Securing History: Privacy and Accountability in Database Systems

Gerome Miklau

(Joint work with Brian Levine & Patrick Stahlberg)

University of Massachusetts, Amherst

History

a record of past data and operations performed on a system.

History

a record of past data and operations performed on a system.

- Arguments **for** preserving history
  - Protection against loss
  - History is useful: accountability
  - Storage is cheap

History

a record of past data and operations performed on a system.

- Arguments **for** preserving history
  - Protection against loss
  - History is useful: accountability
  - Storage is cheap
- Arguments **against** preserving history
  - Threats to privacy and confidentiality
  - Deletion required for compliance with regulation
  - Increasingly, data destruction has real value!

Vision: securing history

- Balance **privacy** and **accountability**
    - Central issue: how and when historical data is retained in systems, who can recover and analyze it.
- For privacy
    - “memory-less” systems and applications
- For accountability
    - preserve needed history efficiently, permit analysis, protect

Plan for securing history in a DBMS

Step 1 — **Forensic analysis** of database systems

Step 2 — Build **transparency** into database systems

Step 3 — Build **accountability** into database systems

Securing history in a DBMS

Step 1 — **Forensic analysis** of database systems

Step 2 — Build **transparency** into database systems

Step 3 — Build **accountability** into database systems

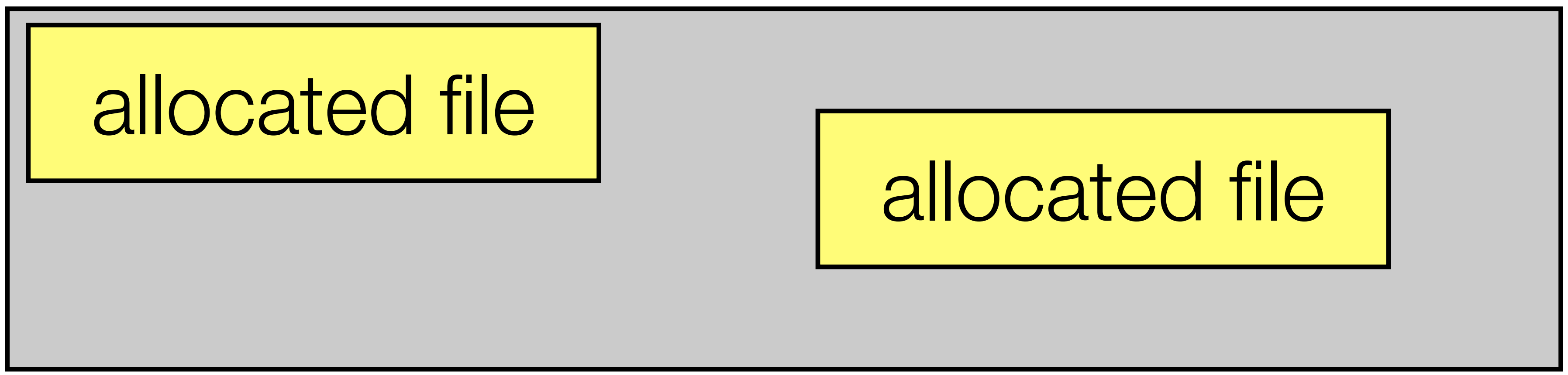
Computer forensics

- Analysis of system state to validate hypotheses about past activities.
- Threat model
  - Investigator has uncontrolled access to disk
  - Same capabilities as privileged insider or hacker

- What does the disk image of DBMS reveal about history?
  - How much expired data is retained?
  - How long does it persist?

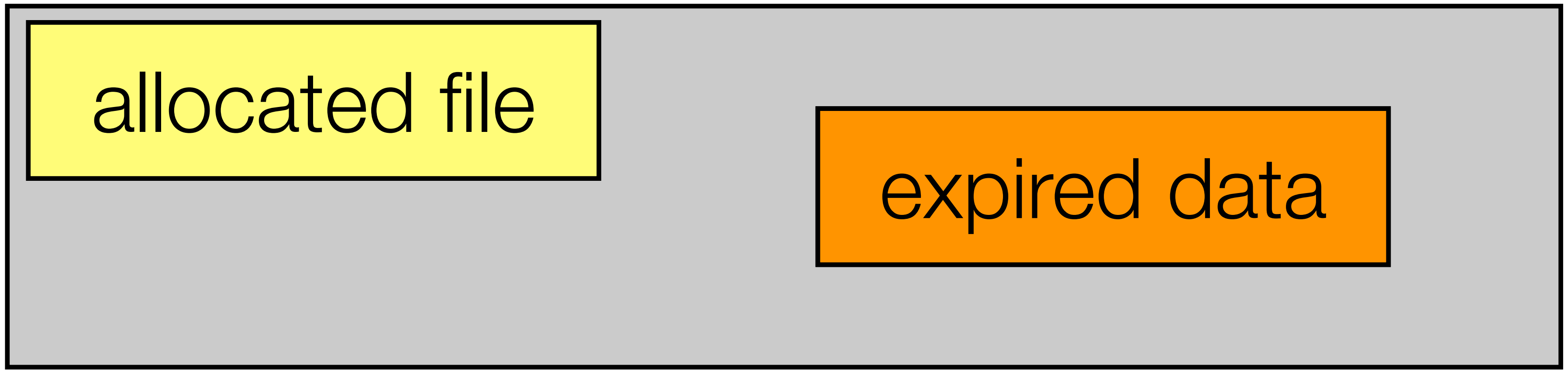
Slack data

- File system slack



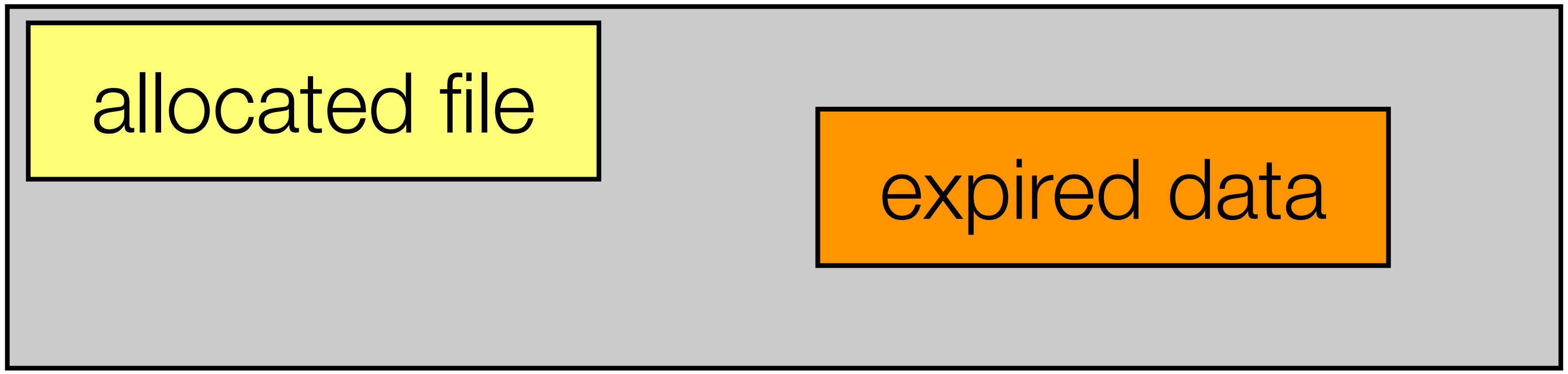
Slack data

- File system slack

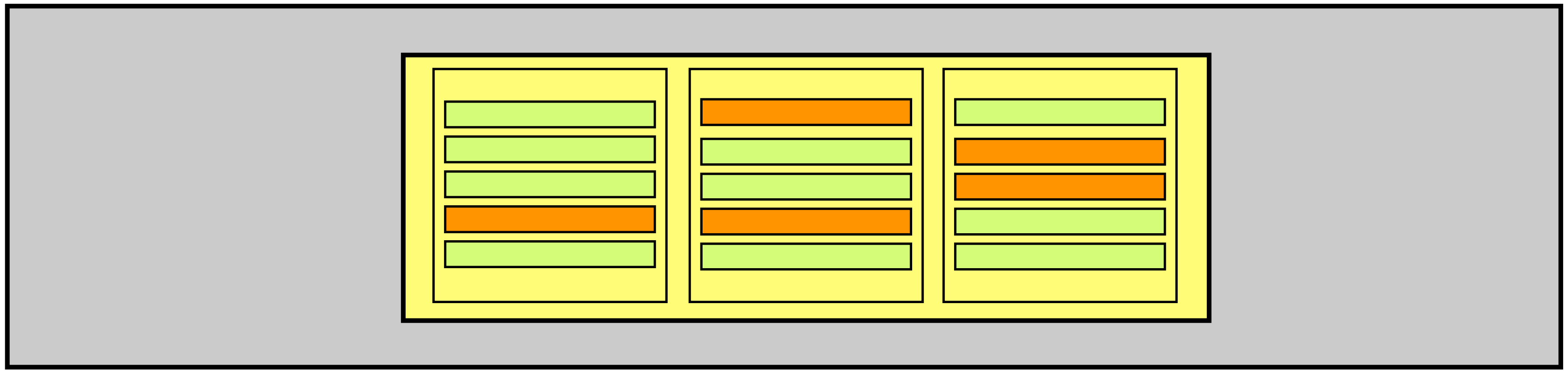


Slack data

- File system slack

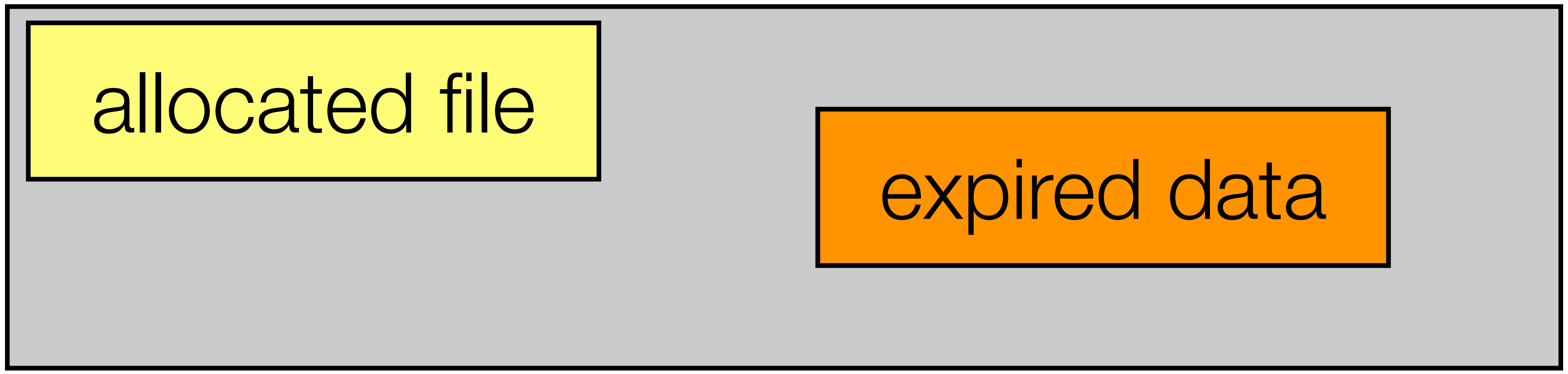


- Database slack

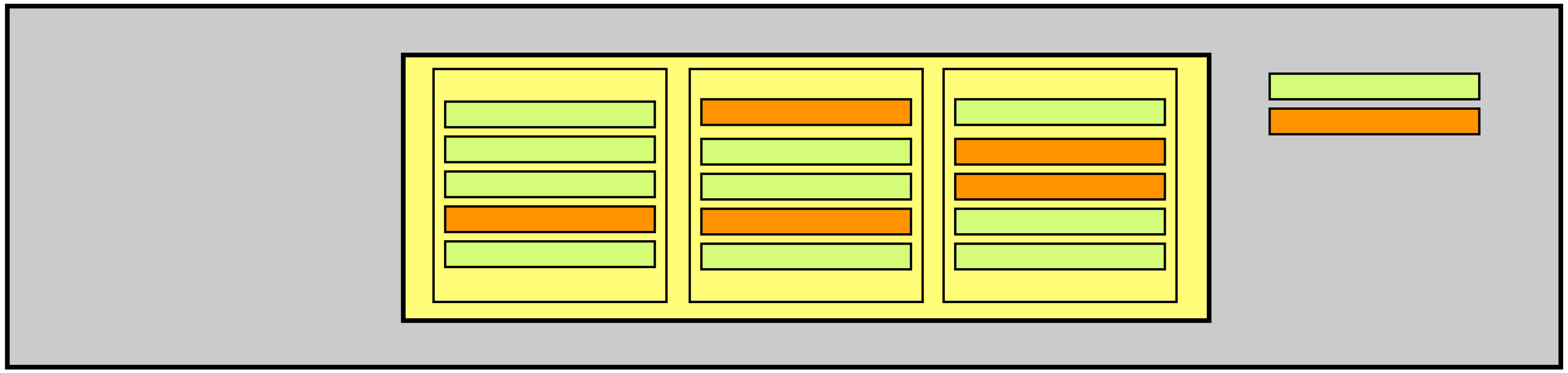


Slack data

- File system slack



- Database slack



Forensic analysis of DBMS

Forensic analysis of DBMS

- **Table storage**
  - deletion is insecure (MySQL, Postgres, DB2, SQLite)
  - database and file system slack data generated in proportion to
    - workload, vacuum, clustering.

Forensic analysis of DBMS

- **Table storage**
  - deletion is insecure (MySQL, Postgres, DB2, SQLite)
  - database and file system slack data generated in proportion to
    - workload, vacuum, clustering.
- **Transaction log**
  - no bounds on retention

Forensic analysis of DBMS

- **Table storage**
  - deletion is insecure (MySQL, Postgres, DB2, SQLite)
  - database and file system slack data generated in proportion to
    - workload, vacuum, clustering.
- **Transaction log**
  - no bounds on retention
- **Temporary relations** remain as file system slack.

Forensic analysis of DBMS

- **Table storage**
  - deletion is insecure (MySQL, Postgres, DB2, SQLite)
  - database and file system slack data generated in proportion to
    - workload, vacuum, clustering.
- **Transaction log**
  - no bounds on retention
- **Temporary relations** remain as file system slack.
- **Indexes** may reveal history of operations.

Securing history in a DBMS

Step 1	**Forensic analysis** of database systems
Step 2	Build **transparency** into database systems
Step 3	Build **accountability** into database systems

Transparent systems

Interfaces must reliably represent system internals.

Complete deletion

- Deleted data must be destroyed, including copies and derived versions.

Purposeful retention

- Data retained after deletion must have a legitimate purpose, and data should be removed once that purpose is no longer valid.

Bounded lifetime

- The system should provide users with clear, accurate bounds on the persistence of data in the system.

Secure deletion in DBMS

Secure deletion in DBMS

- Two basic strategies for secure deletion:
  - overwrite data with zeroes
  - store data in encrypted form, delete by disposing of keys.

Secure deletion in DBMS

- Two basic strategies for secure deletion:
  - overwrite data with zeroes
  - store data in encrypted form, delete by disposing of keys.
- For table storage:
  - pages are read and written often
  - prefer secure deletion and vacuum using overwriting

Secure deletion in DBMS

- Two basic strategies for secure deletion:
  - overwrite data with zeroes
  - store data in encrypted form, delete by disposing of keys.
- For table storage:
  - pages are read and written often
  - prefer secure deletion and vacuum using overwriting
- For transaction log:
  - sequential writes, easily identifiable point of expiry
  - use encryption with key disposal

Securing history in a DBMS

Step 1 — **Forensic analysis** of database systems

Step 2 — Build **transparency** into database systems

Step 3 — Build **accountability** into database systems

Accountability

Who did what to the database, and when?

- Goals
  - Collection, Analysis, Protection
  - “Security provenance”
- Existing capabilities
  - Logs and backups
  - Persistence in databases
    - Postgres, temporal DBs, transaction-time DBs

Accountability challenges

- Integrating and querying historical data
- Accounting for “reads”
- Protecting history
  - Access control model for persistent databases
  - Redaction and expunction operations

Conclusion

- History should be a “first-class” part of a DBMS
- The safe, accurate configuration of the system’s historical memory allows needed balance between **privacy** and **accountability.**
- Transparency requirements:
    - Interface should faithfully represent stored contents.
- Accountability techniques:
    - Collection, integration, protection

Questions?

Does encryption solve forensic threats?

- Encrypted file system:
  - protects historical remnants -- does not destroy data.
  - performance penalty, key manangement
  - in some settings, users/stakeholders cannot choose whether system provides encryption.
- Overall,
  - Encryption has an important role to play, but must be used judiciously.
  - Encryption for protection, destruction should be distinguished.