Auditable Register Emulations

Vinicius V. Cogo
vielmo@lasige.di.fc.ul.pt

Alysson Bessani
http://www.di.fc.ul.pt/~bessani/
bessani@di.fc.ul.pt

Navtalk – May 29, 2019
Private infrastructures
→
Public multi-tenant

**Pros**
- Cost-effectiveness
- High scalability
- Ease of use
...

**Cons**
- Outages
- Security incidents
- Privacy risks
...

Private infrastructures
→
Public multi-tenant
Secure storage systems ... disperse data in multiple objects

E.g., DepSky, SCFS, CHARON, Vawlt
... where **blocks** from a portion of these objects is enough to recover the original **data**
Increasing severity of

Tightening of regulations

demand for further advances on secure storage

(e.g., forensics to identify suspects and measure the damage)
Our goal: Audit who has effectively read data in secure storage systems?
Outline

Preliminaries

Auditable Register Emulations

Resilience Lower Bounds

Alternative Models

Conclusions
Preliminaries

- **System model:**
  - Shared memory
  - Client processes (writers, readers, and auditors)
  - $n$ storage objects

- **Fault model:**
  - Writers and auditors are trusted and can crash
  - Readers and (up to $f$) storage objects can be malicious

- **R/W registers:**
  - Store a value $\nu$
  - Low-level rw-write and rw-read
Preliminaries

• **System model:**
  - Shared memory
  - Client processes (writers, readers, and auditors)
  - $n$ storage objects

• **Fault model:**
  - **Writers** and **auditors** are trusted and can crash
  - **Readers** and (up to $f$) storage objects can be **malicious**

• **R/W registers:**
  - Store a value $\nu$
  - Low-level **rw-write** and **rw-read**
Preliminaries

- **Emulated registers:**
  - Store a value \( \nu \)
  - High-level \( s\text{-write} \) and \( s\text{-read} \)
  - MWMR (Multi-writer multi-reader)
  - **Safe semantics** (read returns value from most recent write---if no concurrent write)
  - **Wait-free liveness** (\( op \) returns in a finite number of steps)
  - **Fast reads** (\( s\text{-read} \) completes in a single communication round-trip)
Preliminaries

- **Information dispersal**: 
  - *s-write*:
  - *s-read*:

  Common assumptions:
  - [Info dispersal] $\tau > f$
  - [Available f-threshold quorums] $q = n - f$
  - [BFT info dispersal] $n \geq \tau + 2f$

Corresponding to the assumptions:
- $|G_1| = f$
- $|G_2| = f$
- $|G_3| = 1$
- $|G_4| = f$

Example:

- $\tau = f + 1$
- $n = 3f + 1$
**Preliminaries**

- **Information dispersal:**
  - _s-write:_ Convert value _v_ into _n_ blocks _b_v1 ... b_vk ... b_vn_ + write one block (_b_v_) per object
  
  - _s-read:_
    
Common assumptions:

- [Info dispersal] _τ > f_ (i.e., malicious objects cannot create arbitrary values)
- [Available f-threshold quorums] _q = n – f_ (i.e., every op waits responses from at least _q_ objects)
- [BFT info dispersal] _n ≥ τ + 2f_ (e.g., _τ_ correct + _f_ malicious + _f_ stale)

**Example**

- _s-write(x)_
  
| _σ_1 | _G_1|=f | _G_2|=f | _G_3|=1 | _G_4|=f |
|-------|--------|--------|--------|--------|
| time  | _v_b_  | _m_b_  | _v_b_  | _v_b_  |

_time ack

- _τ = f + 1_  
- _n = 3f + 1_
Preliminaries

- **Information dispersal**:  
  - **s-write**: Convert value $\nu$ into $n$ blocks $b_{v1} \ldots b_{vk} \ldots b_{vn}$  
    + write one block ($b_v$) per object  
  - **s-read**: $\tau$ correct blocks (from different objects) recovers value $\nu$  
    i.e., an “effective read”

- **Common assumptions**
  - [Info dispersal] $\tau > f$ (i.e., malicious objects cannot create arbitrary values)
  - [Available f-threshold quorums] $q = n - f$ (i.e., every op waits responses from at least $q$ objects)
  - [BFT info dispersal] $n \geq \tau + 2f$ (e.g., $\tau$ correct + $f$ malicious + $f$ stale)
Auditable Register Emulations

- **a-write:**

- **a-read:**

- **a-audit:**

\[
\tau = f + 1 \\
n = 3f + 1 \\
L = 1
\]
- **a-write**: write blocks in a quorum of objects

- **a-read**:

- **a-audit**:

Example

\[
\begin{align*}
\sigma_1 &\quad |G_1|=f \quad |G_2|=f \quad |G_3|=1 \quad |G_4|=f \\
\text{block} &\quad \log &\quad \text{malicious} &\quad \text{block} &\quad \log &\quad \text{block} &\quad \log \\
v_b &\quad \{\} &\quad m_b &\quad \{\} &\quad v_b &\quad \{\} &\quad v_b &\quad \{\} \\
x_b &\quad \{\} &\quad m_b &\quad \{\} &\quad x_b &\quad \{\} &\quad v_b &\quad \{\}
\end{align*}
\]

\[\tau = f + 1 \quad n = 3f + 1 \quad \ell = 1\]
Auditable Register Emulations

- **a-write**: write blocks in a quorum of objects
- **a-read**: read blocks from a quorum of objects
  + log the read operation on each object
    (e.g., record $\nu_r$ with reader $r$ and block of value $\nu$ of the block)
- **a-audit**:

### Example

$$\sigma_1$$

<table>
<thead>
<tr>
<th>block</th>
<th>log</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_b$</td>
<td>{}</td>
</tr>
<tr>
<td>$x_b$</td>
<td>{}</td>
</tr>
</tbody>
</table>

$$|G_1| = f$$

$$|G_2| = f$$

$$|G_3| = 1$$

$$|G_4| = f$$

$$\tau = f + 1$$

$$n = 3f + 1$$

$$\ell = 1$$
Auditable Register Emulations

- **a-write**: write blocks in a quorum of objects

- **a-read**: read blocks from a quorum of objects
  + log the read operation on each object
    (e.g., record \( \nu_r \) with reader \( r \) and block of value \( \nu \) of the block)

- **a-audit**: get logs from a quorum of objects
  + create and return evidences of effective reads
    (e.g., evidence \( \varepsilon_{r,v} = \ell \) records \( \nu_r \))

Example

- **a-write**: write blocks in a quorum of objects
  - \( |G_1| = f \)
  - \( |G_2| = f \)
  - \( |G_3| = 1 \)
  - \( |G_4| = f \)

- **a-read**: read blocks from a quorum of objects
  - \( |G_1| = f \)
  - \( |G_2| = f \)
  - \( |G_3| = 1 \)
  - \( |G_4| = f \)

- **a-audit**: get logs from a quorum of objects
  - \( |G_1| = f \)
  - \( |G_2| = f \)
  - \( |G_3| = 1 \)
  - \( |G_4| = f \)
• How many records do we need to create an evidence? \( l \)

• How much is \( l \)? 1, 2, \( \tau \), or \( n \)?
Auditable Register Emulations

• How many records do we need to create an evidence? \( \ell \)

• How much is \( \ell \)? 1, 2, \( \tau \), or \( n \)?

  ✓ Any audit quorum receives at least \( \tau - 2f \) correct records of every effective read

  ✓ So, the required number of records (\( \ell \)) to produce an evidence is...

\[
\ell \leq \tau - 2f
\]
Auditable Register Emulations

• Guarantees from \textit{a-audit}

1. Completeness:
   • Report every effective read (value \( v \), reader \( r \))
   • Protect the system from readers obtaining data undetectably

2. Accuracy:
   • Do not report actions that have never happened
   • Protect correct readers from malicious objects incriminating them

   ◆ \textbf{Weak accuracy}: Never report a correct reader that has never \textbf{invoked an a-read} before the audit
   ◆ \textbf{Strong accuracy}: Never report a correct reader that has never \textbf{effectively read a value \( v \)} before the audit

Weak auditability = completeness + weak accuracy
Strong auditability = completeness + strong accuracy
Outline

Preliminaries
Auditable Register Emulations
Resilience Lower Bounds
Alternative Models
Conclusions
Completeness

- Impossible with $\tau \leq 2f$
- Malicious reader effectively reads $\nu$
- $a$-audit cannot produce an evidence if it finds no record

Example:

- $\sigma_1$
- $|G_1| = f$
- $|G_2| = f$
- $|G_3| = f$
- $|G_4| = f$

- $\tau = 2f$
- $n = 4f$
- $\ell = \tau - 2f$

 ✓ Our algorithm satisfies completeness with $\tau \geq 2f + 1$
Weak Accuracy

- Impossible with $\ell \leq f$
- No effective read
- $a$-audit receives $f$ records of a non-existent read

 ✓ Our algorithm satisfies weak accuracy with $\ell \geq f + 1$
Weak Auditability (Completeness + weak accuracy)

• Impossible with $\tau \leq 3f$

• $a$-audit receive $f$ records both for non-existent and for existent read

✓ Our algorithm satisfies weak auditability with $\tau \geq 3f + 1$
Strong Accuracy

- Impossible with $\ell < \tau + f$
- Incomplete write and no effective read
- Malicious do not deliver block but log the read
- $a$-audit receives $\tau + f - 1$ records of a non-existent read

✓ Our algorithm satisfies strong accuracy with $\ell \geq \tau + f$
Strong Auditability (Completeness + strong accuracy)

- Impossible

- \( r_1 \) = effective read, \( r_2 \) = non-effective read
- \( a\)-audit receives \((\tau + f - 1)\) records for \( r_2 \) and \((\tau - f - 1)\) for \( r_1 \)

\( \tau \geq 2f + 1 \)
\( n \geq 4f + 1 \)

\( \checkmark \) Impossible to define a single \( \ell \) for a property w/o violating the other
Total Ordering Operations

- Serializing operations with total order broadcast
- Every write complete in a quorum before the invoke of reads
- **Worst case**: $f$ stale + $f$ malicious

- Totally ordering operations in our model allows our algorithm to satisfy strong **accuracy** with $l \geq 2f + 1$
- Totally ordering operations in our model allows our algorithm to satisfy strong **auditability** with $\tau \geq 4f + 1$
Non-fast Reads

- Non-fast read = more than one communication round. E.g. DepSky-CA:
  1. read only metadata (which is the most up-to-date value?)
  2. read the blocks for that value only

- All correct objects from $Q_f$ log the same record

- **Worst case:** $f$ malicious logs for another value

 ✓ Applying our algorithm to DepSky-CA protocol satisfies strong **accuracy** with $\ell \geq f + 1$

 ✓ Applying our algorithm to DepSky-CA protocol satisfies strong **auditability** with $\tau \geq 3f + 1$
Outline

Preliminaries
Auditable Register Emulations
Resilience Lower Bounds
Alternative Models

Conclusions
Conclusions

• “Who has effectively read data?” in secure storage systems
• Auditable register
• Resilience lower bounds:
  - Auditability is impossible if $\tau \leq 2f$ (e.g., most solutions use $\tau \geq f+1$)
  - $\tau \geq 3f + 1$ for weak accuracy (and weak auditability)
  - $\ell \geq \tau + f$ for strong accuracy
  - Strong auditability is impossible
• Alternative models satisfy strong auditability:
  - Total ordering operations ($\tau \geq 4f + 1$)
  - Some non-fast reads ($\tau \geq 3f + 1$)---e.g., DepSky-CA
• https://arxiv.org/abs/1905.08637
Thank you!

Vinicius V. Cogo
vielmo@lasige.di.fc.ul.pt

Alysson Bessani
http://www.di.fc.ul.pt/~bessani/
bessani@di.fc.ul.pt

Navtalk – May 29, 2019
$l \leq \tau - 2f$