NSAC Network Security and Applied Cryptography Laboratory

http://crypto.cs.stonybrook.edu

# Towards Regulatory Compliance in Data Management

Tutorial @ VLDB 2007


**Radu Sion**
Stony Brook NSAC Lab
sion@cs.stonybrook.edu

**Marianne Winslett**
UIUC DAIS Lab
winslett@cs.uiuc.edu


National Science Foundation
WHERE DISCOVERIES BEGIN

STONY BROOK COMPUTER SCIENCE

ver. 2.3 (07/19/2007)

# Finance

National Association of Insurance Commissioners, **Graham-Leach-Bliley Act**, 1999; The U.S. Securities and Exchange Commission, **Rule 17a-3&4, 17 CFR Part 240**: Electronic Storage of Broker-Dealer Records, 2003; U.S. Public Law No. 107-204, 116 Stat. 745, The Public Company Accounting Reform and Investor Protection Act, 2002 (**Sarbanes-Oxley**)

# Healthcare

U.S. Dept. of Health & Human Services, The Health Insurance Portability and Accountability Act (**HIPAA**), 1996; The U.S. Department of Health and Human Services Food and Drug Administration, **21 CFR Part 11**: Electronic Records and Signature Regulations 1997

# Government

U.S. Public Law 107-347. The E-Government Act, 2002 (Federal Information Security Management Act **FISMA**); The U.S. Department of Defense, **Directive 5015.2**: DOD Records Management Program, 2002; The U.S. Department of Education. 20 U.S.C. 1232g; 34 CFR Part 99:The Family Educational Rights and Privacy Act (**FERPA**), 1974

# Title I

Continuing health insurance coverage.

# Title II

- Privacy Rule (all PHI)
- Transactions and Code Sets Rule
- Security Rule (electronic PHI)
  - Safeguards
    - administrative (policies and procedures)
    - physical
    - technical safeguards
- Unique Identifiers Rule
- Enforcement Rule

# e.g., HIPAA (cont'd)



SEC. 1173 (d) ("Security Standards for Health Information") mandates: "safeguards [. . . ] to ***ensure the integrity and confidentiality*** [. . . ] of the information" and "to protect against any reasonably anticipated [. . . ] threats or hazards to the [. . . ] integrity of the information" (e.g., once stored).

***http://www.cms.hhs.gov/HIPAAGenInfo/Downloads/HIPAALaw.pdf***

# Impact Layers



## Hardware

Tamper-resistance, magnetic Residues, emanations

## OS

I/O device drivers and kernel

## Storage

Block level: WORM assurances, secure migration (1)
FS level: secure indexing, secure deletion, secure provenance, history independent data structures, secure migration (2)

## Databases

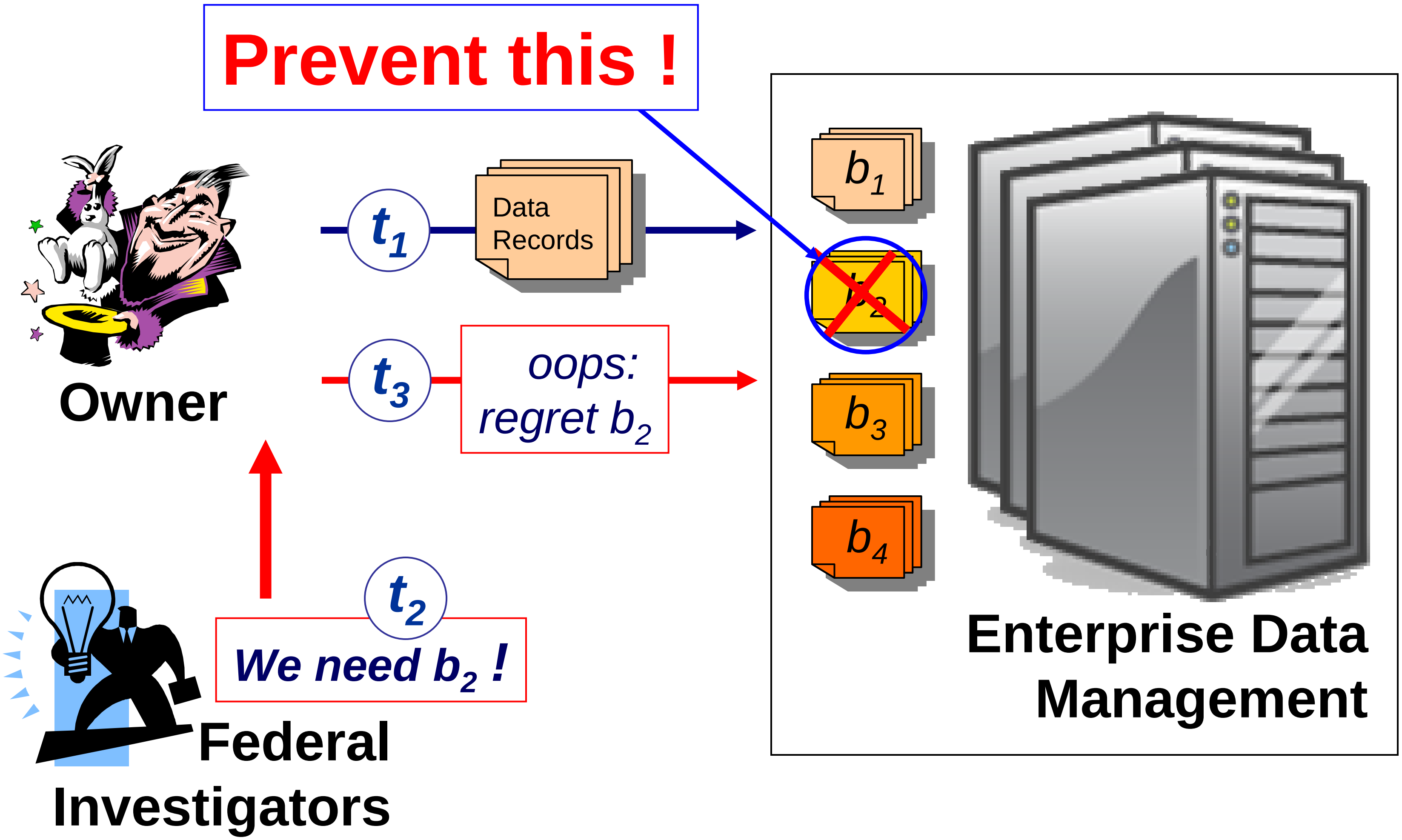
History Independence – novel indexing
ACID still holds ?

## Networks

Physical level: wireless spectrum sharing behavior
Packet level: anti-spam, flow labeling

# WORM: Overview





# WORM: Some Properties



## We do not prevent history. Just history *rewriting*.

A bit artificial in scope – why do we trust the owner to correctly log the entries and then mistrust her later ? If I were a malicious owner, I would simply not log suspicious emails ☺

## Do we trust owner for the next 5 minutes too ?

What is $\Delta t = t_3 - t_1$ ("time warp") ? If we know this, we can deploy all kinds of optimizations.

## Trustworthy Indexing. Really a problem ?

Not sure it is much of a problem ("*outside of very high latency media" - Xiaonan Ma @ IBM Almaden*). Investigators can simply come in and first do a checksum of the entire store in the background, *as investigations usually last months/years*.

## Secure Deletion.

Is a problem only if trustworthy indexing is required.

## Secure Migration.

Relatively straightforward. Build trust chain, deal with obsolete encryption, lack of keys.

## Litigation support.

Need to make sure retention can be prolonged in the case of an ongoing litigation.

## Tape-based

Assumption: specific reader is used. Checksums (keyed) are written onto tape. Keys are managed inside readers.

## Optical Disks

Problem: physical storage space, cost, replication attacks, high latency. No secure deletion.

## Hard Disks

Main problem: “soft”-ware.

# Tape: Quantum



## DLTSage WORM

Assurances of the tape systems are provided under the assumption that compliant tape-readers are deployed. “DLTSage WORM provides features to assure compliance, placing an electronic key on each cartridge to ensure WORM integrity. This unique identifier cannot be altered, providing a tamperproof archive cartridge that meets stringent compliance requirements to ensure integrity protection and full accessibility with reliable duplication.”

### Sony Disk for Data

Holds only 23 GB per disk side. Because it is faster than tape and cheaper than hard disks, optical WORM storage technology is often deployed as a secondary, high-latency storage medium to be used in the framework of a hard disk-based solution. Care needs to be taken in establishing points of trust and data integrity when information leaves the secured hard disk store for the optical media.

## EMC Centera

Content addressed storage (CAS) software solution with regulatory compliance capabilities. Data records have “two components: the content and its associated content descriptor file (CDF) that is directly linked to the stored object […] A digital fingerprint derived from the content itself is the content’s locator. […]

The CDF is used for access to and management of the record. Within this CDF, the application will assign a retention period for each individual business record. Centera will permit deletion of a pointer to a record upon expiration of the retention period. Once the last pointer to a record has been so deleted, the object will be eliminated”, and, in the Plus version, also “shredded” (from the media).

# Disk: Hitachi



## Hitachi Message Archive for Compliance

The Data Retention Utility is a software-based “virtual” WORM mechanism for mainstream Hitachi storage systems. It allows customers to “lock down archived data, making it non-erasable and non-rewritable for prescribed periods, facilitating compliance with governmental or industry regulations”.

# Disk: IBM

## IBM LockVault compliance software

Software layer that operates on top of IBM System Storage N series to provide "disk-based regulatory compliance solutions for unstructured data".

## IBM System Storage Archive Manager

The IBM Tivoli Storage Manager is part of the IBM Total Storage Software and provides certain software data retention protection. It "makes the deletion of data before its scheduled expiration extremely difficult. *Short of physical destruction to storage media or server, or deliberate corruption of data or deletion of the Archive Manager database*, Archive Manager will not allow data [...] to be deleted before its scheduled expiration date."

## Snaplock Compliance/Enterprise Software

A software suite designed to work on top of NetApp NearStore and FAS storage systems. It provides soft-WORM assurances, "preventing critical files from being altered or deleted until a specified retention date". As opposed to other vendors, NetApp SnapLock supports open industry standard protocols such as NSF and CIFS.

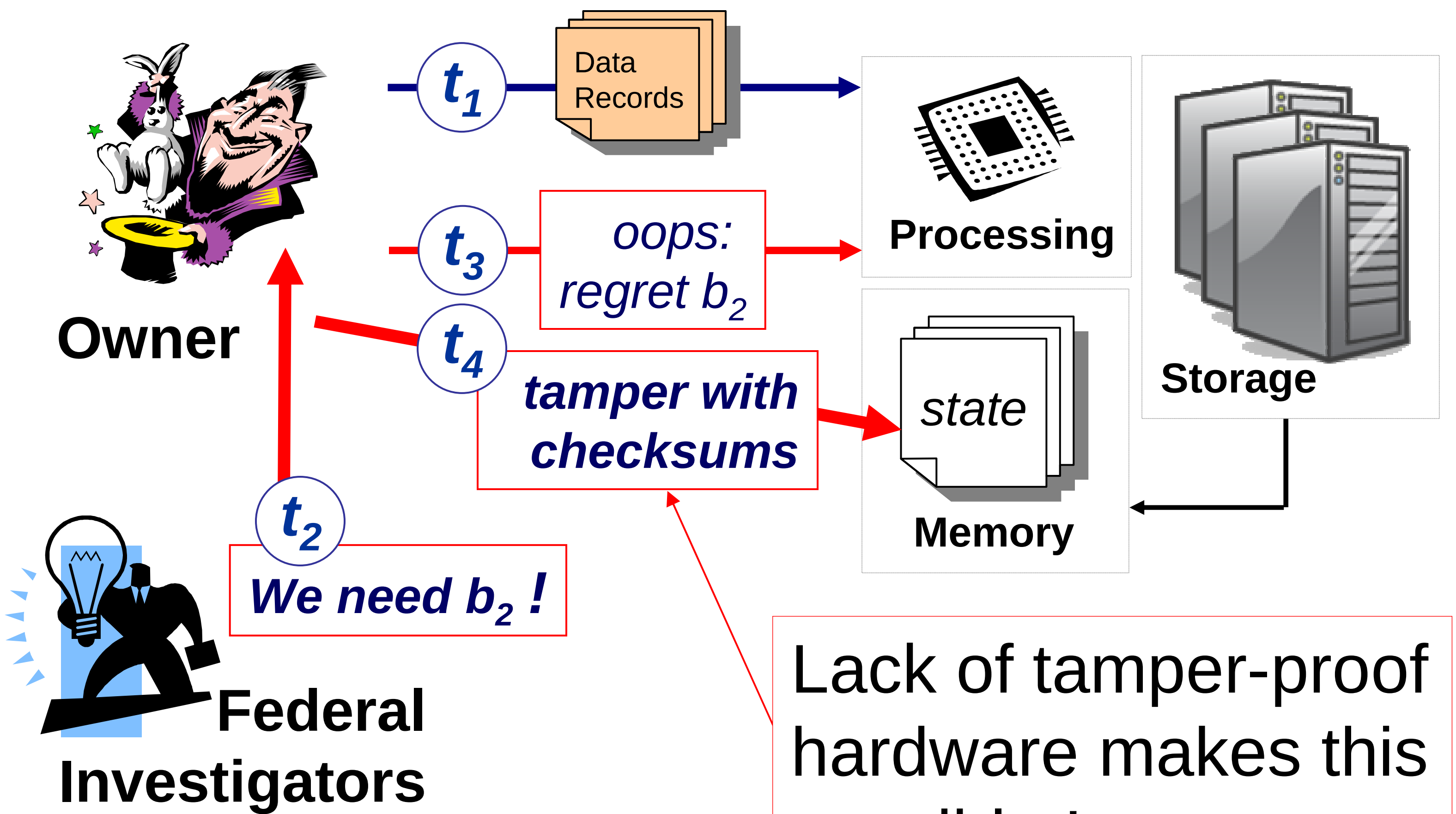
# Disk: Sun

## StorageTek Compliance Archiving Software

Software that runs on top of the Sun StorageTek 5320 NAS Appliance to “provide compliance-enabling features for authenticity, integrity, ready access, and security”.

# “soft”-WORM






Lack of tamper-proof hardware makes this possible !

# WORM: Claim One

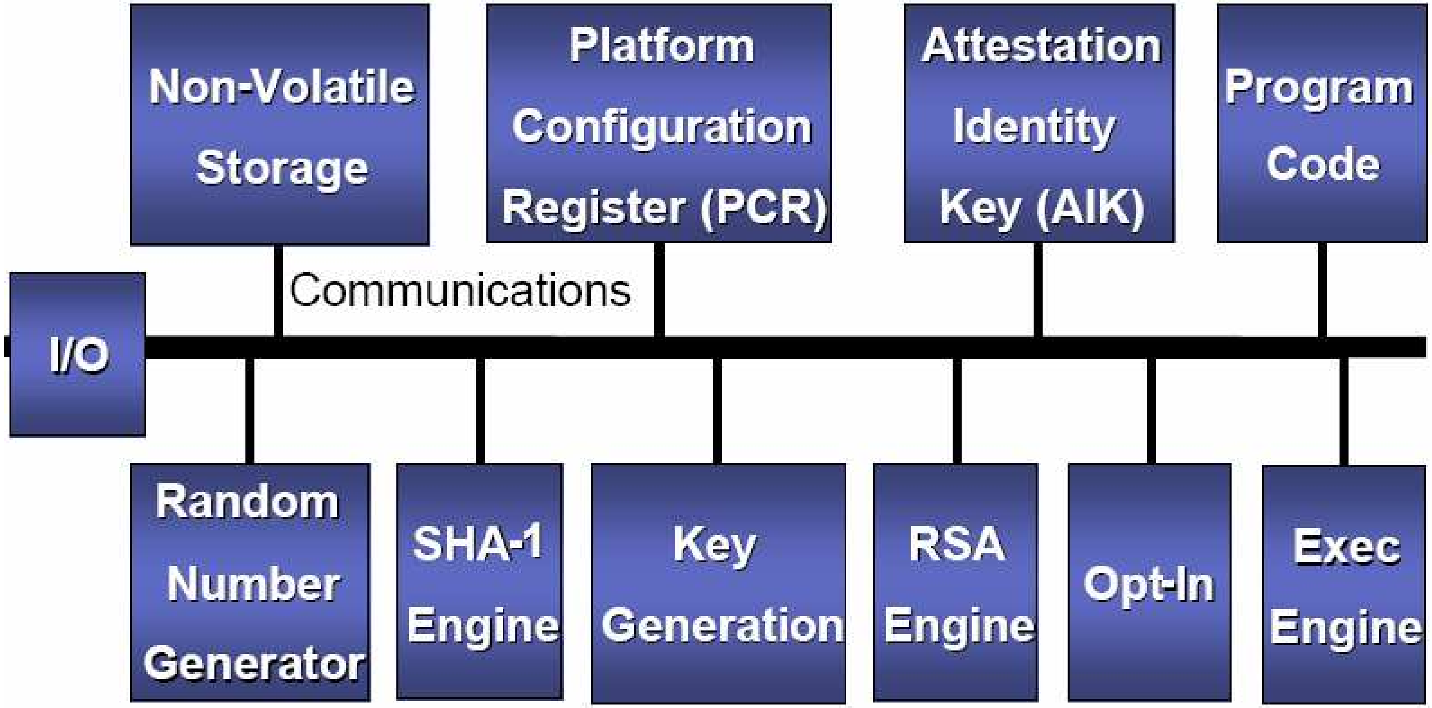
**Tamper-proof Hardware.**
Achieving WORM in the absence of tamper-proof hardware is not possible.

**Q:** *What kind of tamper-proof hardware ?*

# TPM

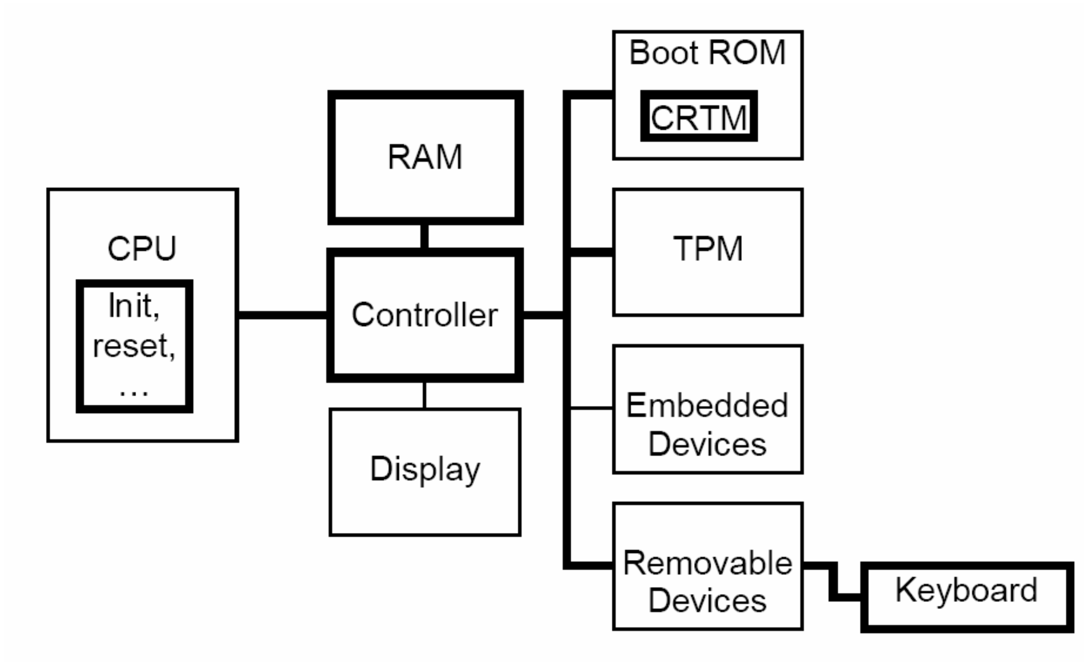


Microcontroller that stores keys, passwords and digital certificates.



# TPM: Trust Chain






**Can the Trusted Platform Module control what software runs?**
No. [… it ] can only act as a 'slave' to higher level services and applications by storing and reporting pre-runtime configuration information. […] At no time can the TCG building blocks 'control' the system or report the status of applications that are running.

# Would a TCG/TPM help ?

The passive nature of a TPM would require an additional point of blank trust in upper layer code. The ability to virtualize makes this hard to achieve.

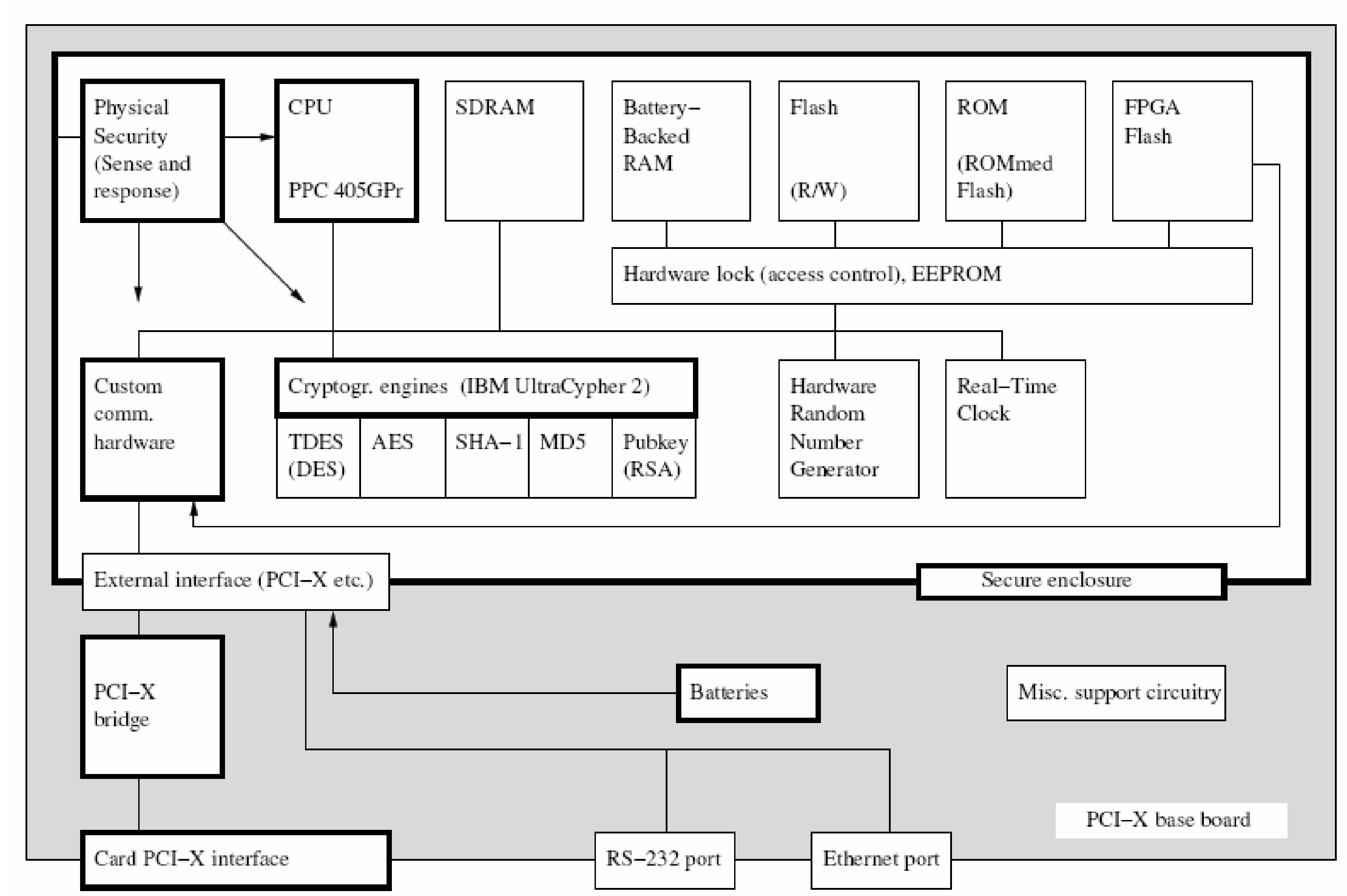
**Discussion:** How would Mallory fake a world view to the TPM. Remember we are talking about *millions of US dollars* worth of incentives here.

And by the way …

… **TPMs have been successfully hacked** by attackers with almost no resources (see refs).

# SCPUs (IBM 4764)





# WORM: Claim Two



**Active Tamper-proof Hardware.**

Achieving WORM in the absence of active tamper-proof hardware is not possible.

# SCPU: Performance

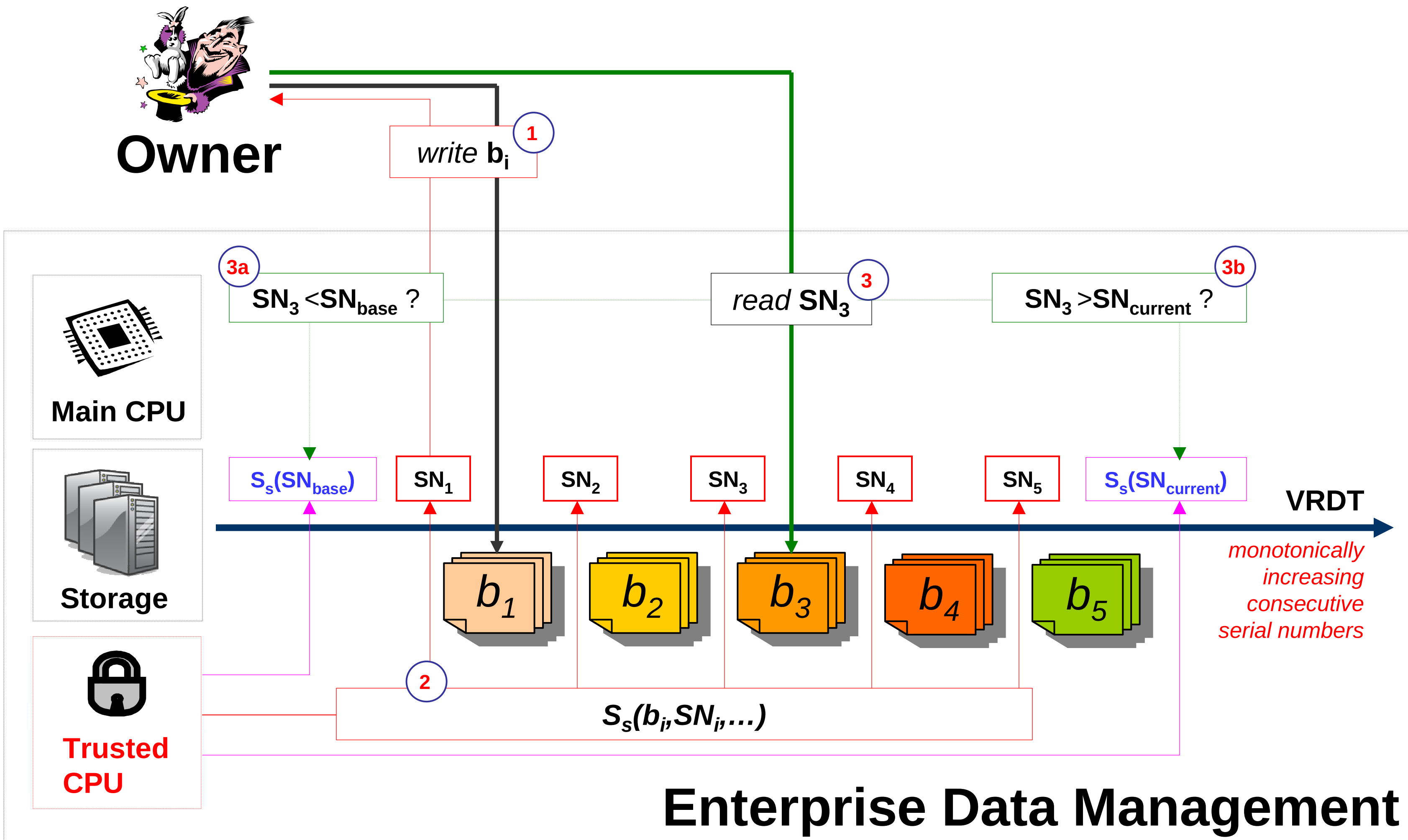


RSA1024 Sign: **848/sec**
RSA1024 Verify: **1157/sec**
3DES: **1-8MB/sec**
DES: **1-8MB/sec**
SHA1: **1-21MB/sec**

IBM 4764-001: 266MHz PowerPC. 64KB battery-backed SRAM storage. Crypto hardware engines: AES256, DES, TDES, DSS, SHA-1, MD5, RSA. FIPS 140-2 Level 4 certified.

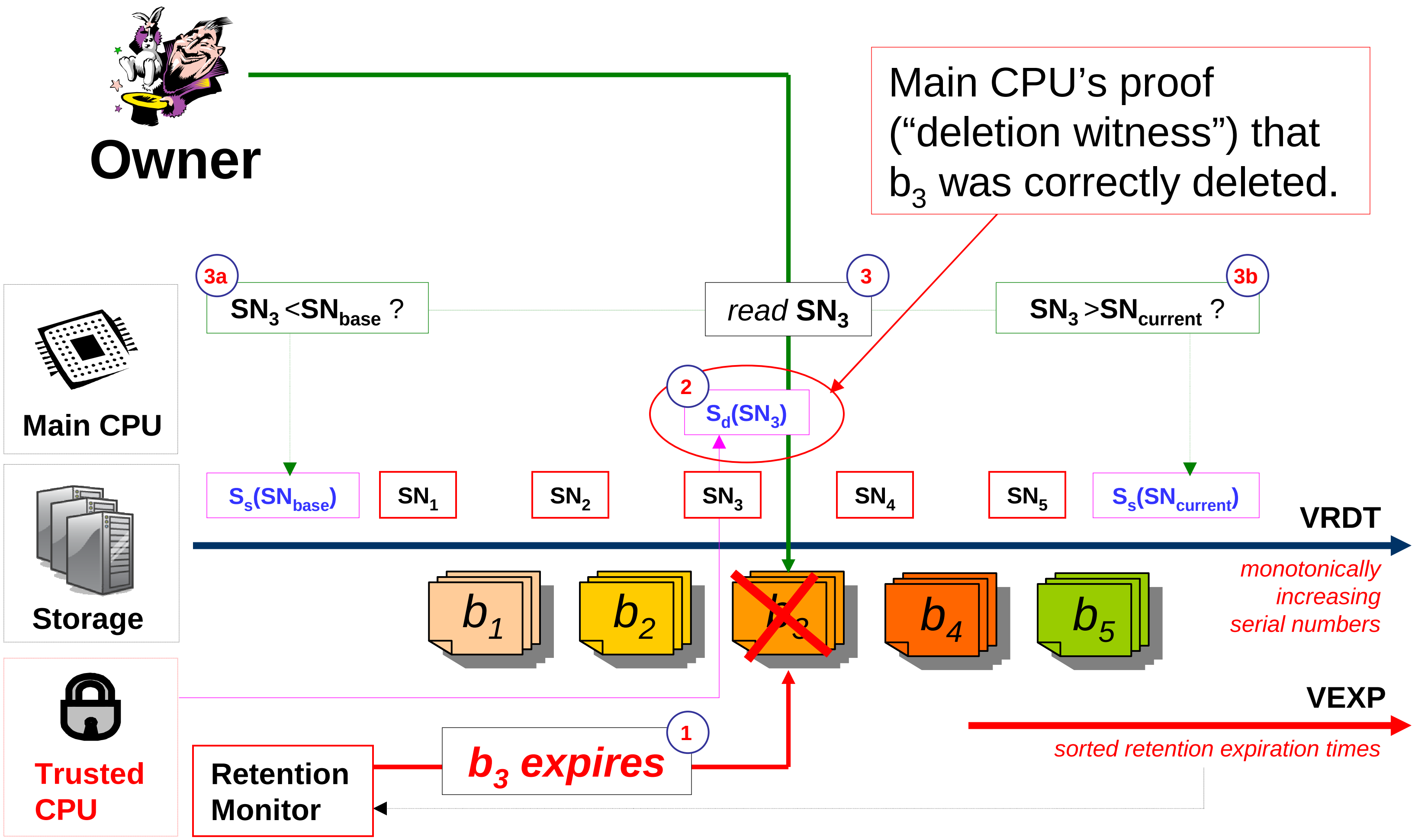
# Some Intuition





# Some Other Intuition





# Efficient Data Digestion ☺



**Issue**

SCPU data digestion (hashing) is not very fast.

**Fact**

We already assume the stored data is accurate.

**Question**

Why not also trust the main CPU to produce correct data digests *at write time*? This should increase throughput.

**How**

To prevent cheating we double check during idle times (or mandatory if too much time passes).

# Can We Eat The Cake Too ?



**How do we maintain the VRDT efficiently.**
Hierarchical. Arbitrary "deletion" windows.

**How does the SCPU/RM enforce deletion efficiently.**
Alarms, efficient index structures of retention expiration times.

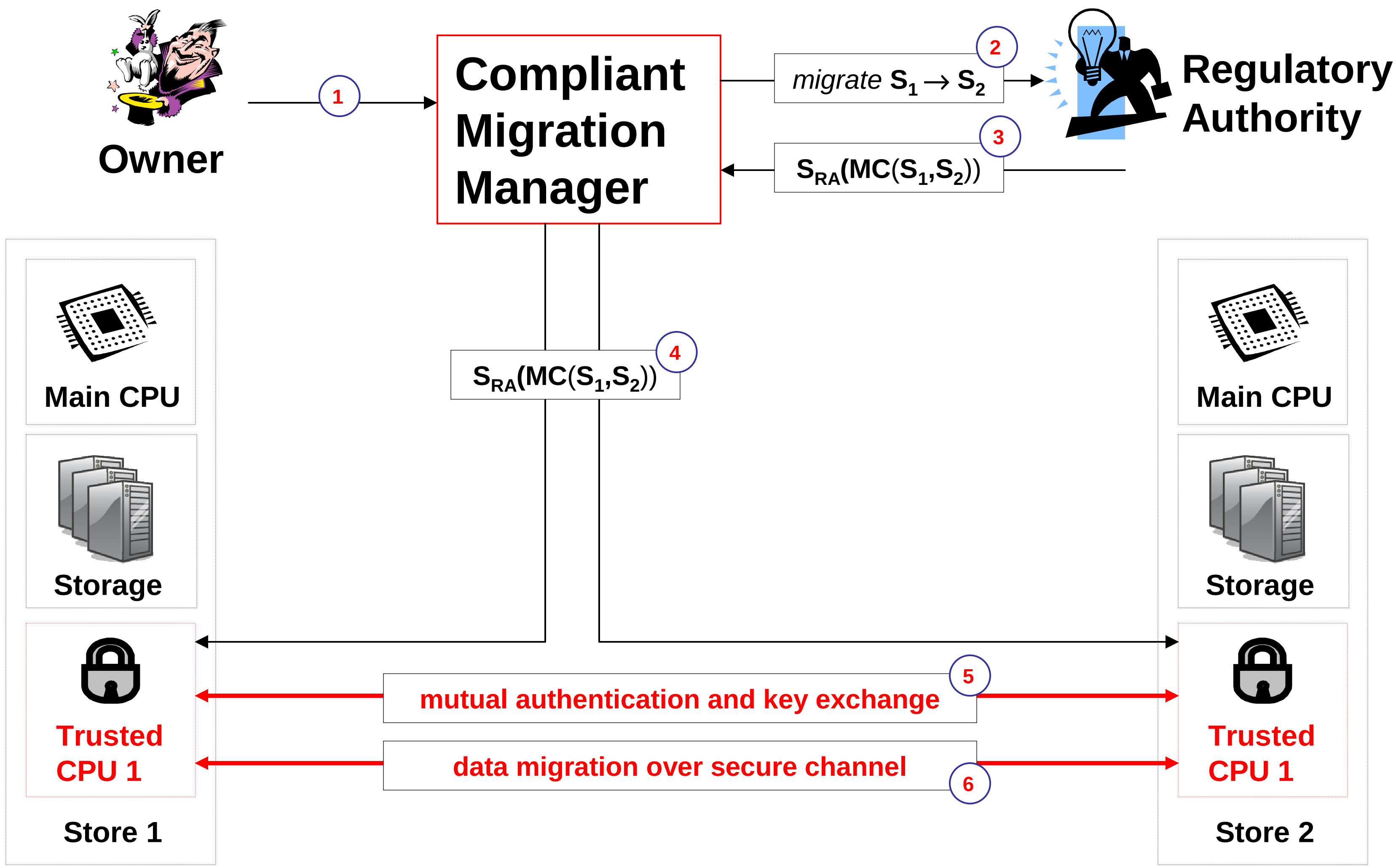
**How can we "witness" things fast: amortization.**
**In times of high-load:** defer expensive witnessing and use short-lived constructs.
**During idle/low-load times:** re-enforce short-lived constructs.

**How fast can we go.**
**Writes:** 3600-3700 updates/second (4-6hrs. bursts), 450-500 updates/sec (sustained).
**Reads:** limited only by un-trusted system segment.

**What about litigation support.**
Authorized regulatory parties present credentials and are allowed to set/reset litigation holds.

# What about migration ?





# What Next ?

**Namespaces, Search Indexes**

Trust-worthy Indexing

**More Complex Migration**

Complex query-driven migration

**Secure Deletion**

History Independent Data Structures, logging etc.

**New Query Languages/Paradigms ?**

Do transactional semantics still hold in the presence of regulatory compliance? Can we extend SQL to deal with e.g., WORM assurances ?

# Refs: Trusted Hardware




1. Attacks on TPMs. Online at http://www.cs.dartmouth.edu/□pkilab/sparks/.
2. TWIRL: The Weizmann Institute Relation Locator. Online at http://www.wisdom.weizmann.ac.il/□tromer/twirl/.
3. IBM PCI-X Cryptographic Coprocessor. Online at http://www-03.ibm.com/security/cryptocards/pcixcc/overperformance.shtml, 2003.
4. IBM 4758 PCI Cryptographic Coprocessor. Online at http://www-03.ibm.com/security/cryptocards/pcicc/overview.shtml, 2006.
5. IBM Common Cryptographic Architecture (CCA) API. Online at http://www-03.ibm.com/security/cryptocards//pcixcc/overcca.shtml, 2006.
6. Trusted Computing Platforms Storage: Compliance, Security, and Policy. Online at https://www.trustedcomputinggroup.org/news/presentations/SNIA Security Summit 2006.pdf, January 2006.
7. IBM Cryptographic Hardware. Online at http://www-03.ibm.com/security/products/,2007.
8. Trusted Computing Group. Online at https://www.trustedcomputinggroup.org/, 2007.
9. Trusted Platform Module (TPM) Specifications. Online at https://www.trustedcomputinggroup.org/specs/TPM,2007.
10. Bernhard Kauer. OSLO: Improving the Security of Trusted Computing. In USENIX Security Symposium, 2007.
11. Smart Card Alliance. HIPAA compliance and smart cards: Solutions to privacy and security requirements. Online at http://www.datakey.com/resources/HIPAA_Compliance_and_Smart_Cards_FINAL.pdf, Sep. 2003.

# Refs: Regulations




1. NIST Federal Information Processing Standards. Online at http://csrc.nist.gov/publications/fips/, 2007.
2. Protiviti Consulting. Frequently Asked Questions About J-SOX. Online at http://www.protiviti.jp/downloads/JSOXOverviewfinal\ E.pdf, 2006.
3. National Association of Insurance Commissioners. Graham-Leach-Bliley Act, 1999. www.naic.org/GLBA.
4. Ministry of Finance. Bill 198 of 2002. An Act to implement budget measures and other initiatives of the Government. Legislative Assembly of Ontario, 2002.
5. British Parliament. Data protection act of 1998. Online at http://www.staffs.ac.uk/legal/privacy/dp10rules/, 1998.
6. European Parliament. European directives. Online at http://ec.europa.eu/justice\ home/fsj/privacy/law/index\ en.htm, 2006.
7. Australian Securities and Exchange Commission. Clerp 9 corporate reporting and disclosure laws. Online at http://www.asic.gov.au, 2004.
8. The Enterprise Storage Group. Compliance: The effect on information management and the storage industry. Online at http://www.enterprisestoragegroup.com/, 2003.
9. The U.S. Department of Defense. Directive 5015.2: DOD Records Management Program. Online at http://www.dtic.mil/whs/directives/corres/pdf/50152std 061902/p50152s.pdf, 2002.
10. The U.S. Department of Education. 20 U.S.C. 1232g; 34 CFR Part 99:Family Educational Rights and Privacy Act (FERPA). Online at http://www.ed.gov/policy/gen/guid/fpco/ferpa, 1974.
11. The U.S. Department of Health and Human Services Food and Drug Administration. 21 CFR Part 11: Electronic Records and Signature Regulations. Online at http://www.fda.gov/ora/compliance ref/part11/FRs/background/pt11finr.pdf, 1997.
12. The U.S. Securities and Exchange Commission. Rule 17a-3&4, 17 CFR Part 240: Electronic Storage of Broker-Dealer Records. Online at http://edocket.access.gpo.gov/cfr2002/aprqtr/17cfr240.17a-4.htm, 2003.
13. U.S. Dept. of Health & Human Services. The Health Insurance Portability and Accountability Act (HIPAA), 1996. www.cms.gov/hipaa.
14. U.S. Public Law 107-347. The E-Government Act, 2002.
15. U.S. Public Law No. 107-204, 116 Stat. 745. The Public Company Accounting Reform and Investor Protection Act, 2002.
16. N. Lawson, J. Orr, and D. Klar. The HIPAA privacy rule: An overview of compliance initiatives and requirements. Defense Counsel Journal, 70:127–149, 2003.
17. Occupational Safety and Health Administration. Access to employee exposure and medical records. - 1910.1020 regulations (standards - 29 cfr). Online at http://www.osha.gov/pls/oshaweb/owadisp.show_document?p_table=STANDARDS&p_id=10027.
18. U. S. Congress. Federal rules of civil procedure. Online at http://www.law.cornell.edu/rules/frcp/, 2006.

# Refs: Crypto




1. TWIRL: The Weizmann Institute Relation Locator. Online at http://www.wisdom.weizmann.ac.il/□tromer/twirl/.
2. Mihir Bellare and Daniele Micciancio. A New Paradigm for Collision-Free Hashing: Incrementality at Reduced Cost. In Walter Fumy, editor, Advances in Cryptology — Proceedings of EuroCrypt 1997
3. Dwaine E. Clarke, Srinivas Devadas, Marten van Dijk, Blaise Gassend, and G. Edward Suh. Incremental multiset hash functions and their application to memory integrity checking. In Chi-Sung Laih, editor, ASIACRYPT 2003
4. A.K. Lenstra et al. Analysis of Bernstein's Factorization Circuit. Advances in Cryptology, pages 1–26, 2002.
5. J. Franke et al. SHARK: A Realizable Special Hardware Sieving Device for Factoring 1024-Bit Integers. In Cryptographic Hardware and Embedded Systems, 2005.
6. W. Geiselmann et al. Scalable Hardware for Sparse Systems of Linear Equations, with Applications to Integer Factorization. In Cryptographic Hardware and Embedded Systems CHES, 2005.
7. W. Geiselmann et al. A Simpler Sieving Device: Combining ECM and TWIRL. In Intl. Conf. on Information Security and Cryptology, 2006.
8. N. Ferguson and B. Schneier. Practical Cryptography. Wiley & Sons, 2003.
9. W. Geiselmann and R. Steinwandt. Hardware for Solving Sparse Systems of Linear Equations over GF(2). In Cryptographic Hardware and Embedded Systems CHES, 2003.
10. W. Geiselmann and R. Steinwandt. Special Purpose Hardware in Cryptanalysis: The Case of 1024-bit RSA. IEEE Security and Privacy, pages 63–66, January 2007.
11. O. Goldreich. Foundations of Cryptography, Cambridge University Press, 2001.
12. Jetico, Inc. BestCrypt software home page. www.jetico.com, 2002.
13. A.K. Lenstra and A. Shamir. Analysis and Optimization of the TWINKLE Factoring Device. In EuroCrypt, 2000.
14. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Handbook of Applied Cryptography. CRC Press, 2001.
15. R. Merkle. Protocols for public key cryptosystems. In IEEE Symposium on Research in Security and Privacy, 1980.
16. C. Pomerance. A Tale of Two Sieves. Notices of the ACM, pages 1473–1485, December 1996.
17. B. Schneier. Applied Cryptography: Protocols, Algorithms and Source Code in C. Wiley & Sons, 1996.
18. A. Shamir. Factoring Large Numbers with the TWINKLE Device. In Cryptographic Hardware and Embedded Systems, 1999.
19. Robert D. Silverman. A cost-based security analysis of symmetric and asymmetric key lengths. Online at http://www.rsasecurity.com/rsalabs/bulletins/index.html.

# Refs: Industry WORM

Stony Brook Network Security and Applied Cryptography Lab


1. EMC. Centera Compliance Edition Plus. Online at http://www.emc.com/centera/ and http://www.mosaictech.com/pdfdocs/emc/centera.pdf, 2007.
2. Hitachi Data Systems. The Message Archive for Compliance Solution, Data Retention Software Utility. Online at http://www.hds.com/solutions/data life cycle archiving/achievingregcompliance.html, 2007.
3. HP. WORM Data Protection Solutions. Online at http://h18006.www1.hp.com/products/storageworks/wormdps/index.html, 2007.
4. IBM Corp. IBM System Storage N series with LockVault compliance software. Disk-based regulatory compliance solutions for unstructured data. Online at http://www-03.ibm.com/systems/storage/network/software/lockvault/, 2007.
5. IBM Corp. IBM Total Storage Family: Tivoli System Storage Archive Manager. Online at http://www-306.ibm.com/software/tivoli/products/storage-mgr-data-reten/, 2007.
6. IBM Corp. IBM TotalStorage Enterprise. Online at http://www-03.ibm.com/servers/storage/, 2007.
7. IBM Corporation and Daniel James Winarski and Kamal Emile Dimitri. United States Patent 6879454: Write-Once Read-Many Hard Disk Drive, 2005.
8. Network Appliance Inc. SnapLock Compliance and SnapLock Enterprise Software, Online at http://www.netapp.com/products/software/snaplock.html, 2007.
9. Quantum Inc. DLTSage Write Once Read Many Solution. Online at http://www.quantum.com/Products/TapeDrives/DLT/SDLT600/DLTIce/Index.aspx and http://www.quantum.com/pdf/DS00232.pdf, 2007.
10. StorageTek Inc. VolSafe secure tape-based write once read many (WORM) storage solution. Online at http://www.storagetek.com/, 2007.
11. Sun Microsystems. Sun StorageTek Compliance Archiving Software. Online at http://www.sun.com/storagetek/management software/data protection/compliance archiving/,2007.
12. Sun Microsystems. Sun StorageTek Compliance Archiving system and the Vignette Enterprise Content Management Suite (White Paper). Online at http://www.sun.com/storagetek/white-papers/Healthcare Sun NAS Vignette EHR 080806 Final.pdf, 2007.
13. Zantaz Inc. The ZANTAZ Digital Safe Product Family. Online at http://www.zantaz.com/,2007.

# Refs: Research on WORM




1. Malcolm C. Easton. Key-Sequence Data Sets on Indelible Storage. IBM Journal of Research and Development, May 1986.
2. W. Hsu and S. Ong. Fossilization: A Process for Establishing Truly Trustworthy Records. IBM Research Report, (10331), 2004.
3. Lan Huang, Windsor W. Hsu, and Fengzhou Zheng. CIS: Content Immutable Storage for Trustworthy Record Keeping. In Proceedings of the Conference on Mass Storage Systems and Technologies (MSST), 2006.
4. Soumyadeb Mitra, Windsor W. Hsu, and Marianne Winslett. Trustworthy Keyword Search for Regulatory-Compliant Records Retention. In Proceedings of VLDB, 2006.
5. Soumyadeb Mitra and Marianne Winslett. Secure Deletion from Inverted Indexes on Compliance Storage. In Proceedings of the StorageSS Workshop, 2006.
6. Peter Rathmann. Dynamic Data Structures on Optical Disks. In 1st International Conference on Data Engineering, 1984.
7. Douglas J. Santry, Michael J. Feeley, Norman C. Hutchinson, and Alistair C. Veitch. Elephant: The file system that never forgets. In Workshop on Hot Topics in Operating Systems, pages 2–7, 1999.
8. Qingbo Zhu and Windsor W. Hsu. Fossilized index: the linchpin of trustworthy non-alterable electronic records. In SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pages 395–406, New York, NY, USA, 2005. ACM Press.
9. W. Hsu and S. Ong. WORM Storage is not Enough. IBM Systems Journal, April 2007.
10. C. Johnson and T. Grandison. Compliance with data protection laws using hippocratic database active enforcement and auditing. IBM Systems Journal, 46(2), April 2007.
11. M. Mesnier, G. Ganger, and E. Riedel. Object-based storage: pushing more functionality into storage. IEEE Potentials, 24(2):31 – 34, April-May 2005.
12. S. Mitra and M. Winslett. Secure deletion from inverted indexes on compliance storage. In StorageSS '06: Proceedings of the Second ACM Workshop on Storage Security and Survivability, pages 67–72, New York, NY, USA, 2006. ACM Press.
13. D. Reiner, G. Press, M. Lenaghan, D. Barta, and R. Urmston. Information lifecycle management: The EMC perspective. ICDE 2004.

# Refs: Provenance (1)



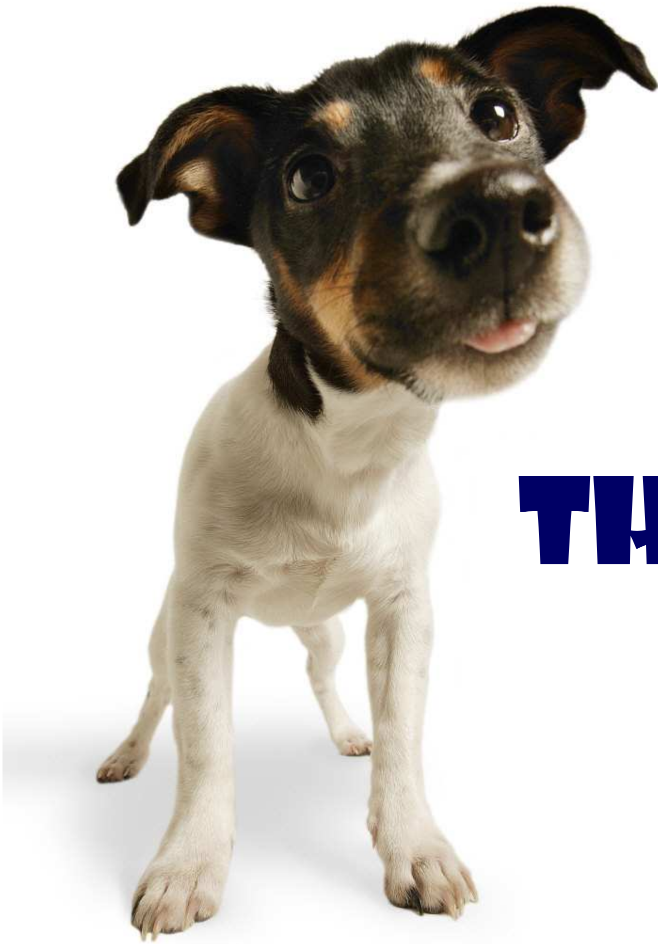

1. U. Braun, S. Garfinkel, D. Holland, K.-K. Muniswamy-Reddy, and M. Seltzer. Issues in automatic provenance collection. In Proceedings of the International Provenance and Annotation Workshop, pages 171–183, 2006.
2. M. Mesnier, G. Ganger, and E. Riedel. Object-based storage: pushing more functionality into storage. IEEE Potentials, 24(2):31 – 34, April-May 2005.
3. Y. Simmhan, B. Plale, and D. Gannon. A survey of data provenance in e-science. SIGMOD Rec., 34(3):31–36, September 2005.
4. R. S. Barga and L. A. Digiampietri. Automatic generation of workflow provenance. In Proceedings of the International Provenance and Annotation Workshop (IPAW), pages 1–9, 2006.
5. U. Braun, S. L. Garfinkel, D. A. Holland, K.-K. Muniswamy-Reddy, and M. I. Seltzer. Issues in automatic provenance collection. In Proceedings of the International Provenance and Annotation Workshop (IPAW), pages 171–183, 2006.
6. P. Buneman, A. Chapman, and J. Cheney. Provenance management in curated databases. In SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, pages 539–550, New York, NY, USA, 2006. ACM Press.
7. P. Buneman, A. Chapman, J. Cheney, and S. Vansummeren. A provenance model for manually curated data. In Proceedings of the International Provenance and Annotation Workshop (IPAW), pages 162–170, 2006.
8. P. Buneman, S. Khanna, and W. C. Tan. Data provenance: Some basic issues. In FST TCS 2000: Proceedings of the 20th Conference on Foundations of Software Technology and Theoretical Computer Science, pages 87–93, London, UK, 2000. Springer-Verlag.
9. P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. Lecture Notes in Computer Science, 1973:316–330, 2001.
10. B. W. Dearstyne. The archival enterprise: Modern archival principles, practices, and management techniques. American Library Association, 1993.
11. E. Deelman, G. Singh, M. Atkinson, A. Chervenak, N. C. Hong, C. Kesselman, S. Patil, L. Pearlman, , and M. Su. Grid-based metadata services. In SSDBM '04: Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM'04), page 393, Washington, DC, USA, 2004. IEEE Computer Society.
12. I. T. Foster, J. Vockler, M. Wilde, and Y. Zhao. Chimera: A virtual data system for representing, querying, and automating data derivation. In SSDBM '02: Proceedings of the 14th International Conference on Scientific and Statistical Database Management, pages 37–46, Washington, DC, USA, 2002. IEEE Computer Society.

# Refs: Provenance (2)




13. C. Goble. Position statement: Musings on provenance, workflow workflow and (semantic web) annotations for bioinformatics. In Workshop on Data Derivation and Provenance, Chicago, 2002.
14. J. Golbeck. Combining provenance with trust in social networks for semantic web content filtering. In Proceedings of the International Provenance and Annotation Workshop (IPAW), pages 101–108, 2006.
15. R. Hasan, S. Myagmar, A. J. Lee, and W. Yurcik. Toward a threat model for storage systems. In Proceedings of the first ACM workshop on Storage security and survivability (StorageSS), pages 94–102, Fairfax, VA, USA, 2005. ACM Press.
16. C. A. Lynch. When documents deceive: Trust and provenance as new factors for information retrieval in a tangled web. Journal of the American Society for Information Science and Technology, 52(1):12–17, 2001.
17. K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. I. Seltzer. Provenance-aware storage systems. In USENIX Annual Technical Conference, General Track, pages 43–56, 2006.
18. J. D. Myers, T. C. Allison, S. Bittner, B. Didier, M. Frenklach, J. William H. Green, Y.-L. Ho, J. Hewson, W. Koegler, C. Lansing, D. Leahy, M. Lee, R. McCoy, M. Minkoff, S. Nijsure, G. von Laszewski, D. Montoya, C. Pancerella, R. Pinzon, W. Pitz, L. A. Rahn, B. Ruscic, K. Schuchardt, E. Stephan, A. Wagner, T. Windus, and C. Yang. A collaborative informatics infrastructure for multi-scale science. clade, 00:24, 2004.
19. C. Sar and P. Cao. Lineage file system. Online at http://crypto.stanford.edu/ cao/lineage.html, January 2005.
20. Y. L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance in e-science. SIGMOD Rec., 34(3):31–36, September 2005.
21. M. Szomszor and L. Moreau. Recording and reasoning over data provenance in web and grid services. In International Conference on Ontologies, Databases and Applications of SEmantics (ODBASE), volume 2888 of Lecture Notes in Computer Science, pages 603–620, Catania, Italy, 2003.
22. N. N. Vijayakumar and B. Plale. Towards low overhead provenance tracking in near real-time stream filtering. In Proceedings of the International Provenance and Annotation Workshop (IPAW), pages 46–54, 2006.
23. J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. In Proceedings of the Second Biennial Conference on Innovative Data Systems Research (CIDR), January 2005.
24. J. Zhao, C. A. Goble, R. Stevens, and S. Bechhofer. Semantically linking and browsing provenance logs for e-science. In ICSNW, pages 158–176, 2004.

# /bin/yes > /dev/wakeup



# THANK YOU !

# Part 2:
# Indexing, Deleting, and Migrating Compliance Records

Marianne Winslett

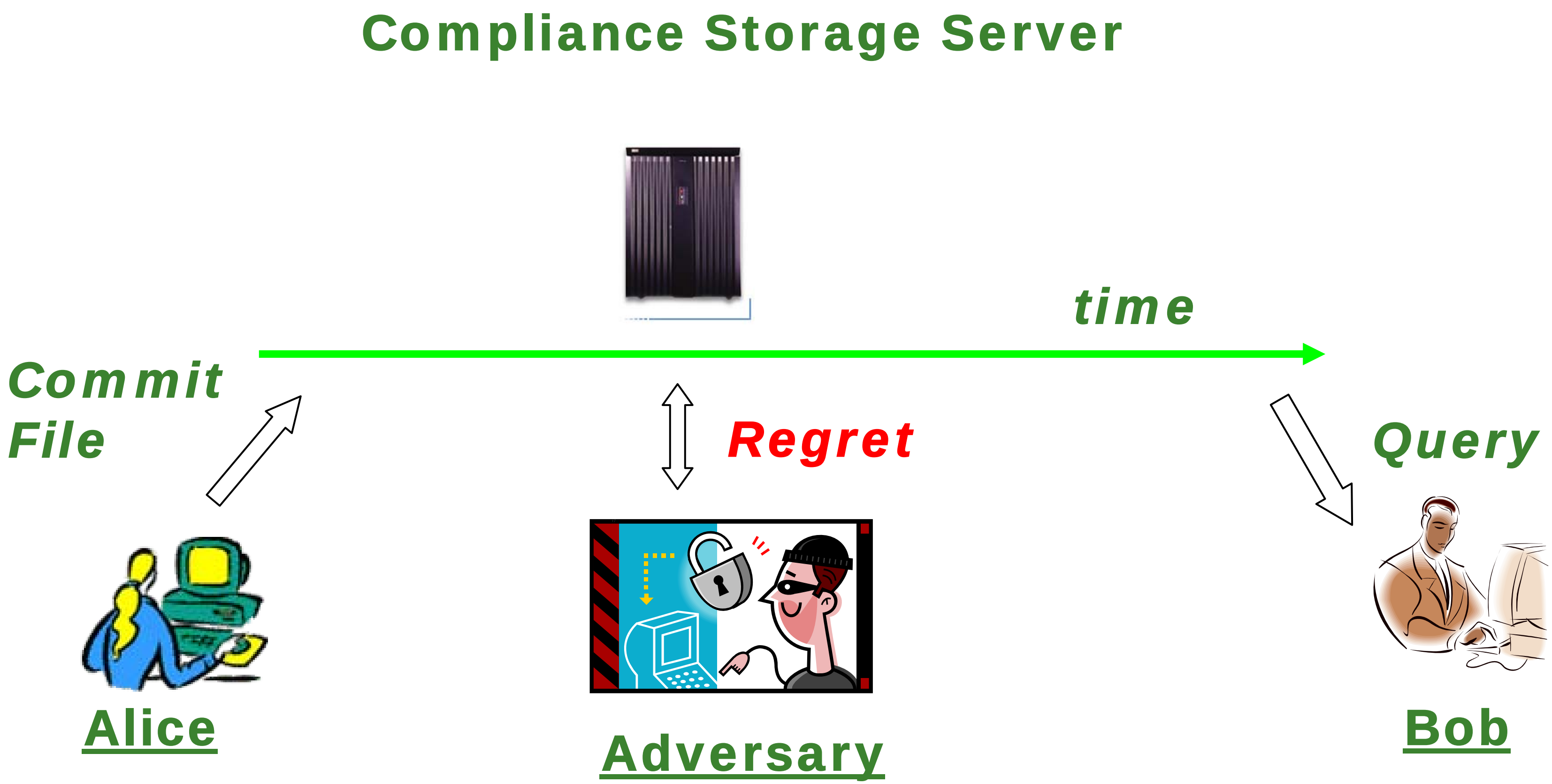
University of Illinois at Urbana-Champaign

# The compliance storage threat model & its implications

Threat model (didn't we hear this already?)

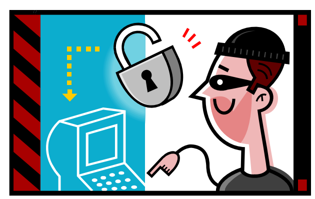
Why traditional approaches to indexing and migration are not trustworthy

# Commit in haste, repent at leisure



# This leads to a unique threat model

***time***



***Commit is trustworthy***

**File is created properly**

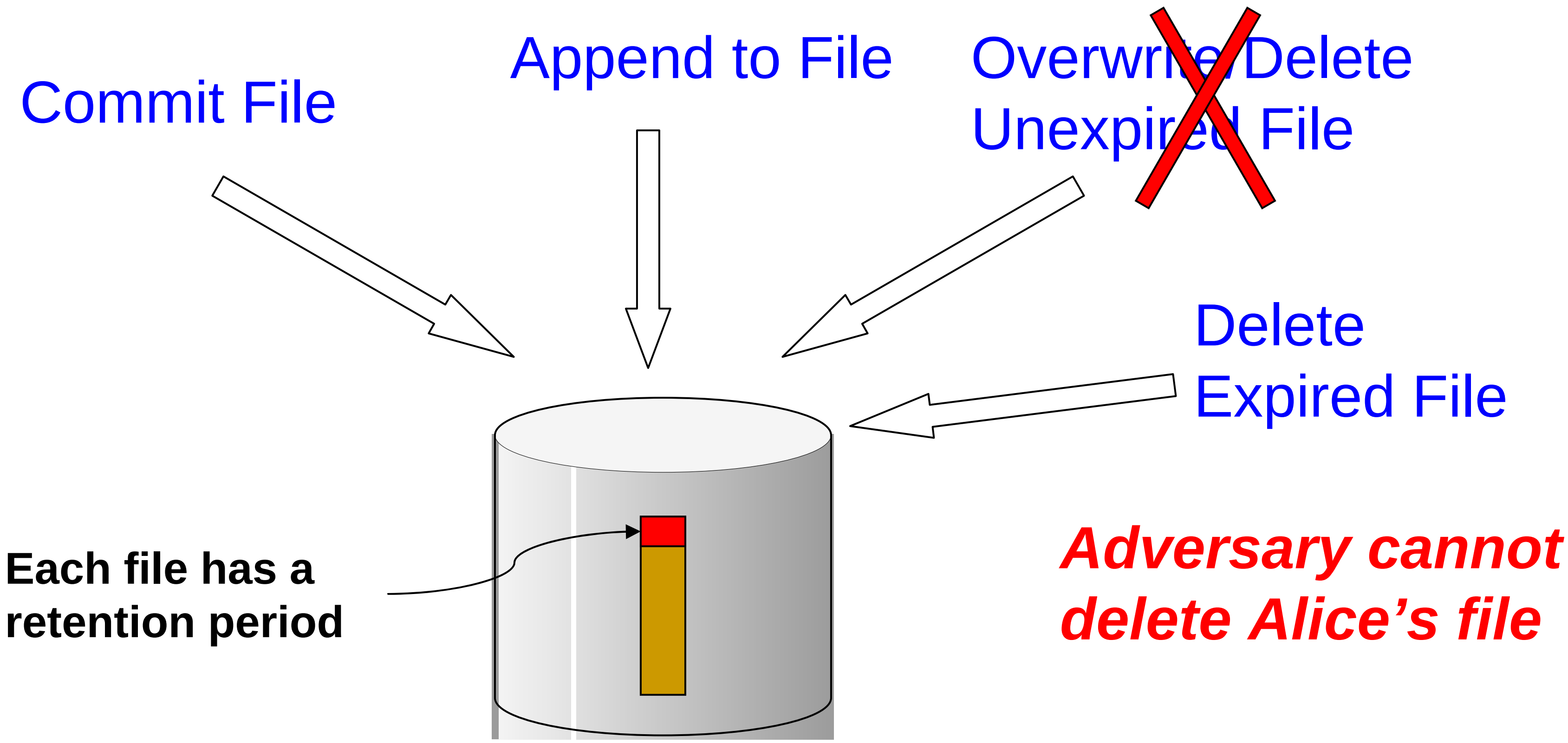
***Adversary has superuser privileges***

**Can access any storage device, any keys**

***Query is trustworthy***

**File is queried properly**

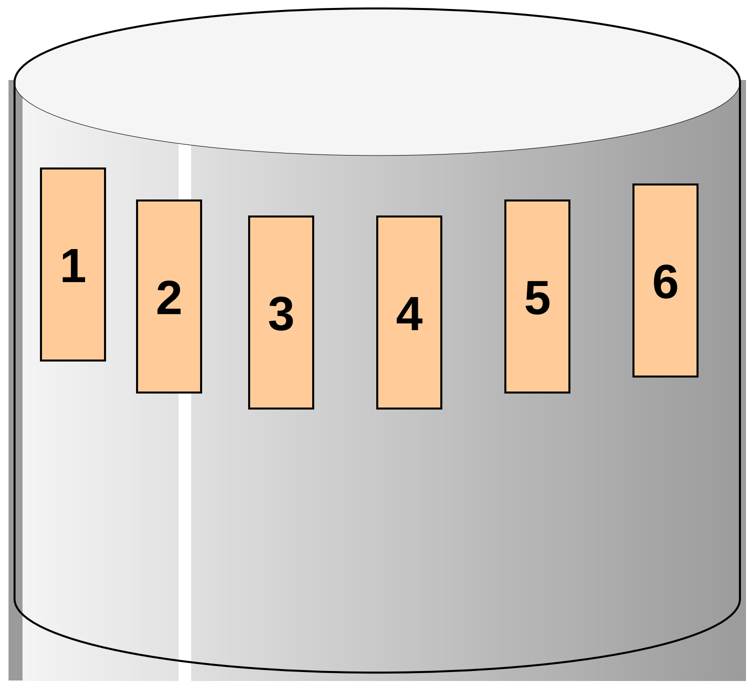
# WORM storage helps address the problem




Write Once, Read Many

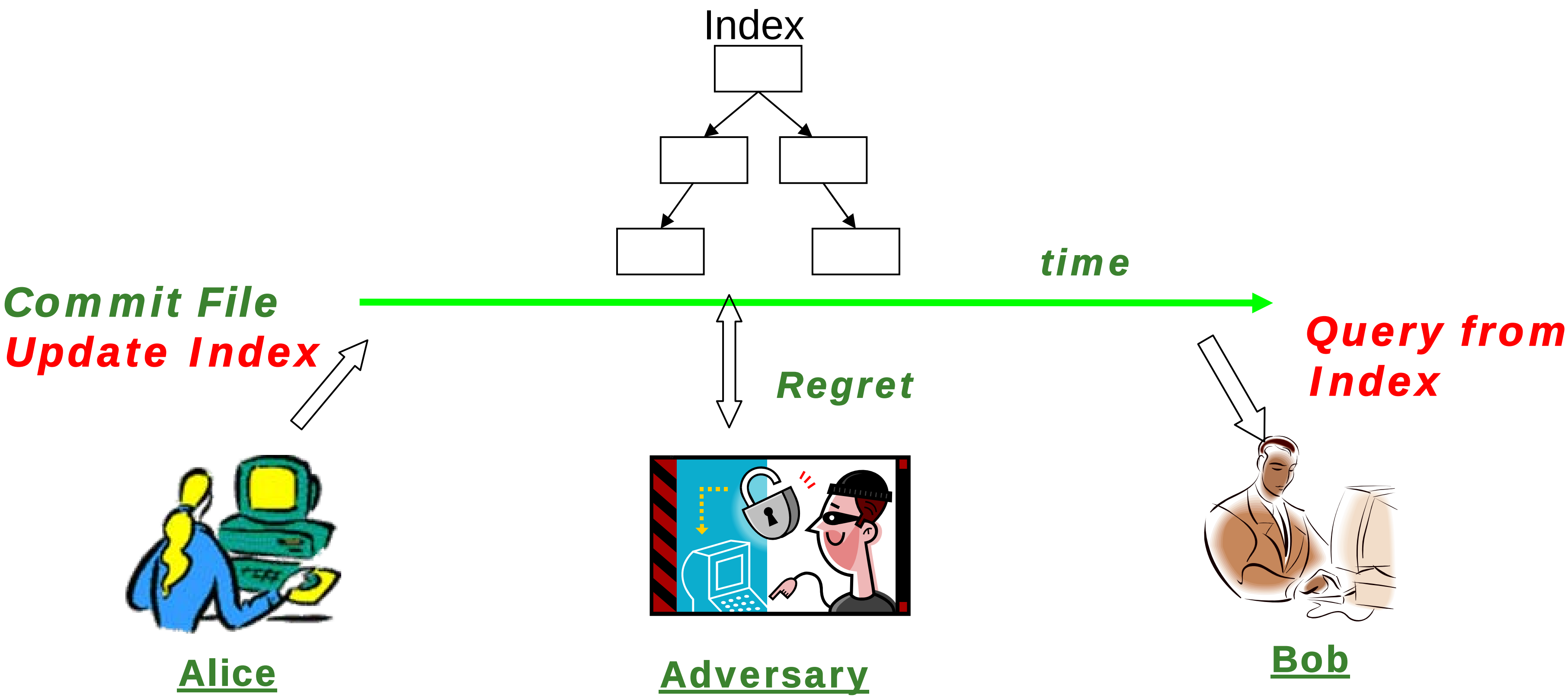
# Data model: File = "record" = document/spreadsheet/email msg = unit of retrieval

**Record IDs increase monotonically over time**




**Object-based compliance storage servers also exist**

# Indexing is needed if there are many compliance records



# In effect, records can be hidden/altered by modifying the index

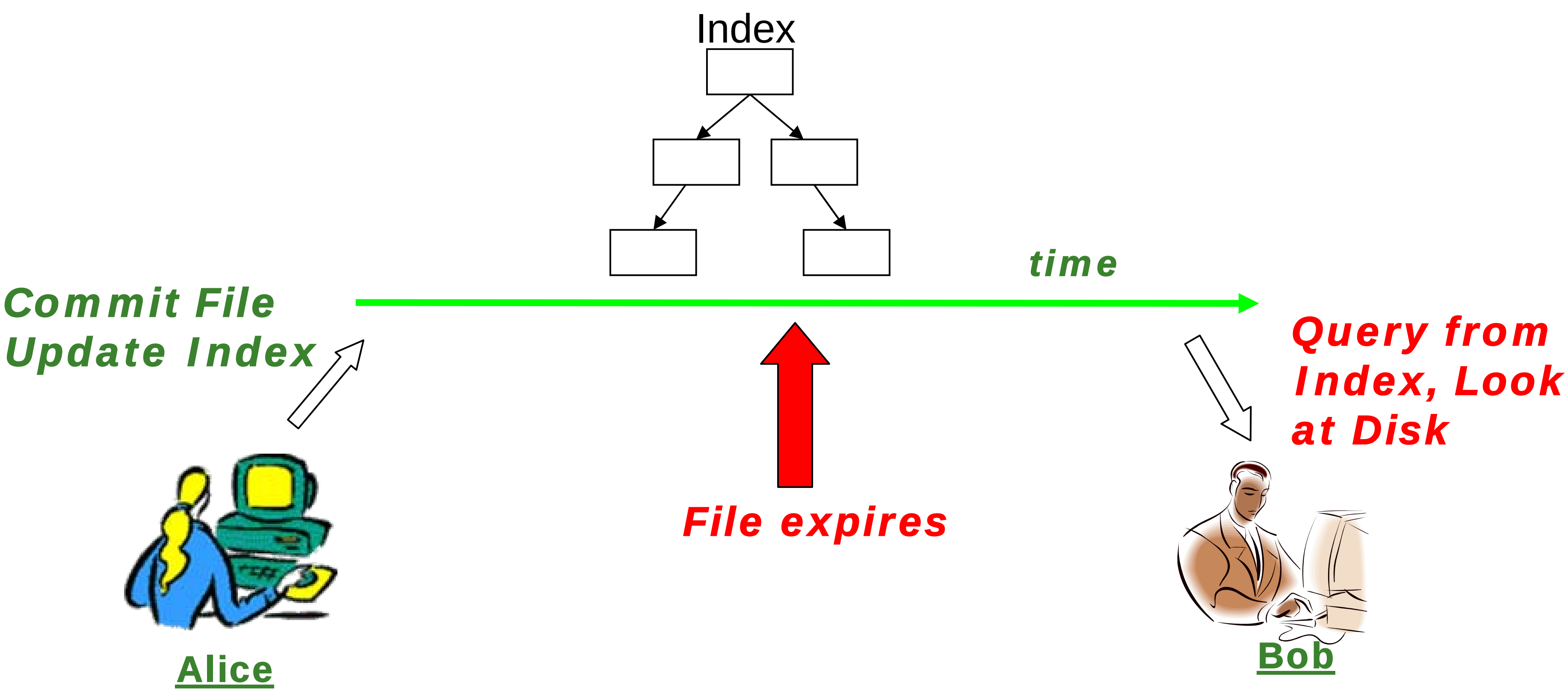
***The index must also be trustworthy!***

# Why don't we run the index code on the (trusted) storage server?

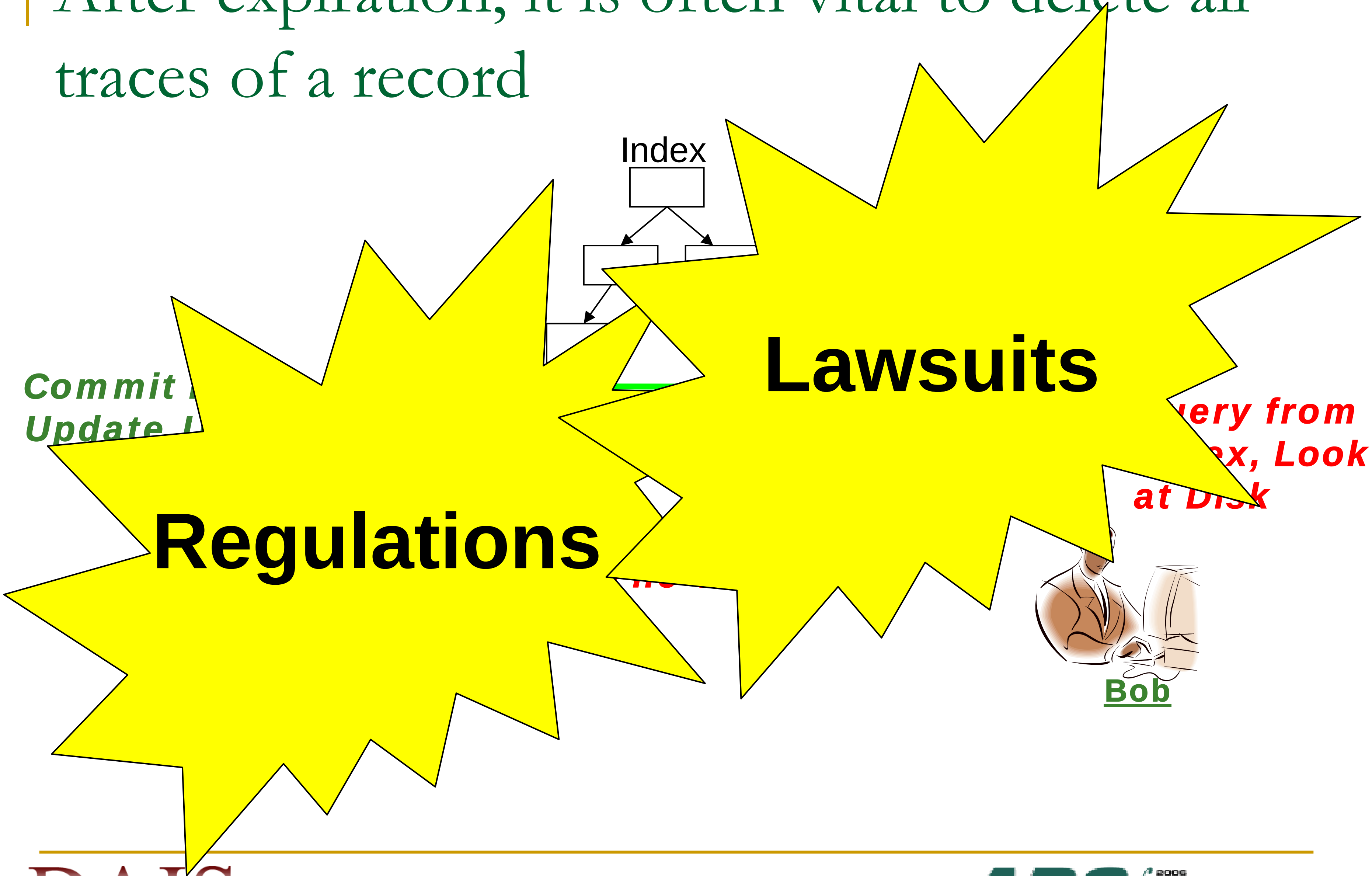
# Why don't we run the index code on the (trusted) storage server?


***The index code should be untrusted!***

# After expiration, it is often vital to delete all traces of a record



# After expiration, it is often vital to delete all traces of a record



# In effect, records can be kept visible by not removing them completely from the index

67: Ralph

67: Martha

67: ImClone

67: Sell

67: Now

# In effect, records can be kept visible by not removing them completely from the index

# Data sometimes must be migrated to a different server

**HIPAA**
**21 years**
**(babies)**

**SOX**
**5 years**

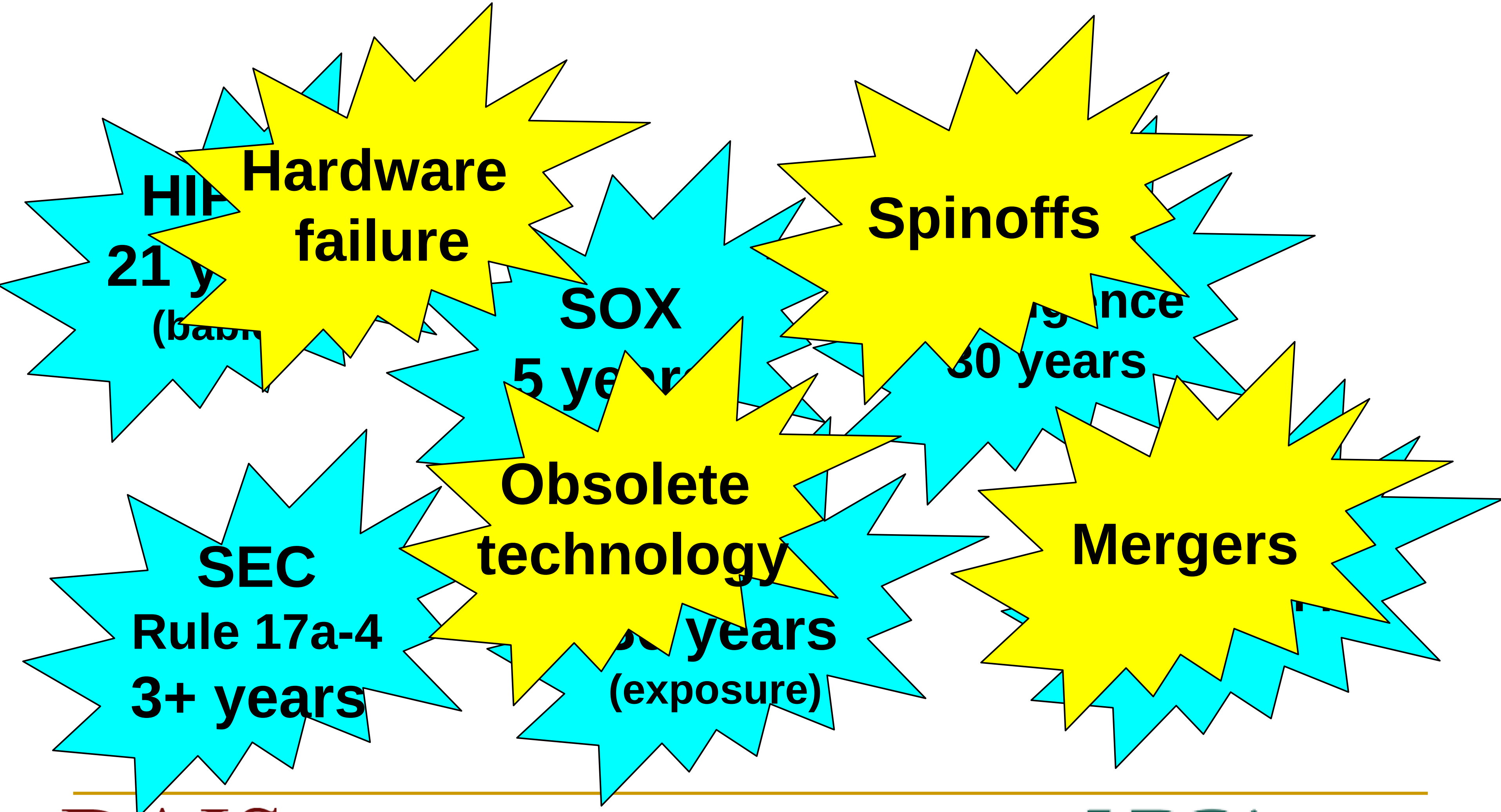
**National Intelligence**
**30 years**

**SEC**
**Rule 17a-4**
**3+ years**

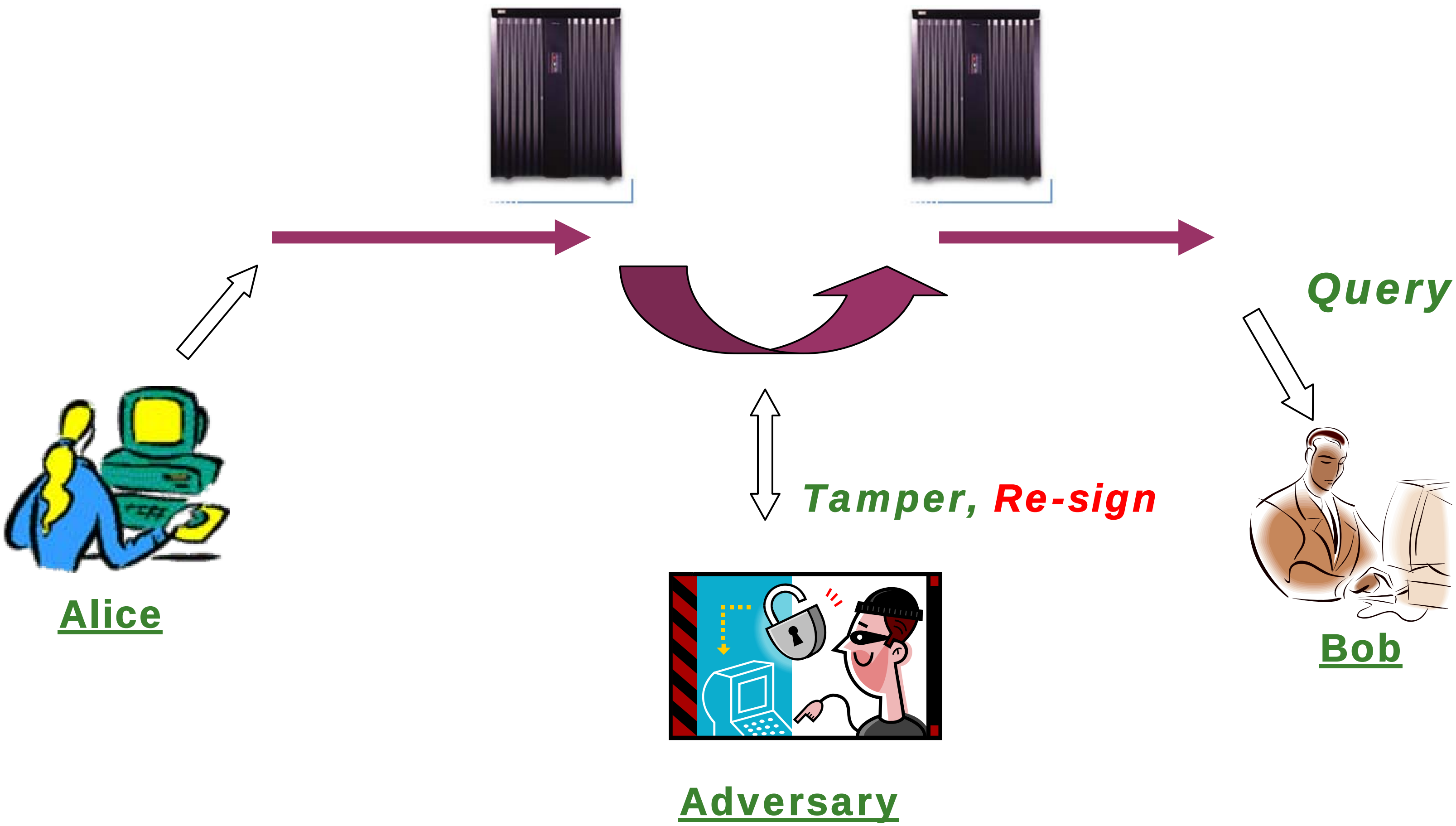
**OSHA**
**30 years**
**(exposure)**

**FERPA**
**Forever?**

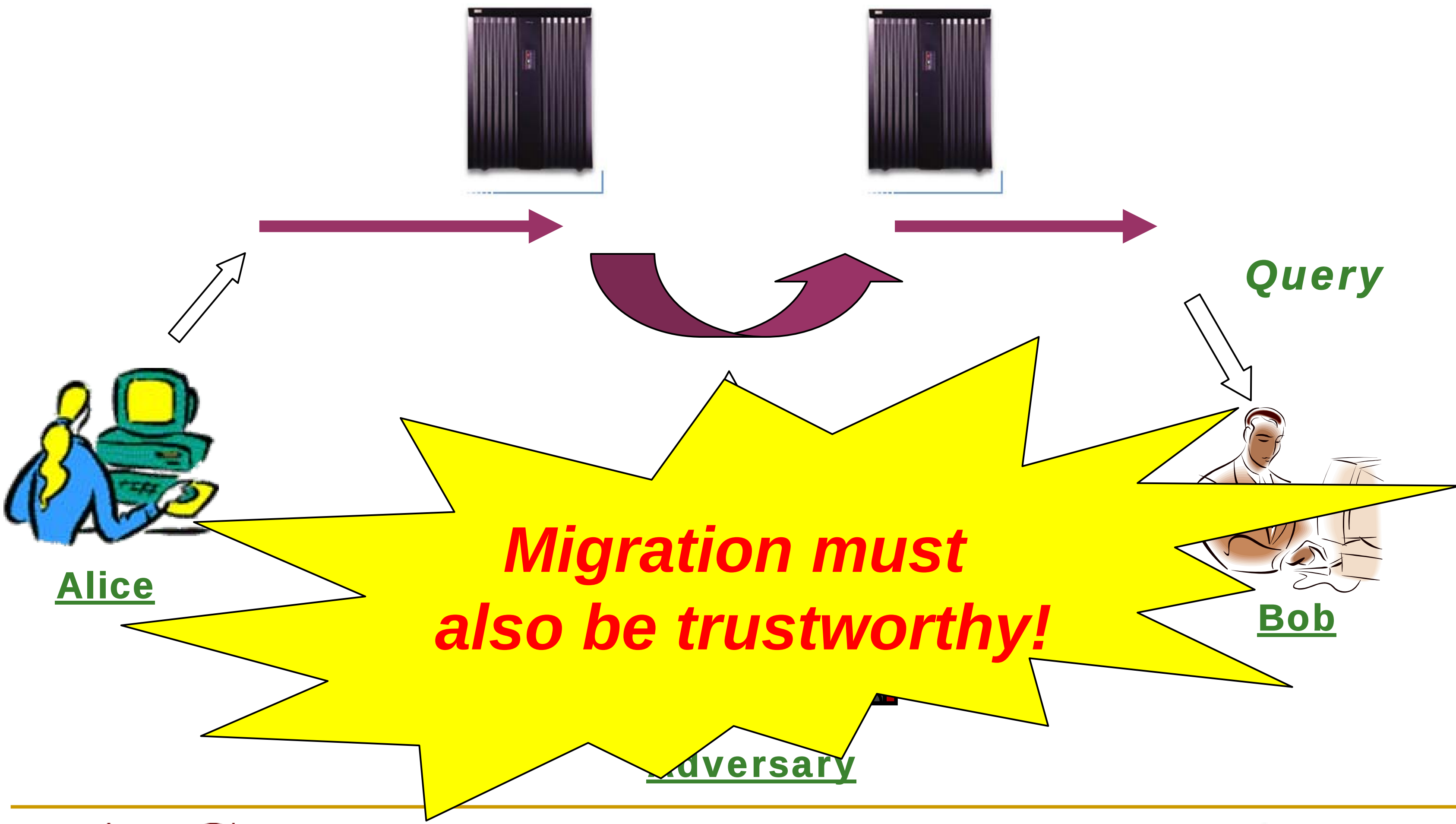
# Data sometimes must be migrated to a different server



# Alice's signature cannot protect a record during migration



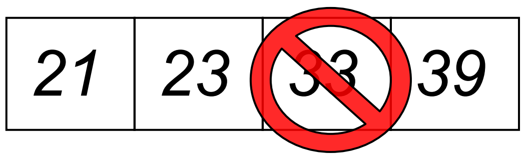
# Alice's signature cannot protect a record during migration



Conclusion: basic compliance data lifecycle needs trustworthy record retention, indexing, migration, and deletion.

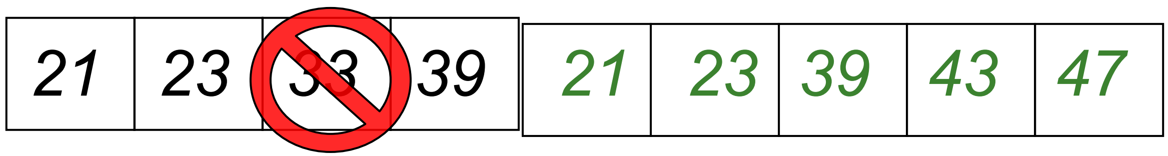
Are traditional indexes trustworthy?

# Binary search is not trustworthy, even on WORM



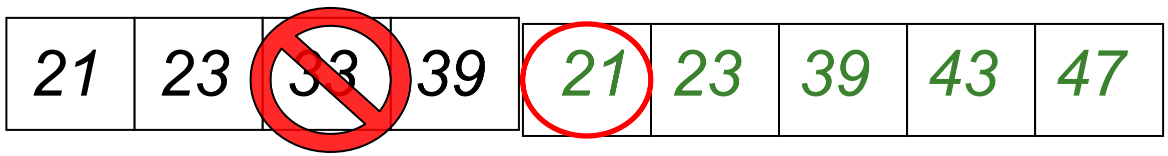

**How to hide 33**
**(and nothing else)**

# Binary search is not trustworthy, even on WORM



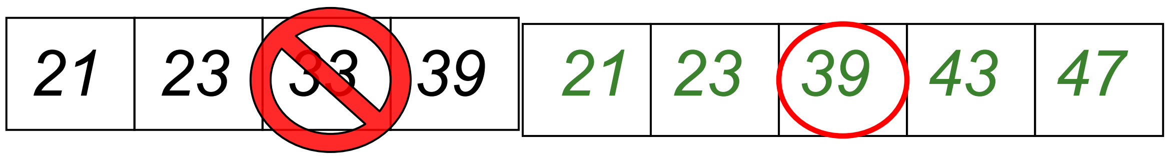

**How to hide 33**
**(and nothing else)**

# Binary search is not trustworthy, even on WORM



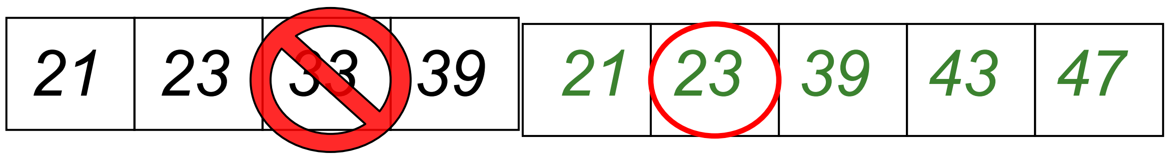

**How to hide 33**
**(and nothing else)**

# Binary search is not trustworthy, even on WORM




**How to hide 33**
**(and nothing else)**

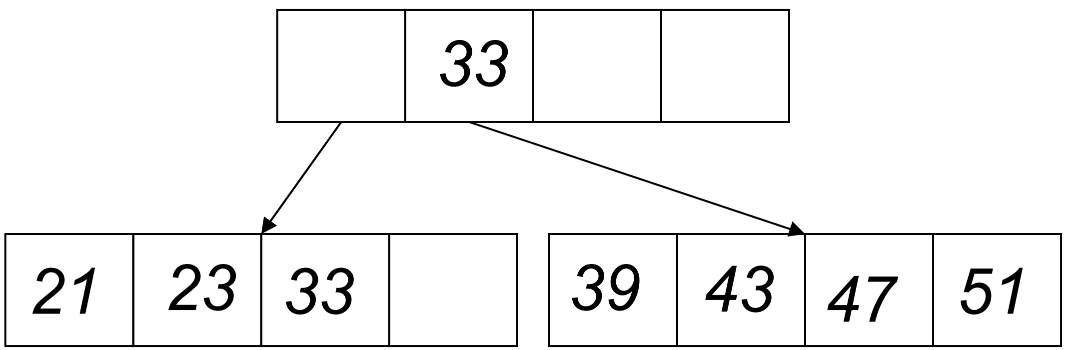
# Binary search is not trustworthy, even on WORM




**How to hide 33**
**(and nothing else)**

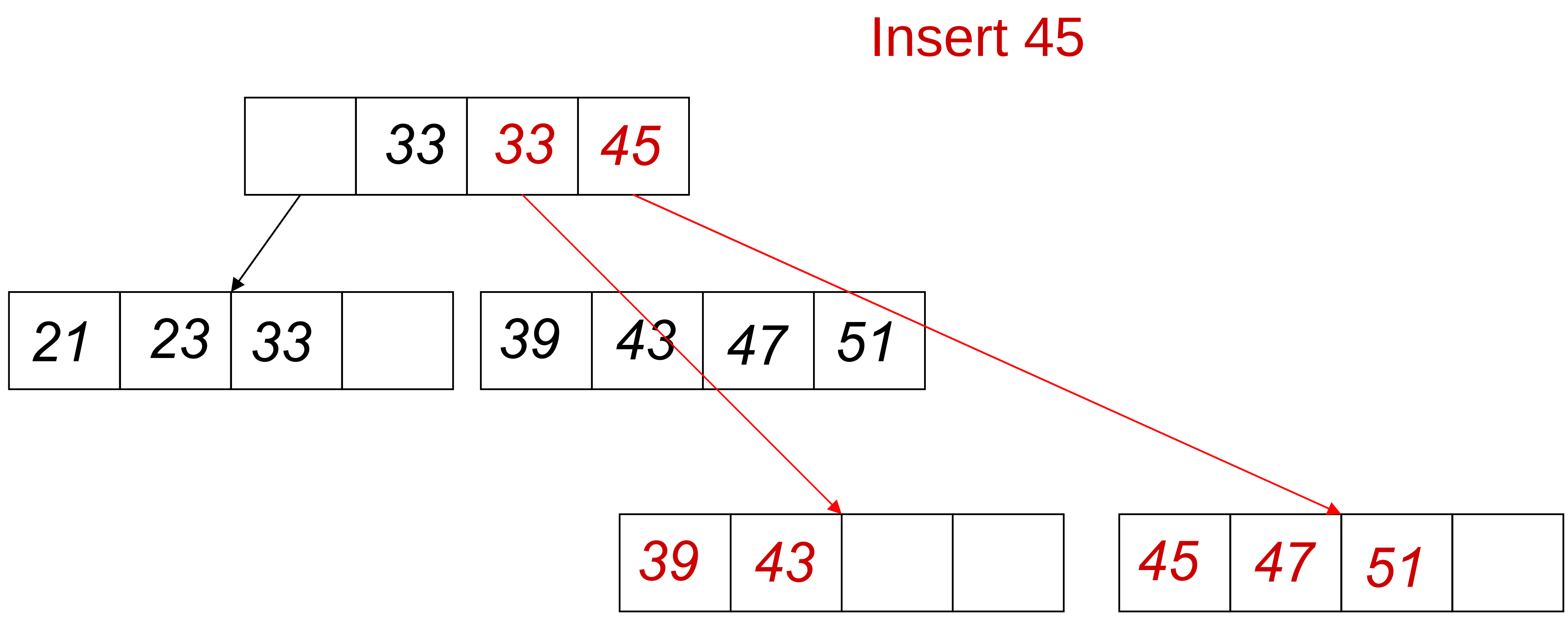
# We can put a B-tree on WORM, either bottom-up (easy) or with node splitting

[Rathmann, ICDE 84; Easton, IBM J. 86]



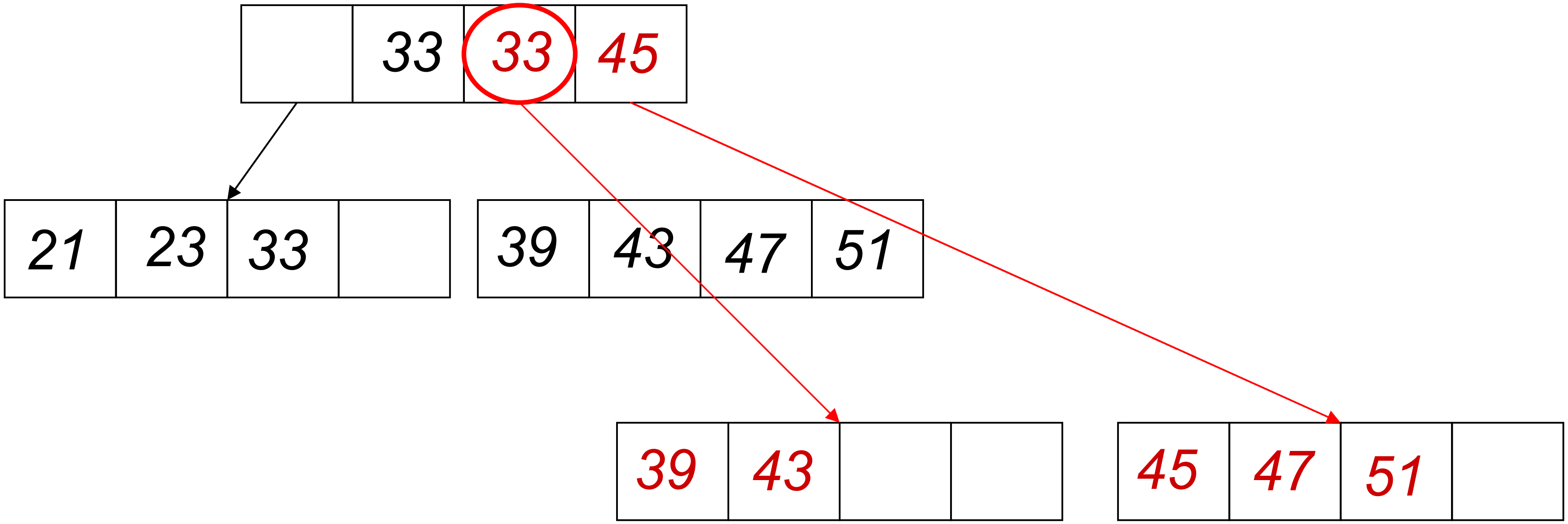

**One B-tree node per (appendable) file**

# Copy a node to split it; append corrected entries to its parent



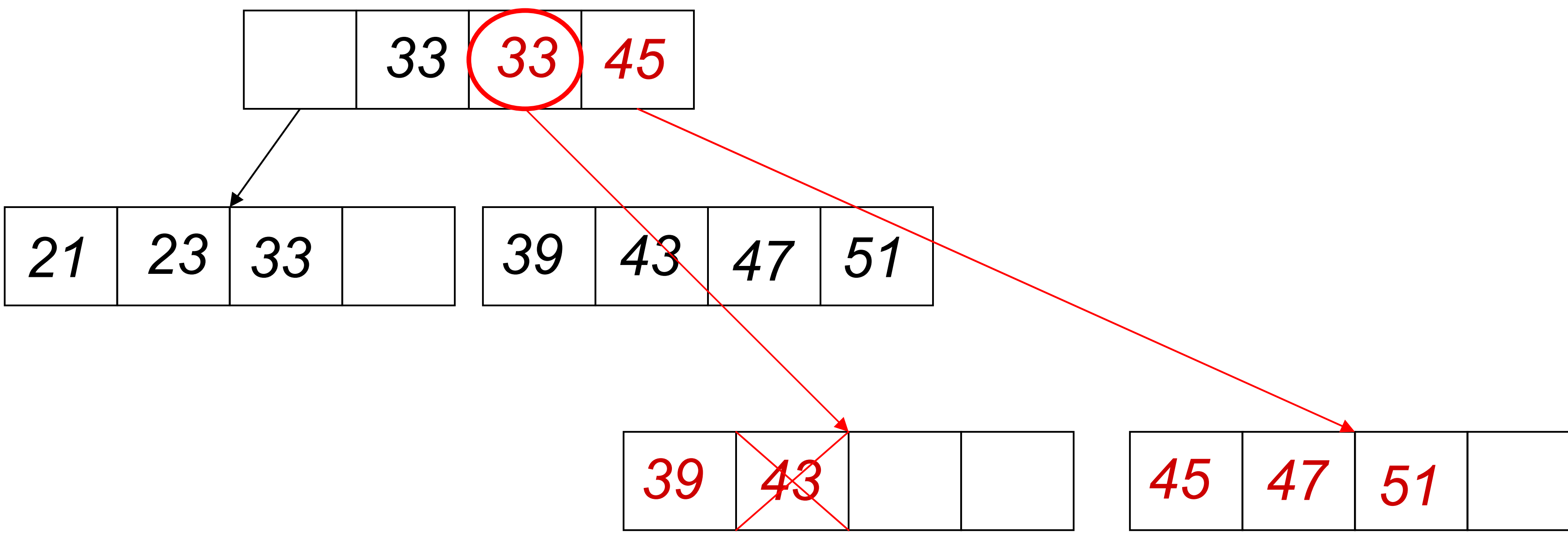
# Lookup follows the newest node

Look up 43

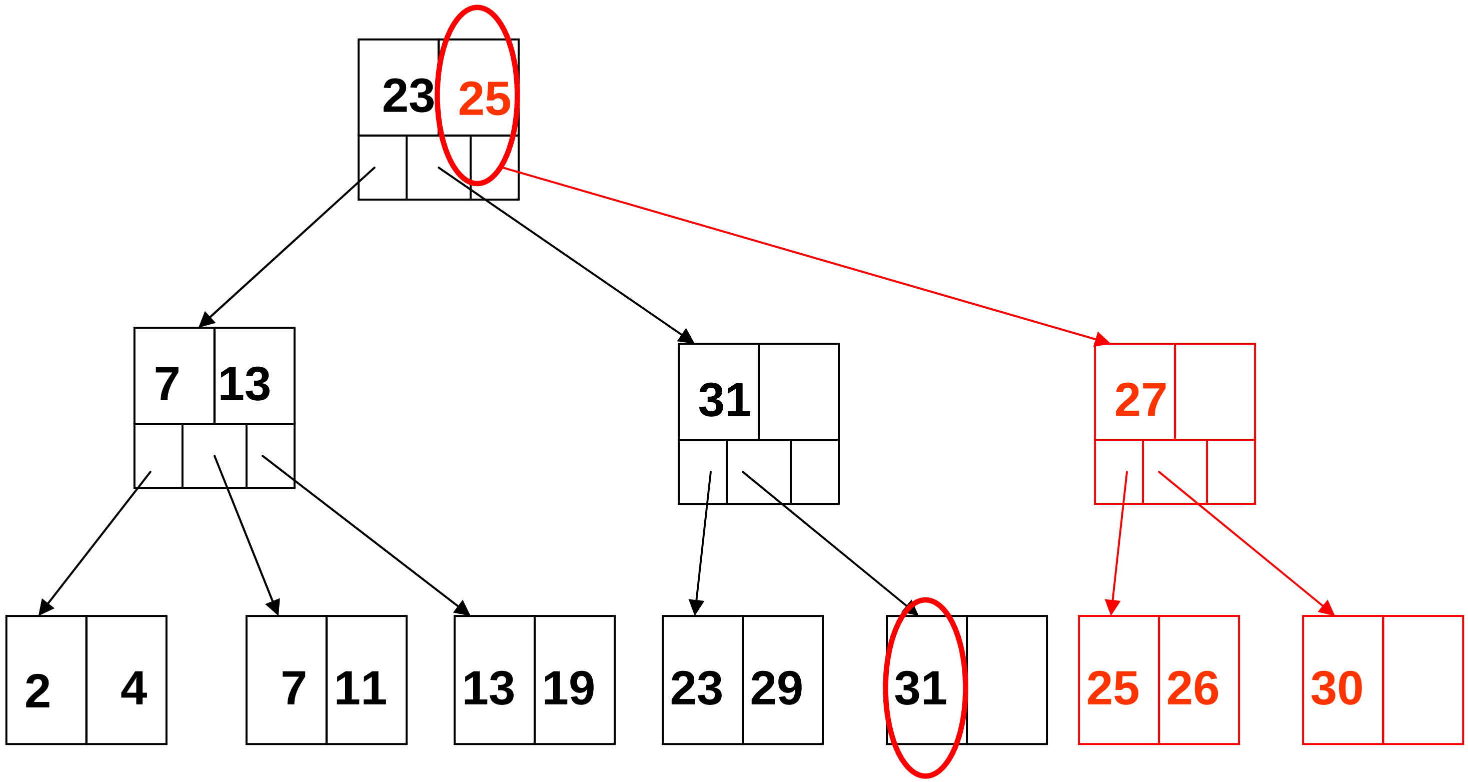


# B-trees are not trustworthy on WORM: the adversary can omit values during copying

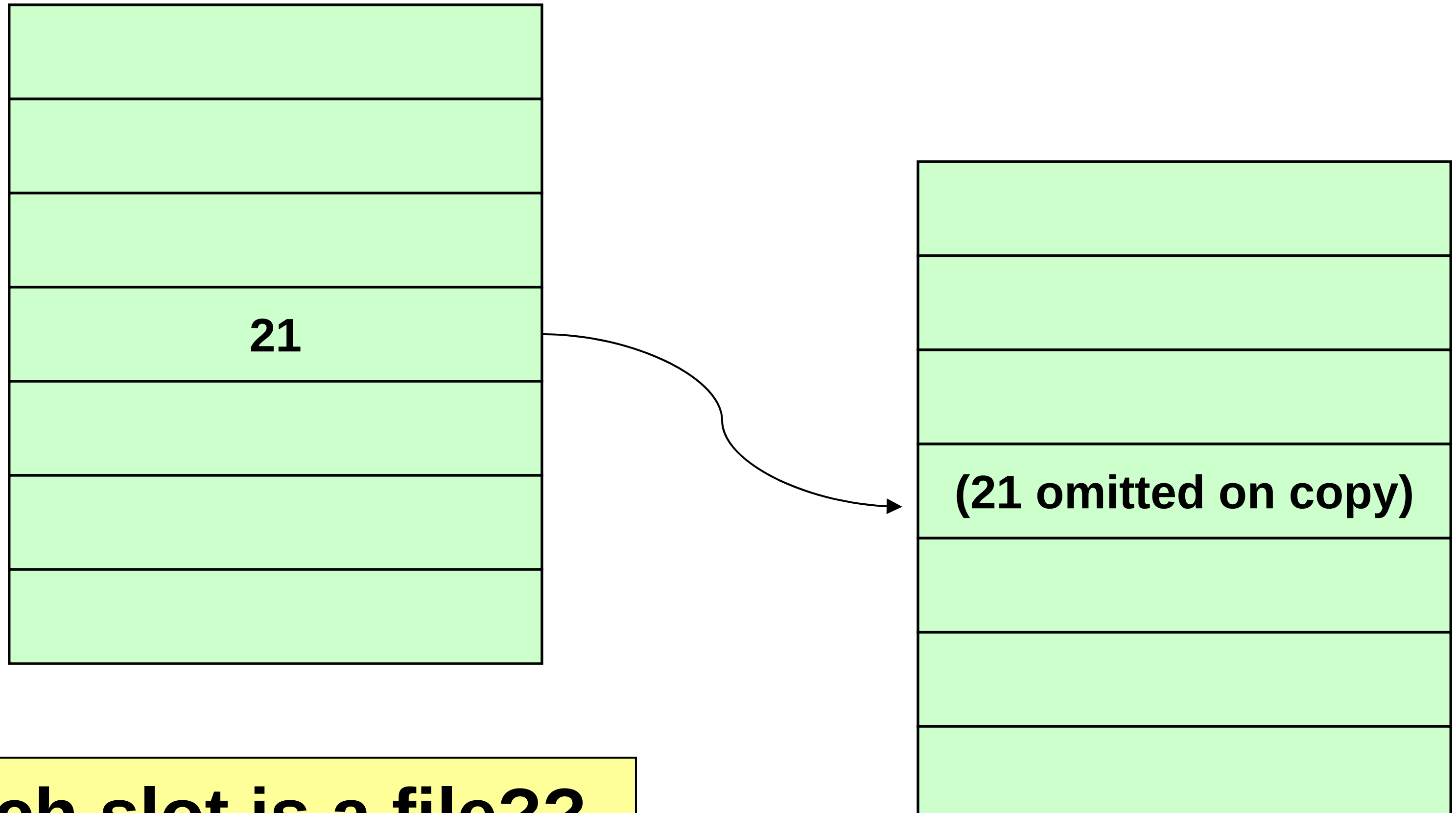
Lookup of 43 fails



# Even built bottom-up without node splits, B-trees on WORM are not trustworthy

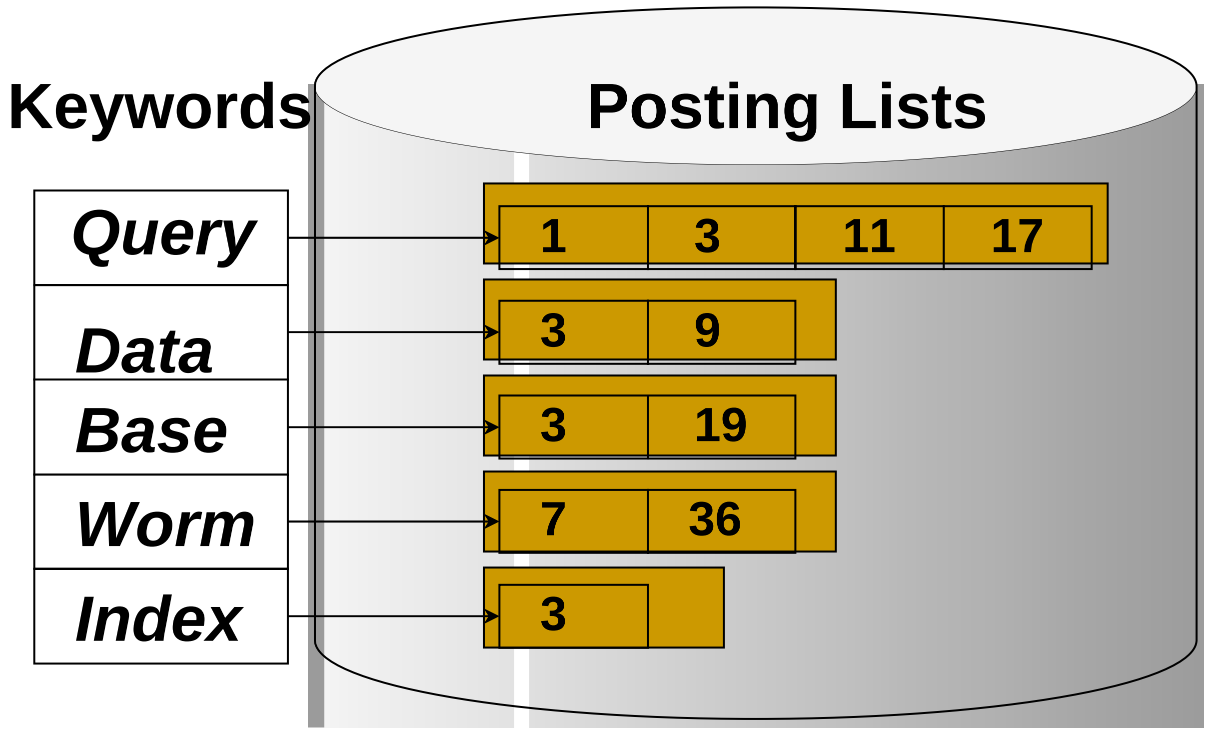


# Linear/extensible/dynamic hashing is bad on WORM, and copying is not trustworthy



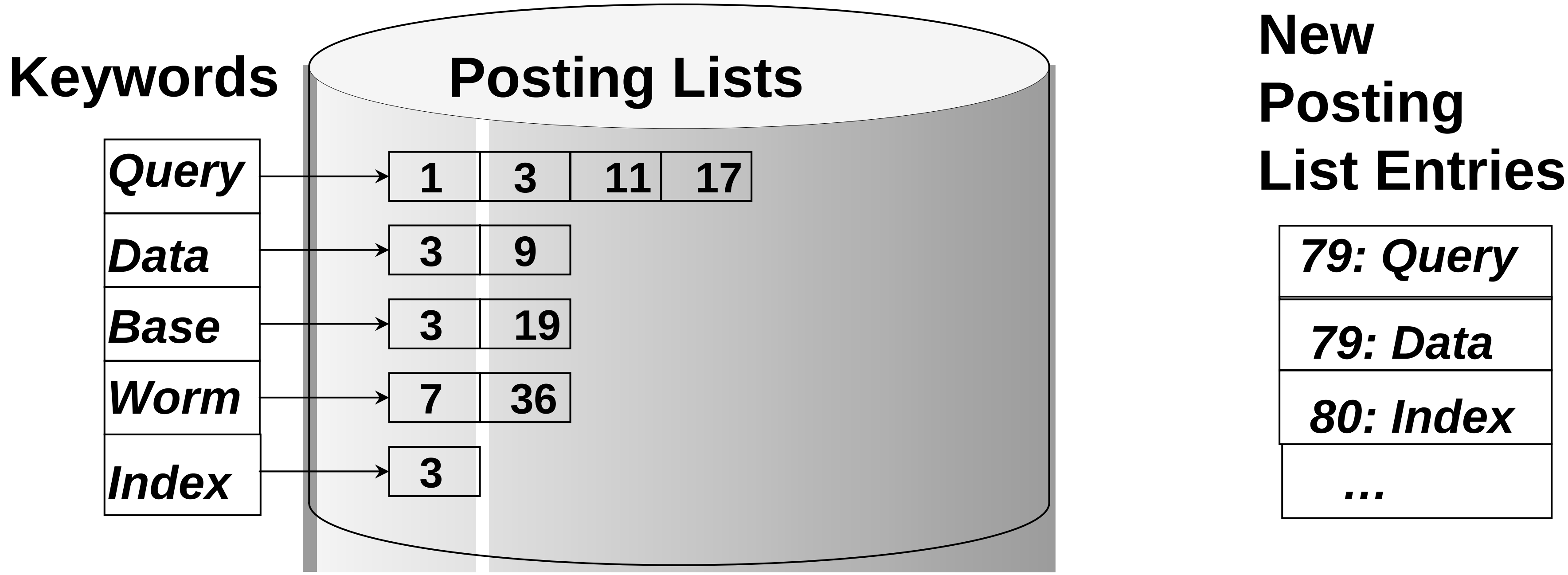

**Each slot is a file??**

# Unstructured/semistructured data need inverted indexes




One WORM file for each posting list

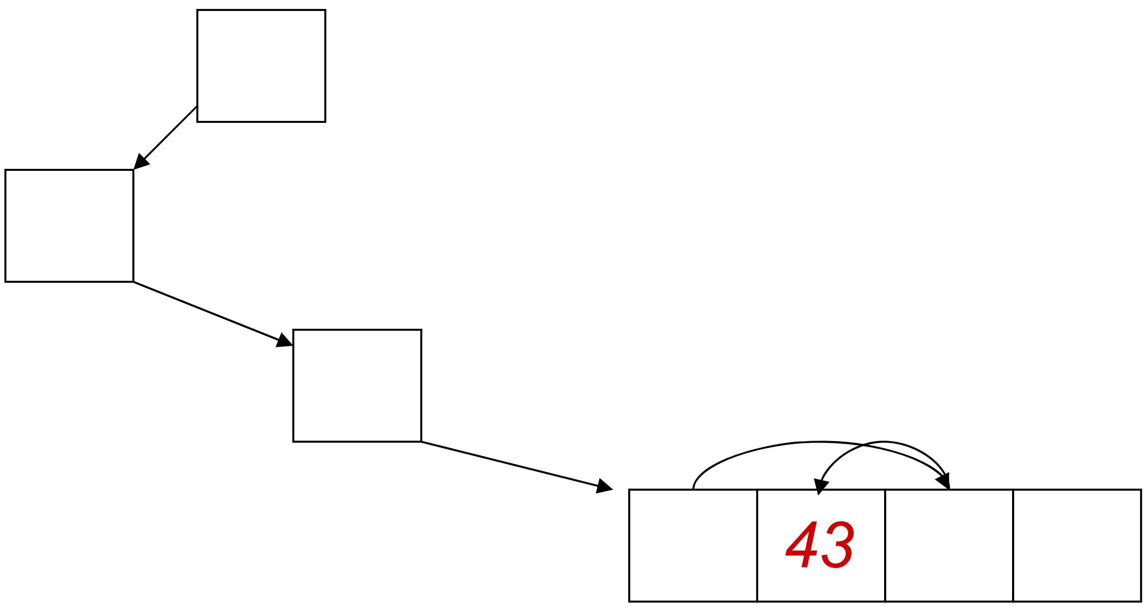
# Traditional inverted indexes are not trustworthy, even on WORM




**Delta file of updates + periodic sort-and-merge = opportunity for adversary**

# Conclusion: ***No*** traditional index is trustworthy

The search path to an item cannot depend on what is inserted later.



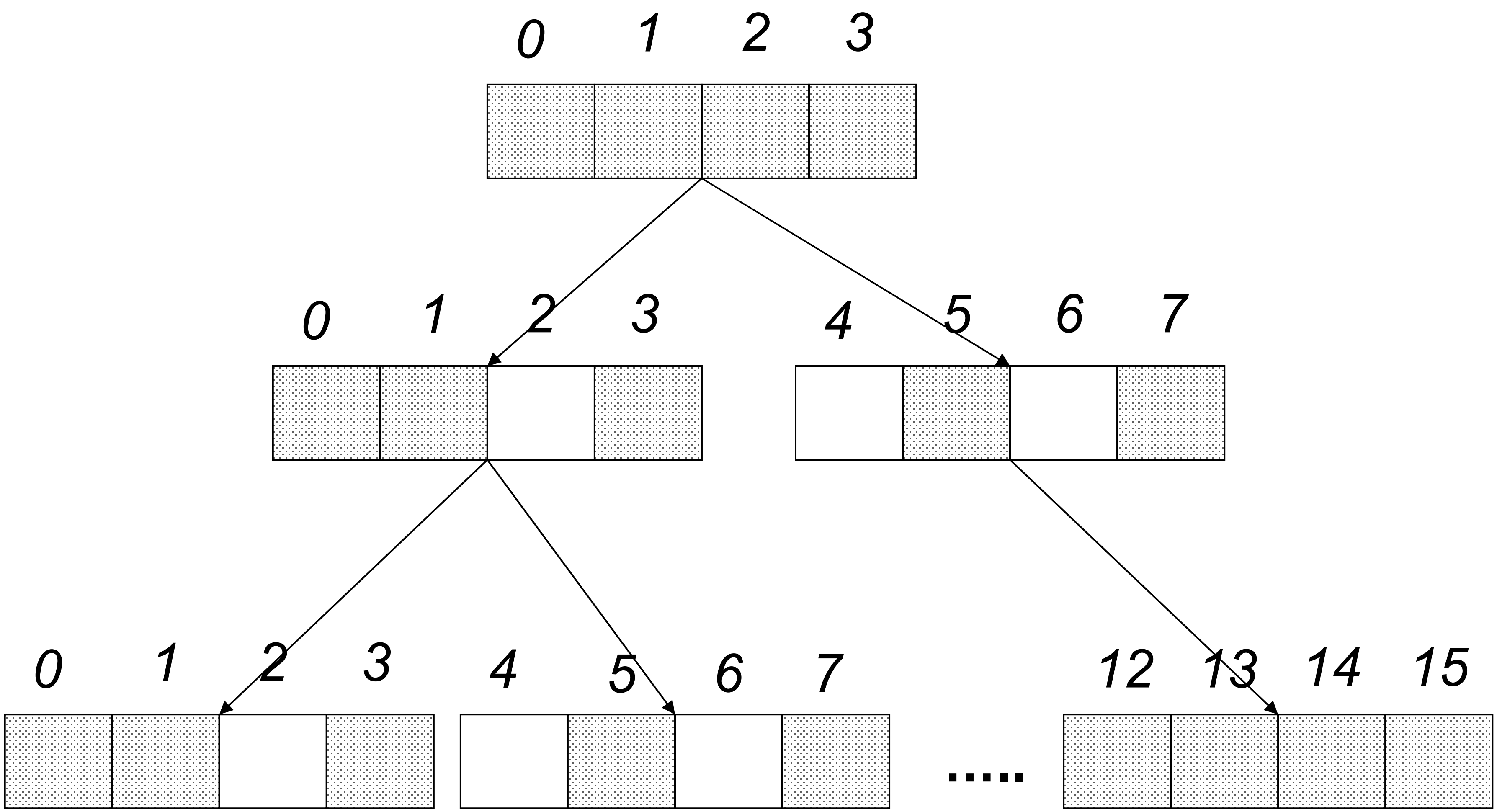
# Trustworthy Indexing:

## Generalized Hash Trees
## Trustworthy Inverted Indexes
## Jump Indexes

[Zhu & Hsu, SIGMOD 05]

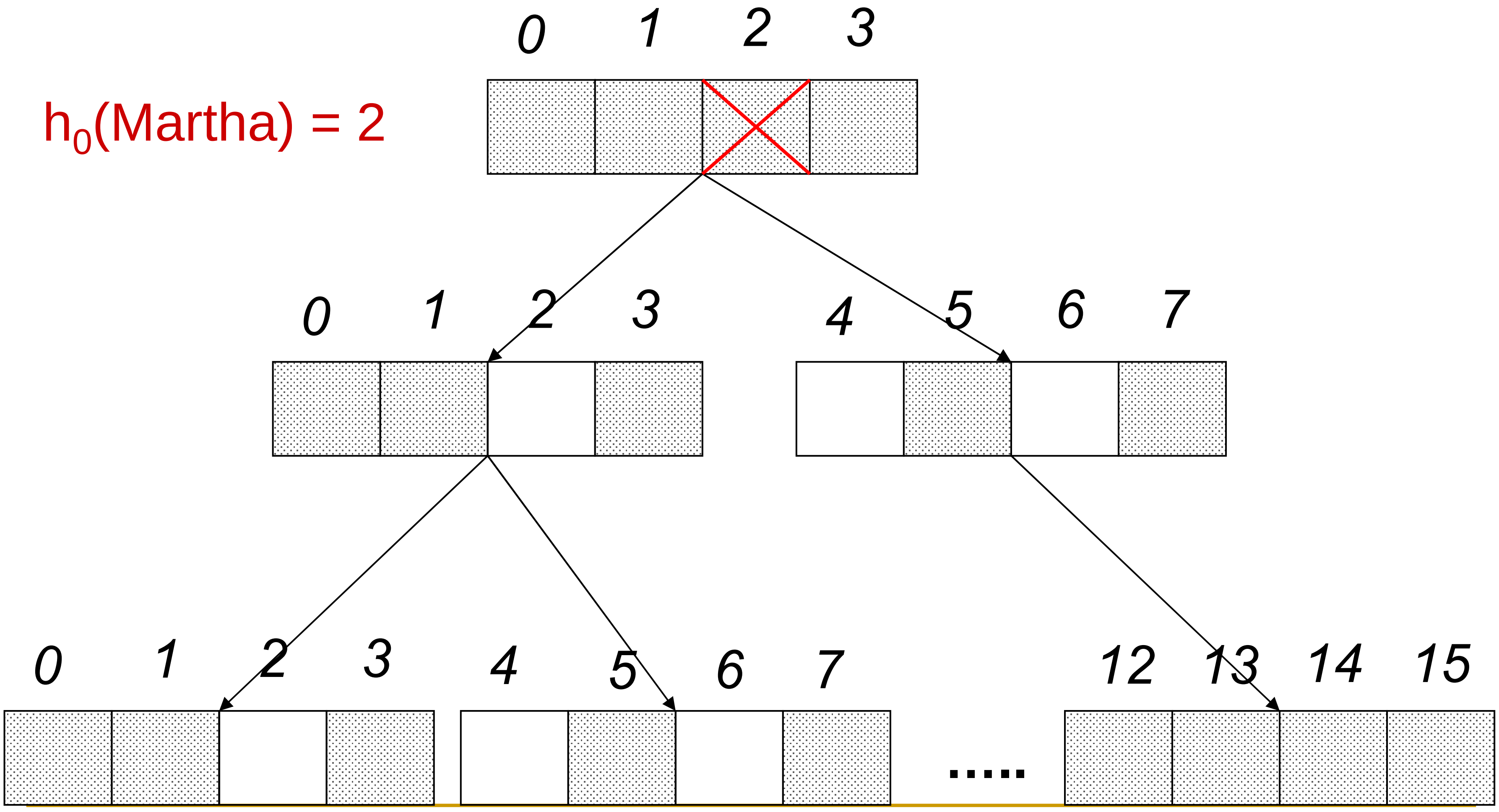
[Mitra, Hsu, & Winslett, VLDB 06 / VLDBJ 07]

# Generalized Hash Trees are trustworthy



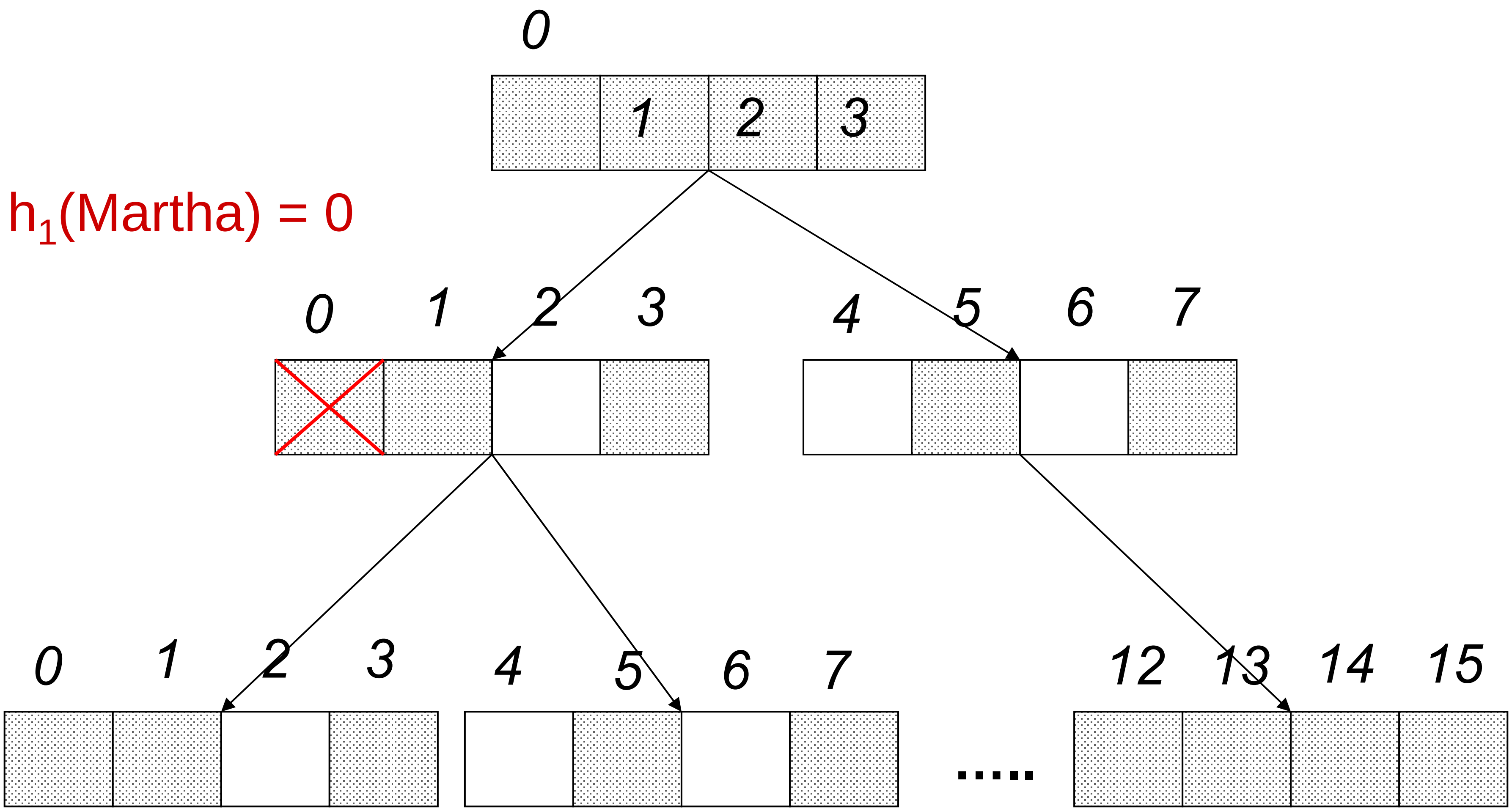
# On collision, go to next level

$h_0(\text{Martha}) = 2$

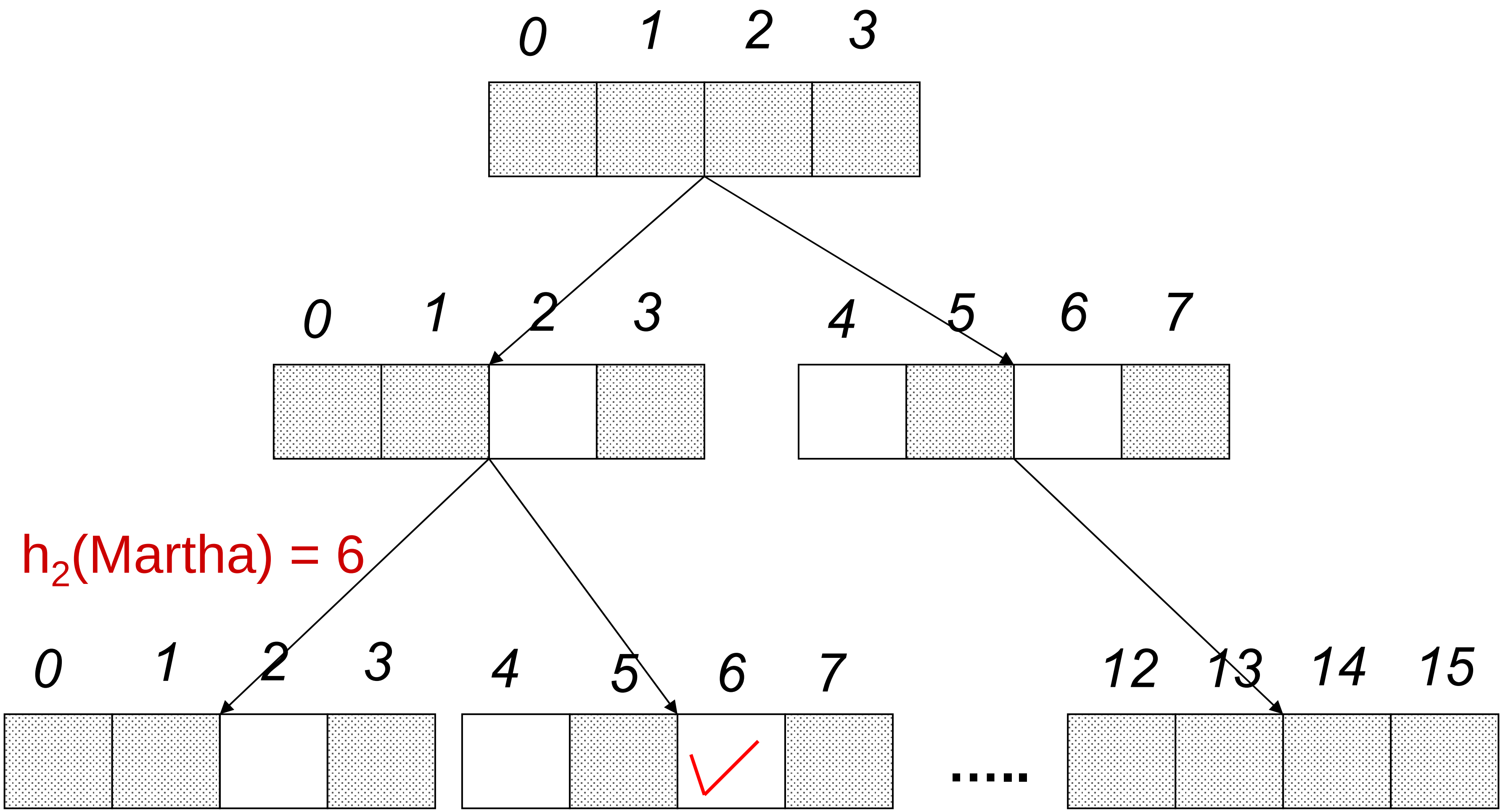


# Rehash at the next level down …

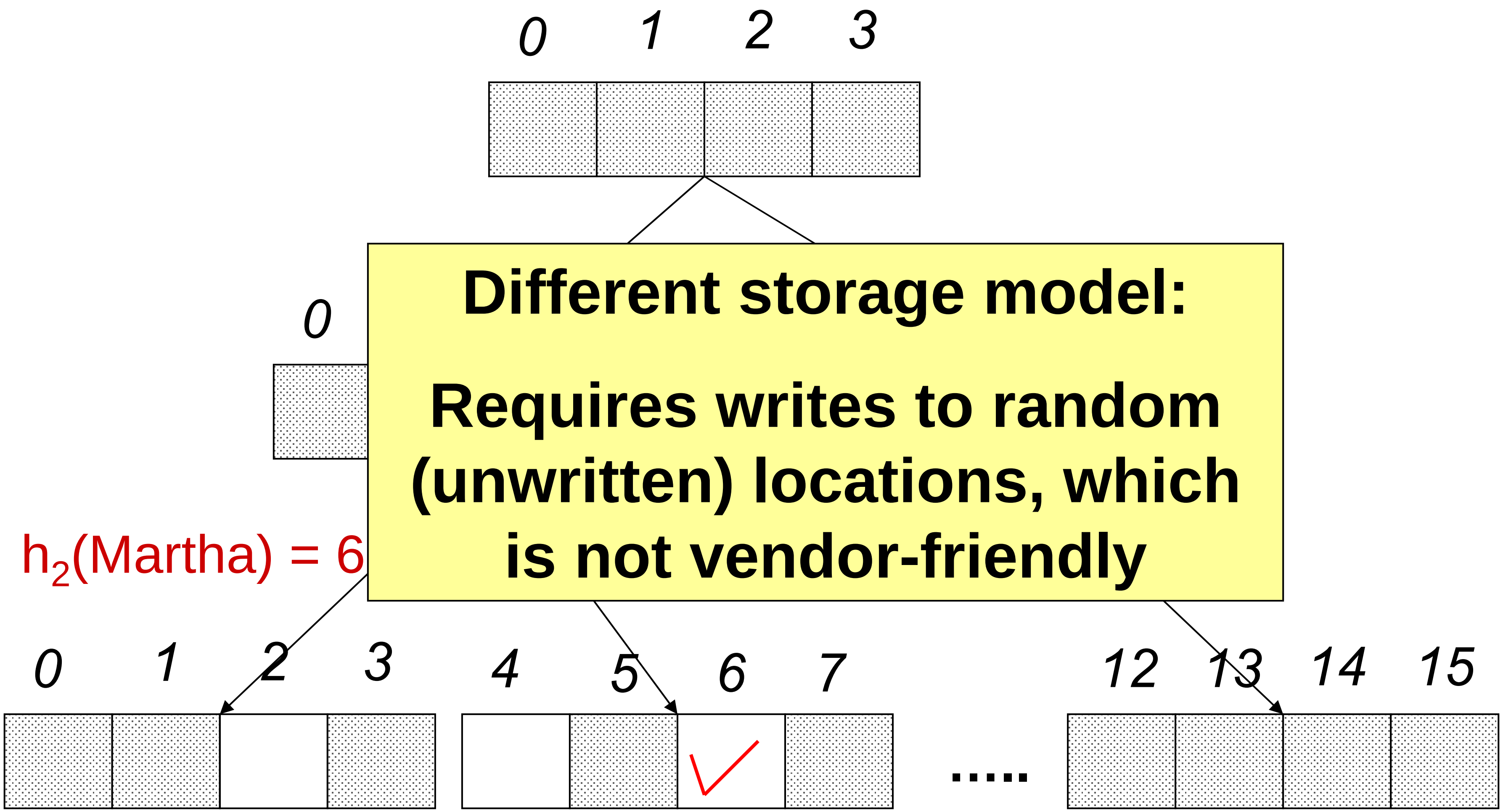
$h_1$(Martha) = 0



# … until you find an empty spot



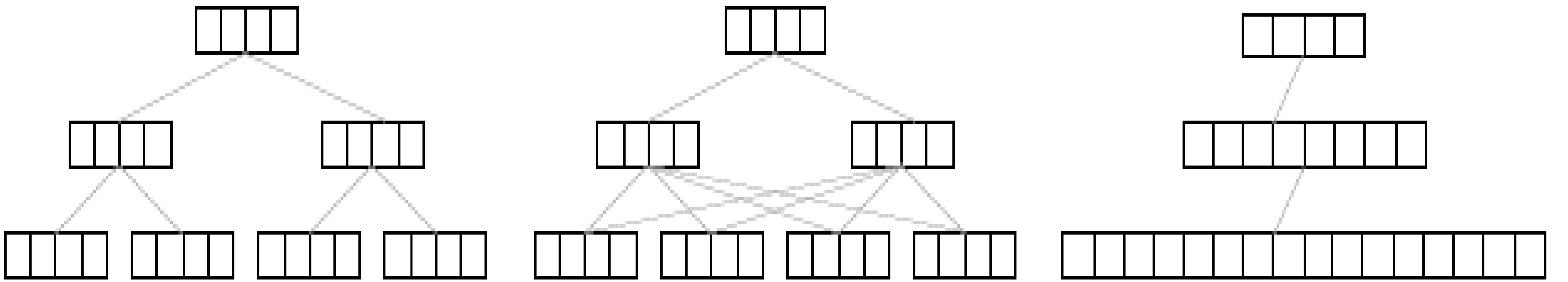
# … until you find an empty spot




**Different storage model:**

**Requires writes to random (unwritten) locations, which is not vendor-friendly**

# GHTs can have many different shapes

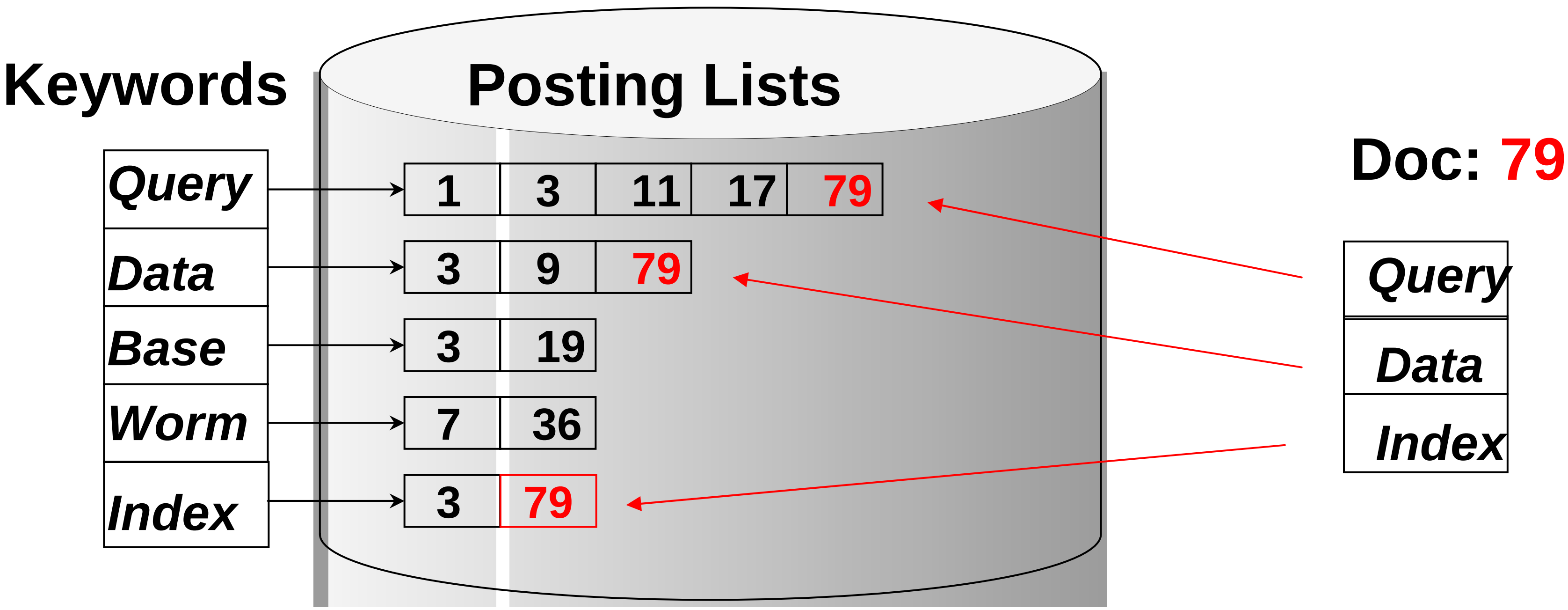


All have O(log n) performance if “good” hash functions are used

Expected size is O(n) to O(n^3), depending on shape

Provably trustworthy

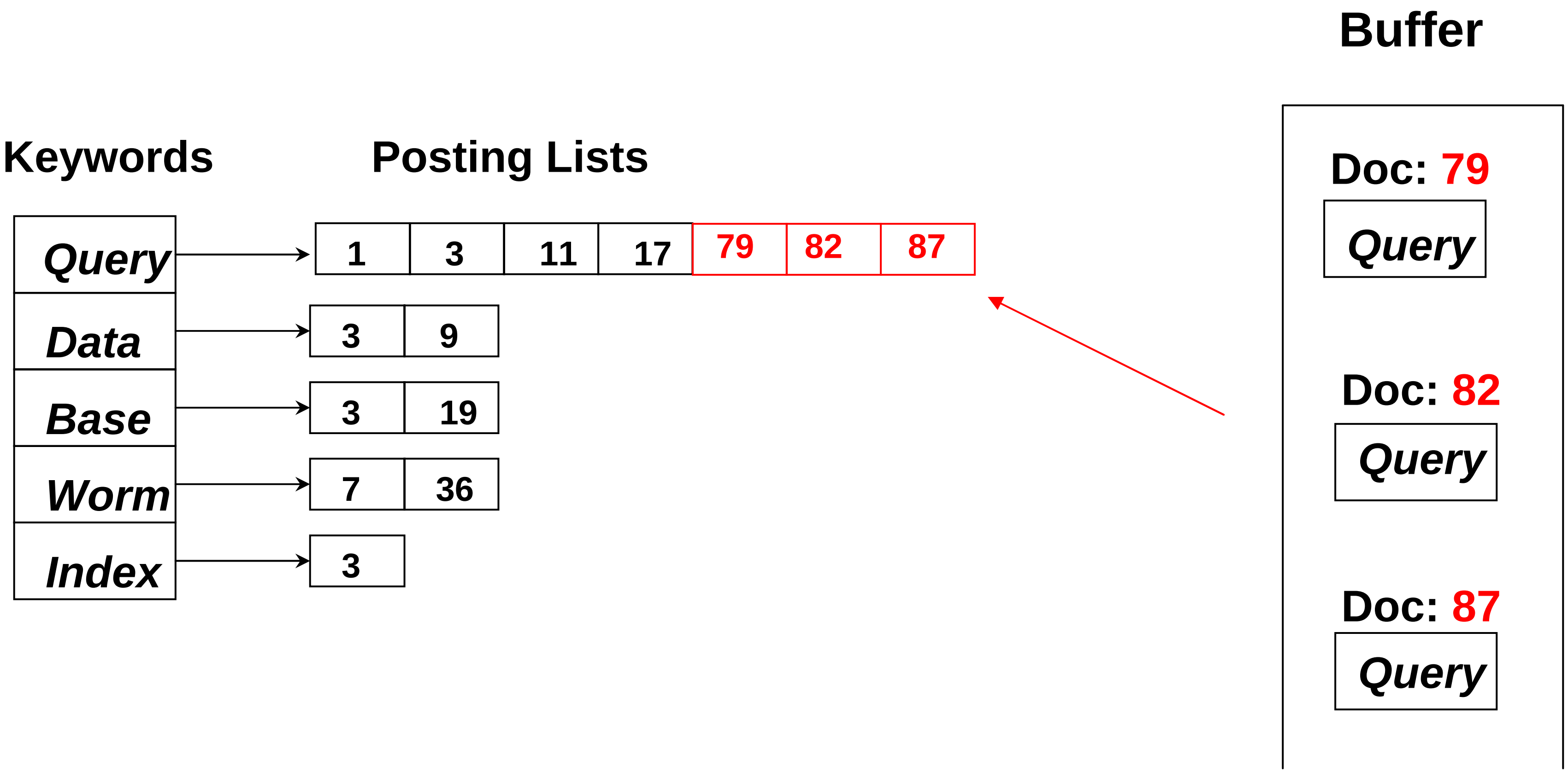
# A trustworthy inverted index must be updated as new documents arrive




500 keywords = 500 disk seeks

~1 sec per document is too slow!

# Amortize the cost by batching updates

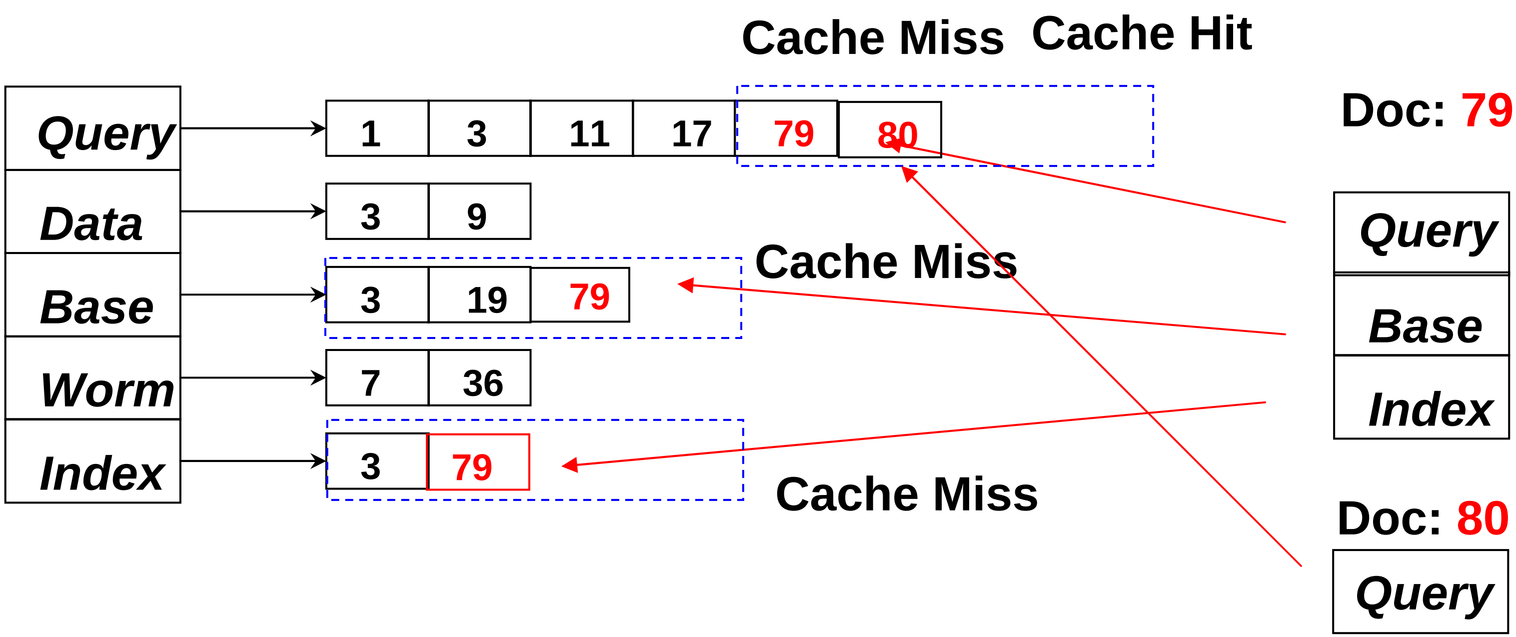



1 seek per keyword in batch

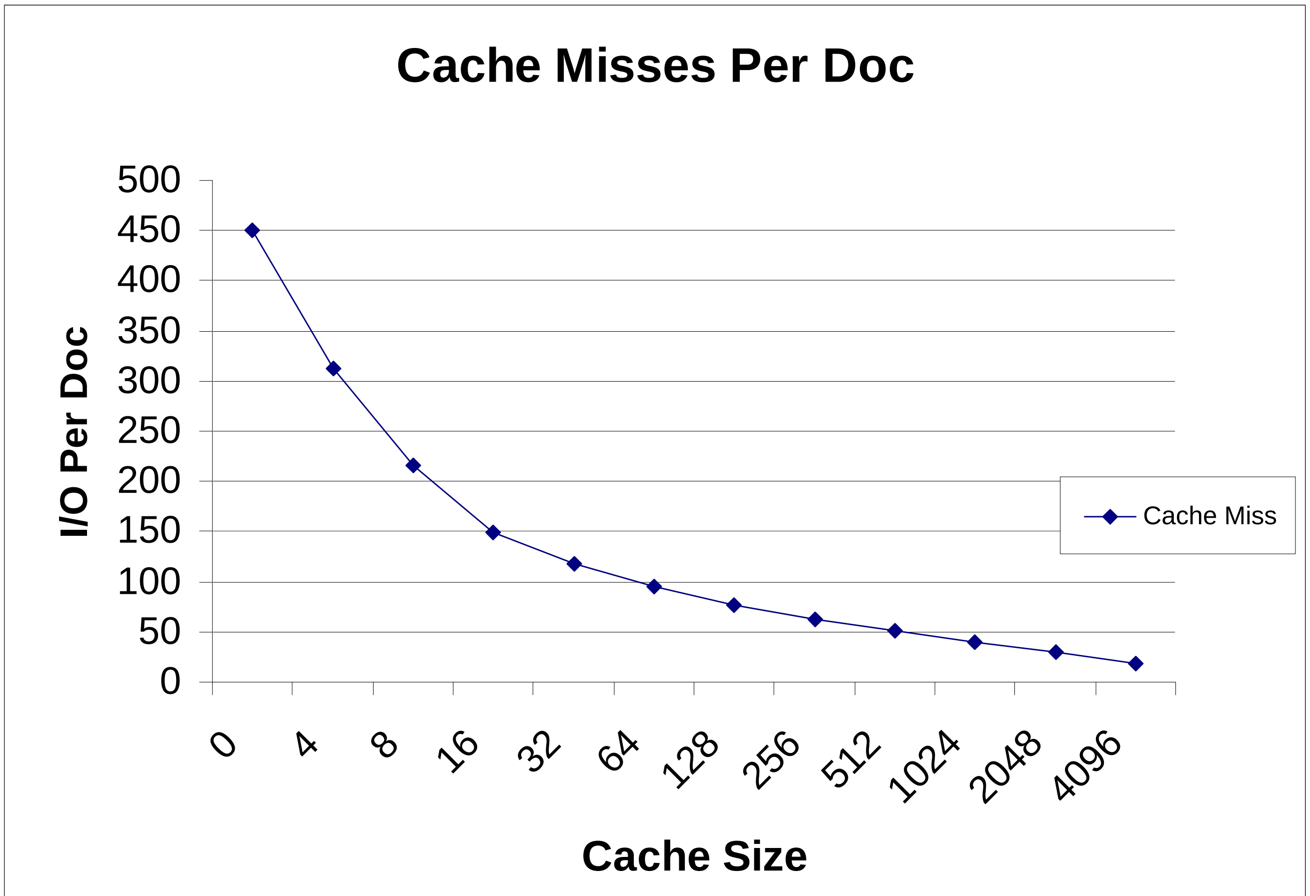
# Buffering opens opportunity for attack when entries are copied to posting lists

- Need > 100,000 buffered documents for update rate of 2 documents/second
- Average document indexed after ½ day
- Window of vulnerability is too large!

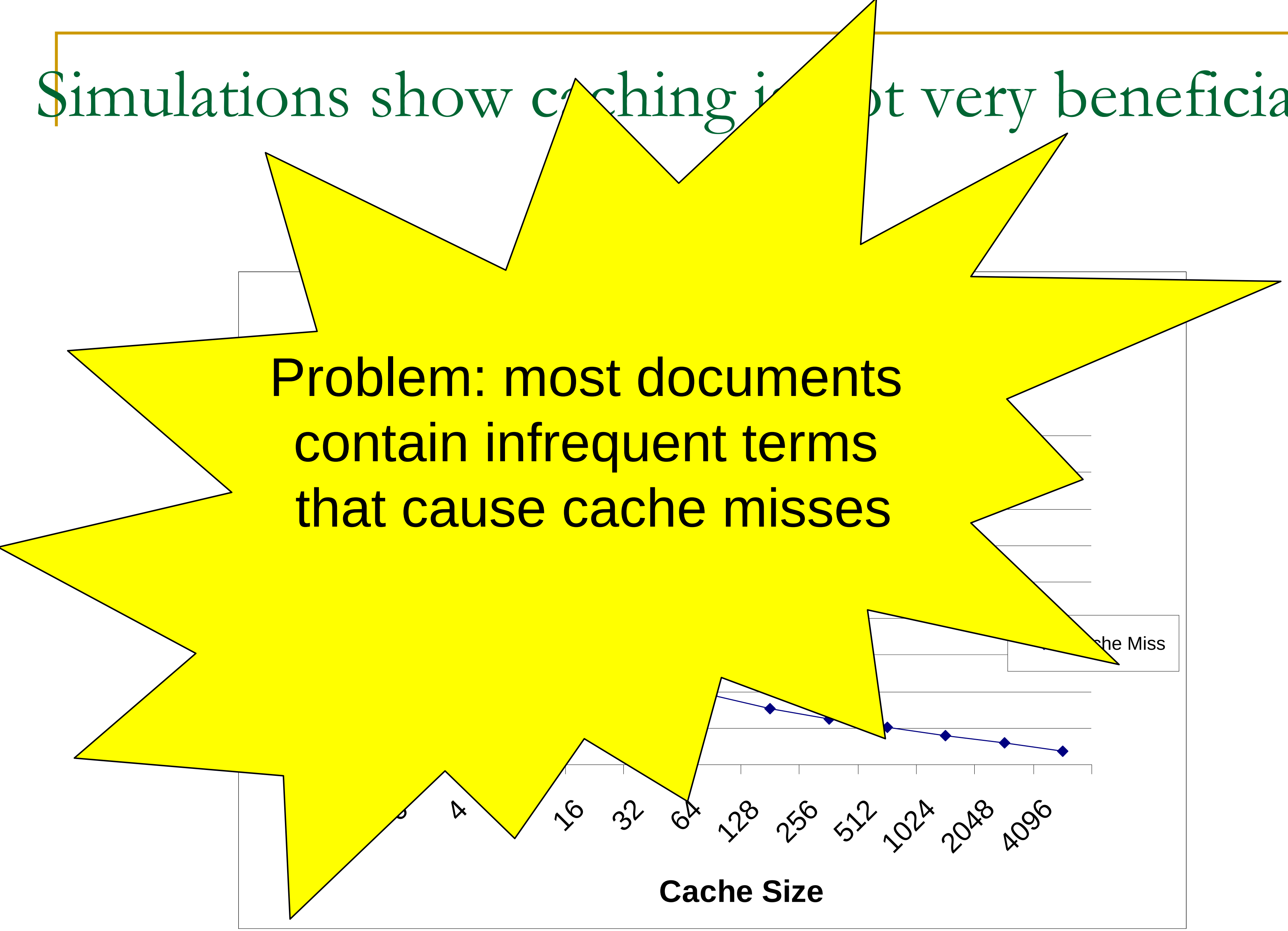
# Storage server cache *is* trustworthy



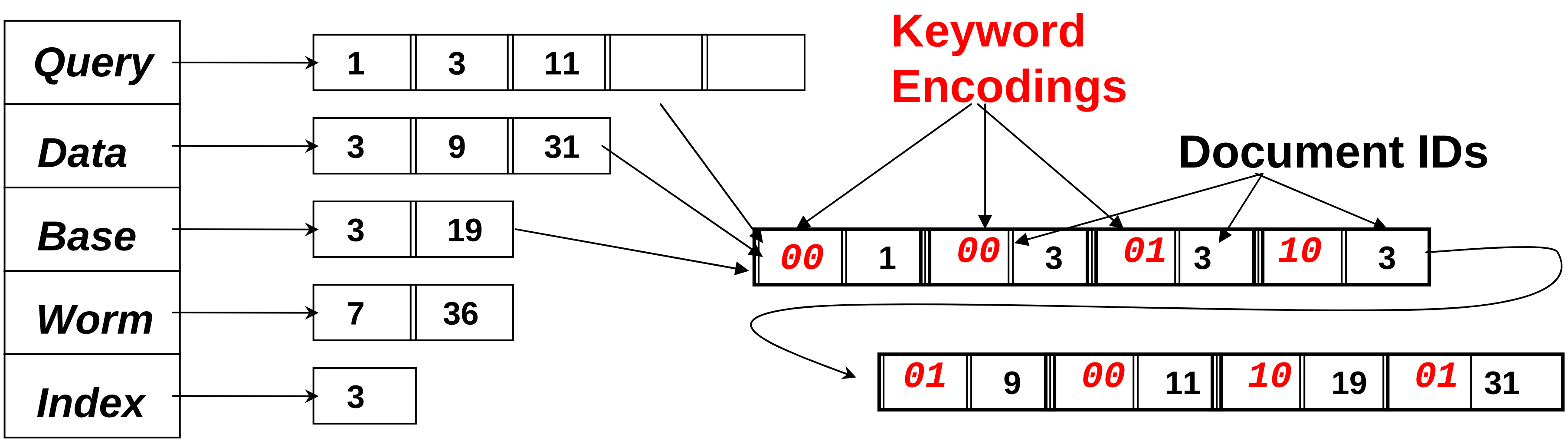
# Simulations show caching is not very beneficial



# Simulations show caching is not very beneficial



# So, merge posting lists until every update hits the cache



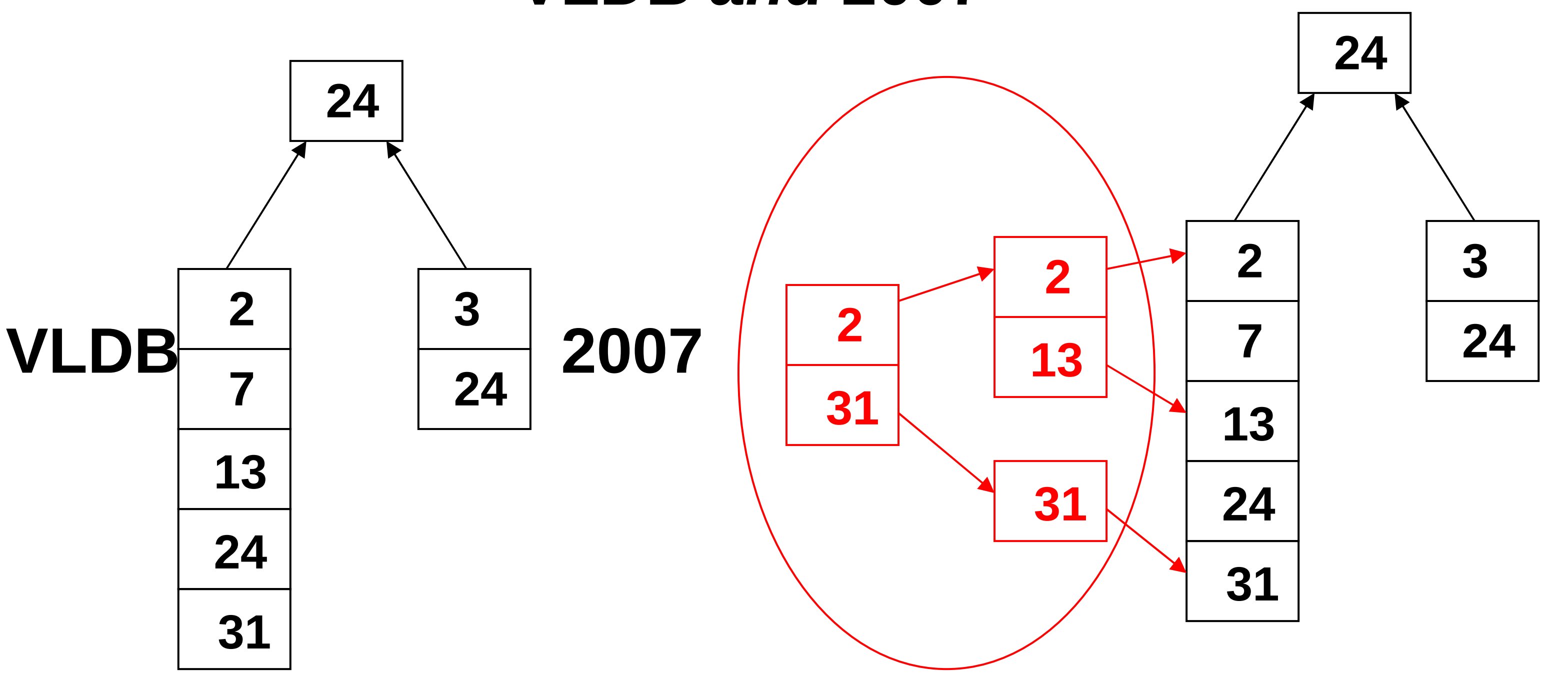

1 random I/O per new document on average, with IBM intranet workload!

# Cost of merging: longer lists to scan during lookup

- ~10% performance hit with uniform random merging, 128 MB server cache, real-world data/queries
  - Only a few popular query terms
  - They have long posting lists and rarely get merged with each other
- “Smarter” merging schemes can do better if needed

# We really could use trustworthy B-trees

**VLDB *and* 2007**

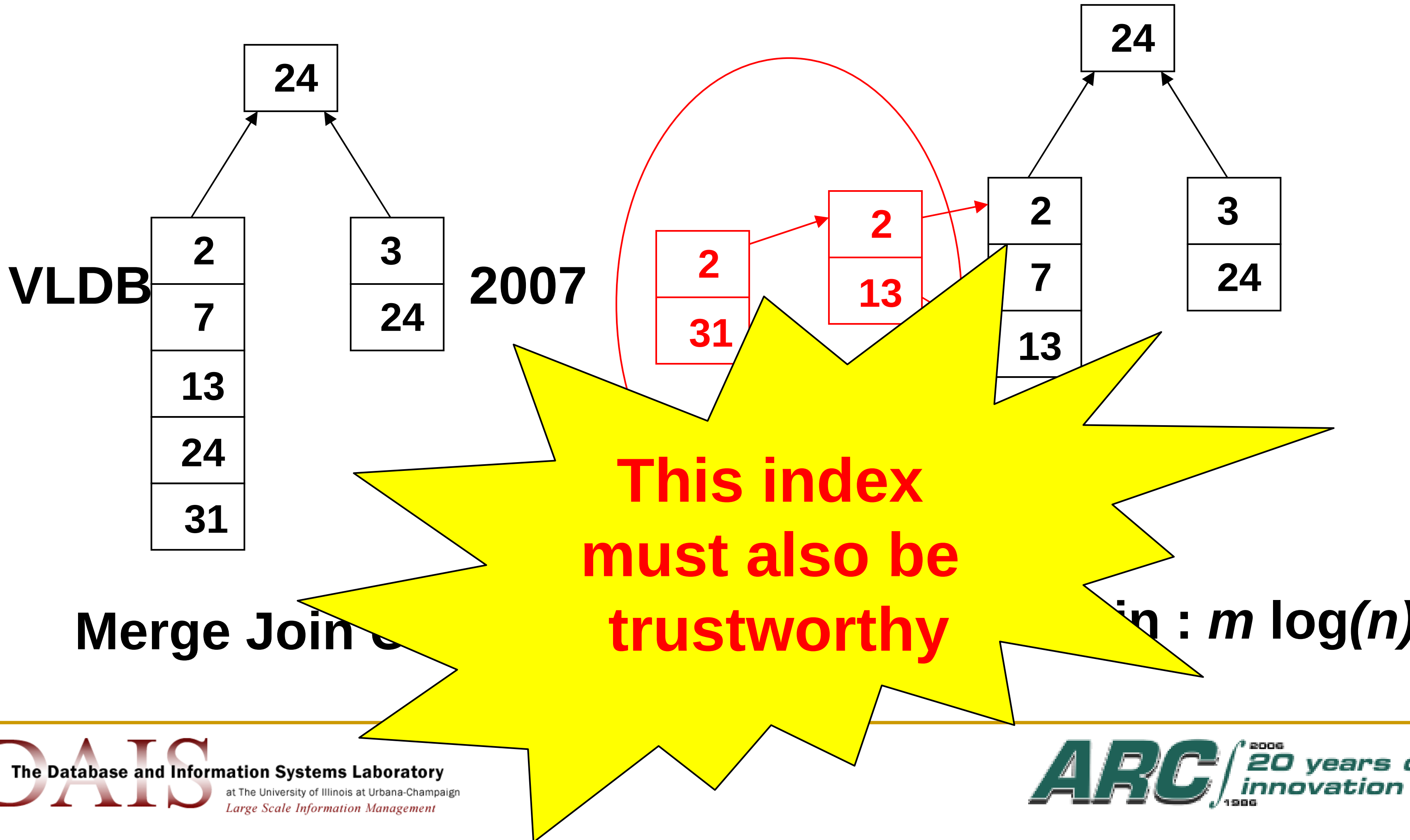



**Merge Join *O* (*m*+*n*)**

**Index Join : *m* log(*n*)**

# We really could use trustworthy B-trees

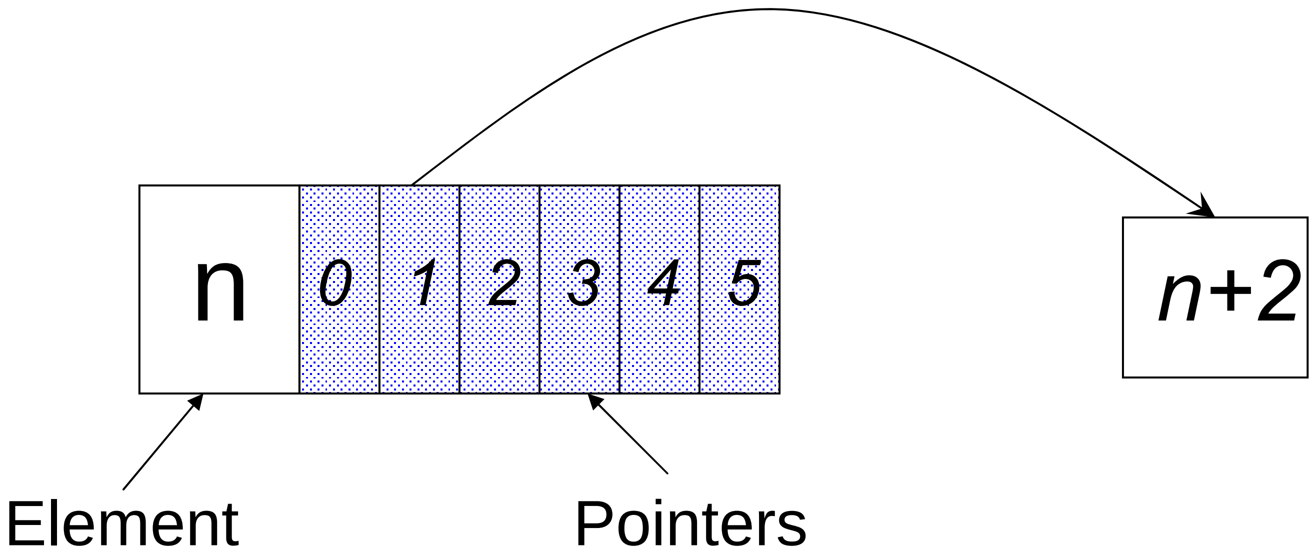
**VLDB *and* 2007**



# *Jump indexes* are a trustworthy version of B-trees for monotone sequences

- Doc IDs, arrival times, …
- O(log *N*) lookup, where *N* is the item being looked up
- Provably trustworthy
- 40% slower than *unmerged* posting lists and B+ trees, on a conjunctive search workload with real-world data, 256 MB server cache
- Avg 2 I/Os/doc insertion with intranet data, 256 MB server cache

# Each jump index node stores extra pointers

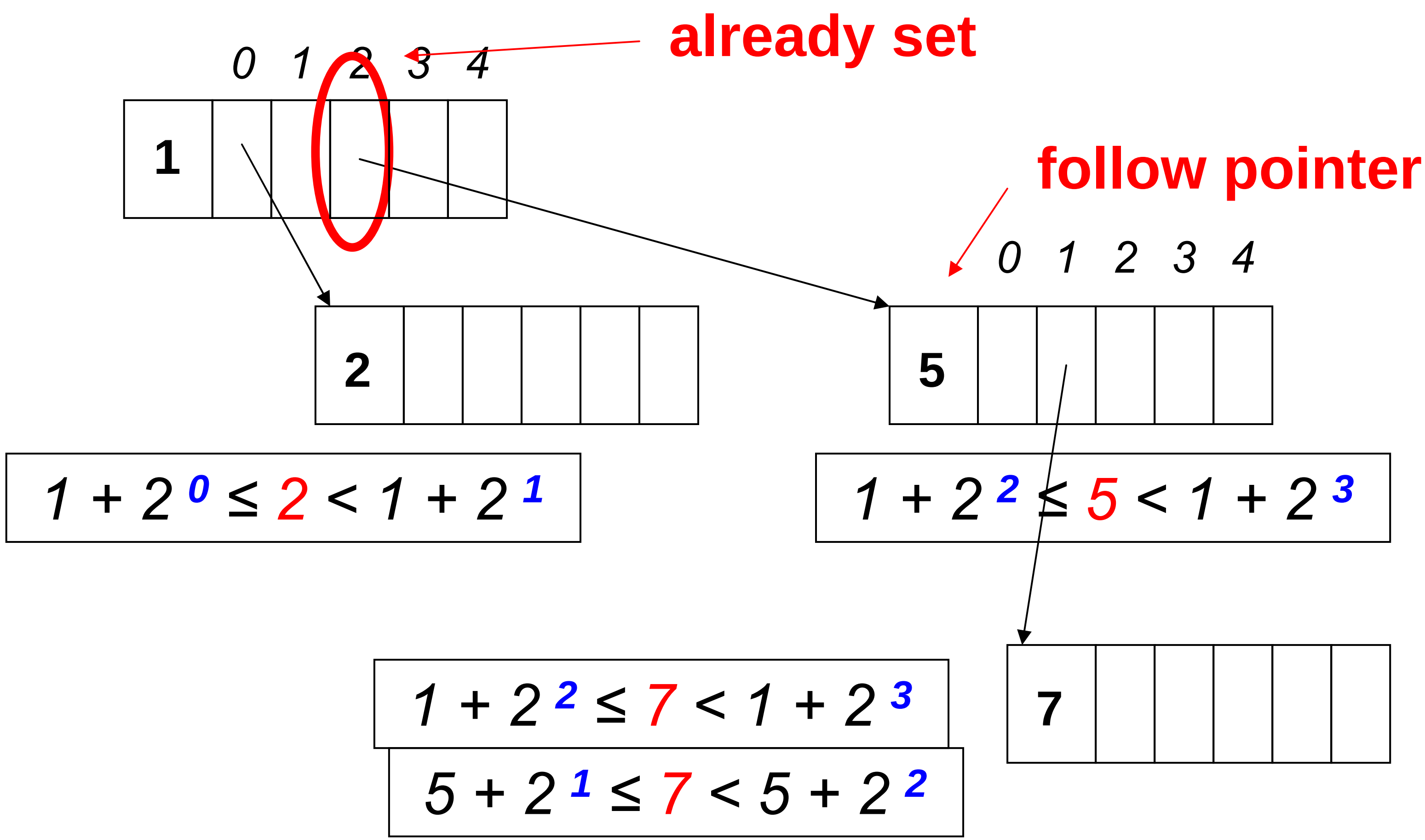



$$n + 2^{1} \leq n+2 < n + 2^{2}$$

$i^{th}$ pointer points to an element $n_i$

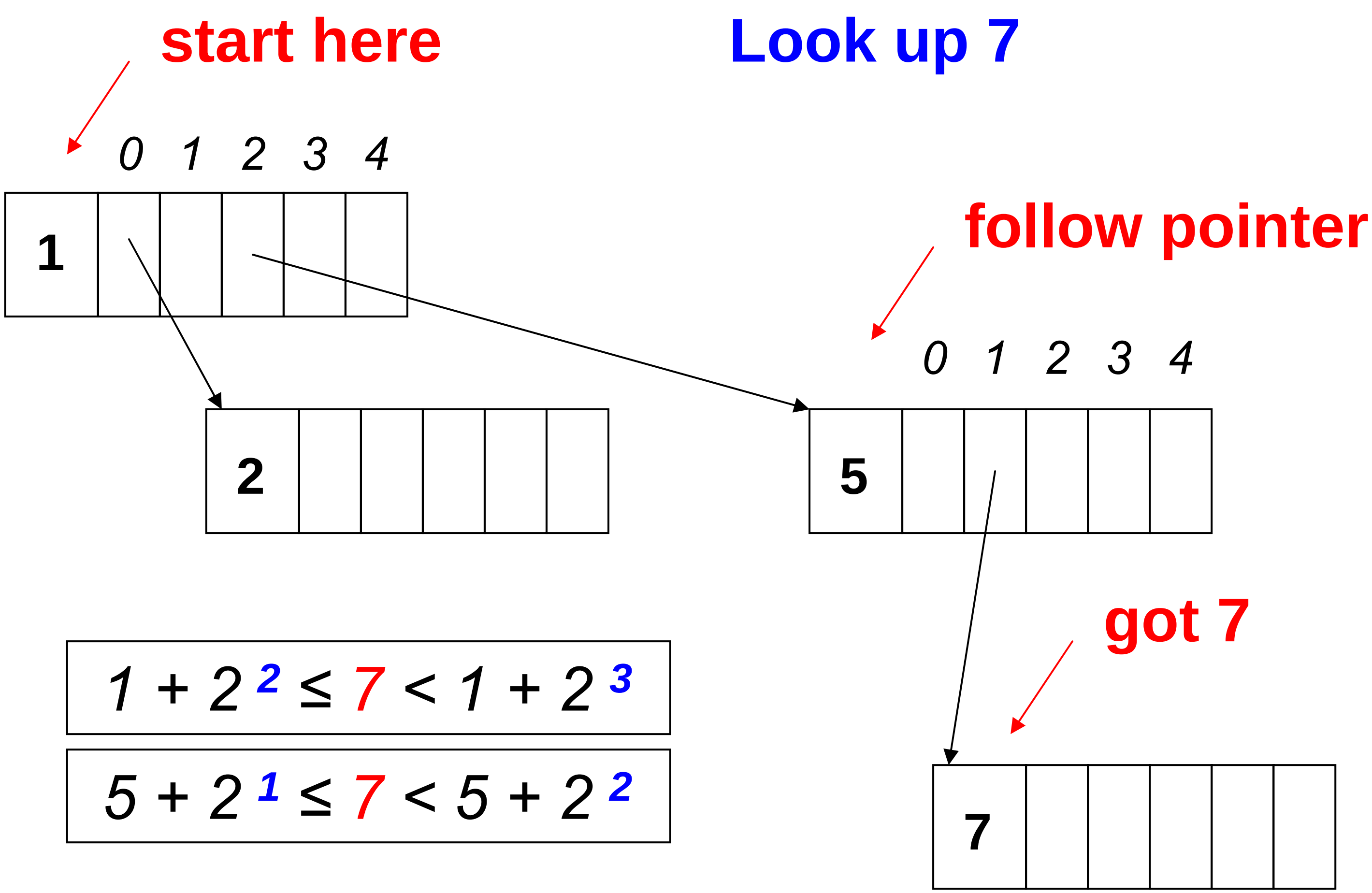
$$n + 2^{i} \leq n_i < n + 2^{i+1}$$

# Jump index in action




log($N$) pointers to reach element $N$

# The path to an element does not depend on future insertions




$$1 + 2^{2} \le 7 < 1 + 2^{3}$$

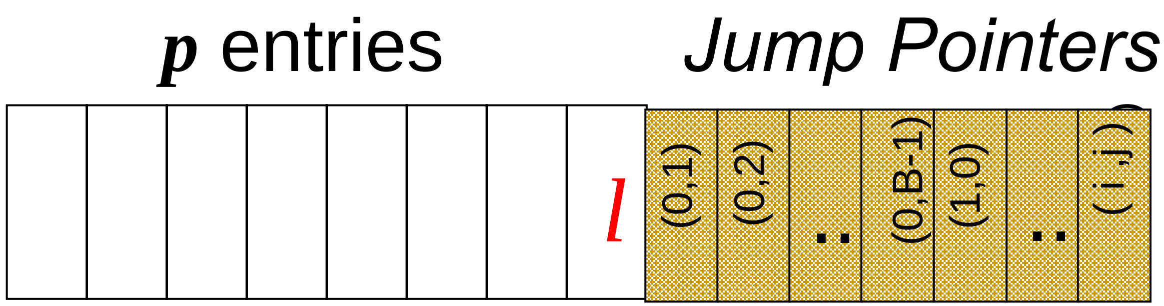
$$5 + 2^{1} \le 7 < 5 + 2^{2}$$

# Jump index elements can be stored in blocks, for greater efficiency

Group $p$ entries together with branching factor $B$

- Pointer $(i, j)$ from block $b$ points to block $b'$ having smallest $x$ such that

$$l + j\,B^i \;\leq\; x \;<\; l + (j+1)\,B^i$$



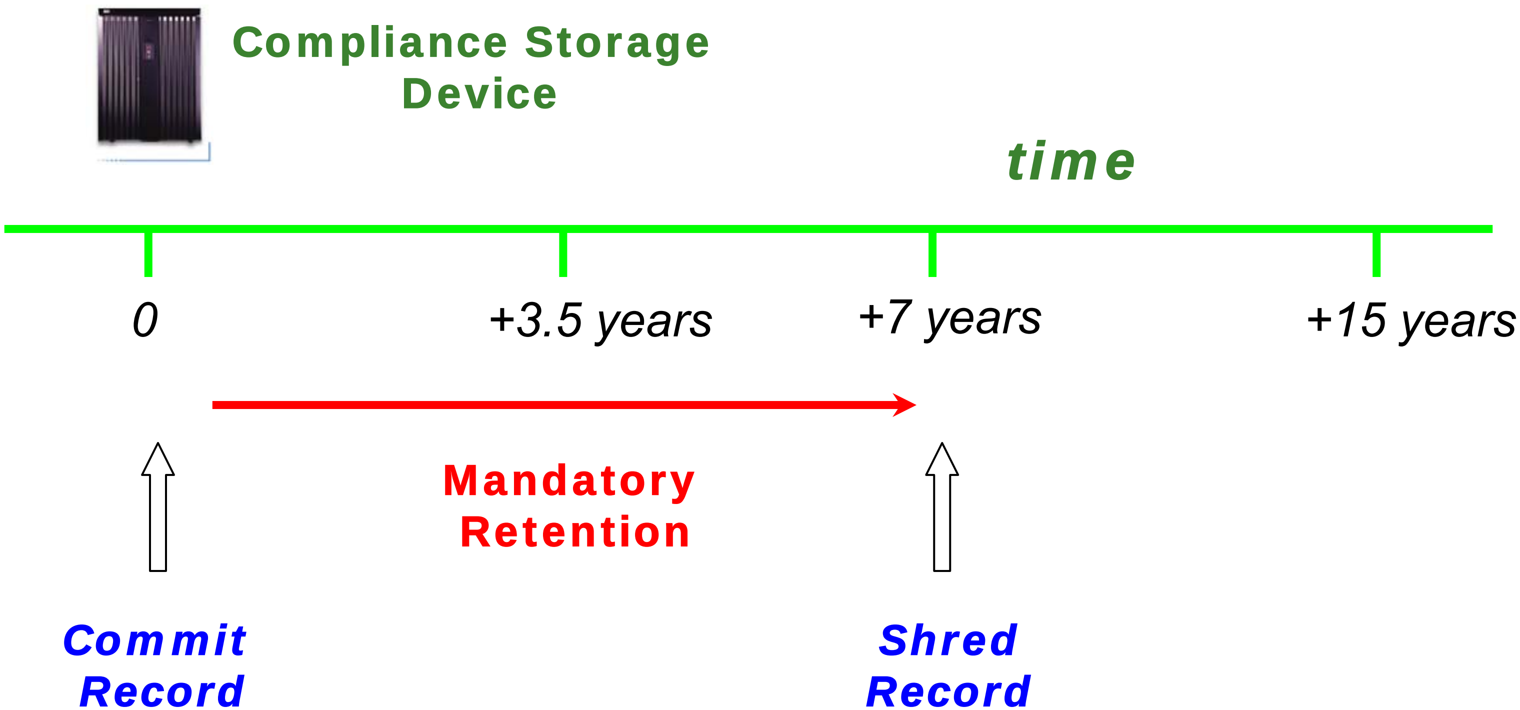
# Indexing conclusions

- Traditional indexes aren't trustworthy
- We can build trustworthy generalized hash trees, inverted indexes and jump indexes for compliance records
- All are reasonably fast

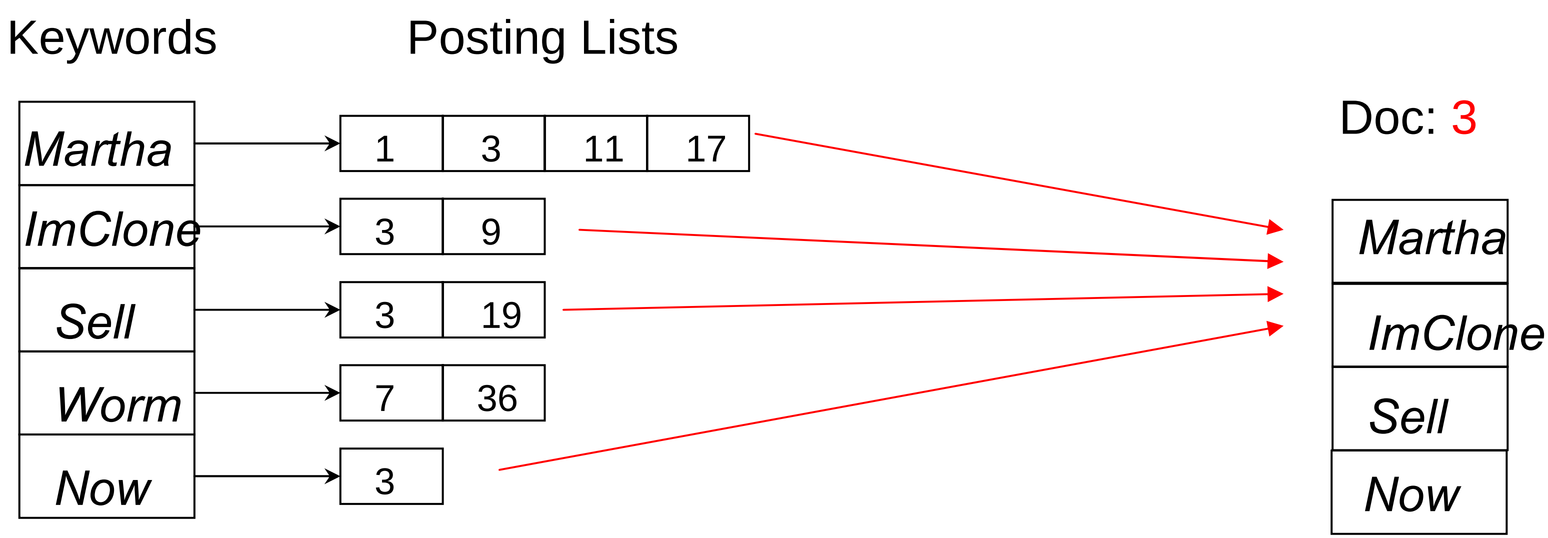
# Trustworthy deletion from indexes

[Zhu & Hsu, SIGMOD 05]
[Mitra, Hsu, & Winslett, SSS 06 and new work]

# Expired documents usually must be deleted

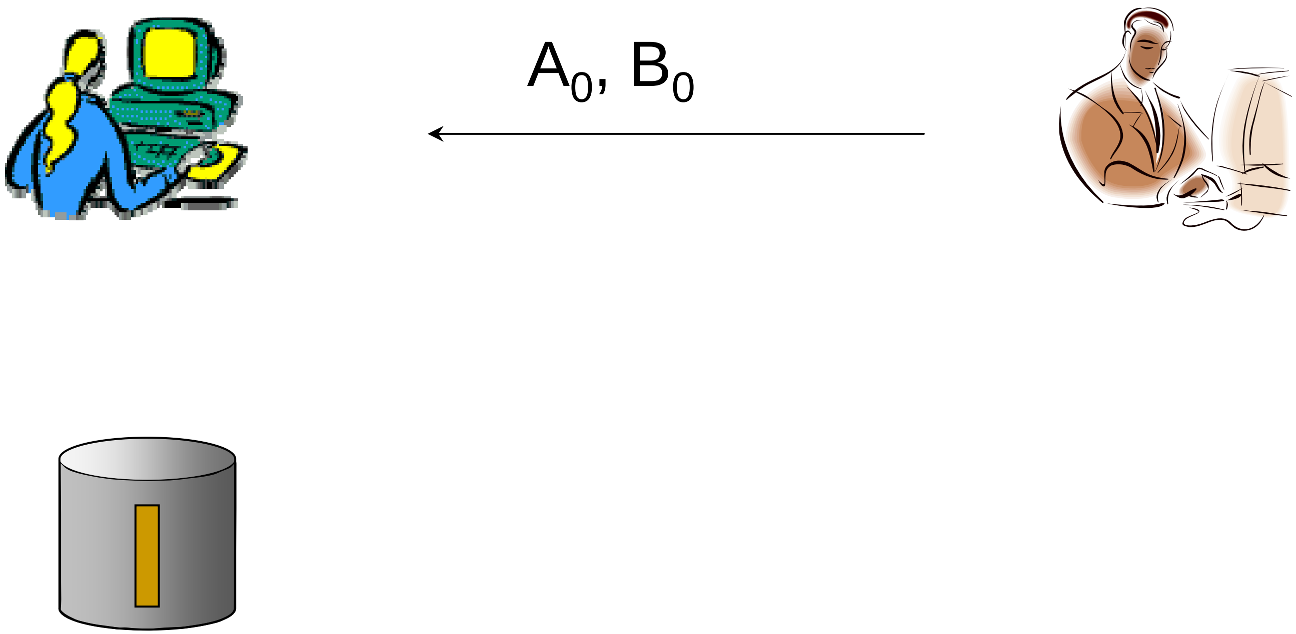


# But a document can be reconstructed from a full text index

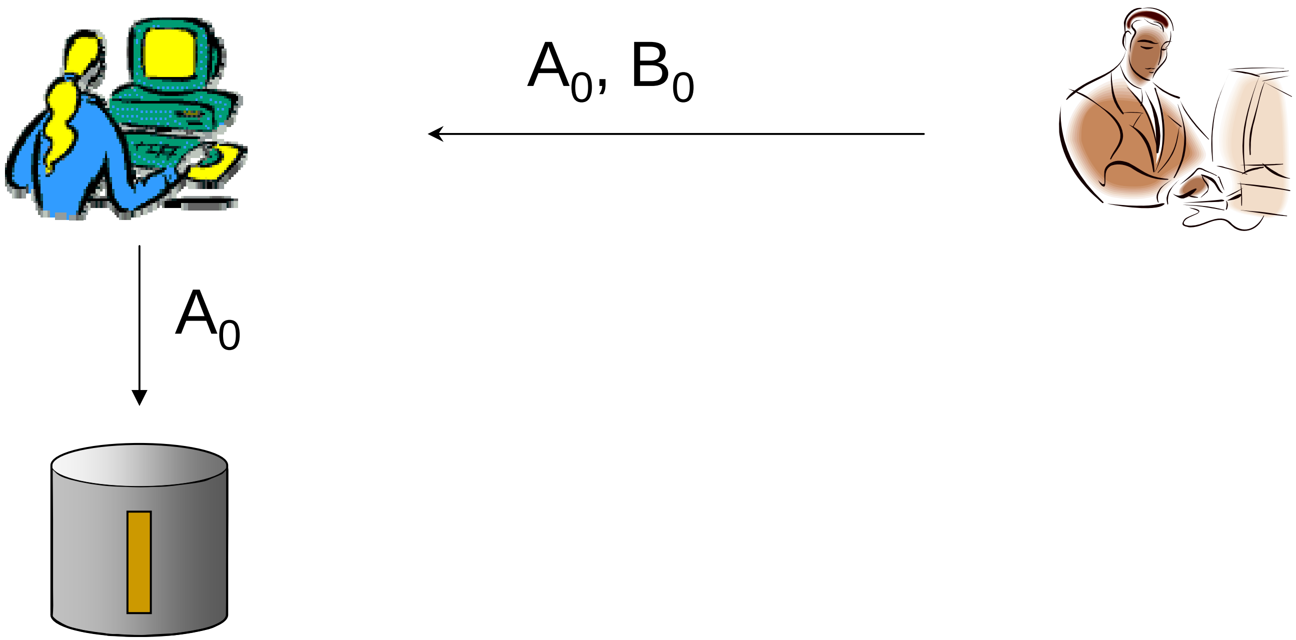


# But a document can be reconstructed from a full text index

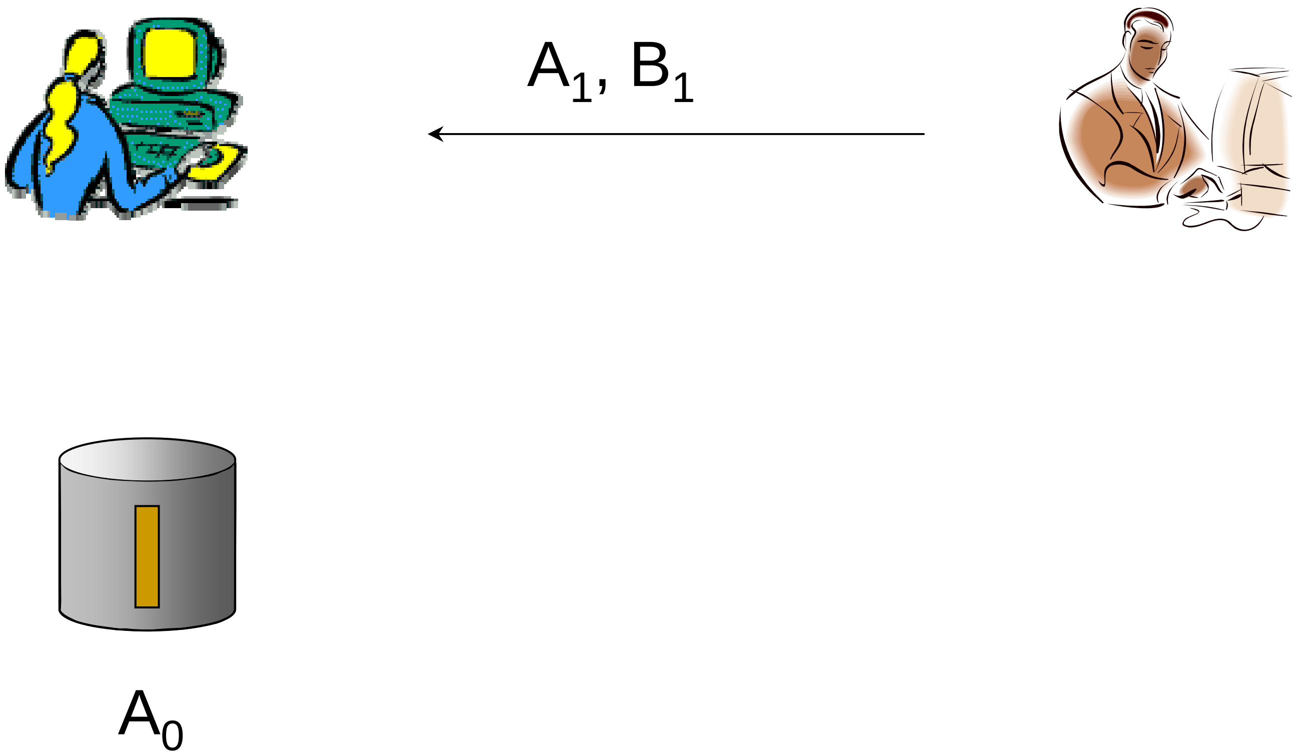
# We can formally define secure deletion through an indistinguishability game



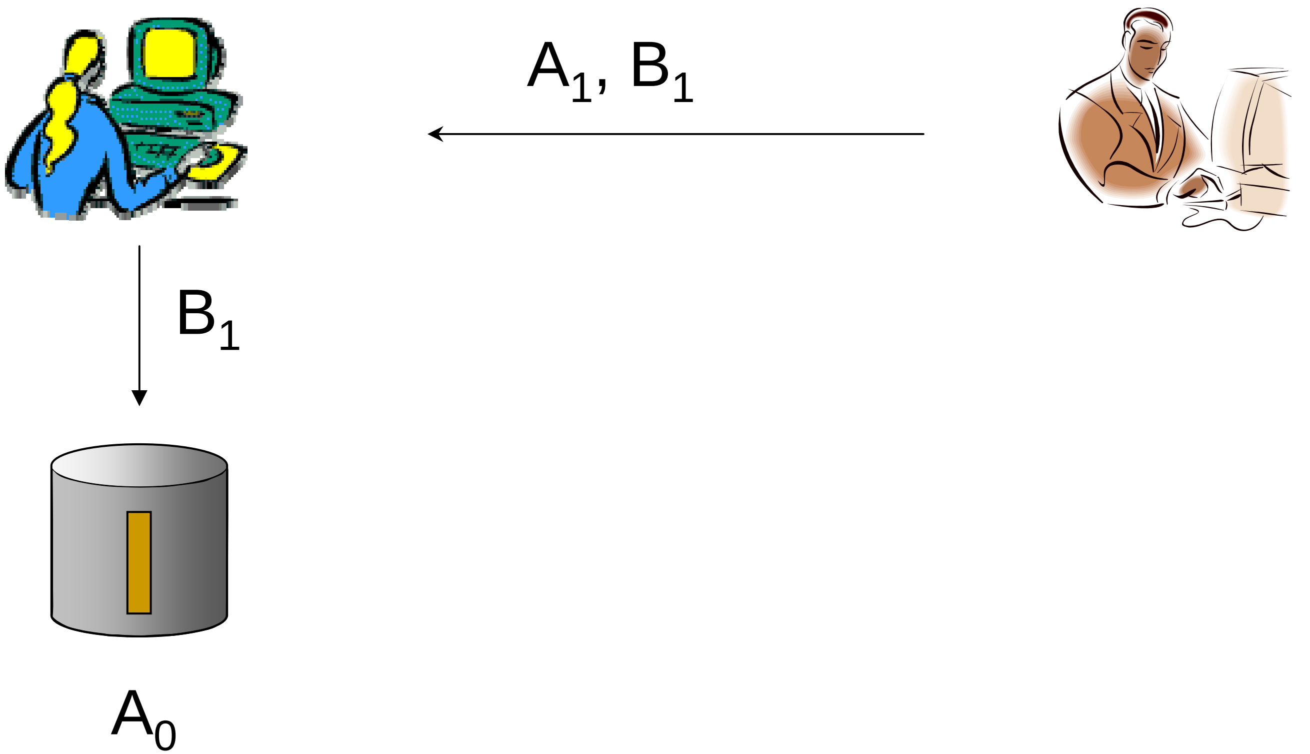
# Hidden from Bob, Alice randomly chooses one document set & stores & indexes it.



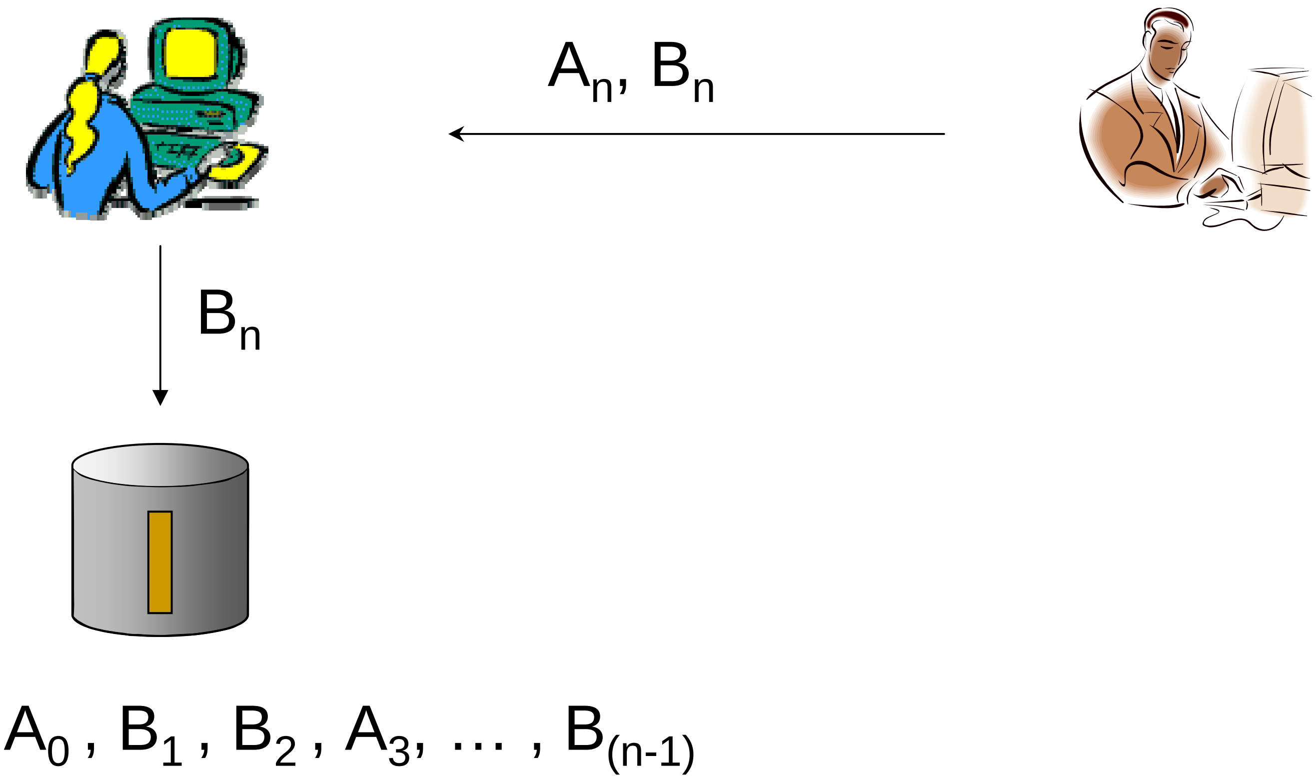
# And the game continues



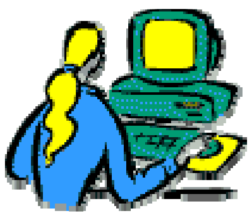
# And the game continues

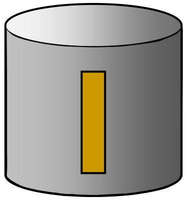


# And the game continues



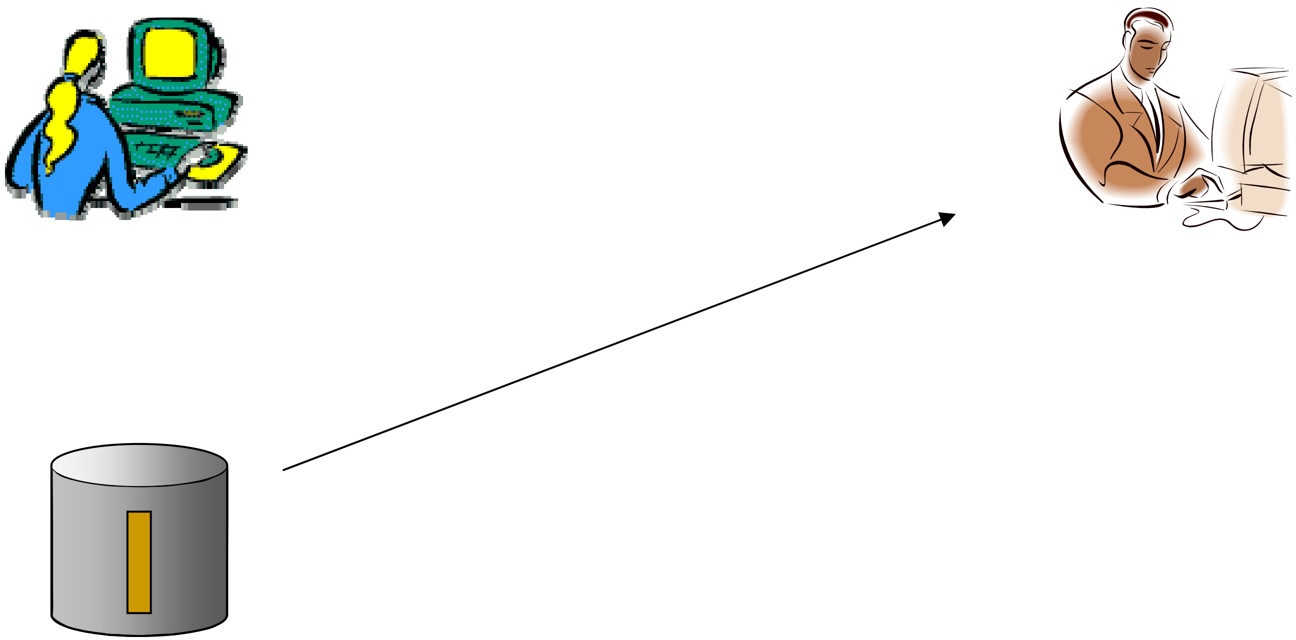
# Alice erases the documents and index entries as documents expire





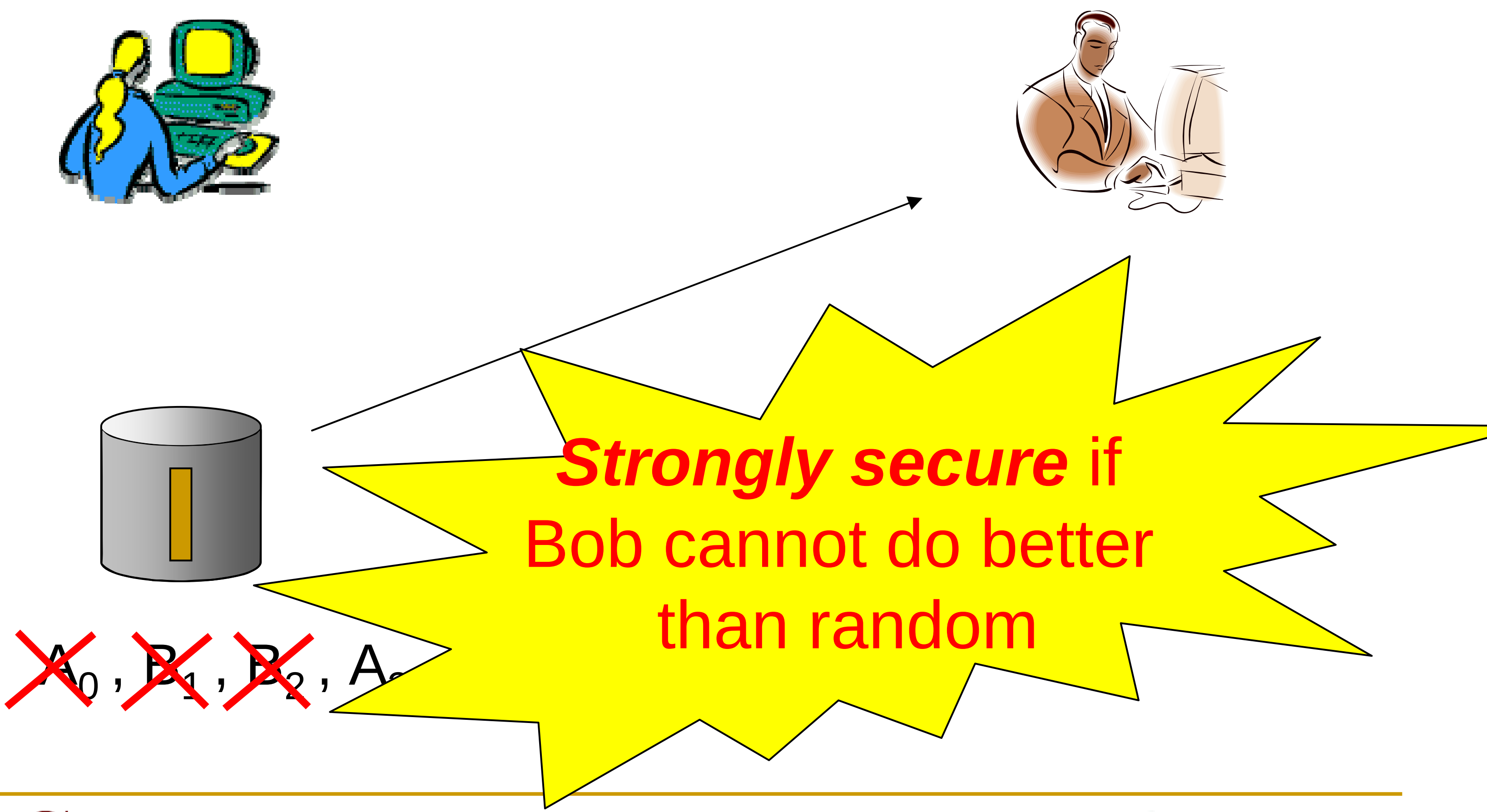
$\cancel{A_0}$ , $\cancel{B_1}$ , $\cancel{B_2}$ , A3, … , $B_n$

# Bob wins the game if he can guess the deleted document sets

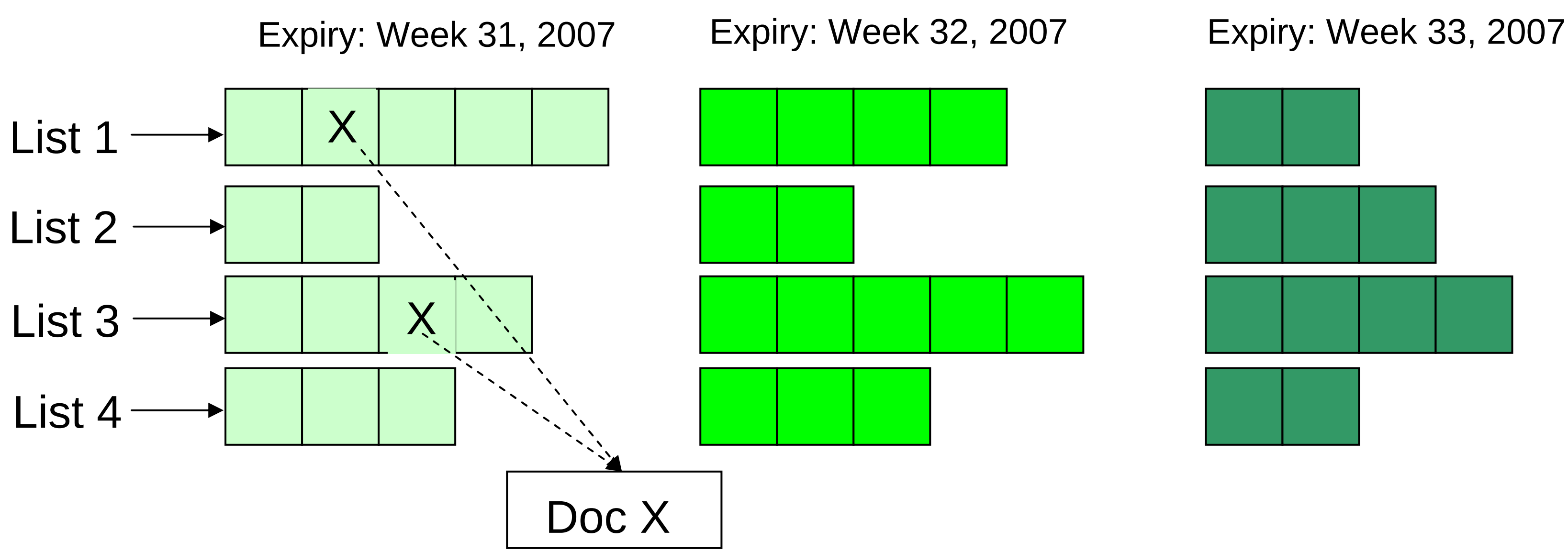


$\cancel{A_0}, \cancel{B_1}, \cancel{B_2}, A_3, \ldots, B_n$

# Bob wins the game if he can guess the deleted document sets



# Separate index for each disposition group: strongly secure, but slow queries

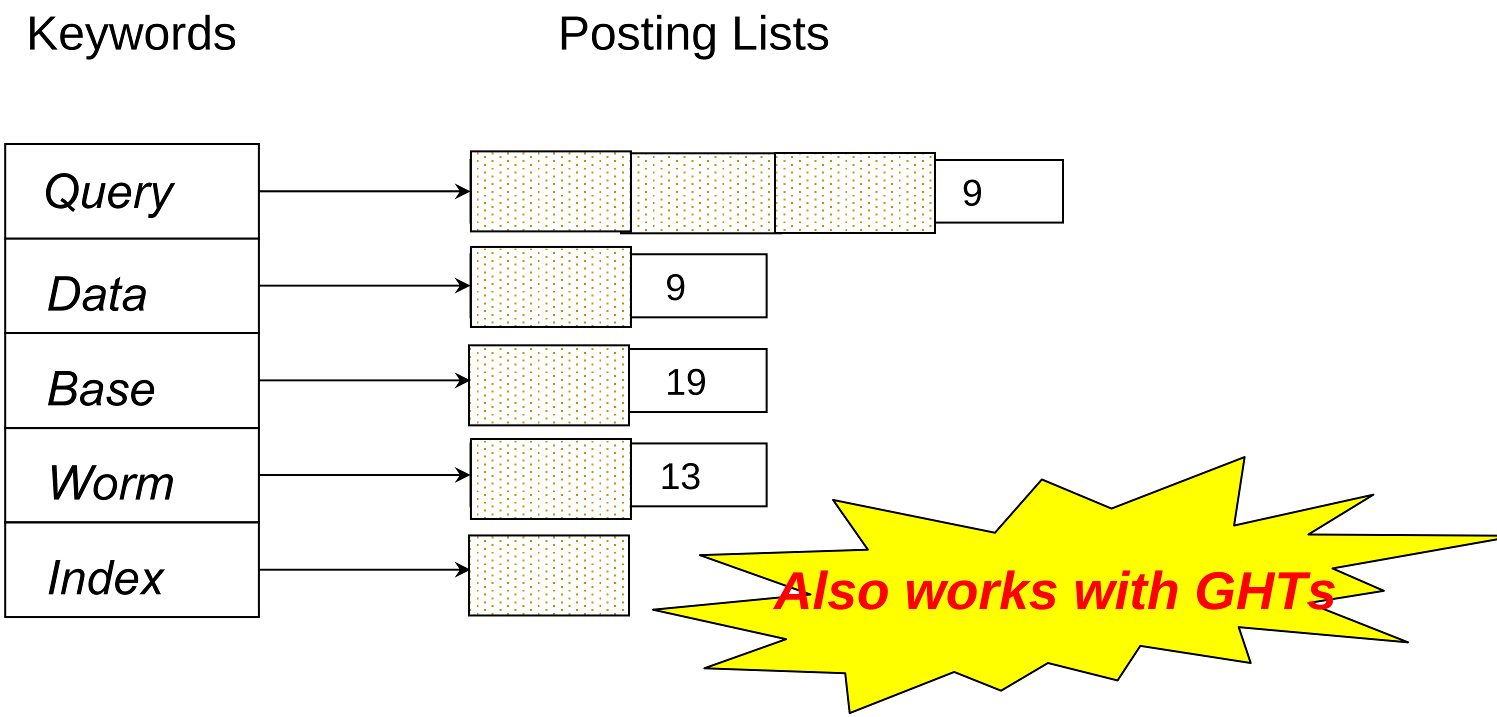



**But is 6x slower than querying a single index, on Enron email w/ 4-month disposition interval**

**Can use the same approach for GHTs**

# Logical disposition: Encrypt the document IDs of a disposition group, discard key

*Delete 1, 3, 5, and 7*



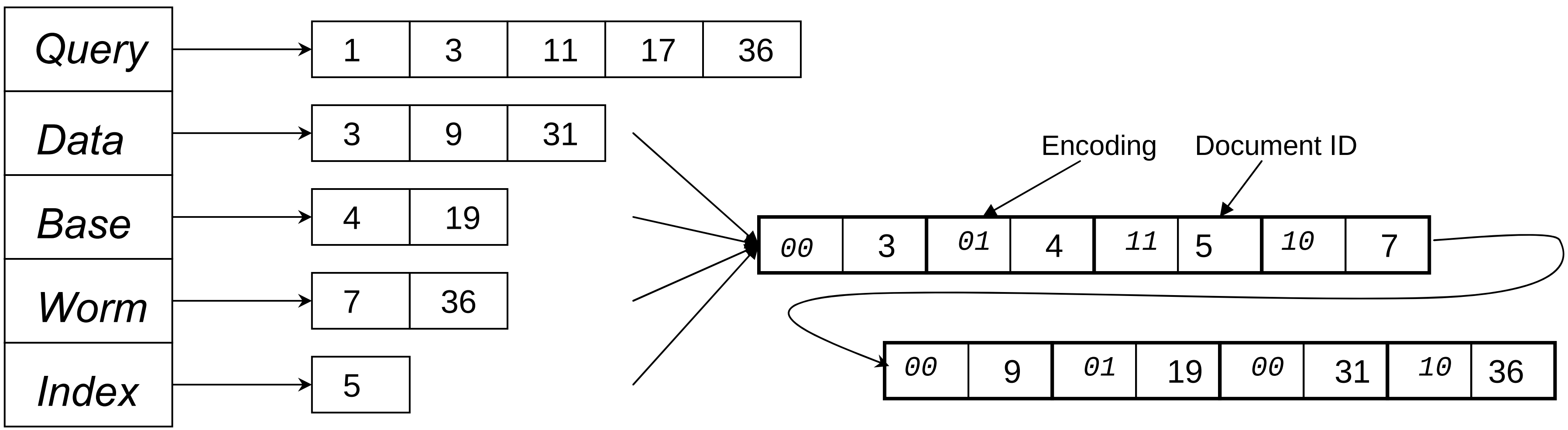

You only learn that *some* disposition group had Query, Data, Base, Worm, and Index (because they have the same key file pointers)

# Logical disposition isn't strongly secure

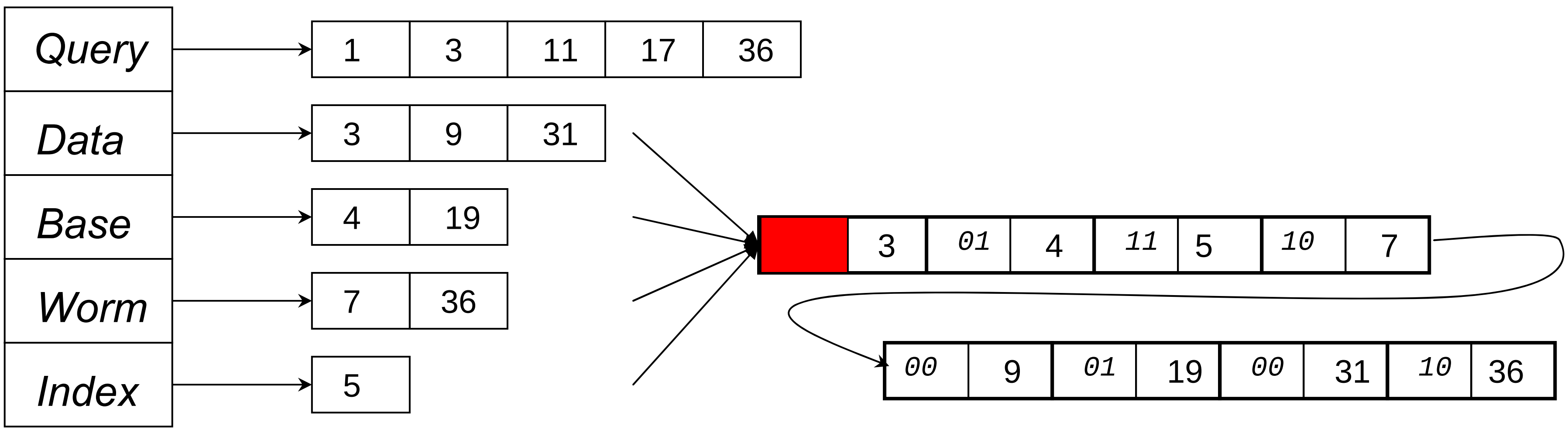
- Presence of *Martha, ImClone* and *Ralph* is suspicious
- **Absence** of *ImClone* can also be suspicious

***Adversary can still reverse engineer terms from posting list lengths***

# Recall that trustworthy inverted indexes have merged posting lists



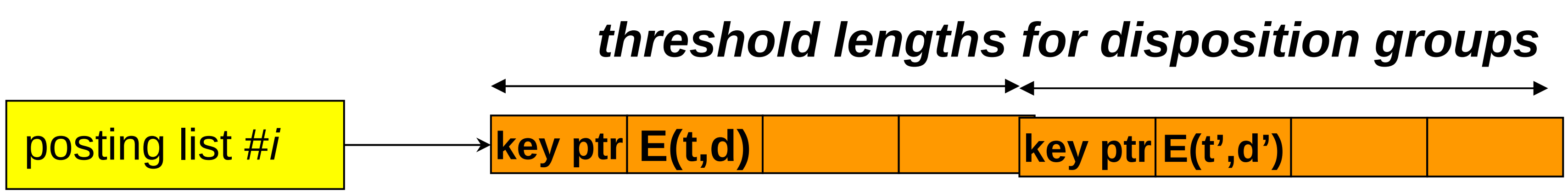
# A term occurrence can be partially hidden by "deleting" its encoding (encrypt it, discard key)



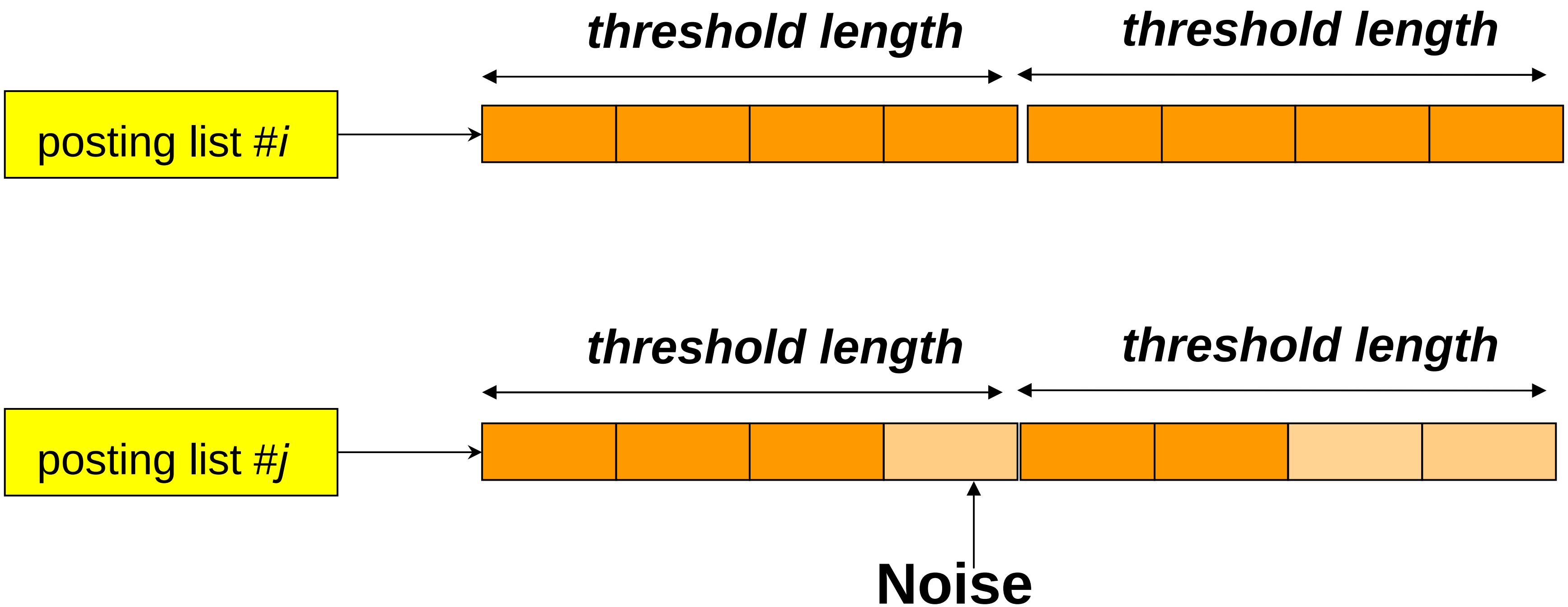

Document *3* had one of the words

Query, Ascertain, McDonald, 55, …

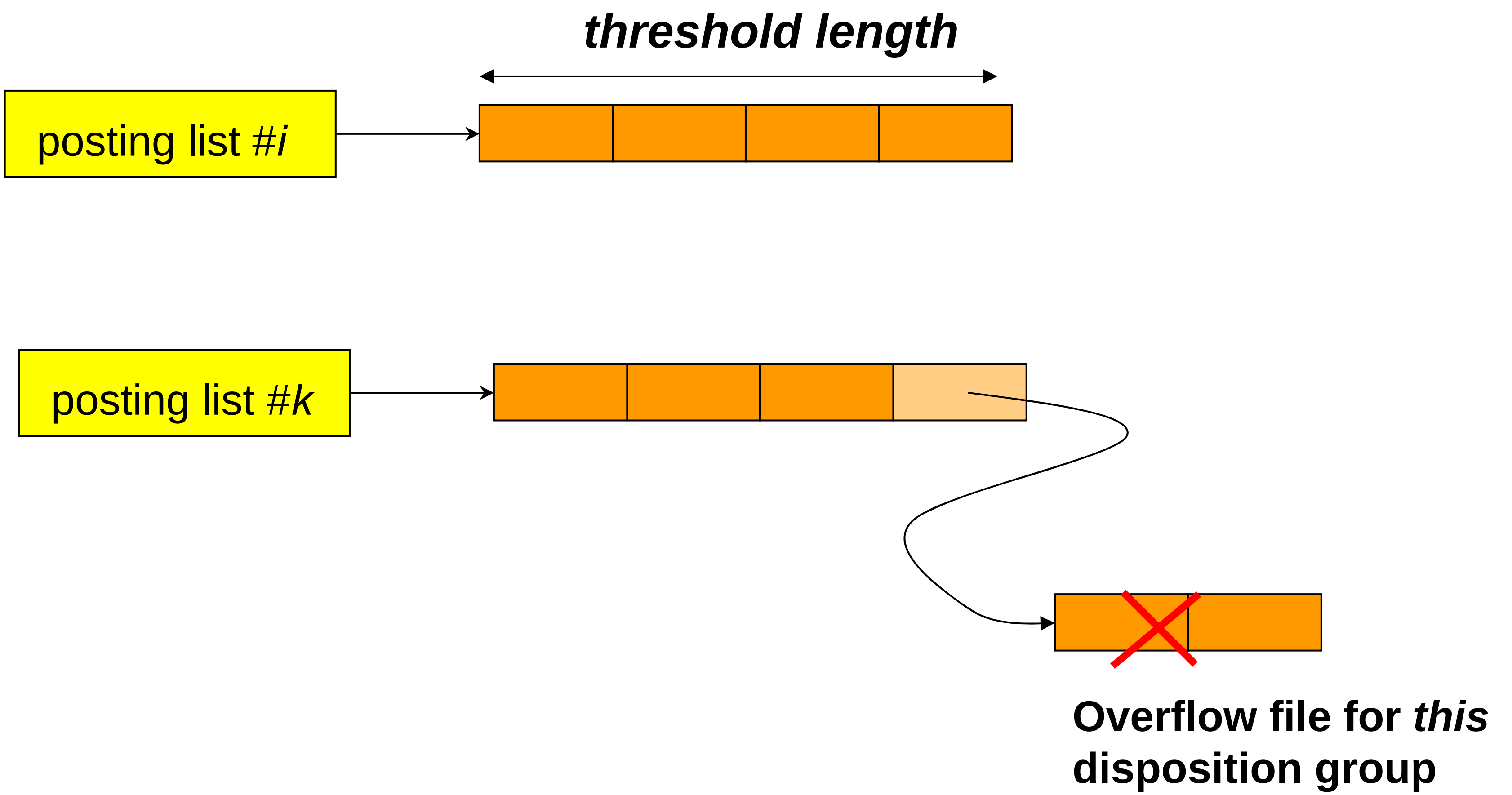
**Uniform posting list lengths** for each disposition group +
merged posting lists +
encrypted posting elements =
provably strongly secure deletion with reasonable performance



# Add noise terms if actual length is less than threshold



# Additional elements for a disposition group are stored in an overflow file



# Index performance depends on the choice of threshold lengths

- Too short → lots of overflows → poor query performance
- Too long → wastes space
- Formulate as a dynamic programming optimization problem

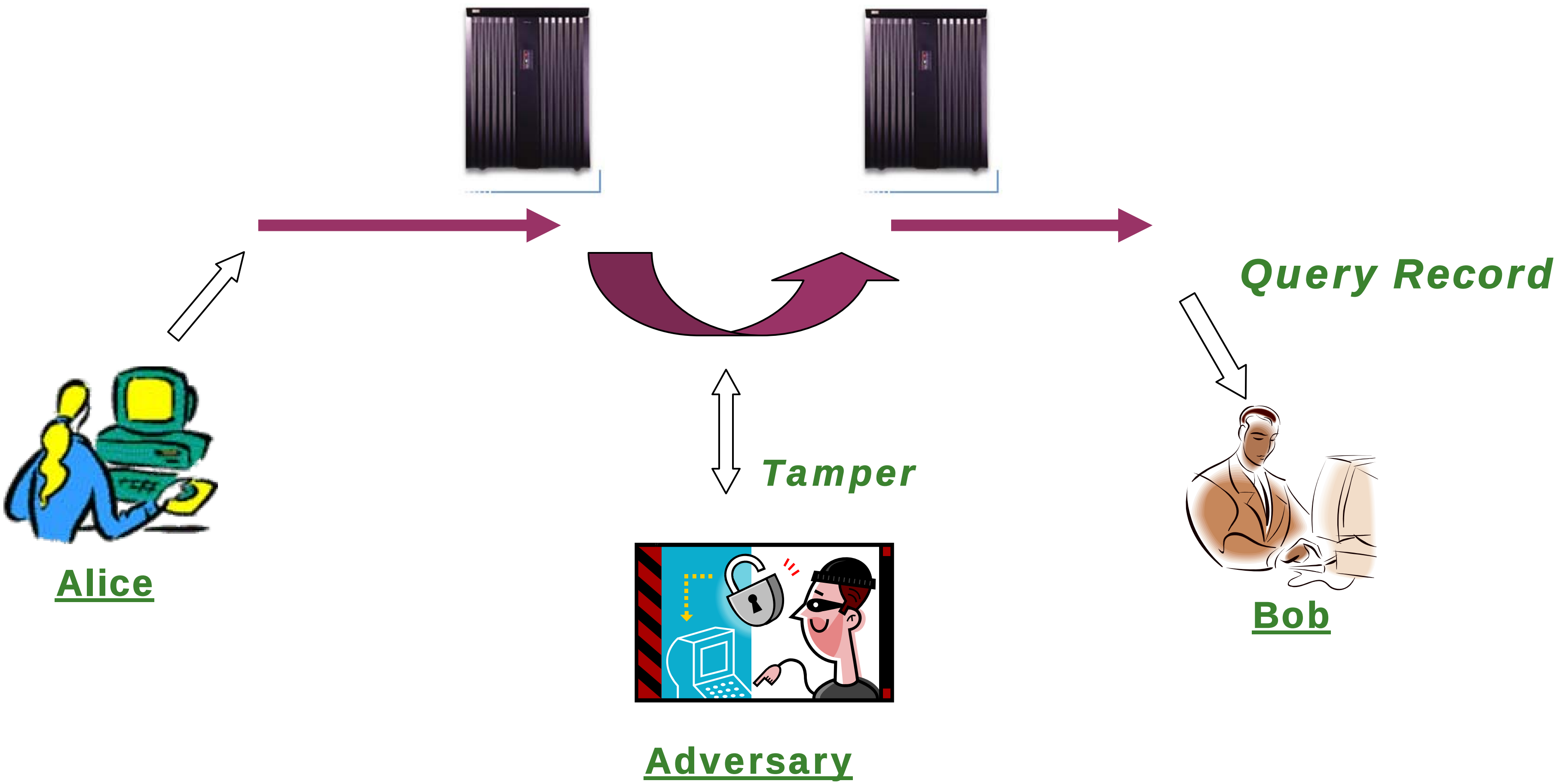
Results on Enron email: queries are 5-6 times faster than with an index for each disposition group

# Trustworthy Migration

[Mitra, Winslett, Hsu, and Ma, MSST 07]

[Sion, tech report 07]

# Data sometimes must be copied to a new server



# Bob must be able to verify that the records have not been tampered with during migration

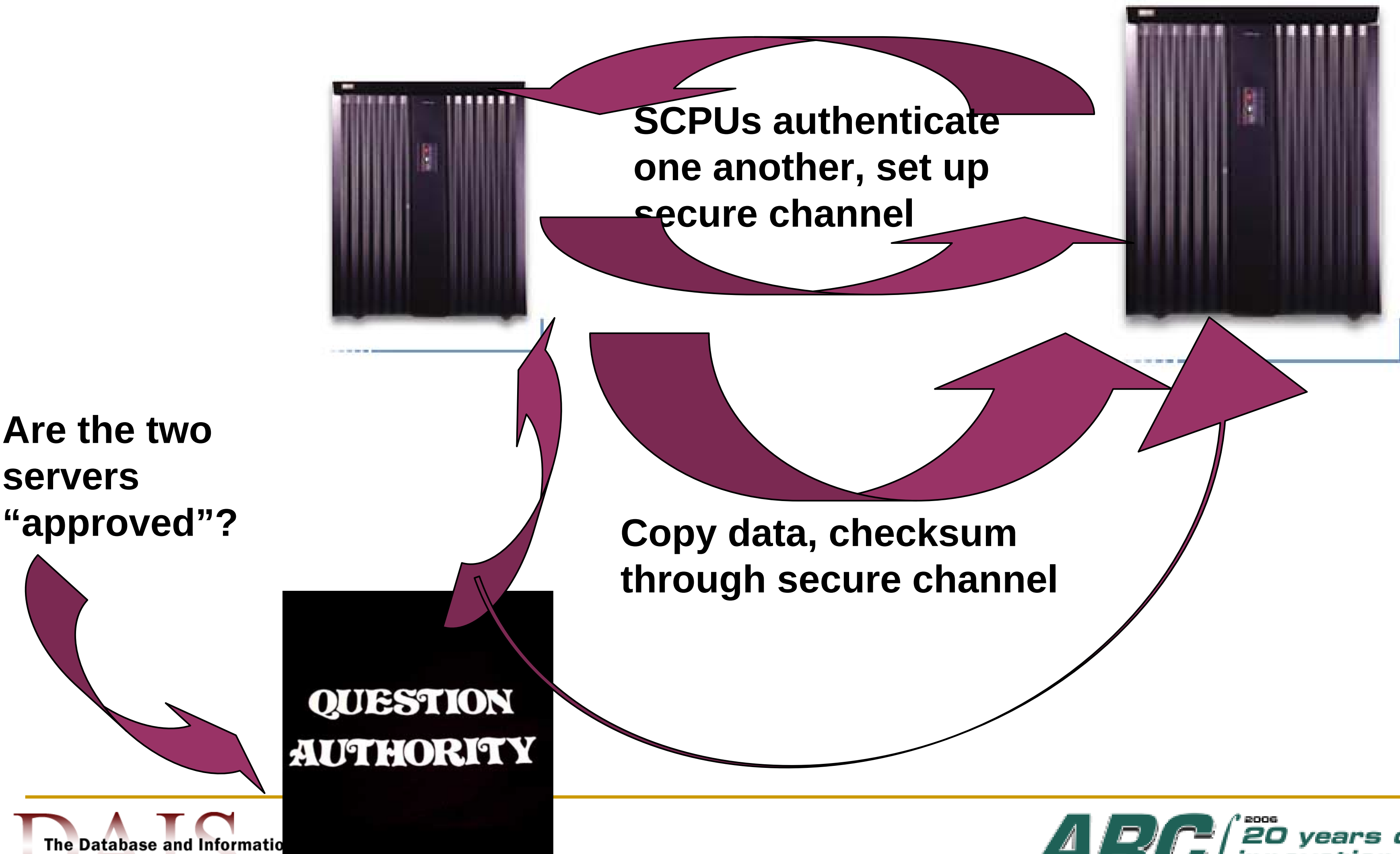
- *Tamper-evident* is good enough

- Detection of tampering leads to automatic presumption of guilt

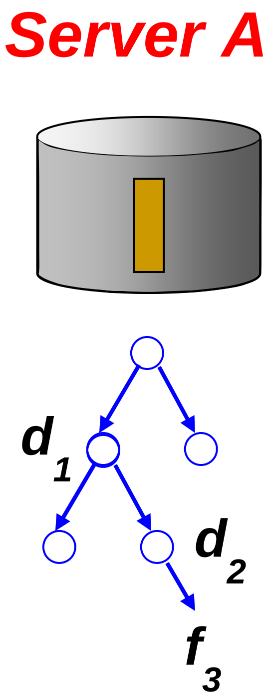
# The bottom line on migration

- Trustworthy migration can be supported by enhancing the storage server with signing capability (via a secure co-processor)
- The concepts can be generalized to an arbitrary sequence of migrations, including directory restructuring and omission
- Policy-driven migration, where certain files are omitted, is hard if we don't want to leak info about omitted files

# If we are using SCPUs, must ensure that new server has one

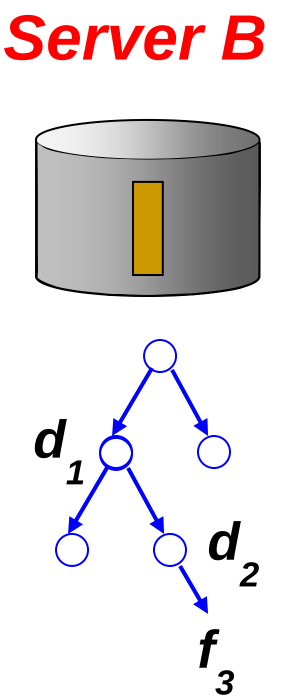


# Bob is looking for a particular file created on server A



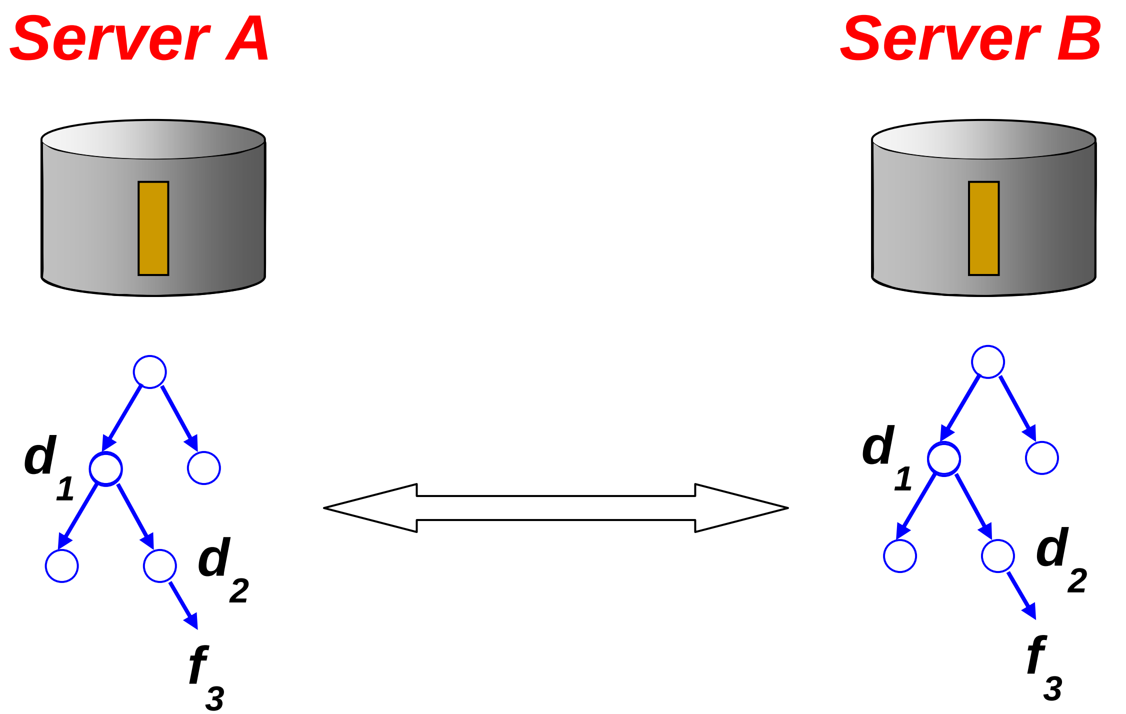

*Bob obtained the path* **$/d_1/\, d_2/\, f_3$** *from a trustworthy index*

# Bob only has access to the files on server B

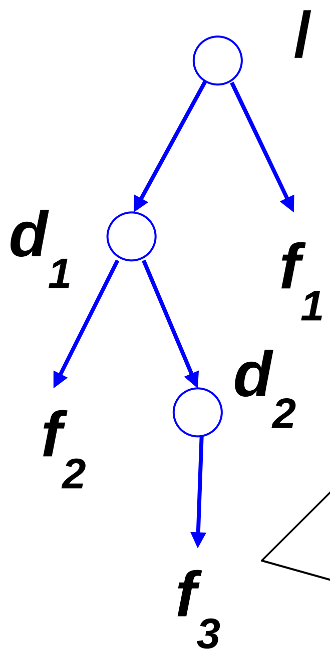



*Bob is looking for file* **$/d_1/\ d_2/\ f_3$**

# Bob has to verify that the file was migrated correctly



# Both the data and metadata must be preserved



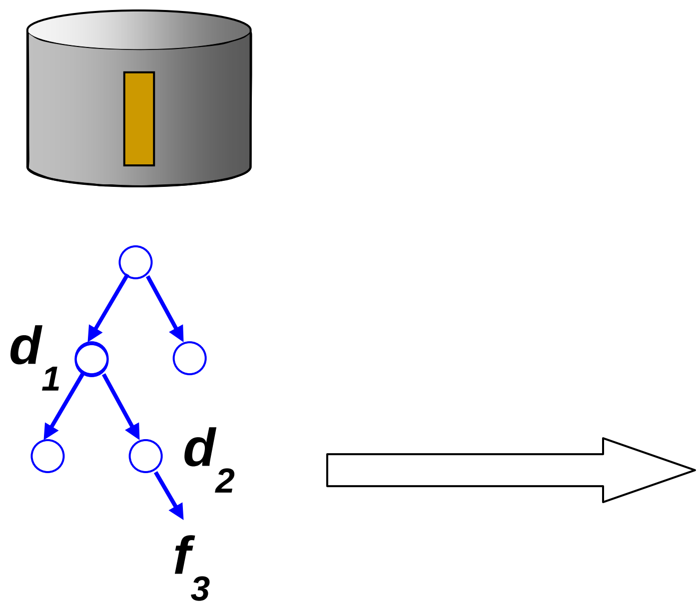

**Meta-data**

| | |
|---|---|
| *owner* | **Alice** |
| *Create-date* | **01/01/1996** |
| *expiry-date* | **01/01/2016** |
| *Size* | **1236** |

**Contents**

# The storage server can be enhanced to issue certificates for files, with a secure coprocessor

**Server A**

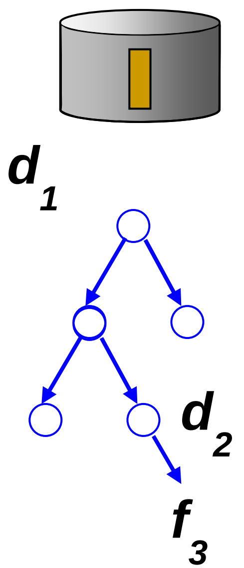



Certificate

The certificate is a proof of file’s contents and metadata

# The certificate includes the metadata and content hashes

**Server A**

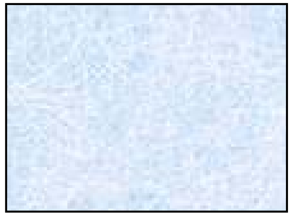



**Meta-data**

| | |
|---|---|
| *path* | /d1/d2/f3 |
| *owner* | Alice |
| *expiry-date* | 01/01/2016 |
| *Size* | 1236 |

***metadata-hash*** **= M**

**Contents**

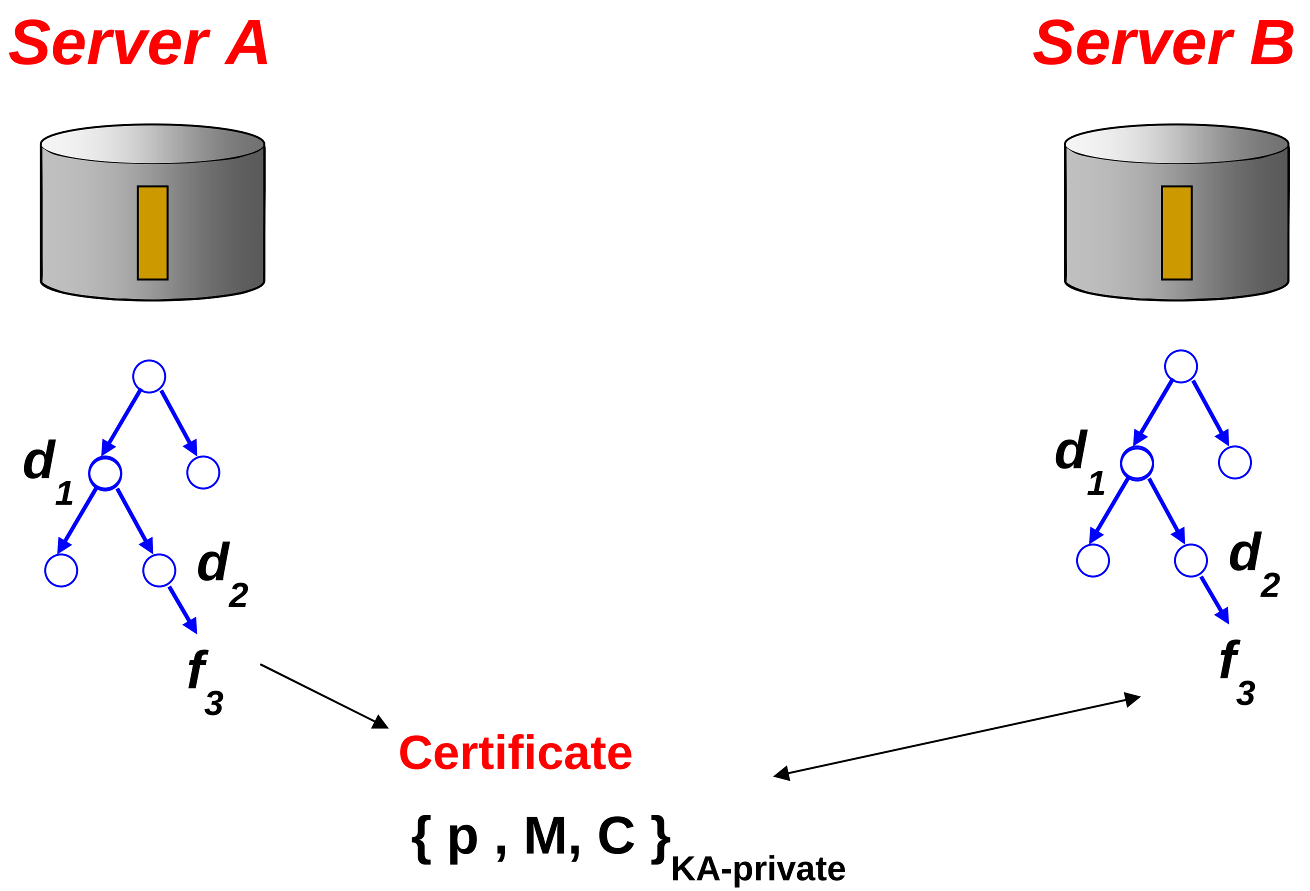


***content-hash*** **= C**

**Certificate**

$\{ p , M , C \}_{\text{KA-private}}$

# The certificate can be generated fairly quickly, used to verify integrity



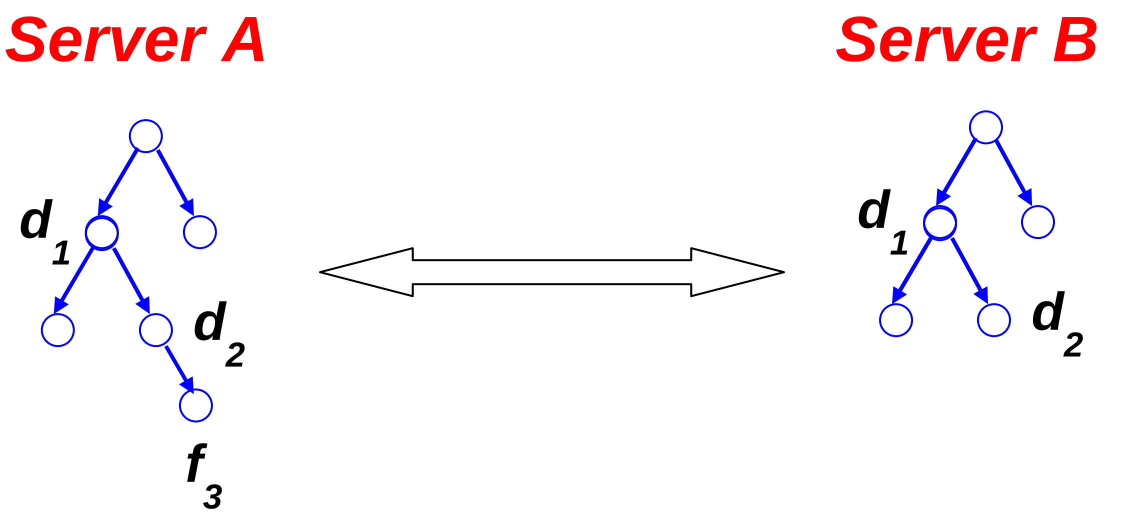
# Migration is solved!! (not really)

- Directory restructuring
- Policy-driven file omission
  - Omit files *owned by user Alice*
  - Omit files *containing specific keywords*

# Keep a (signed) log of the directory restructuring operations

- Given a file path on Server A, Bob can use the migration log to determine its path on Server B
- Bob can verify the file contents against the certificate signed by Server A

# Verification is tricky with file omissions

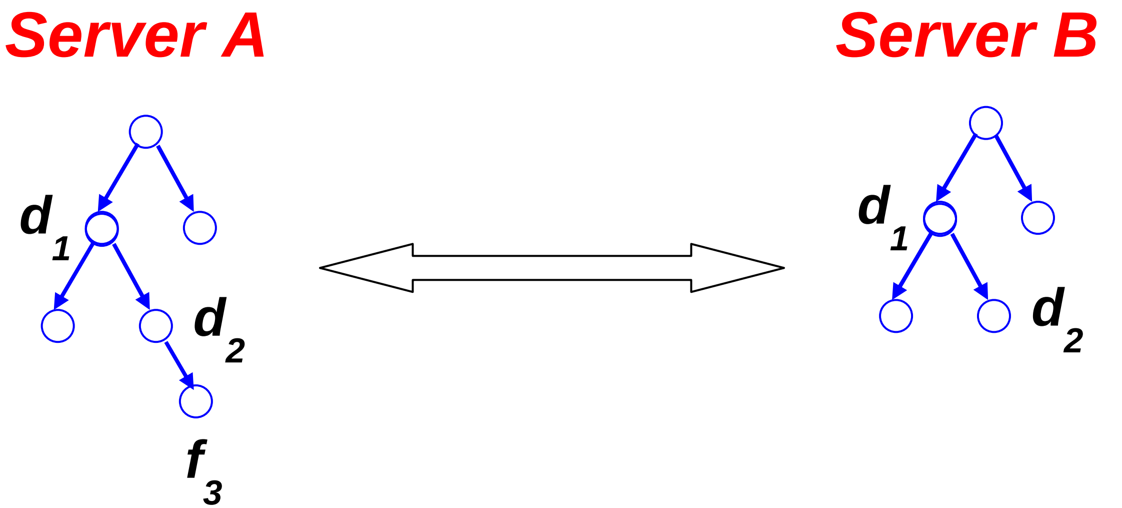



*Bob is looking for file* ***/d1/ d2/ f3*** *--- should not get the file*

*Bob should be able to verify that migration policy is satisfied*

*He should not learn any data/metadata (ex owner) about* $\boldsymbol{f_3}$

# Migration with omission is a 2-step process

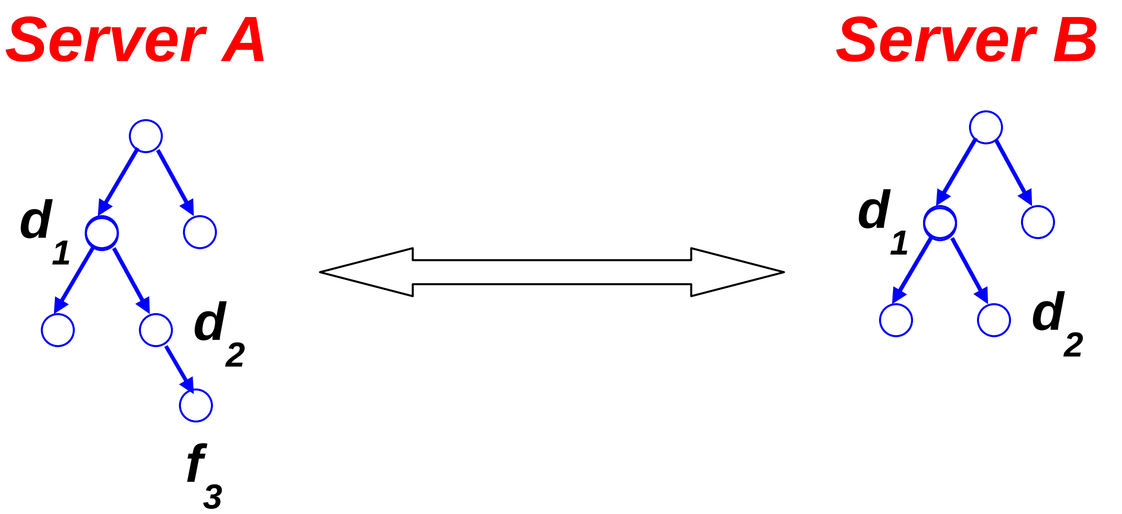



*Migrator records* DEL (***/$d_1$/ $d_2$/ $f_3$***) in the migration log

*Generates and copies its metadata certificate*

*Only the owner field is included in the certificate*

# Bob's verification




*Bob finds* DEL (***/d$_1$/ d$_2$/ f$_3$***) in the migration log

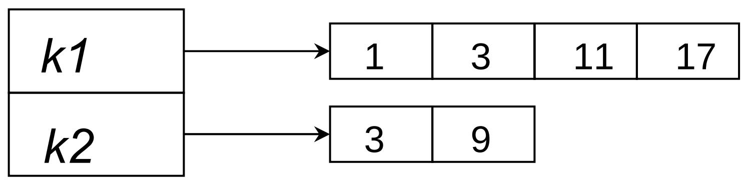
*Verifies that owner was indeed Alice using the certificate*

# The previous approach leaks information

- Bob gets to know that there was a file under /d1/d2/f3
  - Bob is able to prove that a file with that name was present
- Ideally he should not be able to distinguish between these cases:
  - /d1/d2/f3 was not present
  - /d1/d2/f3 was present and was legitimately omitted

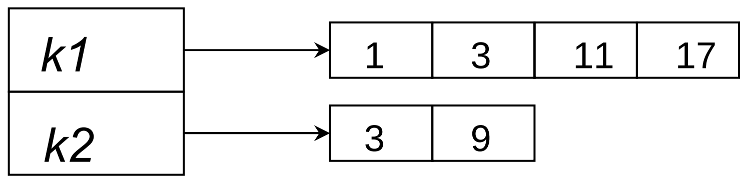
# Policies can be more complex: migrate documents with terms *k1* and *k2*

- Exploit the inverted index




- Migrate the keyword posting lists
- Bob can obtain the list of documents having *k1* and *k2*

# The posting lists leak information




- Bob gets to know the documents that have either *k1* or *k2*
- Ideally he should only learn about documents which have both *k1* and *k2*
- AND queries can be handled using crypto-tricks

# Take-home messages

- Compliance storage research is all about *insider threats*
- A secure coprocessor in the storage server buys a lot in terms of trustworthiness, but must be used very sparingly
- Traditional indexes are not trustworthy
- Trustworthy indexes are complex but usually quite fast

# Everything not mentioned in this tutorial is an open research problem

- Other kinds of trustworthy indexes
- Strongly secure deletion from GHTs, jump indexes
- (Many aspects of) litigation holds
- Index migration
- Removing info about expired files from migration logs
- Trustworthy databases and other high-level functionality
- Trustworthy indexing for object-based compliance storage servers
- …

# Thank you!