<tr>
<td>NE₆</td>
<td></td>
</tr>
</tbody>
</table>

Time of first validation failure ($t_{FVF}$)

Total Chain
Monochromatic Algorithm

When  Where

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

When

Forensic analysis begins

Time of first validation failure ($t_{FVF}$)

$VE_2 = FALSE$

$VE_1 = TRUE$

Total Chain

$NE_0$

$NE_1$

$NE_2$

$NE_3$

$NE_4$

$NE_5$

$NE_6$
Monochromatic Algorithm

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

When

Where

VE_2 = FALSE

Total Chain

Time of first validation failure ($t_{FVF}$)

Forensic analysis begins

$NE_0$, $NE_1$, $NE_2$, $NE_3$, $NE_4$, $NE_5$, $NE_6$
Monochromatic Algorithm

When

\[ UTB \]

time of corruption \((t_c)\)

Where

\[ LTB \]

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

Forensic analysis begins

Time of first validation failure \((t_{FVF})\)

\[ VE_2 = \text{FALSE} \]

\[ VE_1 = \text{TRUE} \]

Total Chain

\[ NE_0 \]

\[ NE_1 \]

\[ NE_2 \]

\[ NE_3 \]

\[ NE_4 \]

\[ NE_5 \]

\[ NE_6 \]
Monochromatic Algorithm

- **When**
  - UTB
  - time of corruption ($t_c$)
- **Where**
  - LTB
    - Corruption Region: captures the uncertainty as to the position of CE

**Forensic analysis begins** at the time of first validation failure ($t_{FVF}$).

- VE$_2$ = FALSE
- VE$_1$ = TRUE
- NE$_6$

**Total Chain**
Monochromatic Algorithm

Forensic analysis begins

Time of first validation failure ($t_{FVF}$)

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

When

$UTB$

time of corruption ($t_c$)

$LTB$

When

$NE_0$

$NE_1$

$NE_2$

$NE_3$

$NE_4$

$NE_5$

$NE_6$

$F$

$VE_1 = TRUE$

$VE_2 = FALSE$

Where

Total Chain
Monochromatic Algorithm

When

UTB

time of corruption \((t_c)\)

LTB

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

Where

Forensic analysis begins

Time of first validation failure \((t_{FVF})\)

\[ \text{VE}_2 = \text{FALSE} \]

\[ \text{VE}_1 = \text{TRUE} \]

Total Chain
Monochromatic Algorithm

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

When

$UTB$

time of corruption ($t_c$)

$LTB$

Forensic analysis begins

Time of first validation failure ($t_{FVF}$)

Total Chain

$NE_0$

$NE_1$

$NE_2$

$NE_3$

$NE_4$

$NE_5$

$NE_6$

$VE_1 = TRUE$

$VE_2 = FALSE$

When

$F F F F F$

Where
Monochromatic Algorithm

When

*UTB*

Time of corruption \(t_c\)

*LTB*

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

Forensic analysis begins

Time of first validation failure \(t_{FVF}\)

\(VE_1 = TRUE\)

\(VE_2 = FALSE\)

Total Chain
Monochromatic Algorithm

When

UTB

Time of corruption ($t_c$)

LTB

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

Where

Forensic analysis begins

Time of first validation failure ($t_{FVF}$)

$V_{E2} = \text{FALSE}$

$V_{E1} = \text{TRUE}$

Total Chain

$NE_0$ $NE_1$ $NE_2$ $NE_3$ $NE_4$ $NE_5$ $NE_6$
Monochromatic Algorithm

Forensic analysis begins

Time of first validation failure ($t_{FVF}$)

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

When

$UTB$

time of corruption ($t_c$)

$LTB$

$NE_0$

$NE_1$

$NE_2$

$NE_3$

$NE_4$

$NE_5$

$NE_6$

$VE_1 = TRUE$

$VE_2 = FALSE$

Total Chain

Where

$LSB$

$USB$
Monochromatic Algorithm

When

Time of corruption ($t_c$)

$UTB$

$LTB$

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

Where

Time of most recent validation success ($t_{RVS}$)

$LSB$  $USB$

When

Forensic analysis begins

Total Chain

Time of first validation failure ($t_{FVF}$)

$VE_2 = FALSE$

$VE_1 = TRUE$

$NE_1$  $NE_2$  $NE_3$  $NE_4$  $NE_5$  $NE_6$

$F$  $F$  $F$  $F$  $F$
Monochromatic Algorithm

When

$UTB$

time of corruption ($t_c$)

$LTB$

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

When

Forensic analysis begins

Where

Time of most recent validation success ($t_{RVS}$)

$t_i$: place of corruption (commit time)

Total Chain

Time of first validation failure ($t_{FVF}$)

$VE_1 = TRUE$

$VE_2 = FALSE$

$NE_0$

$NE_1$

$NE_2$

$NE_3$

$NE_4$

$NE_5$

$NE_6$

$LSB$

$USB$
The a3D Algorithm

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]

\[ \text{\textcolor{blue}{VE}_1 = TRUE} \]
The a3D Algorithm

When

\( R_s = 1 \text{ day} \)
\( N = 2 \)
\( I_N = 2 \text{ days} \)
\( V = 1 \)
\( I_V = 2 \text{ days} \)

\( NE_0 \)

\( VE_1 = \text{TRUE} \)
\( VE_2 = \text{TRUE} \)

Notarization

Events
The a3D Algorithm

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]
The a3D Algorithm

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]
The a3D Algorithm

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]
The a3D Algorithm

When

$R_s = 1$ day
$N = 2$
$I_N = 2$ days
$V = 1$
$I_V = 2$ days
The a3D Algorithm

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]
The a3D Algorithm

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]
The a3D Algorithm

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]
$R_s = 1 \text{ day}$

$N = 2$

$I_N = 2 \text{ days}$

$V = 1$

$I_V = 2 \text{ days}$

When

$VE_1 = \text{ TRUE}$

$VE_2 = \text{ TRUE}$

$VE_3 = \text{ TRUE}$

$VE_4 = \text{ TRUE}$

$VE_5 = \text{ TRUE}$

$VE_6 = \text{ TRUE}$
The a3D Algorithm

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]

Notarization Events

VE_1 = TRUE
VE_2 = TRUE
VE_3 = TRUE
VE_4 = TRUE
VE_5 = TRUE
VE_6 = TRUE
The a3D Algorithm

When

\( R_s = 1 \text{ day} \)
\( N = 2 \)
\( I_N = 2 \text{ days} \)
\( V = 1 \)
\( I_V = 2 \text{ days} \)
The a3D Algorithm

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]

Backdating CE

\[ \text{VE}_1 = \text{TRUE} \]
\[ \text{VE}_2 = \text{TRUE} \]
\[ \text{VE}_3 = \text{TRUE} \]
\[ \text{VE}_4 = \text{TRUE} \]
\[ \text{VE}_5 = \text{TRUE} \]
\[ \text{VE}_6 = \text{TRUE} \]
\[ \text{VE}_7 = \text{TRUE} \]
The a3D Algorithm

When

\( R_s = 1 \) day
\( N = 2 \)
\( I_N = 2 \) days
\( V = 1 \)
\( I_V = 2 \) days

Backdating CE

\( VE_1 = TRUE \)
\( VE_2 = TRUE \)
\( VE_3 = TRUE \)
\( VE_4 = TRUE \)
\( VE_5 = TRUE \)
\( VE_6 = TRUE \)
\( VE_7 = TRUE \)

Notarization Events
The a3D Algorithm

When

- \( R_s = 1 \text{ day} \)
- \( N = 2 \)
- \( I_N = 2 \text{ days} \)
- \( V = 1 \)
- \( I_V = 2 \text{ days} \)

Backdating CE

Notarization

Events
The a3D Algorithm

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]

Forensic analysis begins

Backdating CE

\[ VE_1 = \text{TRUE} \]
\[ VE_2 = \text{TRUE} \]
\[ VE_3 = \text{TRUE} \]
\[ VE_4 = \text{TRUE} \]
\[ VE_5 = \text{TRUE} \]
\[ VE_6 = \text{TRUE} \]
\[ VE_7 = \text{TRUE} \]
\[ VE_8 = \text{FALSE} \]
The a3D Algorithm

When

- $R_s = 1$ day
- $N = 2$
- $I_N = 2$ days
- $V = 1$
- $I_V = 2$ days

Backdating CE

Events

Notarization
The a3D Algorithm

When

\( R_s = 1 \text{ day} \)
\( N = 2 \)
\( I_N = 2 \text{ days} \)
\( V = 1 \)
\( I_V = 2 \text{ days} \)

Backdating CE

\( VE_1 = TRUE \)
\( VE_2 = TRUE \)
\( VE_3 = TRUE \)
\( VE_4 = TRUE \)
\( VE_5 = TRUE \)
\( VE_6 = TRUE \)
\( VE_7 = TRUE \)
\( VE_8 = FALSE \)

Notarization

Events
The a3D Algorithm

When

\( R_s = 1 \text{ day} \)
\( N = 2 \)
\( I_N = 2 \text{ days} \)
\( V = 1 \)
\( I_V = 2 \text{ days} \)

Backdating CE

\( NE_0 \)

\( VE_1 = \text{TRUE} \)
\( VE_2 = \text{TRUE} \)
\( VE_3 = \text{TRUE} \)
\( VE_4 = \text{TRUE} \)
\( VE_5 = \text{TRUE} \)
\( VE_6 = \text{TRUE} \)
\( VE_7 = \text{TRUE} \)
\( VE_8 = \text{FALSE} \)

Notarization Events
The a3D Algorithm

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]

Backdating CE

\[ VE_1 = \text{TRUE} \]
\[ VE_2 = \text{TRUE} \]
\[ VE_3 = \text{TRUE} \]
\[ VE_4 = \text{TRUE} \]
\[ VE_5 = \text{TRUE} \]
\[ VE_6 = \text{TRUE} \]
\[ VE_7 = \text{TRUE} \]
\[ VE_8 = \text{FALSE} \]
The a3D Algorithm

Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

When

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]

Backdating CE

\[ V_E = 1 \]
\[ V_E = 2 \]
\[ V_E = 3 \]
\[ V_E = 4 \]
\[ V_E = 5 \]
\[ V_E = 6 \]
\[ V_E = 7 \]
\[ V_E = 8 \]

NE

\[ V_E = FALSE \]
\[ V_E = TRUE \]
\[ V_E = TRUE \]
\[ V_E = TRUE \]
\[ V_E = TRUE \]

Notarization Events

 NE _0_
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

The a3D Algorithm

When

$R_s = 1$ day

$N = 2$

$I_N = 2$ days

$V = 1$

$I_V = 2$ days

Backdating CE

Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$VE_1 = TRUE$

$VE_2 = TRUE$

$VE_3 = TRUE$

$VE_4 = TRUE$

$VE_5 = TRUE$

$VE_6 = TRUE$

$VE_7 = TRUE$

$VE_8 = FALSE$
The a3D Algorithm

When

$R_s = 1$ day
$N = 2$
$I_N = 2$ days
$V = 1$
$I_V = 2$ days

Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$R_s = 1$ day
$N = 2$
$I_N = 2$ days
$V = 1$
$I_V = 2$ days
The a3D Algorithm

Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

- $R_s = 1$ day
- $N = 2$
- $I_N = 2$ days
- $V = 1$
- $I_V = 2$ days

When

- $t_c$
- $T$
- $F$
- $F$

Backdating CE

- $VE_1 = TRUE$
- $VE_2 = TRUE$
- $VE_3 = TRUE$
- $VE_4 = TRUE$
- $VE_5 = TRUE$
- $VE_6 = TRUE$
- $VE_7 = TRUE$
- $VE_8 = FALSE$

Notarization

Events

- $NE_0$
The a3D Algorithm

Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.
The a3D Algorithm

Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$t_b$: backdating time
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]

**Backdating CE**

\[ t_b: \text{ backdating time} \]
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$t_b$: backdating time
The a3D Algorithm

Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$t_b$: backdating time

$R_s = 1$ day
$N = 2$
$I_N = 2$ days
$V = 1$
$I_V = 2$ days

$VE_1 = TRUE$
$VE_2 = TRUE$
$VE_3 = TRUE$
$VE_4 = TRUE$
$VE_5 = TRUE$
$VE_6 = TRUE$
$VE_7 = TRUE$
$VE_8 = FALSE$

$t_c$: backdating time

Notarization
Events
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$t_b$: backdating time

When

$R_s = 1$ day
$N = 2$
$I_N = 2$ days
$V = 1$
$I_V = 2$ days

The a3D Algorithm

Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$t_b$: backdating time

$NE_0 \rightarrow t_b$
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$t_b$: backdating time

When

$R_s = 1$ day
$N = 2$
$I_N = 2$ days
$V = 1$
$I_V = 2$ days

The a3D Algorithm

$VE_8 = \text{FALSE}$
$VE_7 = \text{TRUE}$
$VE_6 = \text{TRUE}$
$VE_5 = \text{TRUE}$
$VE_4 = \text{TRUE}$
$VE_3 = \text{TRUE}$
$VE_2 = \text{TRUE}$
$VE_1 = \text{TRUE}$

Backdating CE
The a3D Algorithm

Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$t_b$: backdating time

$R_s = 1$ day
$N = 2$
$I_N = 2$ days
$V = 1$
$I_V = 2$ days
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$t_b$: backdating time

$R_s = 1$ day
$N = 2$
$\mathit{I}_N = 2$ days
$V = 1$
$\mathit{I}_V = 2$ days

The a3D Algorithm

When

$t_c$

$\mathit{VE}_1 = \text{TRUE}$
$\mathit{VE}_2 = \text{TRUE}$
$\mathit{VE}_3 = \text{TRUE}$
$\mathit{VE}_4 = \text{TRUE}$
$\mathit{VE}_5 = \text{TRUE}$
$\mathit{VE}_6 = \text{TRUE}$
$\mathit{VE}_7 = \text{TRUE}$
$\mathit{VE}_8 = \text{FALSE}$

Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.
The a3D Algorithm

Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$t_b$: backdating time

$R_s = 1$ day
$N = 2$
$I_N = 2$ days
$V = 1$
$I_V = 2$ days

Backdating CE

$t_c$

Notarization Events

$NE_0$

$N E_0$

$V E_1 = \text{TRUE}$

$V E_2 = \text{TRUE}$

$V E_3 = \text{TRUE}$

$V E_4 = \text{TRUE}$

$V E_5 = \text{TRUE}$

$V E_6 = \text{TRUE}$

$V E_7 = \text{TRUE}$

$V E_8 = \text{FALSE}$
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$t_b$: backdating time

$R_s = 1$ day

$N = 2$

$I_N = 2$ days

$V = 1$

$I_V = 2$ days

Backdating CE
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$R_s = 1$ day
$N = 2$
$I_N = 2$ days
$V = 1$
$I_V = 2$ days

$t_b$: backdating time
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

\[ R_s = 1 \text{ day} \]
\[ N = 2 \]
\[ I_N = 2 \text{ days} \]
\[ V = 1 \]
\[ I_V = 2 \text{ days} \]

**Backdating CE**

\[ t_b : \text{backdating time} \]

\[ NE_0 \]

\[ NE_1 \]

\[ NE_2 \]

\[ NE_3 \]

\[ NE_4 \]

\[ NE_5 \]

\[ NE_6 \]

\[ NE_7 \]

\[ NE_8 \]

\[ VE_1 = \text{TRUE} \]

\[ VE_2 = \text{TRUE} \]

\[ VE_3 = \text{TRUE} \]

\[ VE_4 = \text{TRUE} \]

\[ VE_5 = \text{TRUE} \]

\[ VE_6 = \text{TRUE} \]

\[ VE_7 = \text{TRUE} \]

\[ VE_8 = \text{FALSE} \]
Can use recursive binary search on the hash chains to locate all days during which tampering has occurred.

$t_b$: backdating time

The a3D Algorithm

When

$R_s = 1$ day
$N = 2$
$I_N = 2$ days
$V = 1$
$I_V = 2$ days

Backdating CE

$t_b$: backdating time
## Comparison of Forensic Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Commit-Time-Based</th>
<th>Page-Based</th>
<th>Attribute-Based</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Partitions</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Tables Affected</strong></td>
<td>Any number</td>
<td>Any number</td>
<td>One or several of those containing the designated attribute</td>
</tr>
<tr>
<td><strong>R_s</strong></td>
<td>Time interval</td>
<td>Time interval</td>
<td>Time interval</td>
</tr>
<tr>
<td><strong>R_d</strong></td>
<td>N/A</td>
<td>N/A</td>
<td>Number of subsets of domain values</td>
</tr>
<tr>
<td><strong>Segment</strong></td>
<td>One of the contiguous periods induced by $R_s$, starting from a particular anchor. Contiguous periods form a chronologically ordered partition.</td>
<td>One of the contiguous periods induced by $R_s$, starting from a particular anchor. Contiguous periods form a chronologically ordered partition.</td>
<td>One of the contiguous periods induced by $R_s$, starting from a particular anchor. Contiguous periods form a chronologically ordered partition.</td>
</tr>
<tr>
<td><strong>Granule</strong></td>
<td>Encompasses all tuples with commit times within the associated segment (one granule has tuples from many transactions committing in that segment).</td>
<td>Encompasses all tuples whose physical location is in a page mentioned within the associated segment.</td>
<td>Encompasses all tuples with commit times within the associated segment (one granule has tuples from many transactions committing in that segment).</td>
</tr>
<tr>
<td><strong>Hashing order</strong></td>
<td>Transactions hashed in order of increasing commit time.</td>
<td>Granules hashed in chronological order of “page write” event of the page. Granules not hashed in order of page number.</td>
<td>Transactions hashed in order of increasing commit time.</td>
</tr>
<tr>
<td><strong>Segment Completion Event</strong></td>
<td>When the last tuple in the granule associated with that segment commits</td>
<td>When the last page write event in the segment occurs.</td>
<td>When the last tuple in the granule associated with that segment commits</td>
</tr>
<tr>
<td><strong>Notarization Factor (N)</strong></td>
<td>Specified by DBA</td>
<td>Specified by DBA</td>
<td>Specified by DBA</td>
</tr>
<tr>
<td><strong>Validation Factor (V)</strong></td>
<td>Specified by DBA</td>
<td>Specified by DBA</td>
<td>Specified by DBA</td>
</tr>
<tr>
<td>$I_N$</td>
<td>$N \times R_s$</td>
<td>$N \times R_s$</td>
<td>$N \times R_s$</td>
</tr>
<tr>
<td>$I_V$</td>
<td>$V \times I_N$</td>
<td>$V \times I_N$</td>
<td>$V \times I_N$</td>
</tr>
<tr>
<td><strong>Notarization</strong></td>
<td>Occurs as soon as $N$ granules are hashed.</td>
<td>Occurs as soon as $N$ granules are hashed.</td>
<td>Occurs as soon as $N$ granules are hashed.</td>
</tr>
<tr>
<td><strong>Validation</strong></td>
<td>Occurs as soon as $V$ notarizations have occurred.</td>
<td>Occurs as soon as $V$ notarizations have occurred.</td>
<td>Occurs as soon as $V$ notarizations have occurred.</td>
</tr>
</tbody>
</table>
The Correlated Page-Based Corruption Diagram Depicting the a3D Algorithm

Clock Time

When

Page

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

VE = False

VE = True

\[E(3, 10, 14.3)\]

V = 1

N = 2

\[R_e = 1\]

= Relevant partial chains

Where → Page #

Page Write Events

Time of Write Event

Time of Clock

\[= \text{False}\]

\[= \text{True}\]
The Correlated Page-Based Corruption Diagram Depicting the a3D Algorithm
The Correlated Page-Based Corruption Diagram Depicting the a3D Algorithm

Clock Time

When

Page Write Events

Time of Write Event

Where → Page #

Relevant partial chains

V = 1
N = 2
R_2 = 1

CE_1
CE_2

Page #

Time of Write Event E (3, 10, 14.3)
The Correlated Page-Based Corruption Diagram Depicting the a3D Algorithm
The Correlated Page-Based Corruption Diagram Depicting the a3D Algorithm

Where ➞ Page #

Page Write Events

Clock Time

When

Time of Write Event:

\( CE_1 \)

\( CE_2 \)

\( V = 1 \)

\( N = 2 \)

\( R_e = 1 \)

\( \text{Relevant partial chains} \)
The Correlated Page-Based Corruption Diagram Depicting the a3D Algorithm
The Correlated Page-Based Corruption Diagram Depicting the a3D Algorithm

Clock Time

Where ➔ Page #

When

Page Write Events

{\begin{align*}
V &= 1 \\
N &= 2 \\
R_2 &= 1
\end{align*}}

= Relevant partial chains

Time of Write Event

*{E (3, 10, 14.3)}

Where

Time of Write Event

When

Page Write Events

CE

CE$_2$

{\begin{align*}
= False \\
= True
\end{align*}}
The Correlated Page-Based Corruption Diagram Depicting the a3D Algorithm

Clock Time

When

Time of Write Event

Where

Page #

Page Write Events

= Relevant partial chains

\[ V = 1 \]
\[ N = 2 \]
\[ R_s = 1 \]
The Correlated Page-Based Corruption Diagram Depicting the a3D Algorithm

Clock Time

Where → Page #

Time of Write Event

Page Write Events

VE = False

VE = True

* E (3, 10, 14.3)

= Relevant partial chains

V = 1

N = 2

R = 1

= Relevant partial chains

Page #
The Correlated Page-Based Corruption Diagram Depicting the a3D Algorithm

Clock Time

When

Page Write Events

Time of Write Event

Where → Page #

Page Write Events

= Relevant partial chains

V = 1
N = 2
R_e = 1
Outline

• Information Accountability

• Reference Architecture & Execution Phases

• Forensic Analysis

• Refinements

• Enterprise Considerations
Very Recent Corruptions

\[ t_{FVF} = UTB \]

\[ I_N \]

\[ I_V \]

\[ t_{RVS} = LSB \]

\[ t_{FVF} \]

\[ NE_{11} \]

\[ NE_{10} \]

\[ NE_9 \]

\[ NE_8 \]

\[ CE \]

\[ V = 3 \]

\[ N = 1 \]

\[ R_s = 2 \]
Very Recent Corruptions

- In general $t_c > t_l$
- **Exception**: corruption affects currently executing transaction
• In general  \( t_c > t_l \)

• **Exception**: corruption affects currently executing transaction

• Introduce “envelope” of width \( I_{max\_tran} \)
Very Recent Corruptions (2)
Very Recent Corruptions

- Different solution: Exploit the Regret Interval
Different solution: Exploit the Regret Interval

The regret interval ($I_R$) is the minimum time interval before any adversary can reverse the change they made.
Different solution: Exploit the **Regret Interval**

- The regret interval \( (I_R) \) is the minimum time interval before any adversary can reverse the change they made.

- \( I_R \) is intrinsic to the semantics and social use of application. We have **no control** over it.
**Very Recent Corruptions**

- Different solution: Exploit the *Regret Interval*

- The regret interval \((I_R)\) is the minimum time interval before any adversary can reverse the change they made.

- \(I_R\) is intrinsic to the semantics and social use of application. We have *no control* over it.

- We use an estimate \(I_R^* \leq I_R\)
Very Recent Corruptions \(^{(2)}\)

- Different solution: Exploit the Regret Interval

- The regret interval \((I_R)\) is the minimum time interval before any adversary can reverse the change they made.

- \(I_R\) is intrinsic to the semantics and social use of application. We have no control over it.

- We use an estimate \(I_R^* \leq I_R\)

- No introactive corruptions: \(0 < I_N \leq I_V < I_R^* \leq I_R\)
Shredding
Shredding

- **Transaction time semantics** require that data are never physically deleted.
  - Performance overhead
  - Privacy and liability threat
Shredding

- **Transaction time semantics** require that data are never physically deleted.
  - Performance overhead
  - Privacy and **liability** threat

- **Retention period**: a sliding time frame $I_{RP}$
  - Determined by regulations & company policy
  - Record physically deleted after exiting $now - I_{RP}$
Shredding

• **Transaction time semantics** require that data are never physically deleted.
  – Performance overhead
  – Privacy and **liability** threat

• **Retention period**: a sliding time frame $I_{RP}$
  – Determined by regulations & company policy
  – Record physically deleted after exiting $now - I_{RP}$

• **Shredding** ensures **information restriction**.
  – **Breaks semantics** of information accountability
  – Reconcile shredding with tamper detection and forensic analysis?
Litigation Holds
Litigation Holds

- **Litigation holds** can be issued on the data for a duration of time as specified by a court.
Litigation Holds

- **Litigation holds** can be issued on the data for a duration of time as specified by a court.

- **Override** retention period regulations
Litigation Holds

- **Litigation holds** can be issued on the data for a duration of time as specified by a court.

- **Override** retention period regulations

- **Litigation holds** “restore” info accountability.
Litigation Holds

- **Litigation holds** can be issued on the data for a duration of time as specified by a court.

- **Override** retention period regulations

- **Litigation holds** “restore” info accountability.

- The capability to impose litigation holds prevents indiscriminate shredding and ensures **accountability**.
Outline

• Information Accountability
• Reference Architecture & Execution Phases
• Forensic Analysis
• Refinements
• Enterprise Considerations
Enterprise Architecture GUls
There are three GUIs:
- Chief Security Office (CSO)
- Database Administrator (DBA)
- Crime Scene Investigator (CSI)
Enterprise Architecture GUIs

• There are **three** GUIs:
  - Chief Security Office (CSO)
  - Database Administrator (DBA)
  - Crime Scene Investigator (CSI)

• Configure the security **policies** by
  - selecting a database to be **monitored**
  - setting the **security parameters**, e.g., $R_s, N, V, I_N$
Enterprise Architecture GUIs

- There are three GUIs:
  - Chief Security Office (CSO)
  - Database Administrator (DBA)
  - Crime Scene Investigator (CSI)

- Configure the security policies by
  - selecting a database to be monitored
  - setting the security parameters, e.g., $R_s, N, V, I_N$

- Calculate the forensic cost for normal processing and forensic analysis
Enterprise Architecture GUIs

- There are three GUIs:
  - Chief Security Office (CSO)
  - Database Administrator (DBA)
  - Crime Scene Investigator (CSI)

- Configure the security policies by
  - selecting a database to be monitored
  - setting the security parameters, e.g., $R_s$, $N$, $V$, $I_N$

- Calculate the forensic cost for normal processing and forensic analysis

- Create corruption diagrams
DBA: Database-Specific Settings

Detection Resolution Unit: 0 days 0 hrs 1 mins.
Forensic Algorithm: Monochromatic
Number of Resolution Units Between Notarizations: 1
Time between notarizations: 0 days 0 hrs 1 mins.
Number of Notarizations Between Validations: 1
Time between validations: 0 days 0 hrs 1 mins.
Cost Per Unit: $0.01

Predicted Cost:
Tampering Detection:
Per Day $: 0.02  Per Year $: 7.30
Forensic Analysis (Worst Case):
One Corruption $: 0.00  0 Corruptions $: 0.00

Start On: 7/22/2012 at 23:52
Save These Settings
DBA: Database-Specific Settings

Detection Resolution Unit: 0 days 0 hrs 1 mins.

Forensic Algorithm: Monochromatic

Number of Resolution Units Between Notarizations: 1

Time between notarizations: 0 days 0 hrs 1 mins.

Number of Notarizations Between Validations: 1

Time between validations: 0 days 0 hrs 1 mins.

Cost Per Unit: $0.01

Predicted Cost:
Tampering Detection:
- Per Day: $0.02
- Per Year: $7.30

Forensic Analysis (Worst Case):
- One Corruption: $0.00
- 0 Corruptions: $0.00

Start On: 7/22/2012 at 23:52

Save These Settings
DBA: Database-Specific Settings

Cost Per Unit: $0.01

Predicted Cost:
Tampering Detection:
Per Day: $0.02  Per Year: $7.30

Forensic Analysis (Worst Case):
One Corruption: $0.00  Corruptions: $0.00

Start On: 7/22/2012 at 23:52

Save These Settings
DBA: Database-Specific Settings

Settings

Detection Resolution Unit: 0 ▼ days 0 ▼ hrs 1 ▼ mins.
Forensic Algorithm: Monochromatic ▼
Number of Resolution Units Between Notarizations: 1 ▼
Time between notarizations: 0 ▼ days 0 ▼ hrs 1 ▼ mins.
Number of Notarizations Between Validations: 1 ▼
Time between validations: 0 ▼ days 0 ▼ hrs 1 ▼ mins.
Cost Per Unit: $0.01

Predicted Cost:
Tampering Detection:
Per Day $: 0.02 Per Year $: 7.30
Forensic Analysis (Worst Case):
One Corruption $: 0.00 0 ▼ Corruptions $: 0.00

Start On: 7 ▼ / 22 ▼ / 2012 ▼ at 23 ▼ : 52 ▼

Save These Settings
DBA: Database-Specific Settings

Detection Resolution Unit: 0 days 0 hrs 1 mins.

Forensic Algorithm: Monochromatic

Number of Resolution Units Between Notarizations: 1

Time between notarizations: 0 days 0 hrs 1 mins.

Number of Notarizations Between Validations: 1

Time between validations: 0 days 0 hrs 1 mins.

Cost Per Unit: $0.01

Predicted Cost:

Tampering Detection:
Per Day $: 0.02 Per Year $: 7.30

Forensic Analysis (Worst Case):
One Corruption $: 0.00 0 Corruptions $: 0.00

Start On: 7/22/2012 at 23:52

Save These Settings
DBA: Database-Specific Settings

### Settings

- **Detection Resolution Unit**: 0 days, 0 hrs, 1 mins.
- **Forensic Algorithm**: Monochromatic
- **Number of Resolution Units Between Notarizations**: 1
- **Time between notarizations**: 0 days, 0 hrs, 1 mins.
- **Number of Notarizations Between Validations**: 1
- **Time between validations**: 0 days, 0 hrs, 1 mins.
- **Cost Per Unit**: $0.01

### Predicted Cost:

- **Tampering Detection**:
  - Per Day $: 0.02
  - Per Year $: 7.30
- **Forensic Analysis (Worst Case)**:
  - One Corruption $: 0.00
  - 0 Corruptions $: 0.00

### Start On:

- **Date**: 7/22/2012
- **Time**: 23:52

**Save These Settings**
DBA: Database-Specific Settings

- Detection Resolution Unit: 0 days 0 hrs 1 mins.
- Forensic Algorithm: Monochromatic
- Number of Resolution Units Between Notarizations: 1
- Time between notarizations: 0 days 0 hrs 1 mins.
- Number of Notarizations Between Validations: 1
- Time between validations: 0 days 0 hrs 1 mins.
- Cost Per Unit: $0.01

Predicted Cost:
- Tampering Detection:
  - Per Day $: 0.02
  - Per Year $: 7.30
- Forensic Analysis (Worst Case):
  - One Corruption $: 0.00
  - Corruptions $: 0.00
- Start On: 7/22/2012 at 23:52

Save These Settings
### DBA: Database-Specific Settings

**File**
- acmedb(/home/tau/software/audit/auditdb)
  - CSI: Rick Snodgrass(rts@cs.arizona.edu)

**Settings**
- **Detection Resolution Unit:** 0 days 0 hrs 1 mins.
- **Forensic Algorithm:** Monochromatic
- **Number of Resolution Units Between Notarizations:** 1
- **Time between notarizations:** 0 days 0 hrs 1 mins.
- **Number of Notarizations Between Validations:** 1
- **Time between validations:** 0 days 0 hrs 1 mins.
- **Cost Per Unit:** $0.01

**Predicted Cost:**
- **Tampering Detection:**
  - Per Day $: 0.02
  - Per Year $: 7.30
- **Forensic Analysis (Worst Case):**
  - One Corruption $: 0.00
  - Corruptions $: 0.00

**Start On:** 7/22/2012 at 23:52

**Save These Settings**
DBA: Database-Specific Settings

Detection Resolution Unit: 0 days 0 hrs 1 mins.

Forensic Algorithm: Monochromatic

Number of Resolution Units Between Notarizations: 1

**Time between notarizations:** 0 days 0 hrs 1 mins.

Number of Notarizations Between Validations: 1

Time between validations: 0 days 0 hrs 1 mins.

Cost Per Unit: $0.01

Predicted Cost:
Tampering Detection:
Per Day: $0.02  Per Year: $7.30

Forensic Analysis (Worst Case):
One Corruption: $0.00 0 Corruptions: $0.00

Start On: 7/22/2012 at 23:52

Save These Settings
DBA: Database-Specific Settings

Detection Resolution Unit: 0 days 0 hrs 1 mins.

Forensic Algorithm: Monochromatic

Number of Resolution Units Between Notarizations: 1

Time between notarizations: 0 days 0 hrs 1 mins.

Number of Notarizations Between Validations: 1

Time between validations: 0 days 0 hrs 1 mins.

Cost Per Unit: $0.01

Predicted Cost:

Tampering Detection:
Per Day $: 0.02 Per Year $: 7.30

Forensic Analysis (Worst Case):
One Corruption $: 0.00 0 Corruptions $: 0.00

Start On: 7 / 22 / 2012 at 23:52

Save These Settings
DBA: Database-Specific Settings

Detection Resolution Unit: 0 days 0 hrs 1 mins.

Forensic Algorithm: Monochromatic

Number of Resolution Units Between Notarizations: 1

Time between notarizations: 0 days 0 hrs 1 mins.

Number of Notarizations Between Validations: 1

Time between validations: 0 days 0 hrs 1 mins.

Cost Per Unit: $0.01

Predicted Cost:
- Tampering Detection:
  - Per Day: $0.02
  - Per Year: $7.30
- Forensic Analysis (Worst Case):
  - One Corruption: $0.00
  - Corruptions: $0.00

Start On: 7/22/2012 at 23:52

Save These Settings
DBA: Database-Specific Settings

Detection Resolution Unit: 0 days 0 hrs 1 mins.
Forensic Algorithm: Monochromatic
Number of Resolution Units Between Notarizations: 1
Time between notarizations: 0 days 0 hrs 1 mins.
Number of Notarizations Between Validations: 1
Time between validations: 0 days 0 hrs 1 mins.
Cost Per Unit: $0.01

Predicted Cost:
Tampering Detection:
Per Day $: 0.02 Per Year $: 7.30
Forensic Analysis (Worst Case):
One Corruption $: 0.00 0 Corruptions $: 0.00

Start On: 7/22/2012 at 23:52

Save These Settings
DBA: Database-Specific Settings

**Detection Resolution Unit:** 0 days 0 hrs 1 mins.

**Forensic Algorithm:** Monochromatic

**Number of Resolution Units Between Notarizations:** 1

**Time between notarizations:** 0 days 0 hrs 1 mins.

**Number of Notarizations Between Validations:** 1

**Time between validations:** 0 days 0 hrs 1 mins.

**Cost Per Unit:** $0.01

**Predicted Cost:**

**Tampering Detection:**
- Per Day $: 0.02
- Per Year $: 7.30

**Forensic Analysis (Worst Case):**
- One Corruption $: 0.00
- Corruptions $: 0.00

**Start On:** 7/22/2012 at 23:52

**Save These Settings**
DBA: Database-Specific Settings

File
acmedb(/home/tau/software/audit/auditdb) ▼ CS: Rick Snodgrass(rts@cs.arizona.edu)

Settings
Detection Resolution Unit: ▼ 0 ▼ days ▼ 0 ▼ hrs ▼ 1 ▼ mins.

Forensic Algorithm: Monochromatic ▼

Number of Resolution Units Between Notarizations ▼ 1 ▼

Time between notarizations: ▼ 0 ▼ days ▼ 0 ▼ hrs ▼ 1 ▼ mins.

Number of Notarizations Between Validations ▼ 1 ▼

Time between validations: ▼ 0 ▼ days ▼ 0 ▼ hrs ▼ 1 ▼ mins.

Cost Per Unit: $ 0.01

Predicted Cost:
Tampering Detection:
Per Day $: 0.02 Per Year $: 7.30
Forensic Analysis (Worst Case):
One Corruption $: 0.00 0 ▼ Corruptions $: 0.00

Start On: ▼ 7 ▼ / ▼ 22 ▼ / ▼ 2012 ▼ at ▼ 23 ▼ : ▼ 52 ▼ ▼

Save These Settings
Information Accountability in the Cloud

Diagram:

- User Application
  - Replication Layer
    - DBMS
      - Database (Trans. Time)
- DragondDB Database
  - Notarizer
    - Hash value + notary ID
  - CSO GUI
  - DBA GUI
  - CSI GUI
- Validator
  - New hash value + notary ID
- EDNS

Connections:
- User Application to DBMS: transactions
- DBMS to DragondDB Database: current db state
- DragondDB Database to Notarizer: hash value + notary ID
- Notarizer to DragondDB Database: result
- DragondDB Database to Validator: new hash value + notary ID
- Validator to EDNS: result
Information Accountability in the Cloud
Information Accountability in the Cloud

[Diagram showing the flow of transactions and interactions between different components such as User Application, DBMS, DragoonDB Database, Notarizer, Validator, EDNS, and CSO GUI, DBA GUI, CSI GUI, with annotations for current db state and new hash value + notary ID.]
Information Accountability in the Cloud

- **Notarizer**: hash value + notary ID
- **DBA GUI**: current db state
- **CSI GUI**
- **CSO GUI**: result
- **Validator**: new hash value + notary ID
- **DragoonDB Database**: result
- **EDNS**: result

Overview:

- User Application
- DBMS
- Database (Trans. Time)
- Replication layer
- Transactions

A

B

Connections:

- User Application to DBMS
- DBMS to Database (Trans. Time)
- Notarizer to EDNS
- EDNS to Validator
- Validator to DragoonDB Database
- DragoonDB Database to CSO GUI, DBA GUI, CSI GUI
- Transactions to replication layer
- Transactions from User Application to DBMS
Information Accountability in the Cloud

A: User Application
  replication layer
  transactions
  DBMS
  Database (Trans. Time)

B: DragoonDB Database
   hash value
   notary ID
   new hash value + notary ID
   result

Notarizer
hash value
notary ID
result

Validator
notary ID
result

CSO GUI
DBA GUI
CSI GUI

EDNS
hash value
notary ID
new hash value + notary ID
result
Information Accountability in the Cloud

The new threat model may give rise to other temporal concepts.

Also holds for concurrency, replication, and distribution.
Summary

- Information Accountability
- Reference Architecture & Execution Phases
- Forensic Analysis
- Refinements
- Enterprise Considerations
Summary (2)

- Need to be able to *capture history*.
- Need to be able to *revisit history*.
- Need a *trusted witness* or at least *consensus opinion* to provide *continuous assurance over time*.
The Challenge
The Challenge

As we have seen time arises naturally in many aspects of database information accountability (and in many guises).
The Challenge

As we have seen time arises naturally in many aspects of database information accountability (and in many guises).

What is the deeper structure of the fundamental connection between temporal databases and information security?
Thank You!

Questions?