On Integrating Provenance Data in IPFS

Keno Goertz
Chair of Privacy and Security
TU Dresden
keno.goertz@tu-dresden.de

Erik Daniel
Hasso Plattner Institute
University of Potsdam
erik.daniel@hpi.de

Florian Tschorsch
Chair of Privacy and Security
TU Dresden
florian.tschorsch@tu-dresden.de

***Abstract*—In this demo paper, we present a practical approach for integrating provenance data into IPFS, supporting transparent and verifiable data management in distributed environments. Embedding provenance data into the root block enables efficient retrieval, versioning, and content discovery through a single CID, while remaining fully compatible with the existing network. We demonstrate the approach's practical utility for tracking provenance data of policies in a multi-provider cloud infrastructure.**

***Index Terms*—Data integrity, IPFS, Distributed Computing**

I. Introduction

Large amounts of data is continuously generated and processed to drive autonomous decision-making. Data integrity is becoming more important to ensure that autonomous systems can rely on their input data, to retrace decisions, and adjust their decision-making process.

Zero trust architectures [1] face similar challenges, as they dismantle the long-held notion of perimeter-based security, instead aiming to ensure authenticity and integrity at every step of the way, often across the networks of multiple organizations.

Provenance [2] data addresses these challenges by recording the origin, dependencies, and ownership of data, thereby enabling transparency and verifiability throughout its lifecycle. The *W3C PROV* model provides a framework to represent provenance data across diverse data types [3].

To ensure that such provenance data remains accessible and verifiable, a suitable data exchange infrastructure is required. peer-to-peer (P2P) data networks [4], such as the InterPlanetary File System (IPFS) [5], provide a self-scalable and decentralized mechanism for sharing content. In IPFS, data is exchanged via content-addressed blocks, which naturally enforces data integrity.

Previous work has explored related uses of IPFS for persistent and linked data. Sicilia et al. [6] propose using IPFS to implement decentralized persistent identifiers, but consider the approach only conceptually. The feasibility of sharing linked data in IPFS is more generally examined in [7]. Other research stores provenance data in separate IPFS blocks and makes them discoverable via blockchain-based indexing [8, 9], rather than integrating provenance data directly with the data.

In this demo paper, we show how provenance data can be integrated into IPFS to enable verifiable and transparent data management. Our approach leverages IPFS's Merkle directed acyclic graph (DAG) structure to link provenance data with the corresponding data. By embedding provenance data and metadata into the root block, both can be retrieved using the same content identifier (CID), facilitating lightweight content discovery. Furthermore, our integration method supports version tracking and creates a foundation for a searchable, provenance-aware P2P data network. We demonstrate our approach's practical utility within the context of policy files in a multi-provider cloud. Traceable policy changes are highly relevant in unified cloud infrastructures run by multiple companies, e.g., *8ra initiative*.[1] Such multi-provider infrastructures pose unique challenges to data integrity and auditability.

The remainder of the paper is structured as follows. In Section II, we present our approach for using IPFS as a provenance storage mechanism. Section III describes the setup and requirements of our demonstrator.

II. Provenance Data in IPFS

Provenance data provides detailed contextual information about a data item [2]. It describes how the data was created, by whom, and through which processes it reached its current state. The specific details captured depend on the application domain and intended use. Provenance data can serve multiple purposes [3]: it enables reproducibility of results, supports correct interpretation of data, and helps assess data quality. From a security perspective, provenance data can also record access and modification histories, thereby strengthening accountability and traceability. Typical provenance data records may include information about data sources (e.g., sensors or publishers) and versioning links to previous states of a dataset.

The structure of provenance data can follow the PROV specification [10], a generic and extensible model standardized by the W3C. PROV represents provenance data as a set of entities, activities, and agents, which are connected through well-defined relationships.

For instance, consider a cloud service admin named Alice updating a policy file, which is illustrated in Fig. 1. In this example, the updated policy file is modeled as an entity that `wasGeneratedBy` an activity (the update), which `used` the previous version of the policy file. As a result, policy v2 `wasDerivedFrom` policy v1, and `wasAttributedTo` Alice. The `wasAssociatedWith` relationship assigns responsibility for the update activity to Alice.

This research was supported in part by Telekom Deutschland GmbH under the European Union's IPCEI-CIS.

ISBN 978-3-903176-82-9 © 2026 IFIP

[1] https://8ra.com, *Accessed: 2026-03*

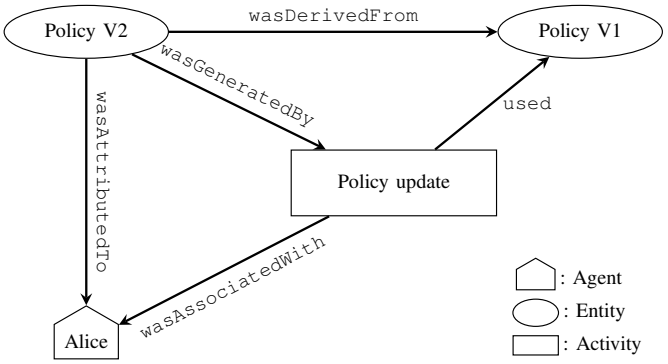

Fig. 1: An example representation of the *PROV* specification.

In a multi-provider cloud architecture, policies that determine access to cloud resources could be written and updated by admins from different organizations. Provenance data is vital to ensure such a system's auditability.

A. Background on IPFS

Our goal is to integrate provenance data directly into the data exchange layer by embedding it in IPFS. IPFS is a P2P overlay network that enables content-addressed data exchange. Data and peers are discovered using a Kademlia distributed hash table (DHT) [11]. The Bitswap protocol manages data transfer by querying neighboring peers for requested content and using the DHT as a fallback to locate additional providers.

The fundamental data unit in IPFS is a content-addressed block. Each block is identified by a CID, which consists of the block's hash, an identifier specifying the hash algorithm, and codec information. Larger datasets are divided into multiple blocks organized as a Merkle DAG, adhering to the InterPlanetary Linked Data (IPLD) model. The DAG's root block can optionally include metadata in form of the file's modification time and Unix permissions.

By nature, blocks in IPFS are immutable. Changing a block's data would change its CID, but a block with a different CID is treated as a different block altogether. For data provenance, however, mutability is a desirable property. Fortunately, the InterPlanetary Name System (IPNS) provides a cryptographically verifiable mechanism for mutable data in IPFS. IPNS records use asymmetric encryption. They map a name, derived from the hash of the public key, to a pointer—typically a CID. IPNS records are signed, ensuring that they can only be updated with access to the private key. Updates are managed using a sequence number. Records are distributed using standard IPFS mechanisms like the Kademlia DHT.

B. Concept

To include provenance data in the data exchange, it must be integrated into the Merkle DAG of IPFS. The Merkle DAG is particularly well-suited for this purpose, as data integrity is inherently ensured through the cryptographic hash used to generate the CID. Since CIDs are immutable, any modification to the data automatically produces a new CID, thereby producing a verifiable history of changes.

Provenance data can be integrated in different ways: as an additional block containing only provenance data, or as part of an existing non-data block. When added as a separate block referencing the root, both the data and its provenance data can be shared by announcing a single CID. In contrast, embedding provenance data deeper in the DAG, e.g., as a leaf block, would require retrieving the entire structure to access it.

To ensure efficient access and verifiability, we propose embedding the provenance data directly into the root block of the DAG. The provenance data is represented as key–value pairs within the block and pinned via the root block's CID. Data distribution and retrieval are then handled by the IPFS network: any peer with knowledge of the root CID or the IPNS name can locate the content provider and retrieve both the data and its associated provenance data.

C. Integration Path

Data is added to IPFS using the `ipfs add` command. The command processes the input, chunks it according to the configured parameters via the IPFS chunker, and generates a set of blocks, each identified by its own CID. These blocks are then organized into a Merkle DAG using the IPLD `DagBuilder`.

We add to `ipfs add` the flag `--provenance`, which allows specifying provenance data in JSON format. The provenance data is added directly to the root block. This is in line with the metadata fields `mtime` and `mode`, which are also present as optional fields directly in the root block.

After adding the block to IPFS, its CID can be published to IPNS with the command `ipfs name publish`.

D. Discussion

A key benefit of our approach is that it makes content discovery more accessible. While IPFS naturally allows data retrieval once a CID is known, locating relevant content across the network can be challenging due to its decentralized design. By embedding provenance data directly in the root block of the Merkle DAG, our approach enables lightweight inspection: retrieving only the root block is sufficient to determine whether the associated data is of interest, without accessing the full dataset.

Beyond improved discoverability, the usefulness of embedded provenance data depends on the credibility of the information. Since provenance data is only as trustworthy as its source, it should originate from an authenticated and verifiable source. Authentication is achieved with the cryptographic signatures of IPNS records. A public key infrastructure should be used to tie an IPNS public key to an entity.

Additional assurance can be provided through auditing mechanisms. For example, authorship or possession at a specific point in time can be confirmed via a timestamping service. Suitable trust anchors may include long-running bootstrap peers, reliable pinning services, or in private deployments

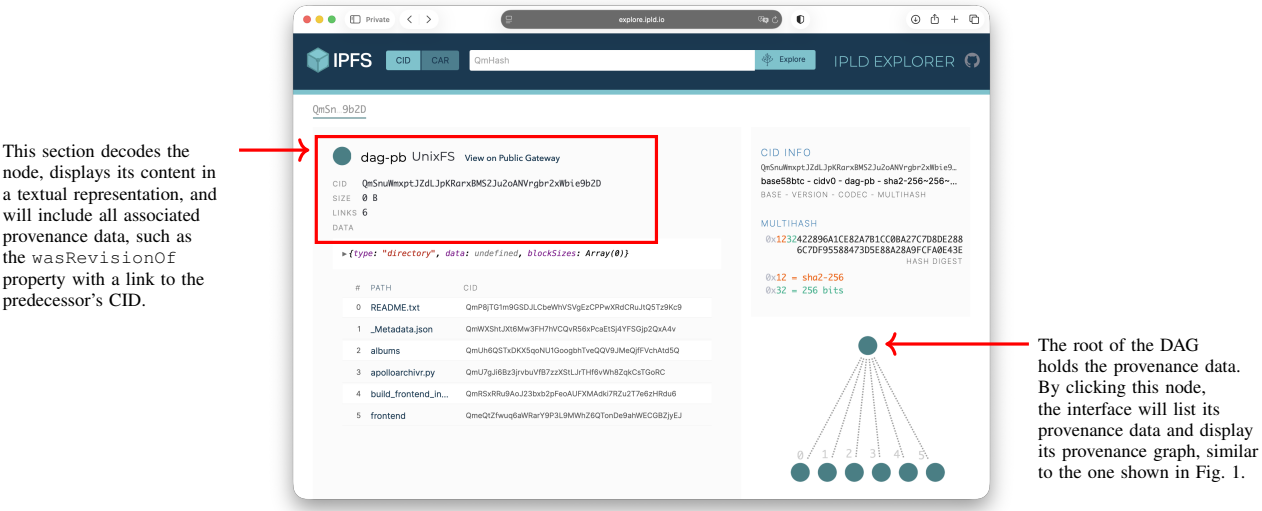

Fig. 2: Demo mockup of a web-based interface for displaying provenance data. The interface is based on the IPLD Explorer,[2] with the red arrows indicating where provenance data will be shown and can be interacted with.

IPFS Cluster nodes, which already manage data availability and can act as stable, trusted entities.

III. Demonstration

In our demonstration, we present an interactive walkthrough of how provenance data is integrated into the IPFS Merkle DAG. We visually illustrate the construction of the DAG and highlight where and how provenance data is embedded. To support exploration at an abstract level, we provide a web-based interface that allows users to inspect the stored data, which is visualized in Fig. 2. The code for the demonstrator is publicly available.[3] Through the interface, users can enter a CID. The corresponding block and all linked blocks are retrieved and displayed as a DAG. Individual blocks can be selected to reveal their associated provenance data.

Since our approach remains fully compatible with the existing IPFS network, it can be used transparently by any participant. During the demo, interested users can upload their own data to a server, enrich it with provenance data, and publish it to the public IPFS network. The resulting CID can then be retrieved by any IPFS node, demonstrating seamless interoperability with real-world deployments.

We motivate our approach's practical utility by using IPFS as a decentralized policy store for a prototypical multi-provider cloud infrastructure. To this end, we present a web page which checks with a policy file published on IPNS to decide which resources the user should get access to. In our demo, we show the transparent traceability of policy changes, due to our mechanism. Easy and transparent auditability is important to debug unintended consequences in a multi-provider infrastructure, in which one organization may base its decisions on the policies written by another organization.

[2]https://github.com/ipld/explore.ipld.io
[3]https://github.com/kenogo/ipfs-prov

References

[1] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, *Zero trust architecture*, en, 2020. DOI: https://doi.org/10.6028/NIST.SP.800-207 [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=930420

[2] F. Zafar, A. Khan, S. Suhail, I. Ahmed, K. Hameed, H. M. Khan, F. Jabeen, and A. Anjum, "Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes," *Journal of Network and Computer Applications*, vol. 94, pp. 50–68, 2017.

[3] B. Pan, N. Stakhanova, and S. Ray, "Data provenance in security and privacy," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 1–35, 2023.

[4] E. Daniel and F. Tschorsch, "IPFS and friends: A qualitative comparison of next generation peer-to-peer data networks," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 31–52, 2022.

[5] J. Benet, "IPFS - content addressed, versioned, P2P file system," *arXiv*, vol. abs/1407.3561, pp. 1–11, 2014. DOI: 10.48550/arXiv.1407.3561

[6] M.-A. Sicilia, E. García-Barriocanal, S. Sánchez-Alonso, and J.-J. Cuadrado, "Decentralized persistent identifiers: A basic model for immutable handlers," *Procedia computer science*, vol. 146, pp. 123–130, 2019.

[7] M.-A. Sicilia, S. Sánchez-Alonso, and E. García-Barriocanal, "Sharing linked open data over peer-to-peer distributed file systems: The case of IPFS," in *Research Conference on Metadata and Semantics Research*, Göttingen, Germany, Nov. 2016, pp. 3–14.

[8] S. Khatal, J. Rane, D. Patel, P. Patel, and Y. Busnel, "Fileshare: A blockchain and IPFS framework for secure file sharing and data provenance," in *ICMLCI '19: Proceedings of International Conference on Machine Learning and Computational Intelligence*, Bhubaneswar, Kantabada, India, Dec. 2019, pp. 825–833.

[9] S. S. Hasan, N. H. Sultan, and F. A. Barbhuiya, "Cloud data provenance using IPFS and blockchain technology," in *Proceedings of the Seventh International Workshop on Security in Cloud Computing*, Auckland, New Zealand, Jul. 2019, pp. 5–12.

[10] P. Missier, K. Belhajjame, and J. Cheney, "The w3c prov family of specifications for modelling provenance metadata," in *EDBT' 13: Proceedings of the 16th International Conference on Extending Database Technology*, Genoa, Italy, pp. 773–776.

[11] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the XOR metric," in *IPTPS'02: Proceedings of the 1st International Workshop on Peer-to-Peer Systems*, Cambridge, MA, USA, Mar. 2002, pp. 53–65.