# Forensic Analysis of Database Tampering

Kyriacos Pavlou and Richard T. Snodgrass

Computer Science Department
The University of Arizona

# Introduction

**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*.

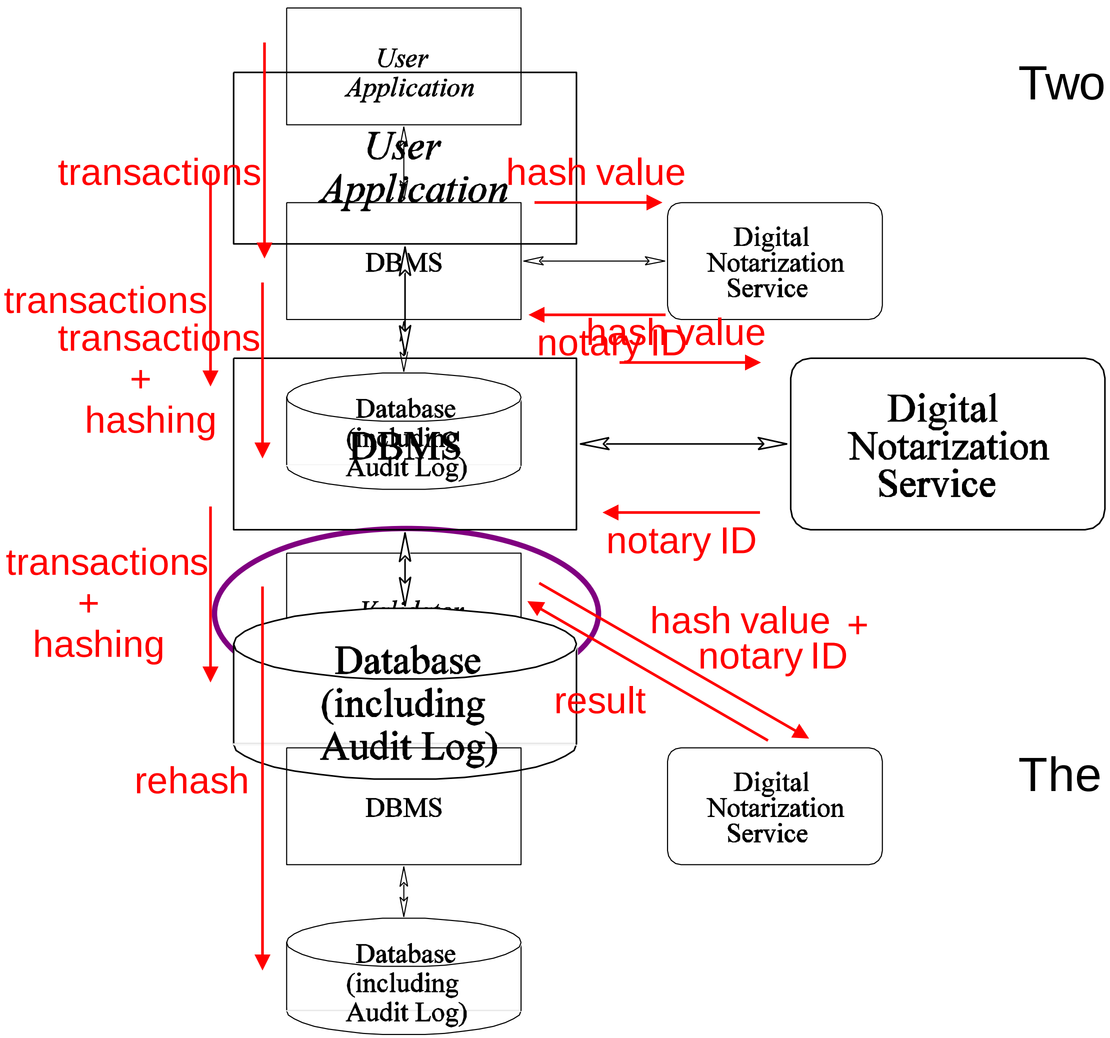
- Snodgrass et al. [*VLDB04*] showed how to detect database tampering. Approach: Hash using a cryptographically strong hash function, *notarize* data manipulated by transactions and periodically *validate*.

- 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

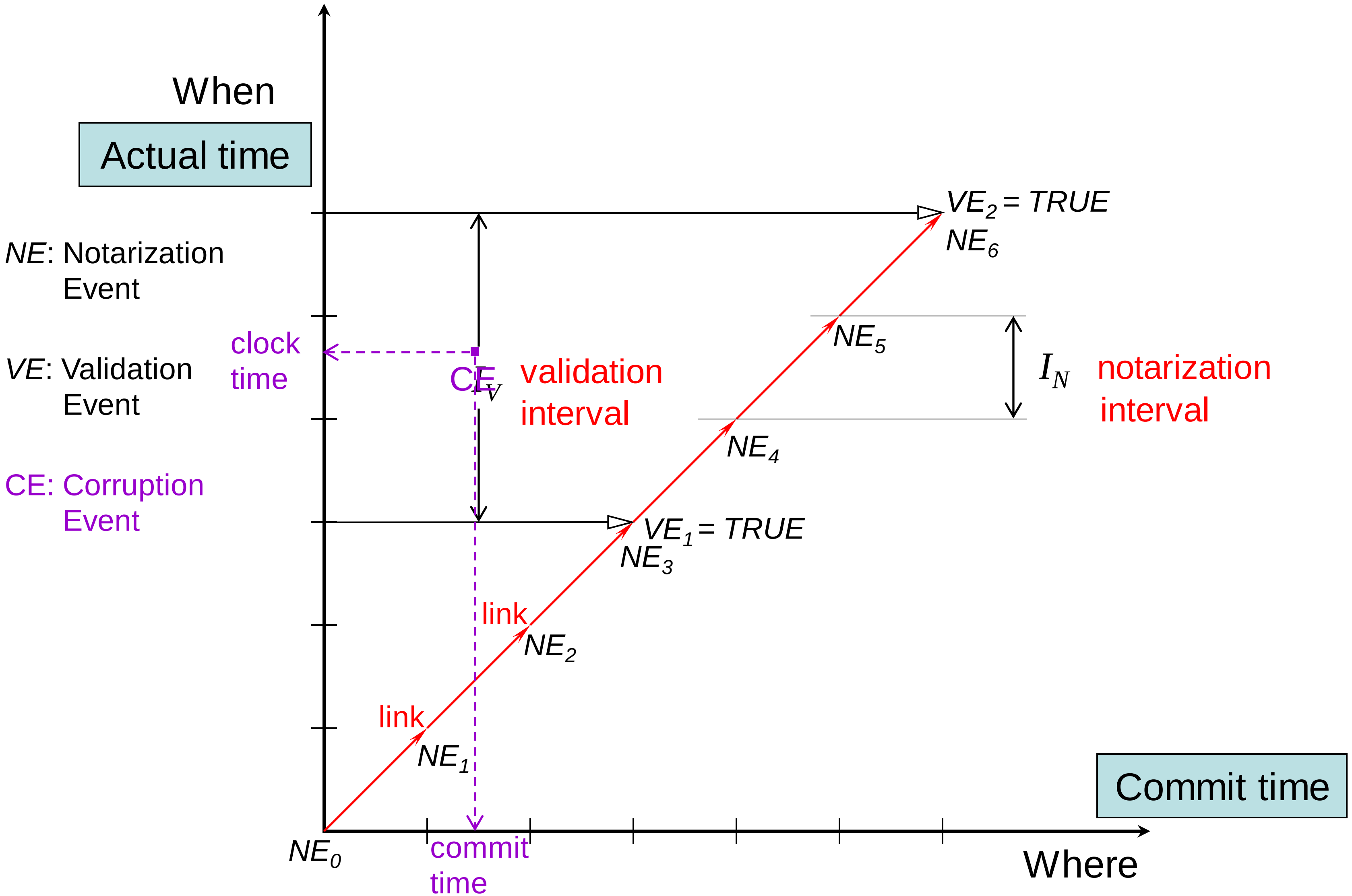



Two phases:

- *Normal Processing*
- *Validation*

The validation result is a **single bit**.

# The Corruption Diagram

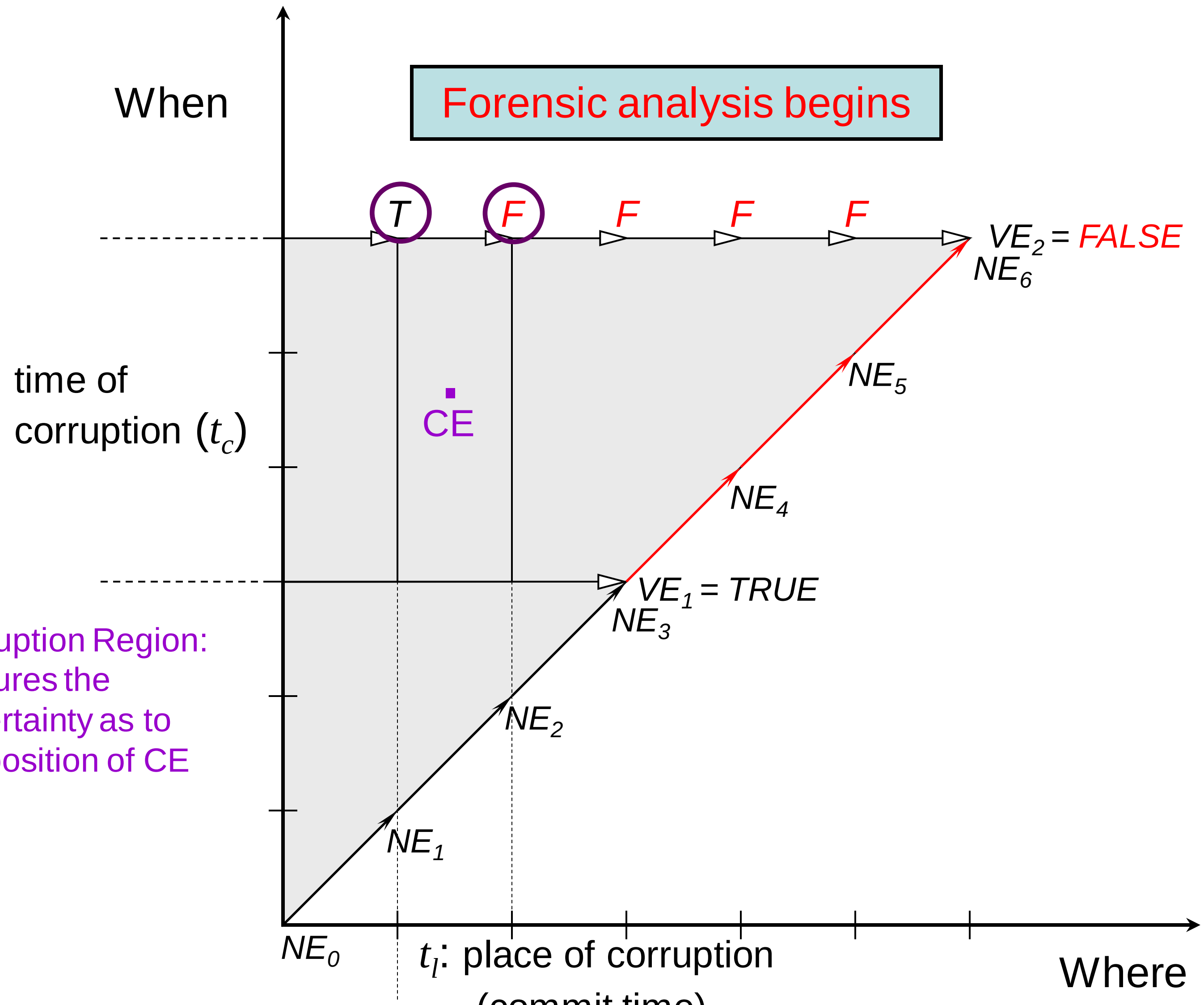


# Forensic Analysis

- 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




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

# 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.

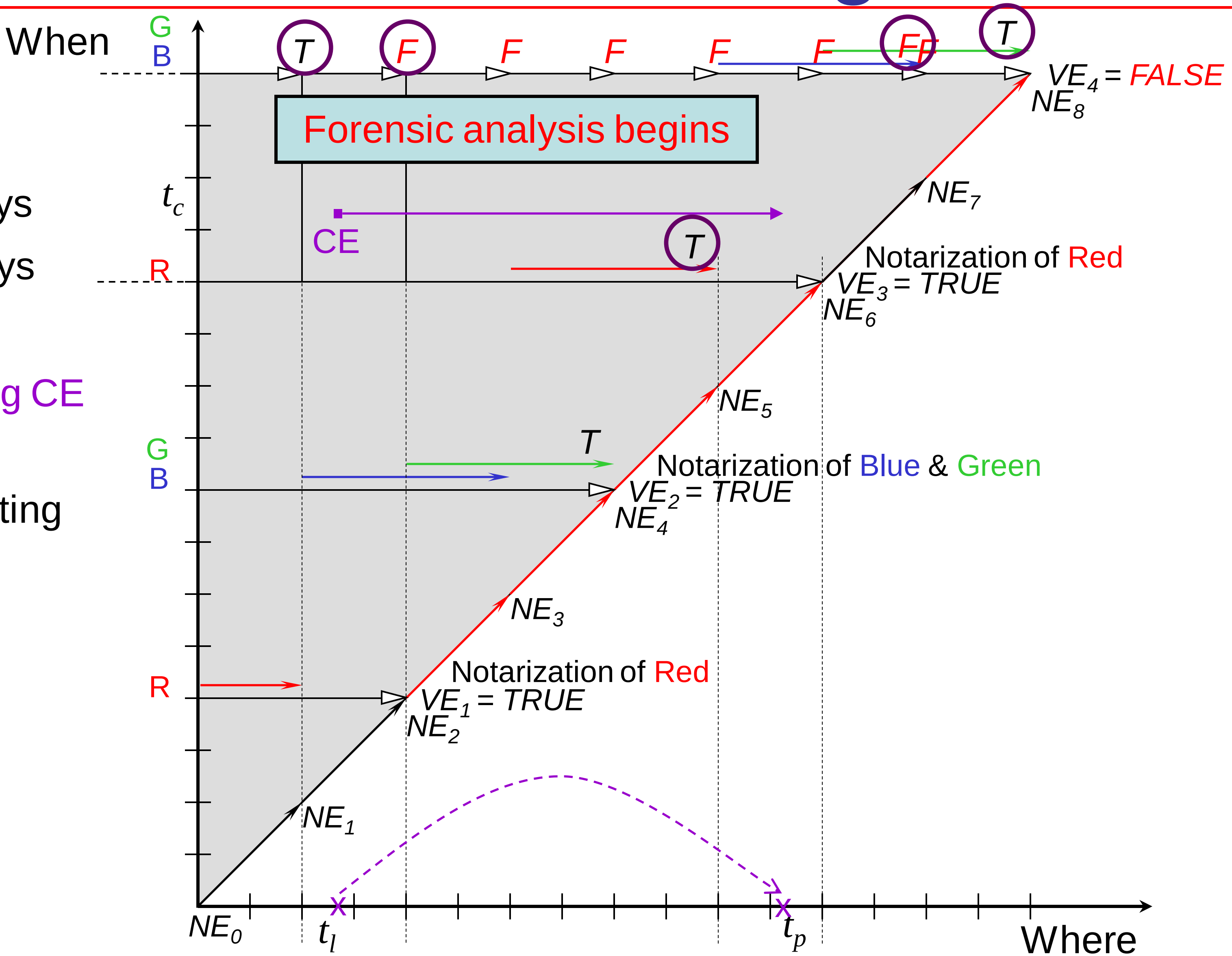
# The RGB Forensic Algorithm

$I_V$ = 4 days

$I_N$ = 2 days

Postdating CE

$t_p$: postdating time



# 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

When

$I_V$ = 4 days

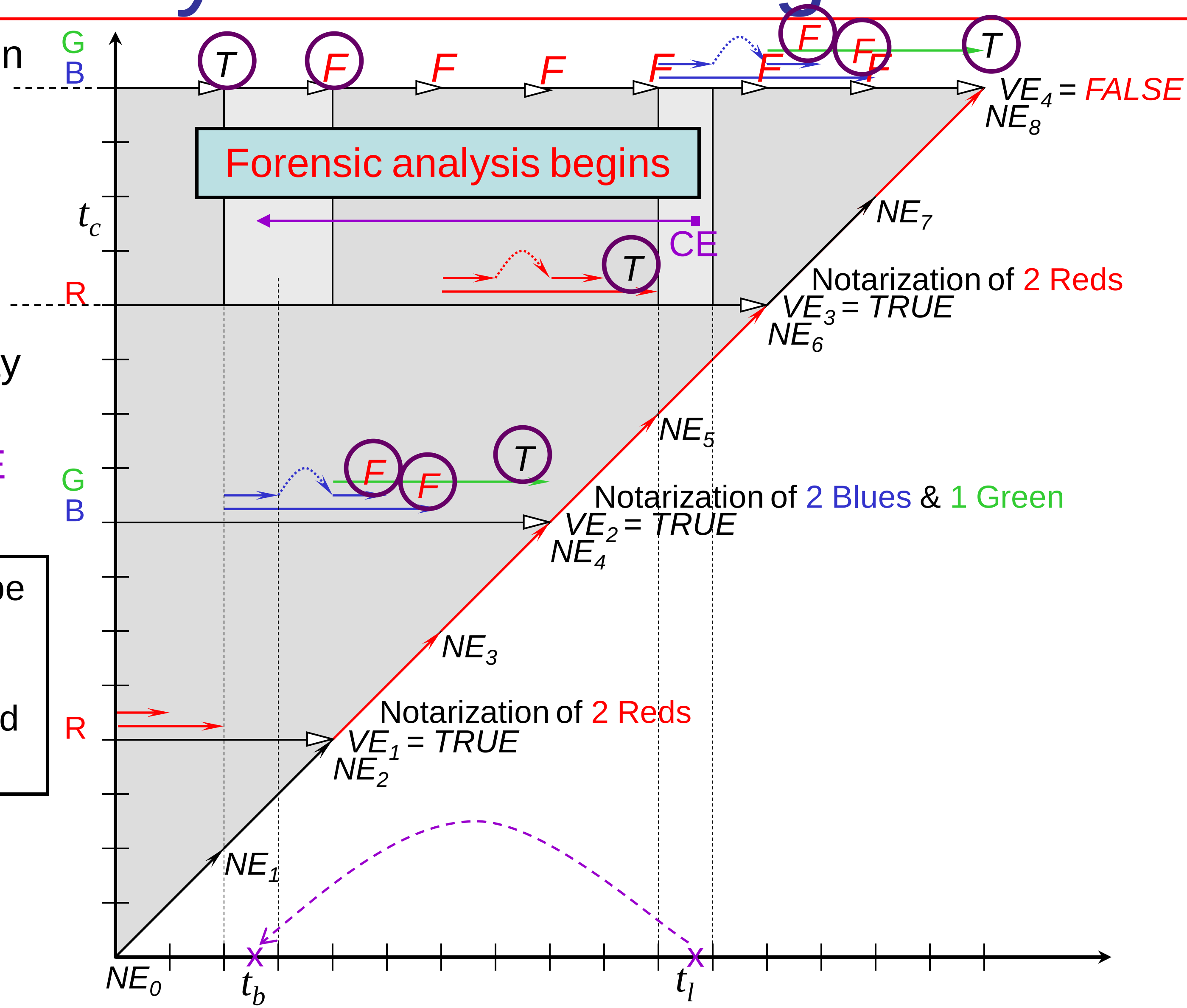
$I_N$ = 2 days

Desired = 1 day

Backdating CE

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

$t_b$: backdating time



# Forensic Strength

Components:

- Work of forensic analysis
- Region-area of *CE*
- Width of postdating / backdating uncertainty

Inverse Forensic Strength:

$$IFS(D, I_N, V) = (NumNotarizes(D, I_N, V) + ForensicAnalysis(D, I_N, V)) \cdot RegionArea(I_N, V) \cdot UncertaintyWidth(D, I_N)$$

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: $O(V \cdot D \cdot I_N^2)$
- Polychromatic: $O((V + lg\ I_N) \cdot D)$

We assume that $D >> I_N$.

# Future Work

- Develop a stronger lower bound for this problem.
- Accommodate multi-locus and complex *CE*s.
- Differentiate postdating and backdating *CE*s.
- 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 *CE*s, termed the corruption diagram.
- We have designed three forensic algorithms.
  - Monochromatic
  - RGB
  - Polychromatic