Security of Cloud Computing

Fabrizio Baiardi
f.baiardi@unipi.it

Syllabus

- Cloud Computing Introduction
  - Definitions
  - Economic Reasons
  - Service Model
  - Deployment Model
- Supporting Technologies
  - Virtualization Technology
  - Scalable Computing = Elasticity
- Security
  - New Threat Model
  - New Attacks
  - Countermeasures

Provenance

Provenance: documented history of an object

Provenance: from Latin *provenire* 'come from', defined as

- "(i) *the fact of coming from some particular source or quarter; origin, derivation.*
- *(ii) the history or pedigree of a work of art, manuscript, rare book, etc.; a record of the ultimate derivation and passage of an item through its various owners*" (Oxford English Dictionary)

In other words, **Who** owned it, **what** was done to it, **how** was it transferred …

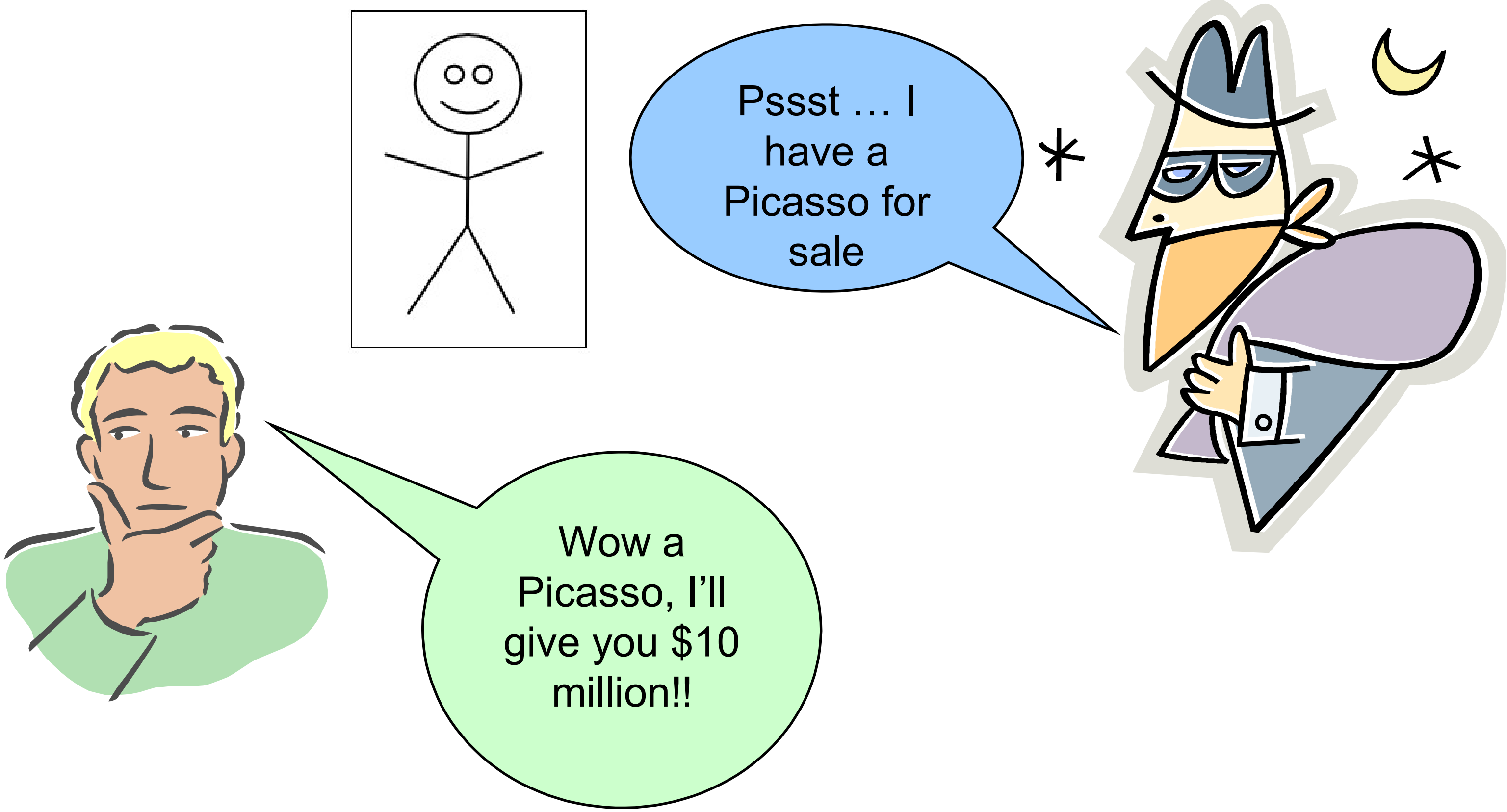
Widely used in arts, archives[1], and archeology

Provides a record of the ownership and operations on an object throughout its existence

Can be used to verify authenticity of the object

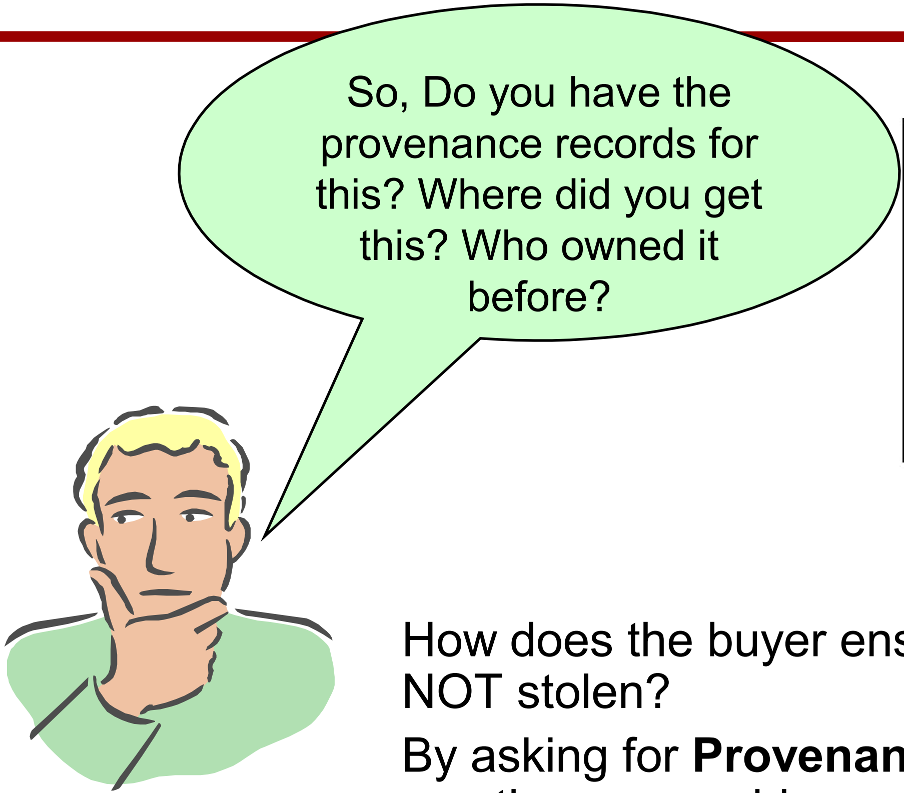
[1] called "Fundamental principle of archival"

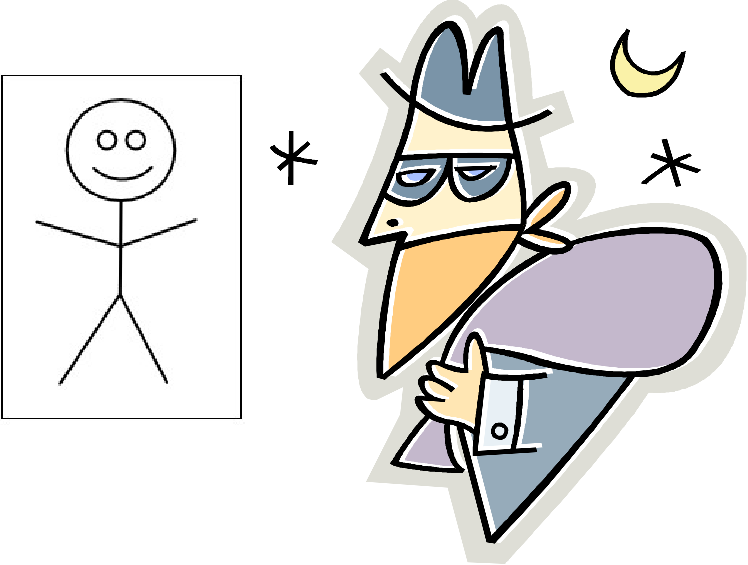
Provenance in arts: an example


Pssst … I have a Picasso for sale
Wow a Picasso, I’ll give you $10 million!!

How does the buyer ensure the painting is authentic and NOT stolen?

Provenance in arts: an example


So, Do you have the provenance records for this? Where did you get this? Who owned it before?

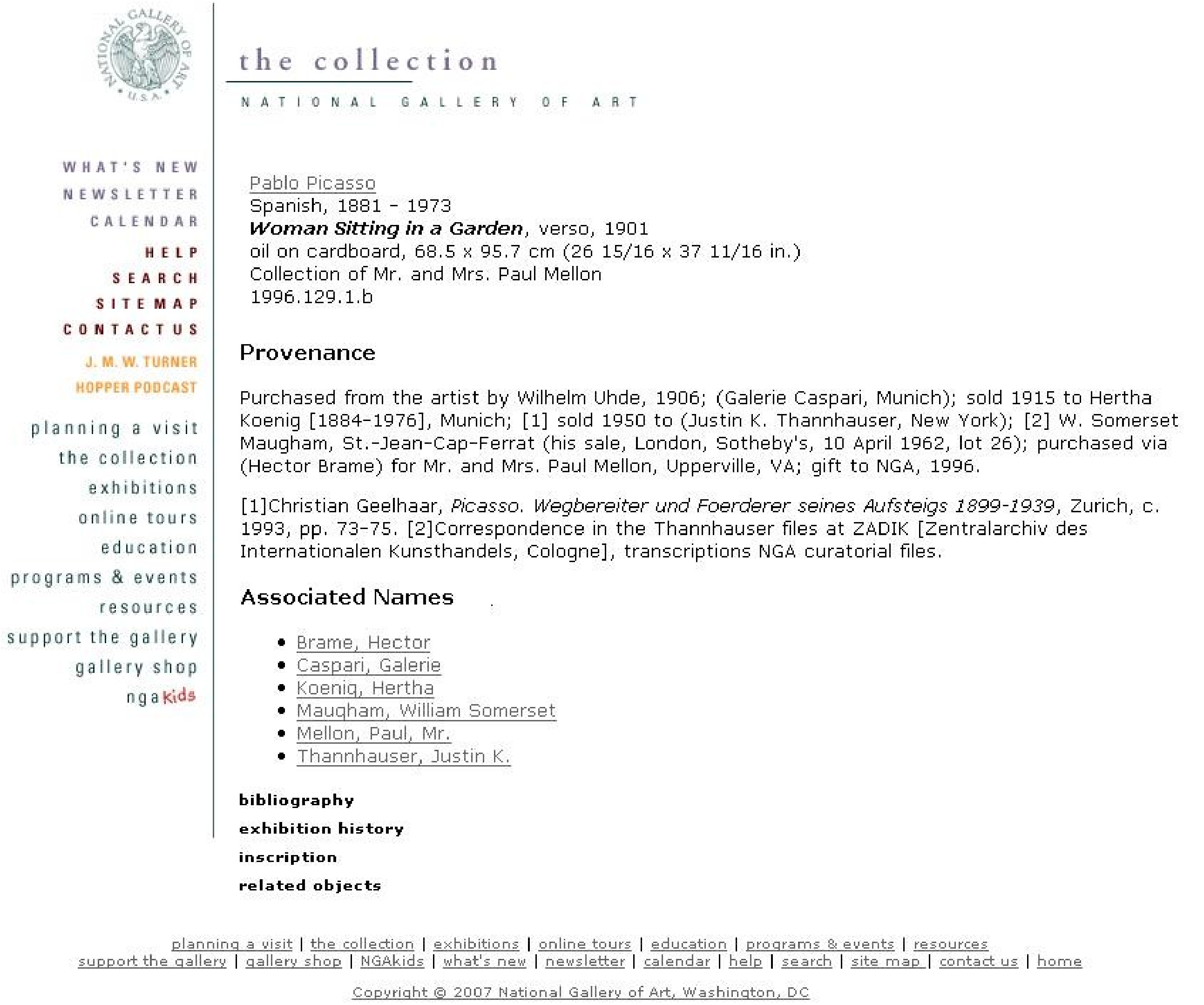


How does the buyer ensure the painting is authentic and NOT stolen?

By asking for **Provenance** records: documenting the creation, ownership, and migration history of the painting

"We can distinguish two meanings for provenance: first, as a concept, it denotes the source or derivation of an object; second, more concretely, it is used to refer to a record of such a derivation".[1]

Example from National Gallery



the collection

NATIONAL GALLERY OF ART

Pablo Picasso
Spanish, 1881 - 1973
Woman Sitting in a Garden, verso, 1901
oil on cardboard, 68.5 x 95.7 cm (26 15/16 x 37 11/16 in.)
Collection of Mr. and Mrs. Paul Mellon
1996.129.1.b

Provenance

Purchased from the artist by Wilhelm Uhde, 1906; (Galerie Caspari, Munich); sold 1915 to Hertha Koenig [1884-1976], Munich; [1] sold 1950 to (Justin K. Thannhauser, New York); [2] W. Somerset Maugham, St.-Jean-Cap-Ferrat (his sale, London, Sotheby's, 10 April 1962, lot 26); purchased via (Hector Brame) for Mr. and Mrs. Paul Mellon, Upperville, VA; gift to NGA, 1996.

[1]Christian Geelhaar, *Picasso. Wegbereiter und Foerderer seines Aufsteigs 1899-1939*, Zurich, c. 1993, pp. 73-75. [2]Correspondence in the Thannhauser files at ZADIK [Zentralarchiv des Internationalen Kunsthandels, Cologne], transcriptions NGA curatorial files.

Associated Names

- Brame, Hector
- Caspari, Galerie
- Koenig, Hertha
- Maugham, William Somerset
- Mellon, Paul, Mr.
- Thannhauser, Justin K.

bibliography

exhibition history

inscription

related objects

planning a visit | the collection | exhibitions | online tours | education | programs & events | resources
support the gallery | gallery shop | NGAkids | what's new | newsletter | calendar | help | search | site map | contact us | home

record for
Vomen sitting
”, from US
llery of Art

Provenance is widely used in Digital Systems

Provenance information used in two ways

- Data source:
  - Where did the data came from?
  - Are the sources trusted?
- Workflow:
  - How was the data created?
  - How did the data migrate?

Existing provenance based systems include

- Chimera (Physics/Astronomy)
- myGrid (Biology)
- CMCS (Chemistry)
- ESSW (Earth science)

Existing provenance systems do not consider security problems

Most of existing research focused towards provenance collection, annotation, and workflow

Without integrity, confidentiality and privacy guarantees, digital provenance would be useless.

So, we need provenance of provenance, i.e. a model for **Secure Provenance**

Introducing secure provenance

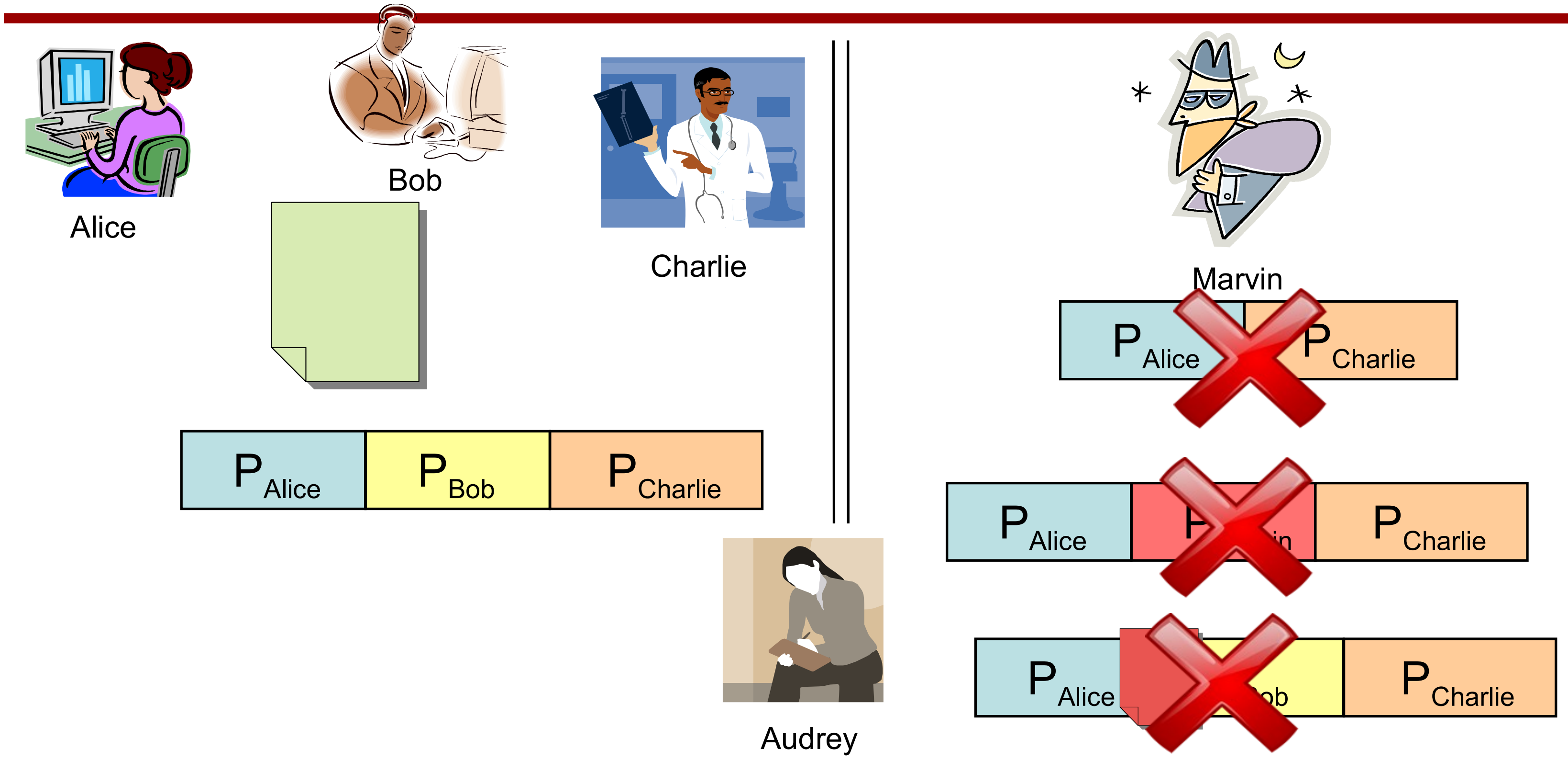
Model:

- Provenance information entries (P_i)contain information about document modifications by a user
- A time ordered sequence of P entries form the provenance chain C_D for the document D
- Authenticity of a claimed provenance is verified by an auditor

Secure provenance problem is to ensure that:

- An adversary (or a group of adversaries) cannot modify the chain by adding fake entries, or removing entries from valid users without being detected
- Users cannot repudiate their activities on the document
- Users can selectively preserve the privacy of their activities on the document
- Auditors can verify authenticity of a chain without requiring to learn the details

A scenario



Alice

Bob

Charlie

Marvin

Audrey

P_{Alice} P_{Bob} $P_{Charlie}$

P_{Alice} $P_{Charlie}$

P_{Alice} P_{Marvin} $P_{Charlie}$

P_{Alice} P_{Bob} $P_{Charlie}$

Marvin shouldn’t be able to add or remove entries in the chain, nor read contents, nor use this chain for a different document. Neither Alice, Bob, Charlie should be able to repudiate their actions.

Threat model

Who can be an adversary?

- Anyone, including Insiders, with access to local storage in the untrusted machines

Motivation of adversaries:

- To forge a provenance chain to claim a particular origin for a document
  - Example: Claim a fake purchase order to be processed through the mandated workflow
- To add or remove entries from the sequence
  - Example: Claim herself to have taken part in creating the data
- To gain information about the users, or the actions taken by them
  - Example: Figure out how a confidential report was created, and what process workflows were performed on it

Insight from art forgers: A forger gains most by claiming a painting to be from Picasso, rather than claiming a Picasso to be from him.

Secure provenance = Integrity + Confidentiality

Integrity: Auditors should be able to detect the following:

- Forgery of individual provenance records
- The sequence of records in the chain

Confidentiality: Privacy and confidentiality guarantees must be provided to prevent:

- Leakage of content of sensitive provenance entries
- Identity of principals involved

Issues in provenance lifecycle

Collection:

- How to collect provenance information in an untrusted environment

Storage:

- How to store provenance in an untrusted machine
- How to reduce the explosion of provenance data

Verification:

- How to verify provenance chain when a subset of auditors are not trusted

Migration

- How to handle migration of a document across organizational boundaries

We gain a lot by securing provenance information

Law enforcement

- In legal system, secure provenance can be used to secure the *Chain of Custody* for digital evidence, making them admissible in court

Scientific data management

- Can ensure trustworthiness of data

Digital forensics

- Trace the lineage of tainted binaries

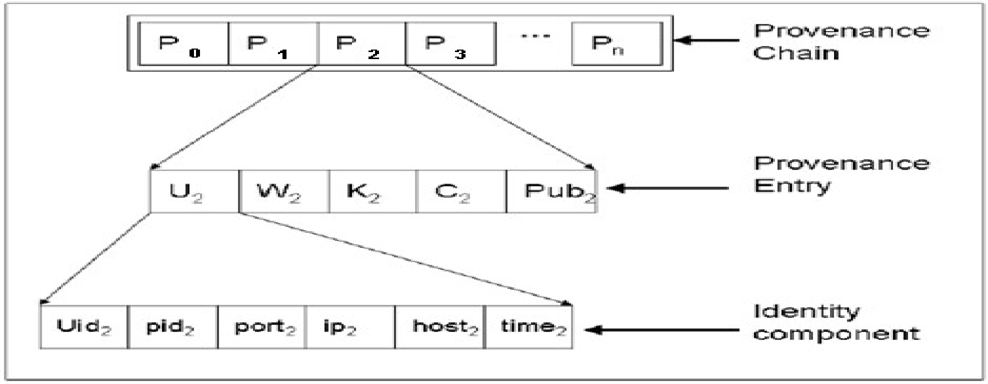
Regulatory compliance

- Provide mandated change and access history for compliance with regulations (such as HIPAA)

Authorship

- Provide proof of prior work and chronology for research in patent litigation
- Help resolve disputes in authorship of scientific research

A solution: Overview

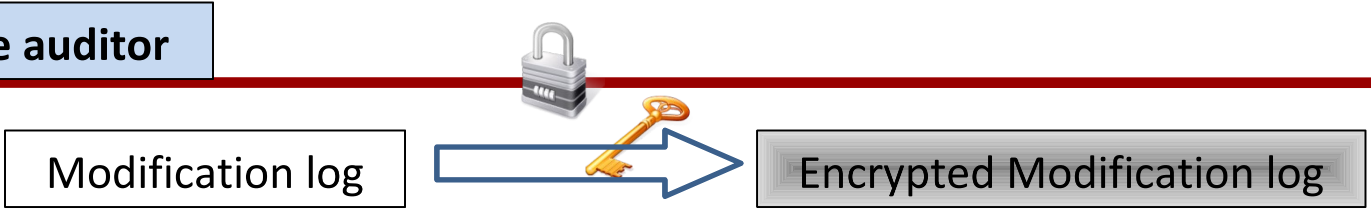


A *provenance entry* **Pi = <Ui,Wi, Ki, Ci, publici>**, where

- Ui the identity of the principal (lineage)
- Wi the (encrypted) document modifications in this entry
- Ki the confidentiality locks for Wi
- Ci the provenance entry integrity checksum(s)
- Publici the public key certificate for Ui (optional)

Confidentiality of chain

A single auditor

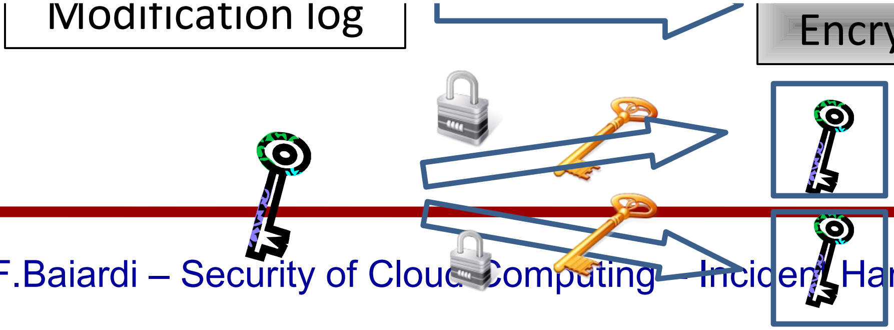


Multip

Issues

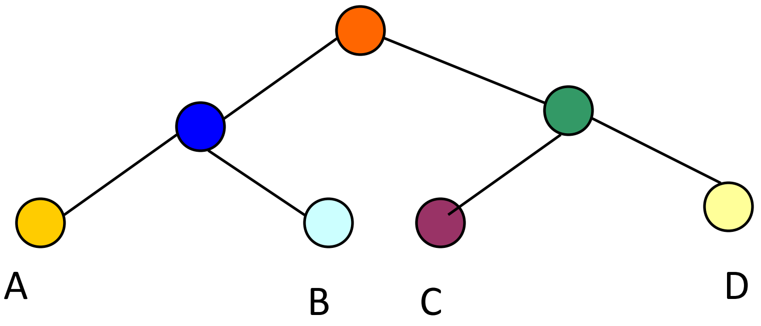
Each user trusts a **subset** of the auditors

Only the auditor(s) **trusted** by the user can see the user’s actions on the document



Optimization: Use broadcast encryption tree to reduce number of required keys

Sidenote: Broadcast Encryption



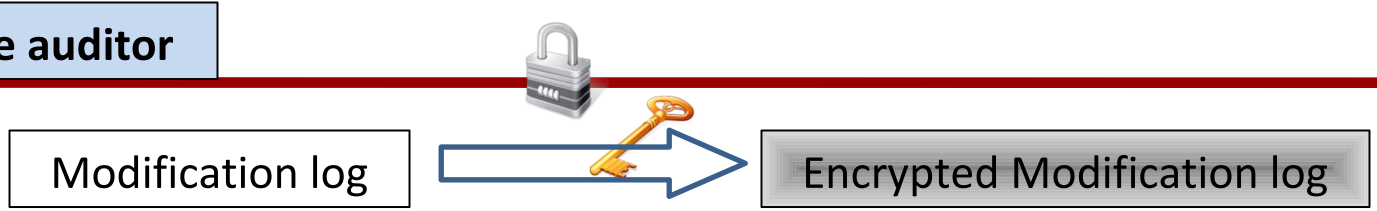
Organize the keys as a tree, with auditors at leaves

Each principal knows all the keys from leaf to root

- ● To trust everyone, use the root key
- ● To trust only D
- ● ● To trust B and C
- ● ● To trust A and C and D

Solution: Confidentiality

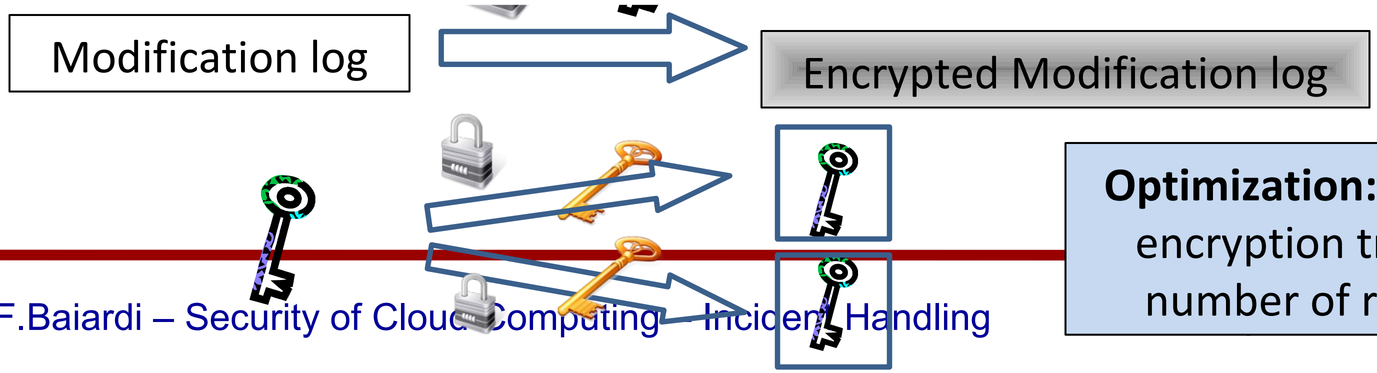
A single auditor



Issues

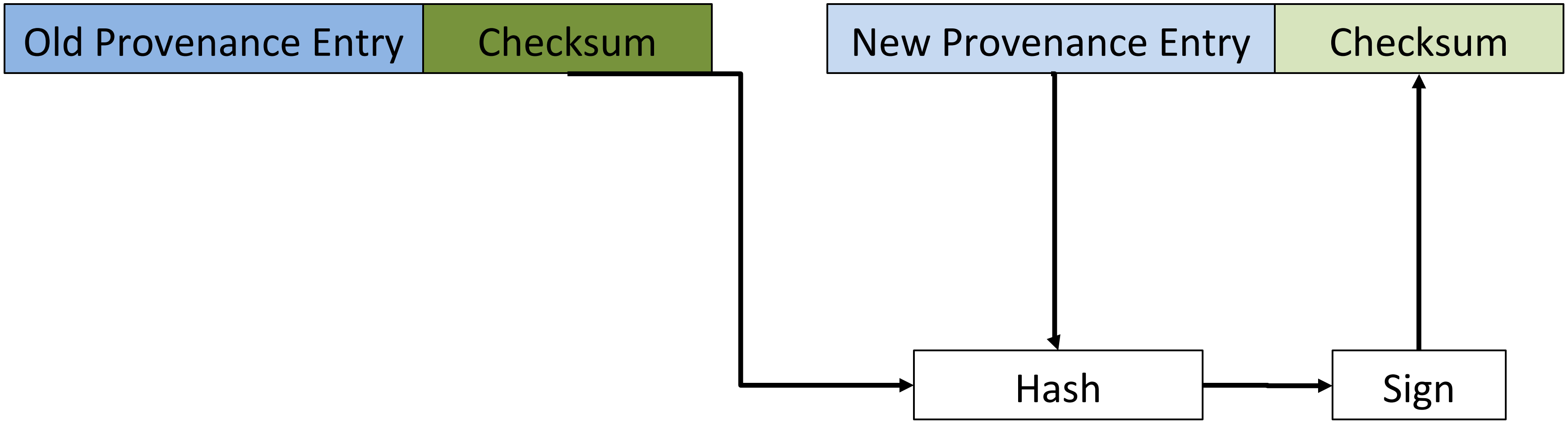
Multip

- Each user trusts a **subset** of the auditors
- Only the auditor(s) **trusted** by the user can see the user’s actions on the document



Optimization: Use broadcast encryption tree to reduce number of required keys

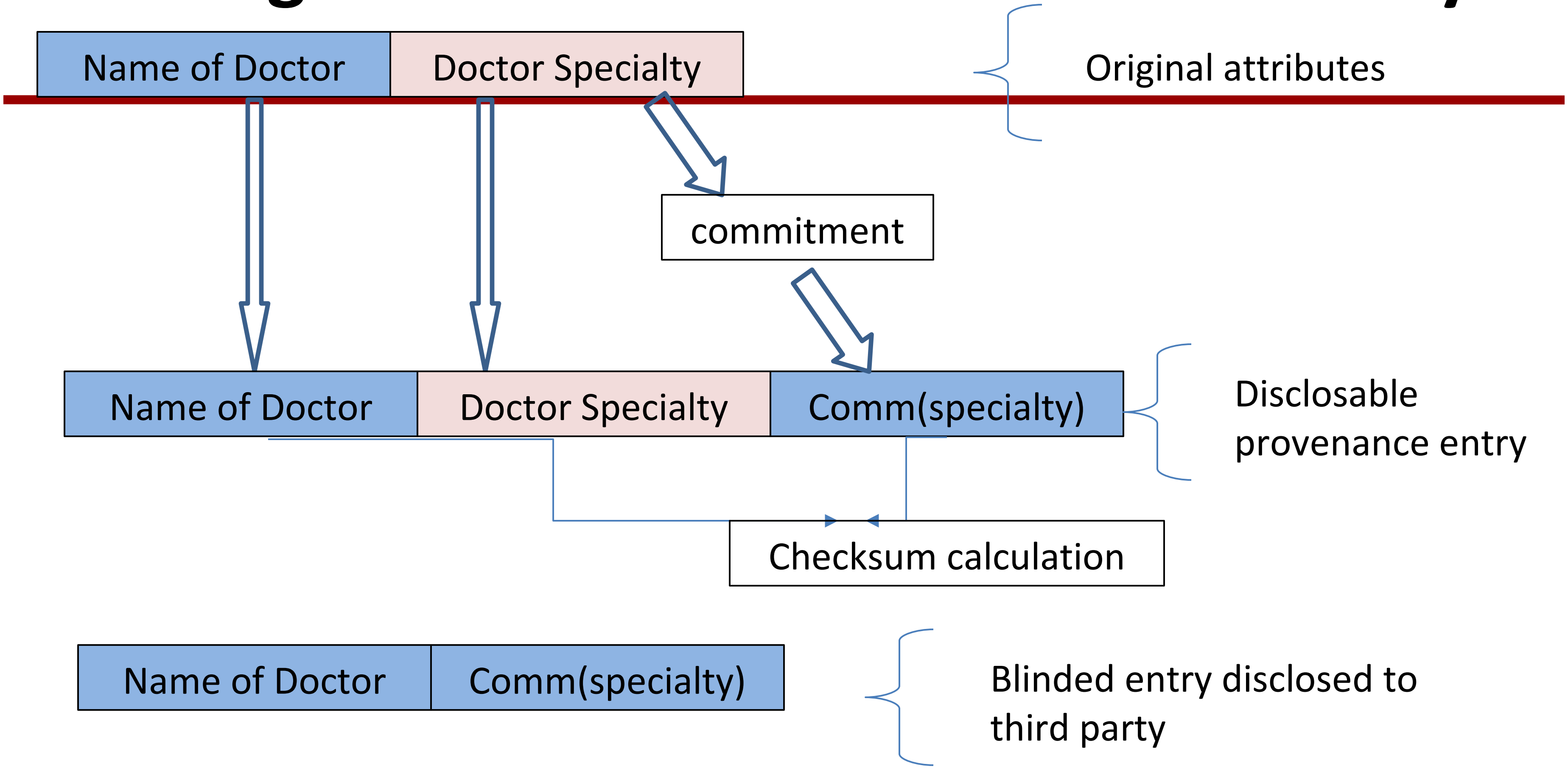
Solution: Integrity



- Each Pi contains a checksum computed as

Ci = Sprivate_i (hash(Ui,Wi,Ki)|Ci–1)

Fine grained control over Confidentiality



- We may need selective disclosure of provenance chain elements/attributes
- To allow future disclosures to third parties, we provide fine grained confidentiality control through **cryptographic commitments** as above

Cryptographic commitment

A scheme that allows one to commit to a chosen value while keeping it hidden to others, with the ability to reveal the committed value later.

Commitment schemes are designed so that a party cannot change the value or statement after they have committed to it

$\Rightarrow$ commitment schemes are binding.

Interactions in a commitment scheme take place in two phases:

a) the commit phase during which a value is chosen and specified

b) the reveal phase during which the value is revealed and checked

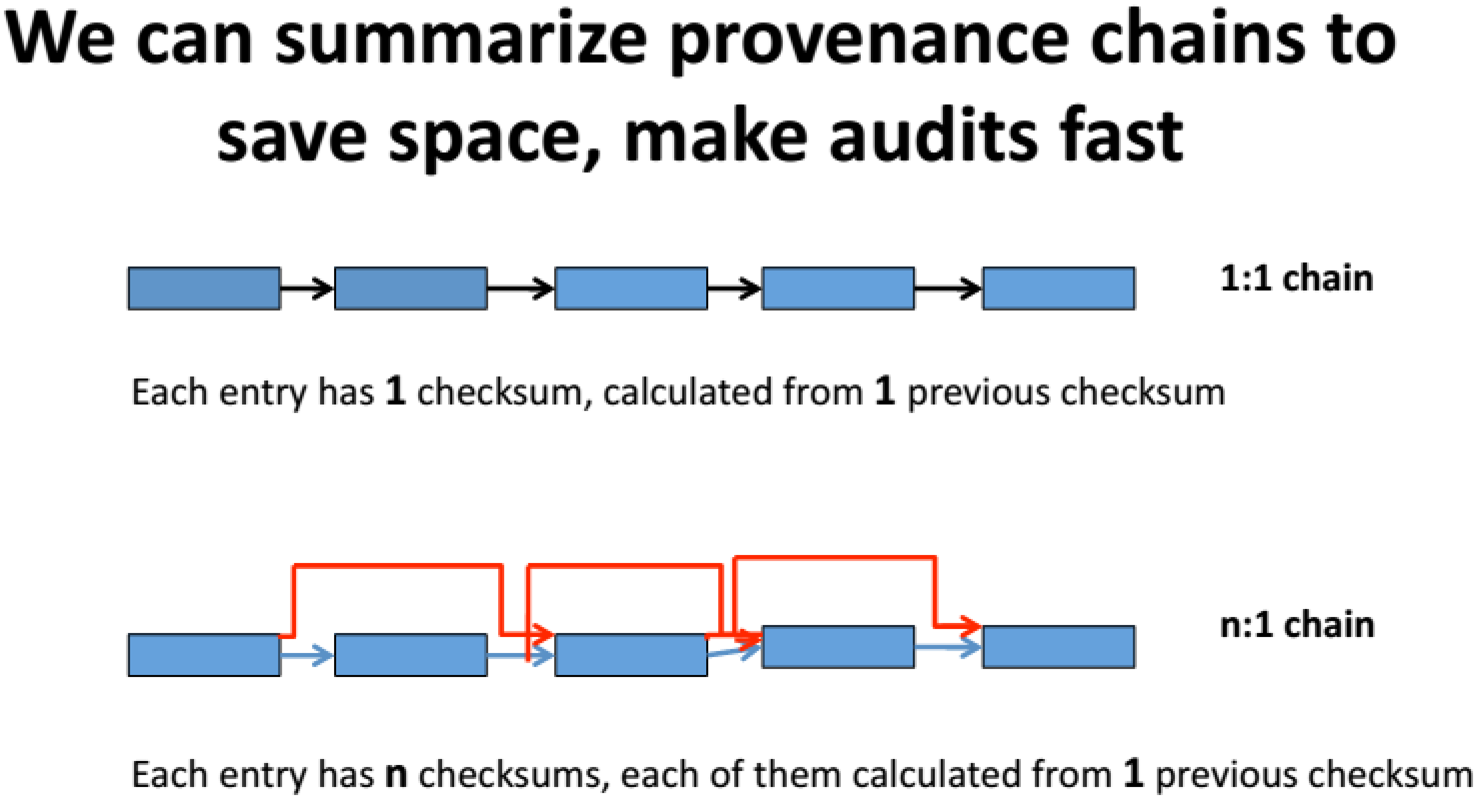
Cryptographic commitment

- To provide flexibility in such situations without a proliferation of broadcast encryption keys, cryptographic commitments are used for sensitive subfield and field data.
- We can omit plaintext data entirely when sending D's chain to a new organization, regardless of whether such a need was foreseen when setting up the session key(s) for D.
- The plaintext information can be restored to the chain if D later finds its way back to its original organization.
- To achieve this level of control without a proliferation of encryption keys, we replace each potentially sensitive subfield s inside Ui or Wi by its commitment before computing the checksum for Pi:

$$comm(s) = hash(s,rs) ,$$

where rs is a sufficiently large random number.

1:1, n:1, …



We can summarize provenance chains to save space, make audits fast

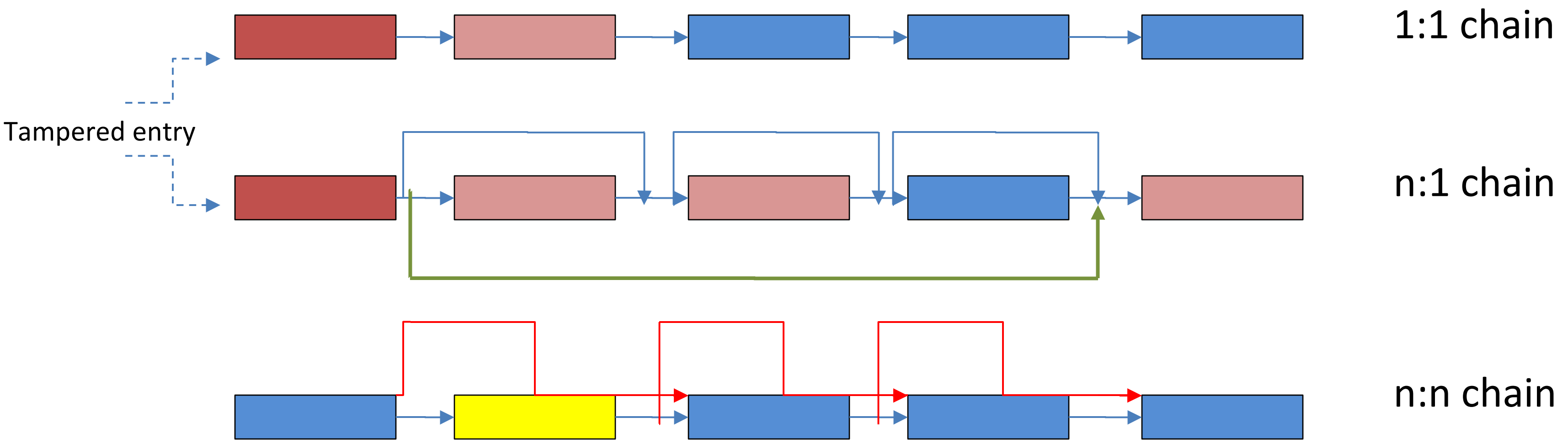
1:1 chain

Each entry has **1** checksum, calculated from **1** previous checksum

n:1 chain

Each entry has **n** checksums, each of them calculated from **1** previous checksum

Augmenting provenance chains



- The **n:1** chain scheme ups the ante for the attacker, as they have to modify more than one entry to evade any modification

- The **n:n** parallel chain scheme allows systematic removal of entries from the chain while being able to prove integrity of chain order

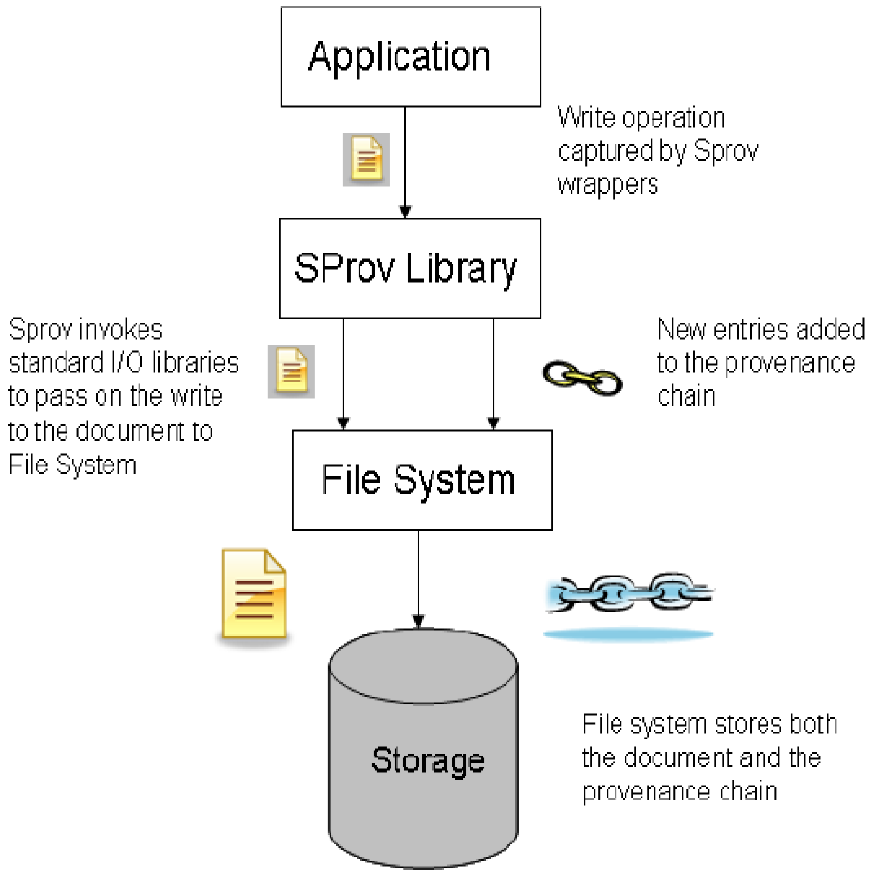
Augmenting provenance chains

- Quick detection of forgery of entries.
- In the singly-linked mechanism, the auditor has to verify the entire chain up to and including the entry Pi
- Quick local verification if multiple entries will be dependent on the checksum Ci of Pi.
- The new added checksum C'i will cause the checksums of these dependent entries to fail, and therefore expose the forgery.
- The **n:n** parallel chain scheme allows systematic removal of entries from the chain while being able to prove integrity of chain order

Implementation Options

- Secure provenance management can be implemented in
  - **Kernel layer**
    - Transparent to user apps
    - But less portable
  - **File System layer**
    - Also transparent to user apps
    - Less portable
  - **Application layer**
    - Needs (slight) modification of apps
    - Highly portable

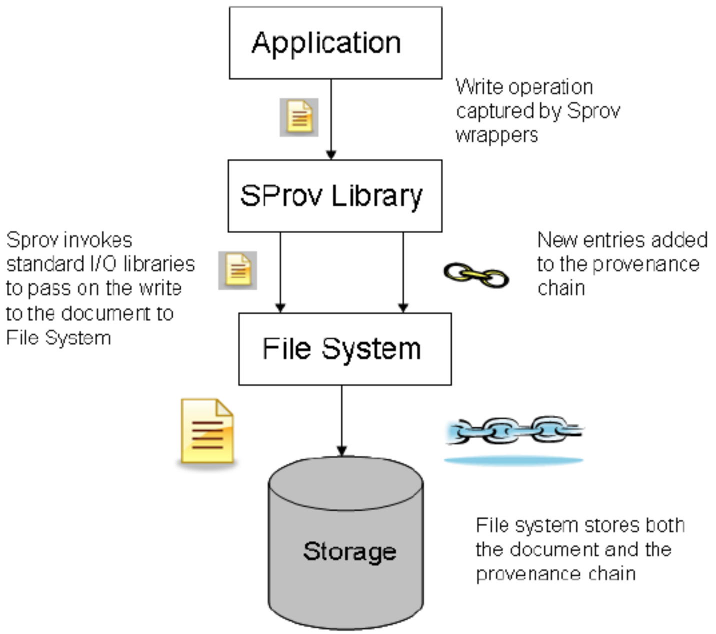
Implementation: Sprov library


Application
Write operation captured by Sprov wrappers
SProv Library
Sprov invokes standard I/O libraries to pass on the write to the document to File System
New entries added to the provenance chain
File System
Storage
File system stores both the document and the provenance chain

- Automated capture of writes to files
- Record all write data, and related context information in the provenance entry
  - SPROV is an application layer library in C
  - Can be added to existing applications with almost no change to code
  - Provides the file system APIs from stdio.h
  - To add secure provenance, simply relink applications with sprov library instead of stdio.h

Implementation: How Sprov works

- Modified **fopen / fwrite / fclose** calls
  - File I/O calls trigger provenance chain handling mechanism
  - Calls to fwrite log the modifications
  - Calls to fclose writes a new provenance entry, computes new checksum



Application
Write operation captured by Sprov wrappers
SProv Library
Sprov invokes standard I/O libraries to pass on the write to the document to File System
New entries added to the provenance chain
File System
Storage
File system stores both the document and the provenance chain

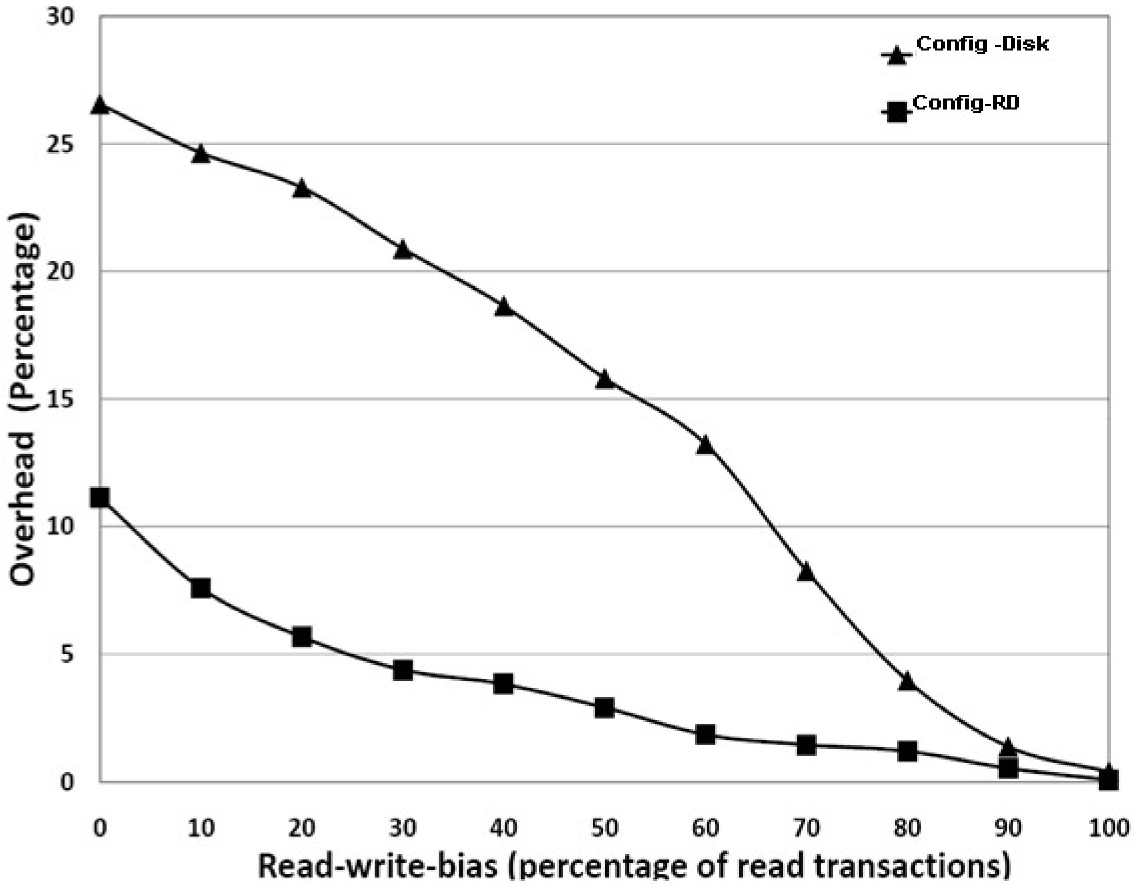
Experiment goals

- Measure run time overhead of secure provenance
  - Run the benchmarks with and without secure provenance
  - Calculate the % overhead
- Observe the effects of different types of workloads

Experimental settings

- **Crypto settings**
  - 1024 bit DSA signatures
  - 128 bit AES encryption
  - SHA-1 for hashes
- **Experiments**
  - **Postmark benchmark** : performance of writes on small files
  - **Hybrid workload benchmark** : performance with real life file system distribution and workloads

Postmark: small files



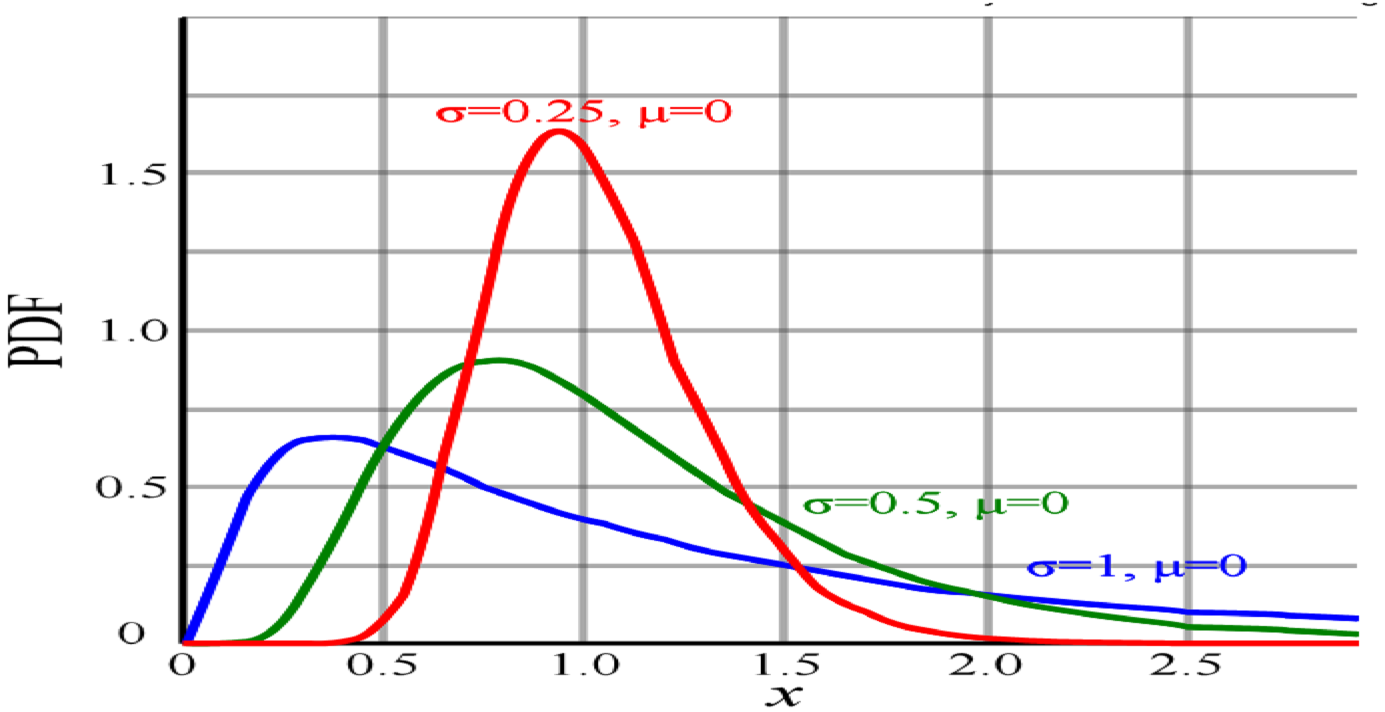
- **20,000** small files (8KB-64KB) subjected to 100% to 0% write load with the Postmark benchmark
- At 100% write load, execution time overhead of using secure provenance over the no-provenance case is approx. 27% (12% with RD)
- At 50% write load, overheads go down to 16% (3% with RD)
- Overheads are less than **5%** with 20% or less write load

Hybrid workloads: Simulating real file systems

File system distribution:

- File size distribution in real file systems follows the log normal distribution [Bolosky and Douceur 99]
- We created a file system with 20,000 files, using the lognormal parameters mu = 8.46, sigma = 2.4
- Median file size = 4KB , mean file size = 80KB
- In addition, we included a few large (1GB+) files

Hybrid workloads: Simulating real file systems



a log-normal (or lognormal) distribution is a probability distribution of a random variable whose logarithm is normally distributed. Thus, if X is log-normally distributed, then Y = \ln(X) has a normal distribution.

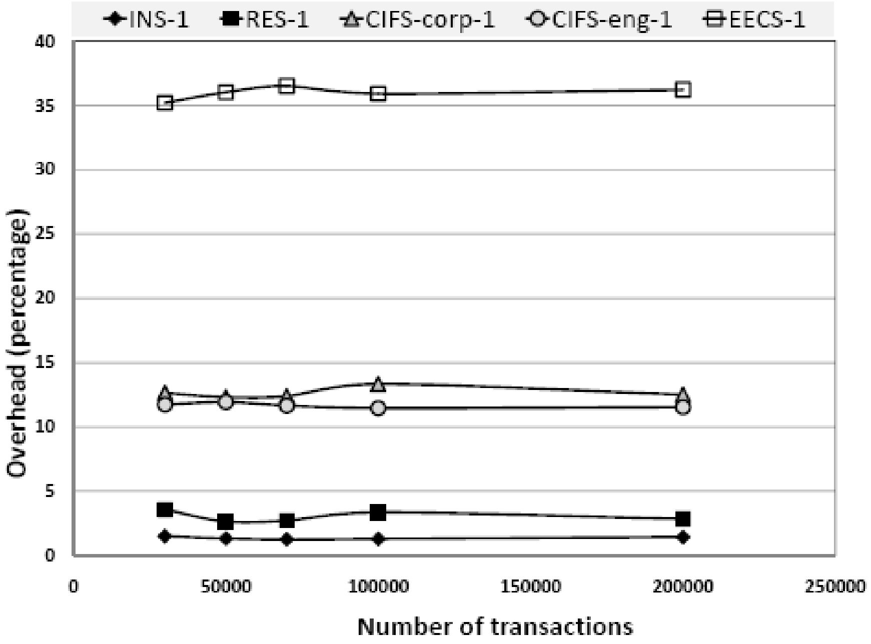
A variable as log-normal if i is the multiplicative product of many independent, positive random variables, . This is justified by considering the central limit theorem in the log domain.

Hybrid workloads: Data from real systems

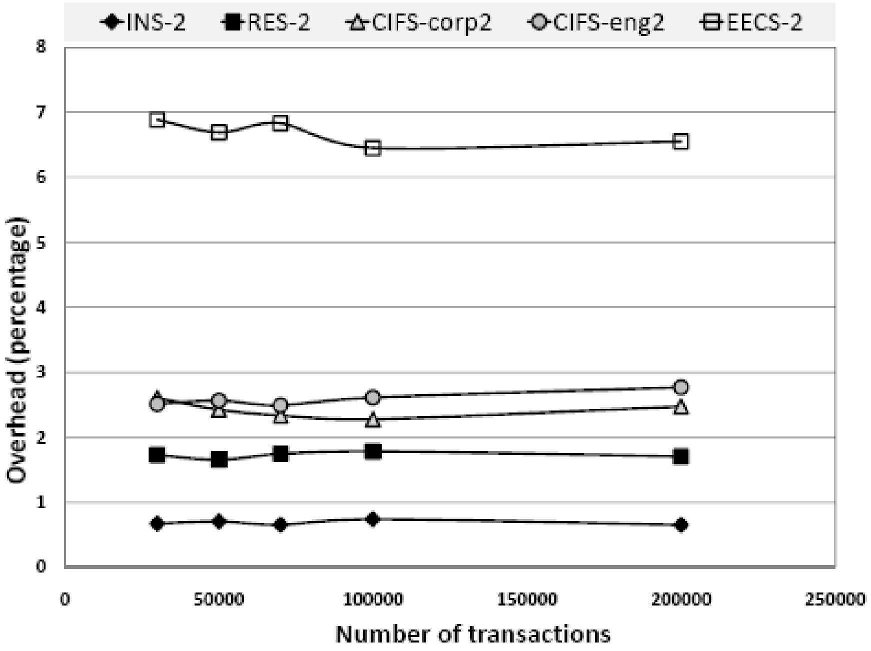
Workload

- **INS**: Instructional lab (**1.1**% writes) [Roselli 00]
- **RES**: A Research lab (**2.9**% writes) [Roselli 00]
- **CIFS-Corp**: (**15**% writes) [Leung 08]
- **CIFS-Eng**: (**17**% writes) [Leung 08]
- **EECS**: (**82**% writes) [Ellard 03]

Effect of real life workloads



Config-Disk



Config-RD

- **INS** and **RES** are read-intensive (80%+ reads), so overheads are very low in both cases.
- **CIFS-corp** and **CIFS-eng** have 2:1 ratio of reads and writes, overheads are still low (range from 12% to 2.5%)
- **EECS** has very high write load (82%+), so the overhead is higher, but still less than 35% for Config-Disk, and less than 7% for Config-RD

PASS: Provenance Aware Storage System [Muniswamy-Reddy et al, 2006]

A modified Linux File System for provenance

- Automatically collects provenance by intercepting system calls at the Kernel level
- Provenance information from input files are added to provenance of output files

Summary

- We can achieve secure provenance with integrity and confidentiality assurances
- For most real-life workloads, overheads are between 1% and 15% only

Final thoughts

Provenance information for digital documents can be used in various application scenarios

Securing the provenance chain will provide integrity, confidentiality, and privacy guarantees to the provenance chain

Any practical application of provenance information must implement secure provenance, at minimal performance overhead

Papers

- Ragib Hasan, Radu Sion, and Marianne Winslett, "Introducing Secure Provenance: Problems and Challenges", ACM StorageSS 2007
- Ragib Hasan, Radu Sion, and Marianne Winslett, "The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance", USENIX FAST 2009
- Ragib Hasan, Radu Sion, and Marianne Winslett, "Remembrance: The Unbearable Sentience of Being Digital", CIDR 2009