Securing logs in operation-based collaborative editing

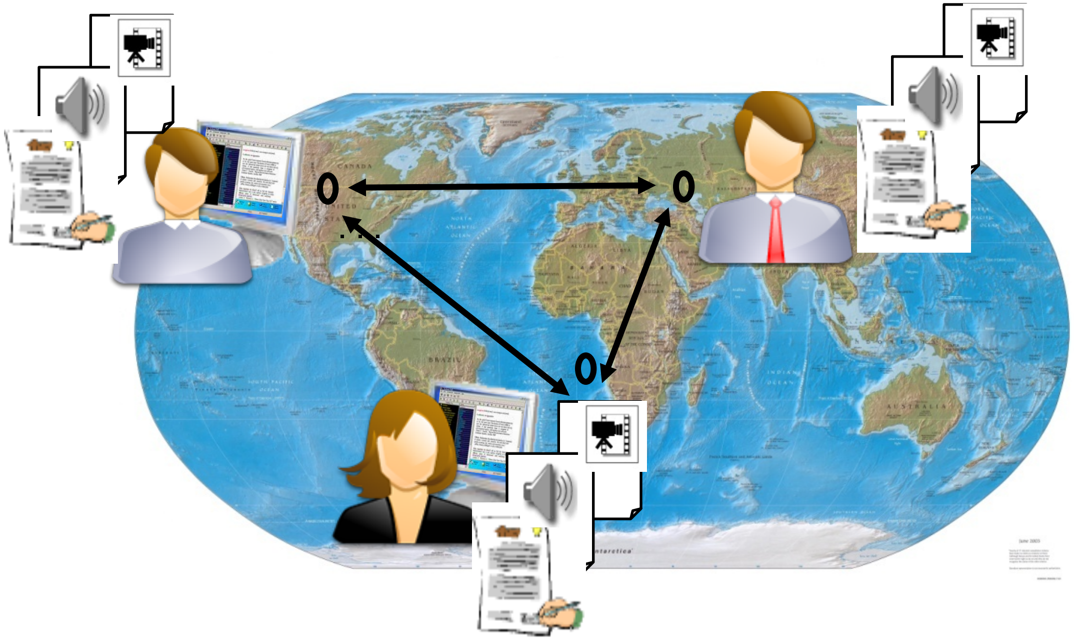
Hien Thi Thu Truong[1], Claudia-Lavinia Ignat[1], Pascal Molli[2]

[1] INRIA Nancy-Grand Est, France

[2] University of Nantes, France

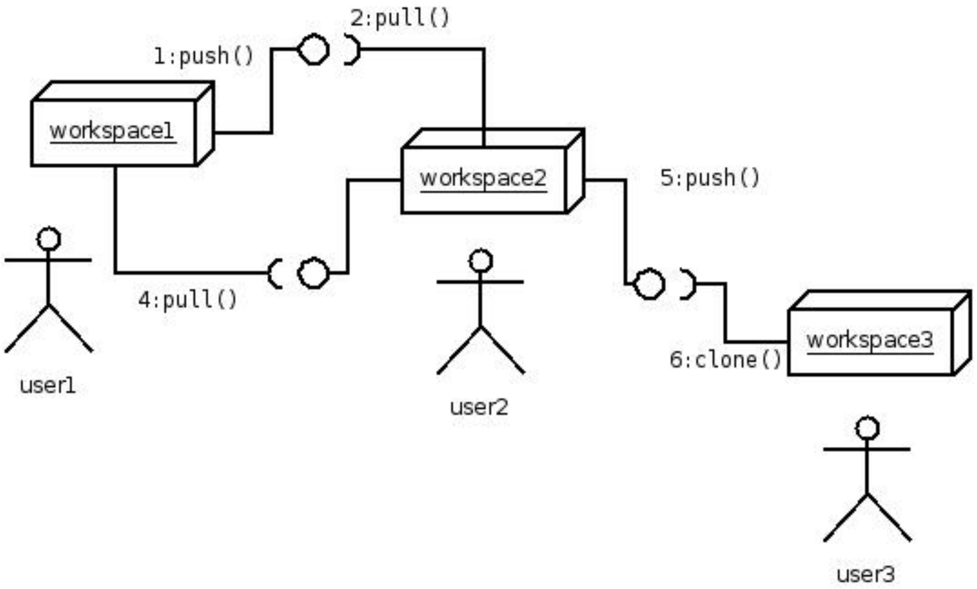
ignatcla@loria.fr

Context



- Collaborative editors: GoogleDocs, Wikis, version control systems

Push-Pull-Clone Collaboration model



- Distributed version control systems: Git, Darcs, Mercurial

Optimistic replication

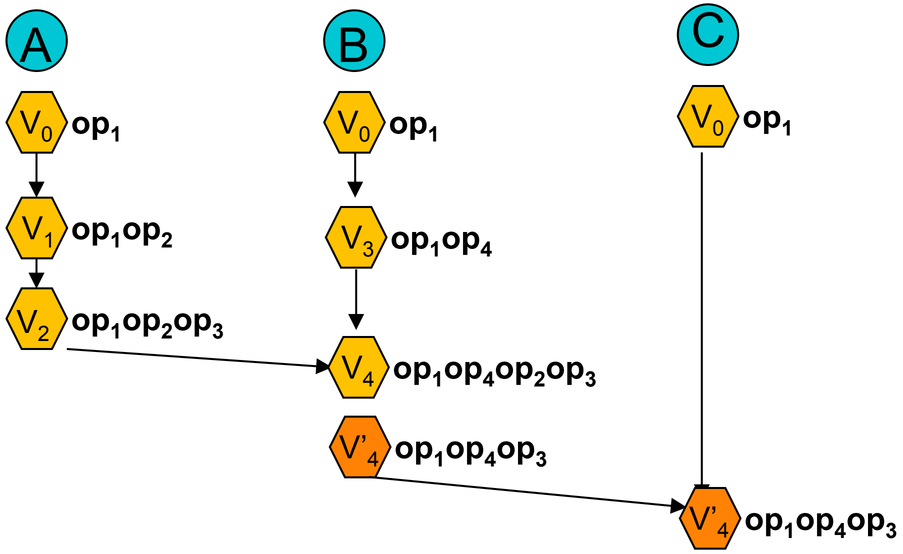
- State-based
  - No change log
  - Active Directory in Windows Server, Coda
- Operation-based
  - Change log
  - Used when cost to transfer state is high
  - Operation semantics
  - Bayou, GoogleDocs
- In this work operation-based optimistic replication

Merging algorithms

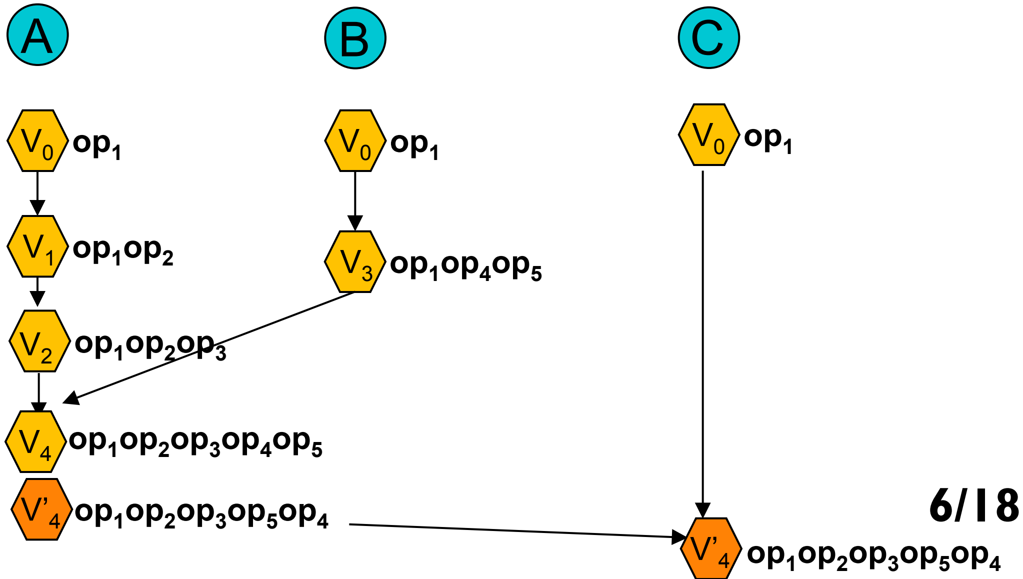
- Operational transformation (OT)
  - Transforms non-commuting operations to make them commute
  - Genericity
  - Limited scalability (state vectors)
- Commutative replicated data types (CRDT)
  - Designs operations to be commutative from the start
  - Document = linear sequence of elements
    - Each element has a unique identifier for the lifetime of the document
    - Total order of identifiers consistent with element order
    - forall M,P: M<P => exists N: M<N<P
  - Scalability
  - Storage cost of identifiers

Securing logs in operation-based collaboration

Deletion of operations



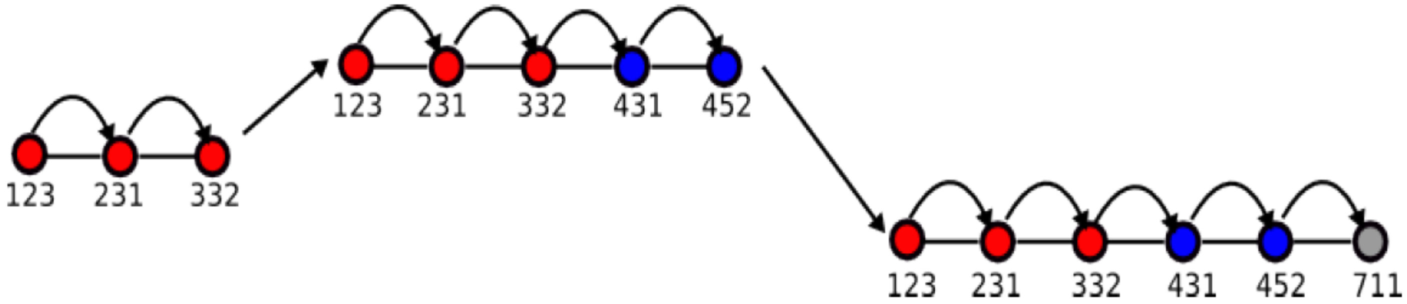
Changing order of operations



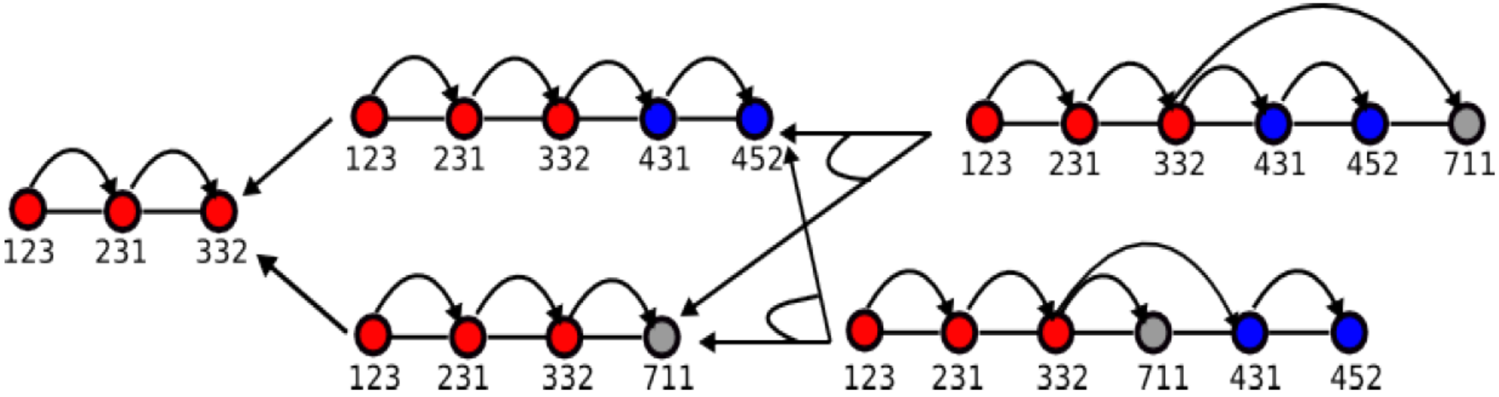
Securing logs in operation-based collaboration

- How to secure logs?
- Ensure the security properties
  - *Integrity* – infeasibility to forge a log operation (modify content, introduce new forged operations, etc.)
  - *Authenticity* – any user can verify the validity of operations
  - *Concurrency-tolerant property*

Concurrency-tolerant property



Totally ordered log



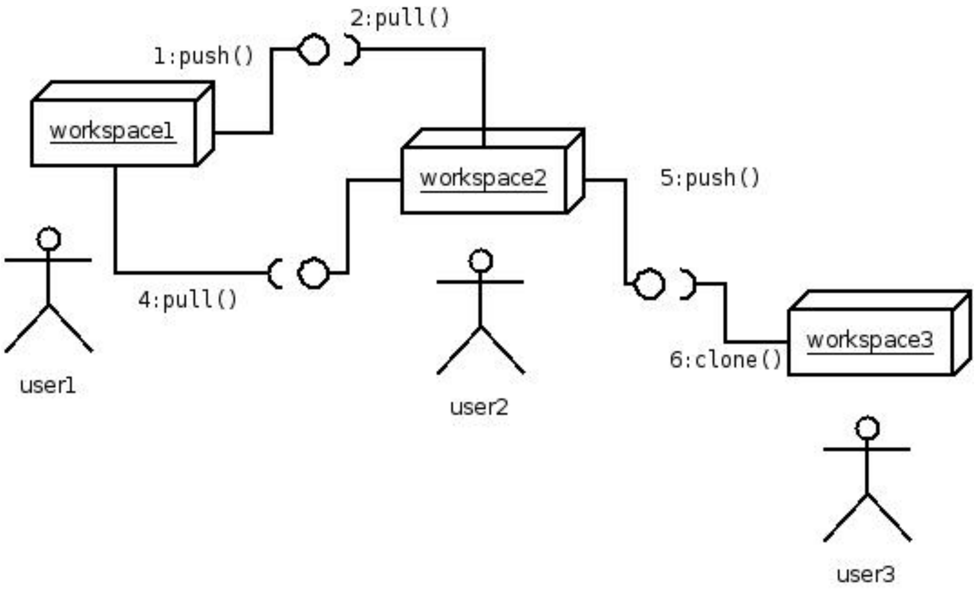
Partially ordered log

Concurrency-tolerant property

- Causal order (Lamport happens before relation)
  - $op_1 \rightarrow op_2$ if op_2 generated after the execution of op_1
- Partially ordered set:
  - (L, →) with L the set of operations and → the causal relation
- A linear extension of (L,→) is $(L,<_t)$ s.t.
  - For all op1, op2 in L either op1 $<_t$ op2 or op2 $<_t$ op1
  - If op1 → op2 then op1 $<_t$ op2
- Σ (L) the set of linear extensions
- Concurrency-tolerant property

$$F(L_i) = F(L_j)\ \forall L_i, L_j \in \Sigma(L).$$

Authenticators



- A public key/site
- An authenticator created when pushing changes
- An authenticator created when pulling changes

Authenticators

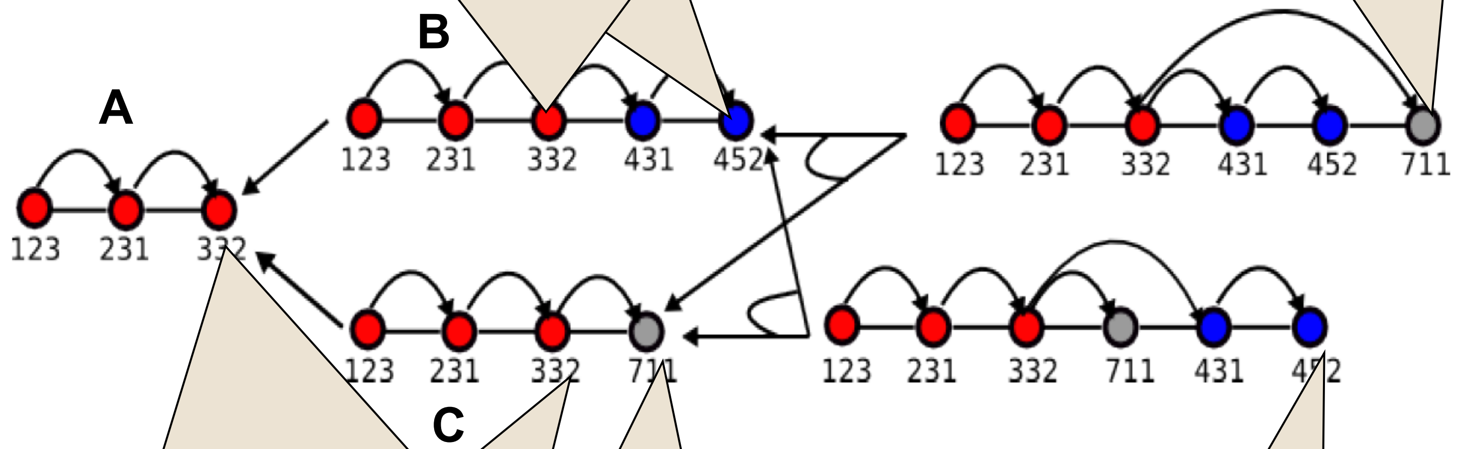
- Use authenticator as a log-tamper evident for updates
- Each authenticator is a tuple of <ID, SIG, IDE, PRE, SYN>:
  - ID: identifier of the authenticator
  - SIG: signature of author of authenticator
  - IDE: list of operation identifiers the authenticator refers to
  - PRE, SYN: identifiers of preceding and remote authenticators
- Compute SIG for each authenticator:
  - $Y^A_n.SIG = Y^A_{n-1}.SIG \mid\mid E \mid\mid Y^B_m.SIG$
- The order of operations in Y^A_{n-1} and E has to be respected
- The order of operations in Y^B_m and E does NOT need to be respected

Creation of authenticators

Υ_{B}^{3}=<(711,B),SIG$_{B}$(Υ^{2}_{B}.SIG||Υ^{2}_{C}.SIG),Ø,Υ_{B}^{2}, Υ_{C}^{2}>

Υ_{B}^{2}=<(452,B),SIG$_{B}$(Υ^{1}_{B}.SIG||431||452),{431,452},Υ_{B}^{1},Ø>

Υ_{B}^{1}=<(332,B),SIG$_{B}$(Υ^{1}_{A}.SIG),Ø,Ø,Υ_{A}^{1} >



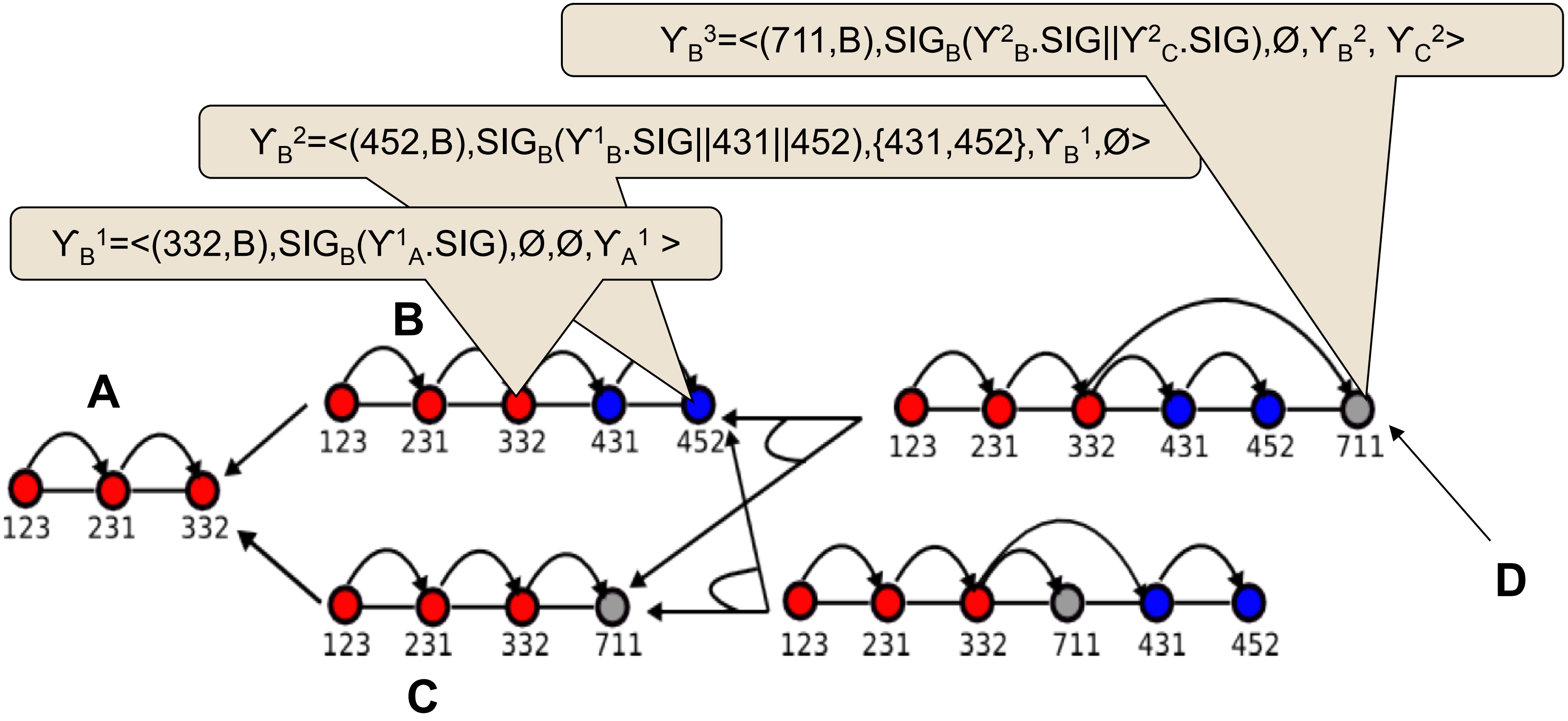
Υ_{A}^{1}=<(332,A),SIG$_{A}$(123||231||332),{123,231,332},Ø,Ø>

Υ_{C}^{1}=<(332,C),SIG$_{C}$(Υ^{1}_{A}.SIG),Ø,Ø,Υ_{A}^{1} >

Υ_{C}^{2}=<(711,C),SIG$_{C}$(Υ^{1}_{C}.SIG||711),{711},Υ_{C}^{1},Ø>

Υ_{C}^{3}=<(452,C),SIG$_{C}$(Υ^{2}_{C}.SIG||Υ^{2}_{B}.SIG),Ø,Υ_{C}^{2}, Υ_{B}^{2}>

Verification of authenticators



Υ_B^3=<(711,B),SIG$_B$(Υ^2_B.SIG||Υ^2_C.SIG),Ø,Υ_B^2, Υ_C^2>

Υ_B^2=<(452,B),SIG$_B$(Υ^1_B.SIG||431||452),{431,452},Υ_B^1,Ø>

Υ_B^1=<(332,B),SIG$_B$(Υ^1_A.SIG),Ø,Ø,Υ_A^1 >

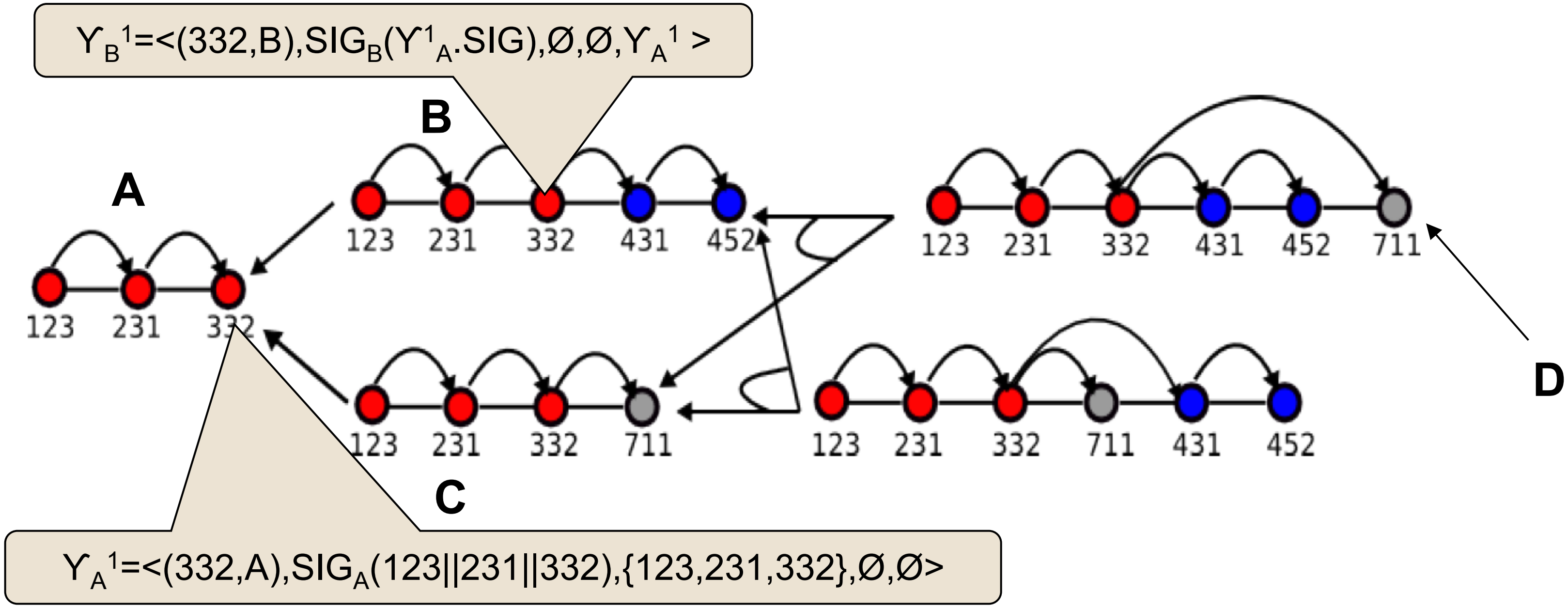
Verify Υ_B^3

- Verify Υ_B^3.SIG
- Verify Υ_B^2
- Verify Υ_C^2

Verify Υ_B^2

- Verify Υ_B^2.SIG
- Verify Υ_B^1
- Verify order op Υ_B^1 (332) before {431,452}

Verification of authenticators



Υ_B^1=<(332,B),$SIG_B(\Upsilon^1_A.SIG)$,Ø,Ø,Υ_A^1 >

Υ_A^1=<(332,A),SIG_A(123||231||332),{123,231,332},Ø,Ø>

Verify Υ_B^1

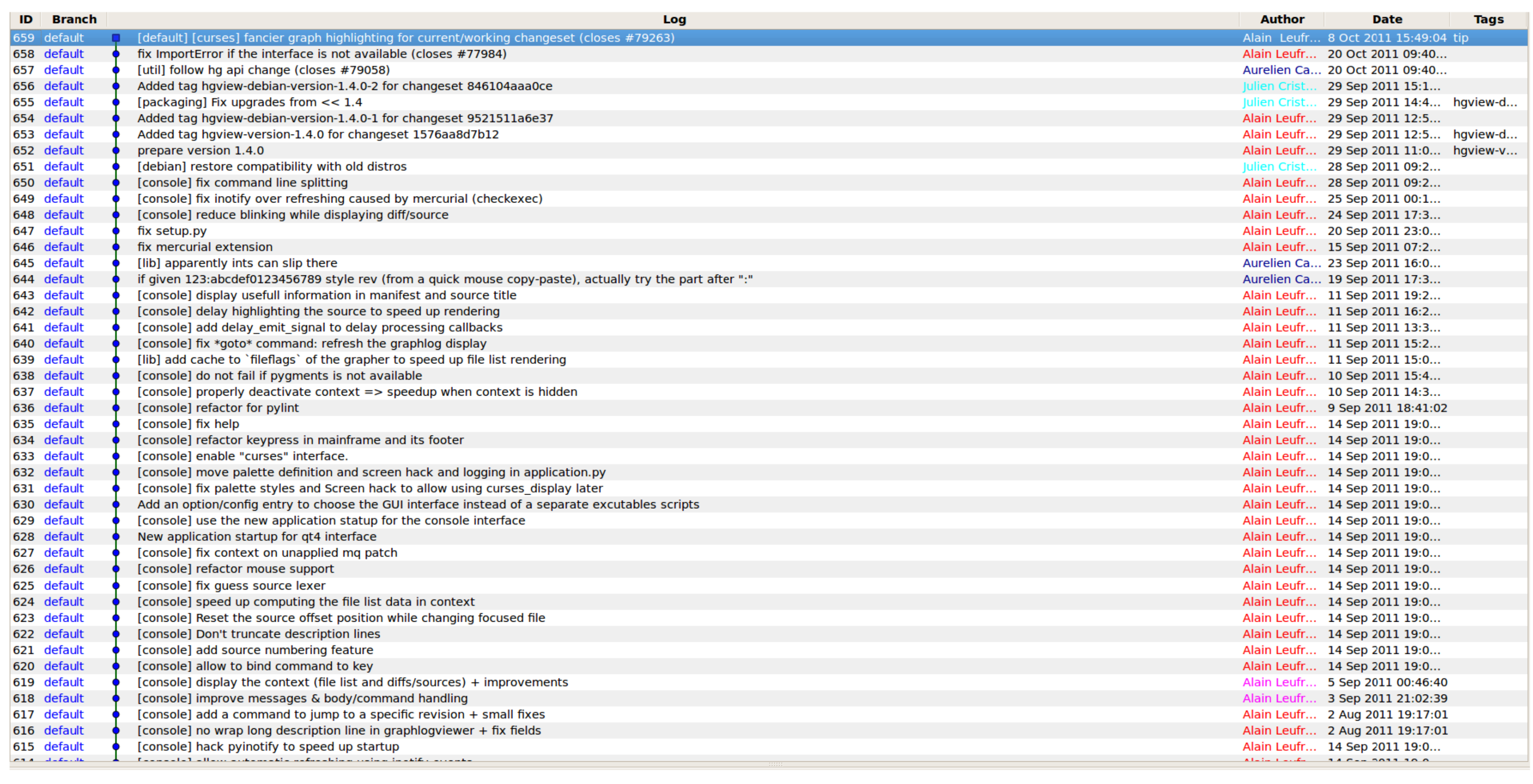
- Verify Υ_B^1.SIG
- Verify Υ_A^1

Verify Υ_A^1

- Verify Υ_A^1.SIG
- Verify order {123,231,332} respected

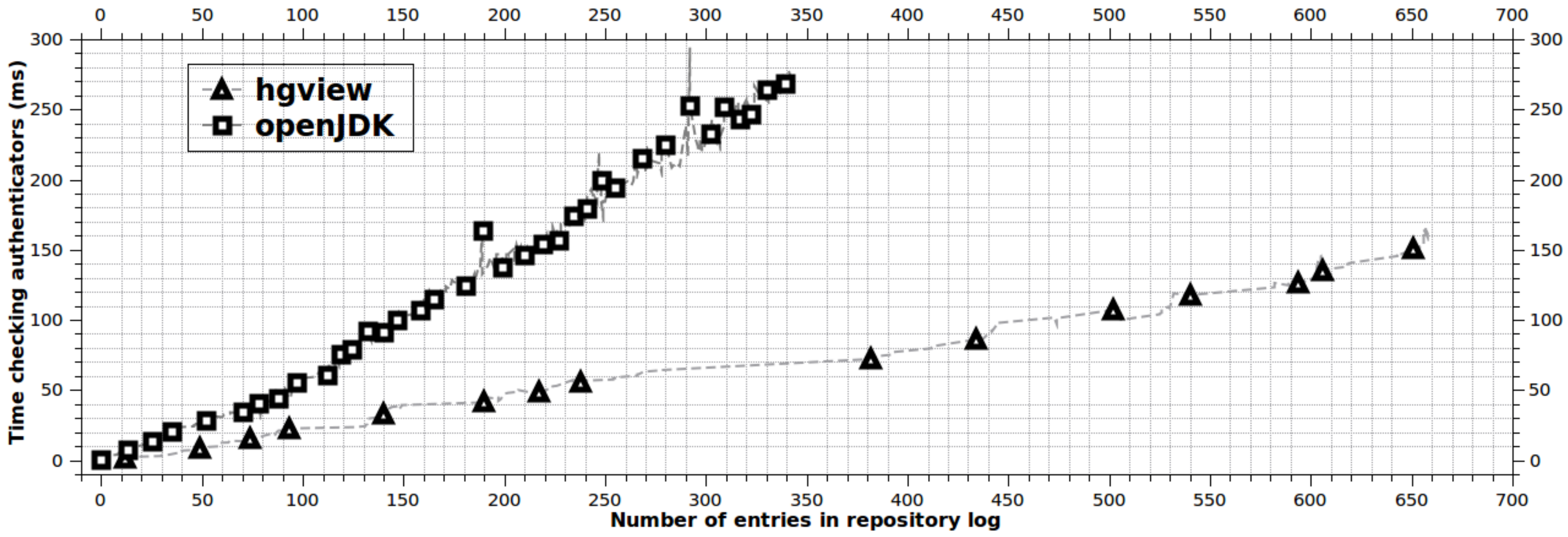
Evaluation

- Mercurial traces



ID	Branch	Log	Author	Date	Tags
659	default	[default] [curses] fancier graph highlighting for current/working changeset (closes #79263)	Alain Leufr...	8 Oct 2011 15:49:04	tip
658	default	fix ImportError if the interface is not available (closes #77984)	Alain Leufr...	20 Oct 2011 09:40...	
657	default	[util] follow hg api change (closes #79058)	Aurelien Ca...	20 Oct 2011 09:40...	
656	default	Added tag hgview-debian-version-1.4.0-2 for changeset 846104aaa0ce	Julien Crist...	29 Sep 2011 15:1...	
655	default	[packaging] Fix upgrades from << 1.4	Julien Crist...	29 Sep 2011 14:4...	hgview-d...
654	default	Added tag hgview-debian-version-1.4.0-1 for changeset 9521511a6e37	Alain Leufr...	29 Sep 2011 12:5...	
653	default	Added tag hgview-version-1.4.0 for changeset 1576aa8d7b12	Alain Leufr...	29 Sep 2011 12:5...	hgview-d...
652	default	prepare version 1.4.0	Alain Leufr...	29 Sep 2011 11:0...	hgview-v...
651	default	[debian] restore compatibility with old distros	Julien Crist...	28 Sep 2011 09:2...	
650	default	[console] fix command line splitting	Alain Leufr...	28 Sep 2011 09:2...	
649	default	[console] fix inotify over refreshing caused by mercurial (checkexec)	Alain Leufr...	25 Sep 2011 00:1...	
648	default	[console] reduce blinking while displaying diff/source	Alain Leufr...	24 Sep 2011 17:3...	
647	default	fix setup.py	Alain Leufr...	20 Sep 2011 23:0...	
646	default	fix mercurial extension	Alain Leufr...	15 Sep 2011 07:2...	
645	default	[lib] apparently ints can slip there	Aurelien Ca...	23 Sep 2011 16:0...	
644	default	if given 123:abcdef0123456789 style rev (from a quick mouse copy-paste), actually try the part after ":"	Aurelien Ca...	19 Sep 2011 17:3...	
643	default	[console] display usefull information in manifest and source title	Alain Leufr...	11 Sep 2011 19:2...	
642	default	[console] delay highlighting the source to speed up rendering	Alain Leufr...	11 Sep 2011 16:2...	
641	default	[console] add delay_emit_signal to delay processing callbacks	Alain Leufr...	11 Sep 2011 13:3...	
640	default	[console] fix *goto* command: refresh the graphlog display	Alain Leufr...	11 Sep 2011 15:2...	
639	default	[lib] add cache to `fileflags` of the grapher to speed up file list rendering	Alain Leufr...	11 Sep 2011 15:0...	
638	default	[console] do not fail if pygments is not available	Alain Leufr...	10 Sep 2011 15:4...	
637	default	[console] properly deactivate context => speedup when context is hidden	Alain Leufr...	10 Sep 2011 14:3...	
636	default	[console] refactor for pylint	Alain Leufr...	9 Sep 2011 18:41:02	
635	default	[console] fix help	Alain Leufr...	14 Sep 2011 19:0...	
634	default	[console] refactor keypress in mainframe and its footer	Alain Leufr...	14 Sep 2011 19:0...	
633	default	[console] enable "curses" interface.	Alain Leufr...	14 Sep 2011 19:0...	
632	default	[console] move palette definition and screen hack and logging in application.py	Alain Leufr...	14 Sep 2011 19:0...	
631	default	[console] fix palette styles and Screen hack to allow using curses_display later	Alain Leufr...	14 Sep 2011 19:0...	
630	default	Add an option/config entry to choose the GUI interface instead of a separate excutables scripts	Alain Leufr...	14 Sep 2011 19:0...	
629	default	[console] use the new application statup for the console interface	Alain Leufr...	14 Sep 2011 19:0...	
628	default	New application startup for qt4 interface	Alain Leufr...	14 Sep 2011 19:0...	
627	default	[console] fix context on unapplied mq patch	Alain Leufr...	14 Sep 2011 19:0...	
626	default	[console] refactor mouse support	Alain Leufr...	14 Sep 2011 19:0...	
625	default	[console] fix guess source lexer	Alain Leufr...	14 Sep 2011 19:0...	
624	default	[console] speed up computing the file list data in context	Alain Leufr...	14 Sep 2011 19:0...	
623	default	[console] Reset the source offset position while changing focused file	Alain Leufr...	14 Sep 2011 19:0...	
622	default	[console] Don't truncate description lines	Alain Leufr...	14 Sep 2011 19:0...	
621	default	[console] add source numbering feature	Alain Leufr...	14 Sep 2011 19:0...	
620	default	[console] allow to bind command to key	Alain Leufr...	14 Sep 2011 19:0...	
619	default	[console] display the context (file list and diffs/sources) + improvements	Alain Leufr...	5 Sep 2011 00:46:40	
618	default	[console] improve messages & body/command handling	Alain Leufr...	3 Sep 2011 21:02:39	
617	default	[console] add a command to jump to a specific revision + small fixes	Alain Leufr...	2 Aug 2011 19:17:01	
616	default	[console] no wrap long description line in graphlogviewer + fix fields	Alain Leufr...	2 Aug 2011 19:17:01	
615	default	[console] hack pyinotify to speed up startup	Alain Leufr...	14 Sep 2011 19:0...	
614	default	[console] allow automatic refreshing using inotify events	Alain Leufr...	14 Sep 2011 19:0...	

Evaluation


hgview
openJDK
Time checking authenticators (ms)
Number of entries in repository log
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
0
50
100
150
200
250
300

Existing approaches

- State-based approaches
  - *Summary Hash History (SHH)*[1]
- Secure total ordered logs
  - No support for concurrent work => no preservation of concurrency-tolerant property
  - *Secure Provenance History*[2]
  - *Secure History through Time Entanglement*[3]
- Fork-join causal consistency
  - (key, value) store + version vectors + signature for each operation
  - *Depot*[4]

[1] B.Kang, R.Wilensky, J. Kubiatowicz. The hash history approach for reconciling mutual inconsistency. ICDCS 2003

[2] R. Hasan, R. Sion, M. Winslett. The case of the fake Picasso: Preventing history forgery with secure provenance. FAST 2009

[3] P. Maniatis, M. Baker. Secure history preservation through timeline entanglement. USENIX 2002

[4] Prince Mahajan, Srinath T.V. Setty, Sangmin Lee, Allen Clement, Lorenzo Alvisi, Michael Dahlin, Michael Walfish: Depot: Cloud Storage with Minimal Trust. OSDI 2010

Conclusion

- A hash chain based approach to secure partially ordered logs
  - Tamper-detection
  - Accountability of users
  - Concurrency-tolerant property
- Evaluation based on real-traces