Data provenance in security and privacy

BOFENG PAN and NATALIA STAKHANOVA, University of Saskatchewan, Department of Computer Science, Canada
SUPRIO RAY, University of New Brunswick, Faculty of Computer Science, Canada

Provenance information corresponds to essential metadata that describes the entities, users, and processes involved in the history and evolution of a data object. While the benefits of tracking provenance information have been widely understood in a variety of domains, only recently provenance solutions have gained interest in security community. Indeed, on the one hand, provenance allows for a reliable historical analysis enabling security-related applications such as forensic analysis and attribution of malicious activity. On the one hand, the unprecedented changes in the threat landscape place demands for securing provenance information to facilitate its trustworthiness.

With the recent growth of provenance studies in security, in this work, we examine the role of data provenance in security and privacy. To set this work in context, we outline fundamental principles and models of data provenance and explore how the existing studies achieve security principles. We further review the existing schemes for securing data provenance collection and manipulation known as secure provenance and the role of data provenance for security and privacy, which we refer to as threat provenance.

CCS Concepts: • **Security and privacy** → **Database and storage security**.

Additional Key Words and Phrases: Data provenance, Security, Privacy, Secure provenance, Threat provenance

1 INTRODUCTION

How has the environment changed since the intrusion alert was generated? Who had access to private user data and when? Why does the final version of a file look like that? Who has made the most effort to maintain the entire database?

In essence, answering these questions requires an understanding of the origins and the history of data in its life cycle. Such information is known as *data provenance*. Broadly speaking, provenance (also called *lineage*) refers to metadata describing the origins, history, or evolution throughout the life cycle of an end product. This includes the whole spectrum of entities, data, processes, activities and users involved throughout the process.

Introduced in the database community [20], provenance has evolved to elicit significant interest in many fields: tracking of products in supply chain [85, 86], validation and reproducibility of results in scientific experiments [26, 30, 31], and debugging and refinement of data processing [62].

Driven by the explosion of digital data constantly created, copied, transferred and manipulated through online platforms, provenance has played a significant role in security. Indeed, provenance provides assurance about the correctness of data modifications, enables data forensics, and allows us to verify access through a historical perspective.

Authors' addresses: Bofeng Pan, panbofeng@hotmail.com; Natalia Stakhanova, natalia@cs.usask.ca, University of Saskatchewan, Department of Computer Science, Saskatoon, Saskatchewan, Canada, S7N 5C9; Suprio Ray, sray@unb.ca, University of New Brunswick, Faculty of Computer Science, Fredericton, New Brunswick, Canada, E3B 5A3.

0360-0300/2023/4-ART $15.00
https://doi.org/10.1145/3593294

Due to the versatility of provenance use-cases, it has recently gained significant interest in the security community. Data provenance has shown tremendous value in numerous security and privacy applications. For example, for privacy-preserving network analysis to explore system behavior while providing guarantees for organizations' privacy preferences [148], for security protection of data by controlling users' access using provenance [100], or to protect system integrity [120].

Although historically, provenance has been primarily used with legitimate data, more recently, researchers started leveraging provenance as a means for tracking anomalous events, including the detection of intrusions [17, 139] and malware [127]. With unprecedented changes in the threat landscape, the need for securing provenance information has emerged. Given the rapid evolution of provenance applications, provenance data have become a lucrative target. Indeed, compromising provenance data might not only leak sensitive information (e.g., the components of a product), but also potentially undermine the trustworthiness of the system (e.g., bank records).

In this paper, we present a comprehensive survey on the role of data provenance in security and privacy. Data provenance and provenance, in general, have been actively explored in the research literature [25]. Bose et al. [25] provided a broad overview of provenance in various domains from the lineage retrieval perspective. More recently, Herschel [62] offered a more focused view of provenance studies across provenance types and proposed a classification of provenance research. Oliveira et al. [98] surveyed provenance analysis techniques. Beyond these studies, over the past two decades, several surveys have explored the use of provenance in various application domains: e-science [112], distributed systems [50], database systems [34, 122], the cloud [146], and scientific experiments [25, 98]. While extensive, these surveys predate a recent growth of provenance studies in security and hence do not discuss security and privacy challenges of provenance research. Lee et al [76], Tan et al. [123], and Zipperle et al. [149] are the only surveys that discussed the security and data accountability implications of provenance solutions. Lee et al. [76] gave a brief overview of ten secure data provenance schemes. Tan et al. [123] outlined the security requirements of distributed systems and briefly explored how they can be achieved through the capabilities of the existing provenance models. Zipperle et al. [149] narrowed its focus on intrusion detection provenance systems.

We believe that a comprehensive, systematic and up-to-date survey of the existing research is essential for researchers planning to initiate research in this direction.

Our contributions are as follows:

- We provide an overview of data provenance and its related concepts. We provide essential background knowledge for data provenance security and privacy properties and highlight threats to data provenance models and technologies.
- We provide a comprehensive, systematic and up-to-date overview of the existing data provenance research in the security and privacy field focusing on two aspects: *threat provenance* and *secure provenance*. We discuss the existing threat provenance studies and the associated mechanisms for tracing threats outlining their advantages and limits. We analyze the state-of-the-art secure provenance solutions that address the existing security and privacy of the provenance data and the provenance users. To the best of our knowledge, this is the first attempt at this scale to systematize the existing studies in the area.
- We analyze research gaps in the area of secure provenance and threat provenance. Our analysis serves as a guide through the existing research exposing underexplored areas.

The remainder of the paper is organized as follows: In Section 2, we introduce the basic concepts of the provenance, its properties, and models. In Section 3, we introduce the theory of secure provenance and discuss high level categories. Section 4 surveys the existing approaches for secure provenance, and Section 5 discusses existing threat provenance solutions. Finally, Section 6 provides practical insights into gaps in the existing mechanisms and Section 7 concludes the work.

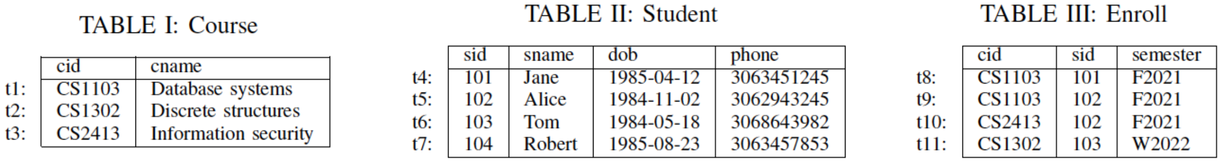


TABLE I: Course

	cid	cname
t1:	CS1103	Database systems
t2:	CS1302	Discrete structures
t3:	CS2413	Information security

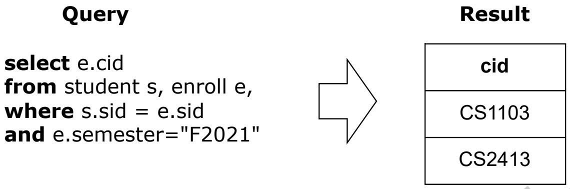
TABLE II: Student

	sid	sname	dob	phone
t4:	101	Jane	1985-04-12	3063451245
t5:	102	Alice	1984-11-02	3062943245
t6:	103	Tom	1984-05-18	3068643982
t7:	104	Robert	1985-08-23	3063457853

TABLE III: Enroll

	cid	sid	semester
t8:	CS1103	101	F2021
t9:	CS1103	102	F2021
t10:	CS2413	102	F2021
t11:	CS1302	103	W2022

Fig. 1. Provenance representation example



Query

```
select e.cid
from student s, enroll e,
where s.sid = e.sid
and e.semester="F2021"
```

Result

cid
CS1103
CS2413

Fig. 2. A query for provenance representation example provided in Figure 1

2 PROVENANCE OVERVIEW

The definition of provenance has evolved over the years, often changing based on the application context. Historically, in database systems, provenance was known as *lineage* and referred to the source of data as a result of query processing, i.e., *data provenance*. The concept of data lineage was first formalized by Cui et al. [40], and in the context of relational databases, it was used to identify each tuple *t* in the set of input tables that contributed to the output of a query [35].

The data provenance or lineage was viewed from three angles [34]: *why-provenance*, i.e., the set of minimal input tuples that contributed to the result; *how-provenance*, specifying how the output was generated from the minimal input set; and *where-provenance*, a mapping between the specific output fields and the input fields.

To illustrate these concepts, consider a set of tables given in Figure 1 and an example query in Figure 2, which retrieves all courses in which students were enrolled during the F2021 semester. The output of this query includes two tuples, and the lineage of the first output tuple (CS1103) is {Student (*t4*,*t5*) and Enroll (*t8*,*t9*)}. Here, Student (*t4*,*t5*) represents sub-instances of the source table Student with the tuples *t4* and *t5* (see Figure 1). Similarly, Enroll (*t8*,*t9*) are the sub-instances of the corresponding source table. The tuples in each sub-instance can be said to serve as the 'witness' for the output tuples, because they justify the existence of the output tuples.

The idea of a witness is the basis of why-provenance, which was formalized by Buneman et al. [28] in the context of a semi-structured data model. With why-provenance different witnesses of output tuples are identified. For instance, the why-provenance of (CS1103) is {*t4*,*t5*,*t8*,*t9*}.

The why-provenance, however, does not clarify how an output tuple is derived from the input table based on the execution of the query. To support this, the how-provenance was introduced, which is based on the formal notion of provenance semirings [54]. The provenance of each output can be described by a polynomial. For the considered query, the provenance of the output tuple (CS1103) is $t4 \times t8 + t5 \times t9$.

While data provenance in general was mostly concerned with content, in scientific and collaborative environments provenance was emphasized as the flow of execution. More specifically, *workflow provenance* was defined as a set of steps that were executed to achieve the results along with information about the environment used in execution, performed activities and configuration parameters. This view of provenance is often referred to

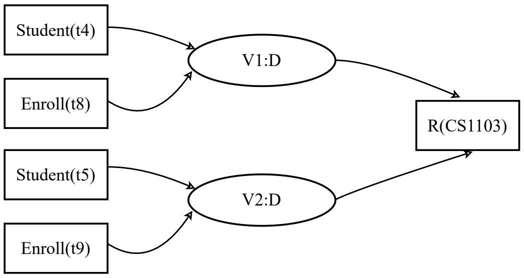


Fig. 3. Provenance graph representation of the how-provenance showing the partial results of the query given in Figure 2.

when issues of reproducibility, reusability and troubleshooting are raised, e.g., in distributed systems or scientific computations.

Along with data and workflow provenance, Herschel and Hlawatsch [63] defined *information systems provenance*, and *provenance metadata* as more general types of provenance, while Anjum [12] took a step further and outlined *process provenance*, that addresses tracking the data dependencies involved in the transformation process that produce a data item. Any metadata about processes within any information system are considered under the purview of process provenance. Therefore, workflow provenance and information systems provenance are subsumed by process provenance.

Whereas process provenance is a coarser-grained form of provenance because of its generality and wider applicability, data provenance can be considered a more fine-grained provenance, due to its focus on low-level (e.g., tuple) transformations.

Over the past decade, a number of different provenance types and granularities have been considered [98]. Along these types, a notion of *secure provenance* has emerged, emphasizing the need for securing provenance information. Indeed, with the unprecedented changes in the threat landscape, securing provenance information became critical to facilitate its trustworthiness. In this work, we narrow our focus to the role of data provenance in security and privacy.

2.1 Provenance representation

Similar to provenance definition, efforts in conceptual modeling of provenance metadata can be traced back to earlier work in the database community. Graphs have traditionally been considered as the most general way to formally represent database provenance. As such, any graph used to model provenance is referred to as a *provenance graph*.

In general, the specific format of captured provenance data depends on the domain. **A directed acyclic graph (DAG)** is a common provenance representation. In a DAG, each node represents an entity, and each edge represents the relationship between two entities. An entity can be represented, for example, by a file or a process, while the relationship denotes information flow from inputs to outputs [27]. DAG is a suitable way to represent provenance, since it can capture the relationship and dependence structure that may be present among the entities.

For example, how-provenance and why-provenance can be represented as DAGs to connect outputs to the inputs [4]. An example of a DAG provenance graph is shown in Figure 3. It represents the provenance of the output tuple (CS1103), where the tuple nodes (square) are connected by derivation nodes (oval). The derivation node represents an algebraic expression based on provenance semirings. For instance, the two tuple nodes labeled Student(*t4*) and Enroll(*t8*) contributed to the generation of the result tuple node R(CS1103). The label of each derivation node contains a unique node id and a derivation expression. For the node labeled *V1:D*, *V1* is its node

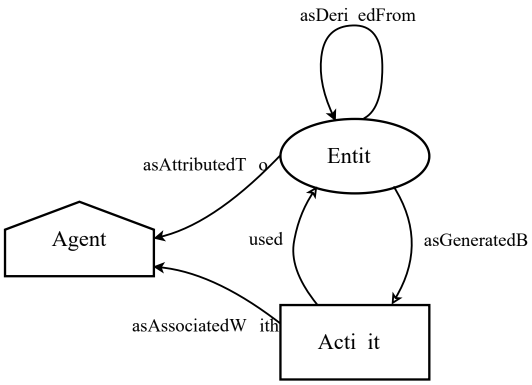


Fig. 4. The PROV model

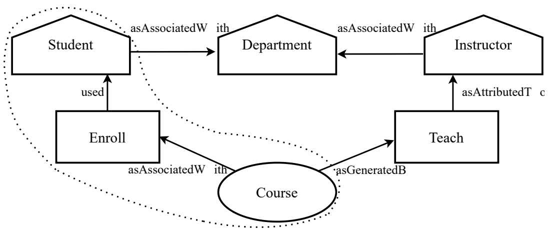


Fig. 5. An example based on the PROV model: objects enclosed in dotted line correspond to the example tables in Figure 1 and query in Figure 2

id and D the derivation expression ($t4 \bowtie t8 \cup t5 \bowtie t9$). This provenance graph can also be formally represented as a bipartite graph, $G = (T, D, E)$, with vertices $T \cup D$ and edges $E \subseteq (T \times D) \cup (D \times T)$. Here, vertices T are the tuple nodes and D are the derivation nodes.

Beyond graphs, provenance systems have used other formats to represent provenance, e.g., XML format and relational tables. However, most of them were domain specific.

The development of a generic domain-agnostic model to represent provenance has been an ongoing effort in the research community. Aiming at establishing the interoperability of systems, **the Open Provenance Model (OPM)** was proposed to exchange provenance information between systems [91]. The OPM models provenance as an annotated DAG. The OPM nodes can represent 1) an *artifact*, i.e., an immutable object, 2) a *process*, i.e., artifacts' actions and causalities, or 3) an *agent*, the initiator of the process. The nodes are linked using causal relationships, representing their dependencies (e.g., used, wasGeneratedBy, wasControledBy). Built on purely syntactical inference rules, OPM was criticized for its lack of completeness [75].

The **PROV** model appeared as a successor of the OPM model aiming to promote interoperability among a diverse variety of provenance types. The PROV model was standardized by the Provenance Working Group at the World Wide Web Consortium (W3C) [89]. While entity-relationships components of PROV are quite similar to those of the OPM, the PROV is capable of expressing a richer set of concepts. the PROV model elements include *entities*, i.e., physical, digital, conceptual objects, *activities*, and *agents* linked through various *relations*. This is shown in Figure 4. The PROV model was designed as a generic model applicable to a variety of data sources. To support this interoperability, PROV was accompanied with a set of models, among them PROV-DM, a provenance data model that captures provenance elements, and PROV-CONSTR, which describes constraints that provenance statements must satisfy. A concrete example that utilizes the PROV model is shown in Figure 5. This represents a student enrolment system within a university. There are three agents: Department, Student, and Instructor. A Student is associated with the Course entity through the Enrolment activity. The Course entity is generated by the Teach activity attributed to the Instructor.

The PROV model remains a widely adopted provenance model. At the same time, different models were subsequently proposed to support PROV applications in different areas. For instance, ProvONE extended PROV to support the DataOne scientific community, which is a large, federated data network for open, persistent, robust, and secure access to Earth observational data [33]. Prov-IoT [68] was geared specifically to Internet of things (IoT) environments and incorporated security metadata along with provenance data. While the model offered

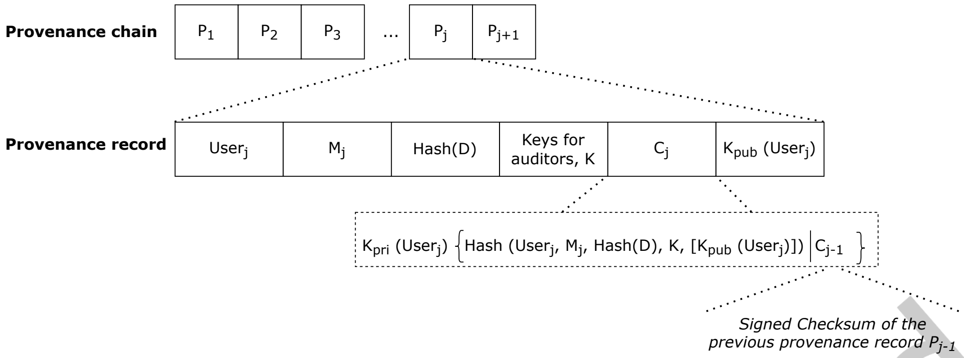

Provenance chain
P_1
P_2
P_3
...
P_j
P_{j+1}
Provenance record
$User_j$
M_j
Hash(D)
Keys for auditors, K
C_j
K_{pub} ($User_j$)
K_{pri} ($User_j$) {Hash ($User_j$, M_j, Hash(D), K, [K_{pub} ($User_j$)]) | C_{j-1} }
Signed Checksum of the previous provenance record P_{j-1}

Fig. 6. Provenance chain representation

little details on how and what security provisions could be included, it outlined the necessity to provide trust, integrity and authenticity of provenance.

Recently, Gao et al. [51] introduced big data provenance model (BDPM), which extends the PROV-DM model, in the context of tackling data transformation processes through various components in big data systems. BDPM allows to incorporate constraints that help ensure the construction of valid provenance while retaining a core PROV structure. While the model does not provide any security properties to secure provenance, it offers data security supervision strategies based on provenance graph analysis to help detect abnormal operations.

There are also numerous derivative models based on OPM and PROV such as OPM v1.01 [92], OPM v1.1 [90], D-OPM [39], PROV-Wf [38] and Wf4Ever [21].

Noting the limitations of PROV and OPM standards, several studies attempted to develop more comprehensive models applicable across various fields. A generic provenance model SimP [67] was designed to represent provenance information at different levels of granularity as requested by users through the Granularity Policy entity. Although the model was viewed as security-aware, it only allowed a controlled access to provenance information through the defined access control policy.

Overall, these models enable the provenance system to organize the provenance data more reliably and in turn implement more useful and convenient functionality (e.g., faster querying, authenticated data querying). Yet, the vast majority of them lack security provisions.

A number of custom data provenance models were proposed and geared toward specific contexts (e.g., documents [59, 61], IoT [99, 111]). Many of them were not mature, and consequently, did not receive wide acceptance. An exception to this was a **provenance chain** introduced by Hasan et al. [59, 61] for securing provenance information of documents has become a widely used model in secure provenance. The provenance chain organized as a time-ordered chain represents a history of document modifications, where each user's modifications were encompassed in a provenance record. Figure 6 gives an example of a provenance chain. A provenance record P_j incorporates information about a user $User_j$ that performed a (optionally encrypted) sequence of modification actions M_j on a document D, hash of the document $Hash(D)$, key information K that auditors can use to decrypt provenance record fields if encrypted, C_j checksum signed by a $User_j$ that stores previous provenance record P_{j-1}, and $User_j$ public key if used $K_{pub}(User_j)$. C_j checksum incorporates a hash of the record P_j and the previous C_{j-1} checksum. Although later studies extended the initial idea of the provenance chain, its main core characteristics remained pertinent.

Table 1 presents a summary of the major provenance models.

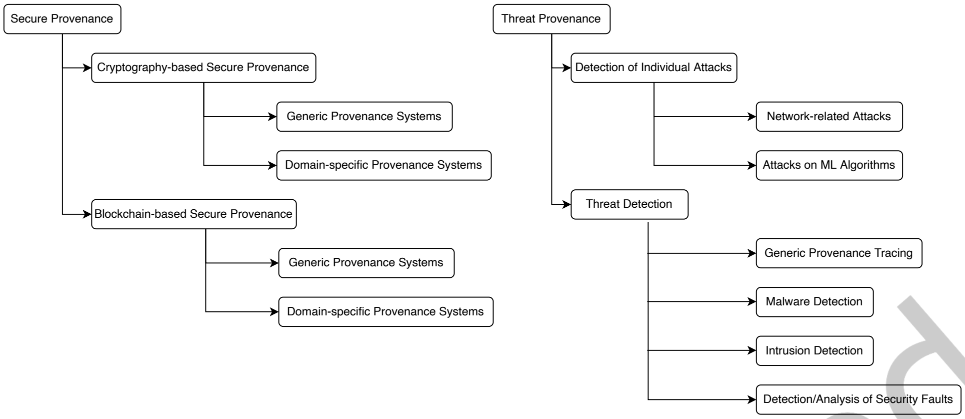

Secure Provenance
Cryptography-based Secure Provenance
Generic Provenance Systems
Domain-specific Provenance Systems
Blockchain-based Secure Provenance
Generic Provenance Systems
Domain-specific Provenance Systems
Threat Provenance
Detection of Individual Attacks
Network-related Attacks
Attacks on ML Algorithms
Threat Detection
Generic Provenance Tracing
Malware Detection
Intrusion Detection
Detection/Analysis of Security Faults

Fig. 7. Categorization of data provenance studies in security and privacy

3 PROVENANCE IN SECURITY

In this section, we present the concepts of secure provenance and introduce high level categories.

3.1 Provenance security properties

Historically, security properties of provenance were not emphasized. Collection and analysis of provenance was viewed in the context of three broad applications: ensuring reproducibility, understandability and quality of monitored data [62]. In other words, provenance was considered as a means for conveying the information about data. For example, the provenance for reproducibility aimed to support replication of the process used to produce the data. Provenance-based understandability emphasized on an explanation of how the results were obtained. Finally, provenance focused on the quality of the obtained data and its process aimed at assessing various quality dimensions, e.g., correctness, and performance.

With the development of security related provenance research, the inadequacy of these properties became apparent. Indeed, practical applications of data provenance call for trustworthy data that can facilitate further analysis. For example, Bertino et al. [22] noted that for privacy-aware analysis, it is not sufficient to simply ensure the confidentiality of data. Indeed, **provenance confidentiality** guards sensitive provenance information (e.g., location data), yet, entities (e.g., devices collecting location information) and users involved in the chain of data production are not protected from exposure. Thus, while confidentiality ensures protection of collected provenance data, **privacy** provides additional guarantees that sources of this provenance are not revealed to unauthorized entities.

In the context of security, data provenance properties have seen less agreement. Tan et al. [123] viewed confidentiality, integrity, authenticity, and reliable collection as four security requirements essential for reliable and trustworthy data provenance. Zafar et al. [142] broadened this list to include chronology, i.e., assurances of preserved chronological order of events, unforgeability of existing provenance records and their non-repudiation. Unforgeability as a means to attest to the ownership and history of data objects was emphasized by Lu et al. [82]. McDaniel et a. [88] posed tamper-proof and non-repudiation as the core principles of secure provenance. Among related studies, availability of provenance has been rarely included in a list of core security properties. One exception is a study by Hasan et al. [60] that viewed the efficiency of provenance mechanisms, the completeness of records, their integrity, availability, and confidentiality as properties of secure data provenance. **Availability** generally implies accessibility of data and is commonly assured through fault-tolerance mechanisms, e.g., data replication across multiple sites. For example, solutions that leverage decentralized ledger technologies such as blockchain, implicitly ensure availability. In the context of provenance, however, availability has been sometimes interpreted as a part of integrity, and therefore required integrity verification that for example, provenance data was not modified [60, 105] or selectively deleted [69, 135]. In this survey, we maintain a traditional definition of availability of provenance data.

Ultimately, provenance security properties should be examined through the lens of provenance application. For example, confidentiality of provenance may not be required for public code repositories, but can be crucial for scientific workflows. Similarly, availability is vital to provenance captured on mobile systems when it is used for location-based services.

Hence, despite different application domains, secure provenance hinges on the *integrity* of provenance data, including the *authenticity* and *non-repudiation* of the provenance data source pillars. In the computer security domain, integrity is commonly understood as an umbrella property for guarding against improper/unauthorized data and system modifications. In the case of provenance, integrity is generally narrowed down to data integrity, as the integrity of the system is challenging to ensure. **Integrity of provenance data** is based on the assumption that collected provenance has not been modified. This refers to individual records as well as the structure of their group. For example, integrity of provenance chain assumes that individual provenance records have not been tampered with, their order within the chain has not been modified, and the provenance chain has not been replaced.

There are several existing mechanisms to ensure these security properties. Cryptographic hashing is one of the most common approaches to ensure the integrity of data. This is the mechanism implemented in the provenance chain. While hashing provides guarantees against data tampering, it gives no assurances on the source of the data. Hence, a specific focus on authenticity and non-repudiation emphasizes the need for stronger guarantees.

In this context, authenticity requires a verification of provenance data sources (e.g., system components, IoT devices), which is commonly achieved through mechanisms such as digital signatures. Provenance non-repudiation takes a step further providing evidence so that the source cannot deny generating provenance data.

For example, provenance records in a provenance chain are hashed providing integrity guarantees, and signed with a user's private key ensuring authenticity of provenance. While non-repudiation is not a part of initial provenance chain design, it was often considered in the improved versions of a provenance chain. Note, confidentiality in the provenance chain is also treated as an optional property.

Table 2 lists examples of how the existing studies ensure the provenance security properties and the mechanisms they employ.

3.2 Categories

Modern provenance solutions in the area of security and privacy tend to focus on two broad objectives: (1) ensuring secure and/or privacy-preserving management of provenance data, which we refer to as *secure provenance*, and (2) leveraging provenance for security, i.e., threat and secure fault analysis, which we denote as *threat provenance*.

Secure provenance solutions address the existing security and privacy challenges of provenance users and provenance data collection and manipulation [36, 64, 79, 81, 104]. There is no uniform agreement on what secure provenance entails, e.g., not all provenance studies support confidentiality or availability as the essential requirements of provenance data, yet, all studies see integrity as a core characteristic. Hence, based on how the integrity of provenance data is ensured, the existing secure provenance studies can be broadly divided into two categories:

- *Cryptography-based secure provenance*: The early studies in the area primarily leveraged cryptographic concepts to provide guarantees for provenance data integrity and, in many cases, confidentiality.

- *Blockchain-based secure provenance*: With the rise of distributed ledger technologies such as blockchain, various blockchain platforms have become the main vehicle for ensuring security and privacy of provenance data. Once deployed on the chain, the blockchain records are immutable, i.e., cannot be modified or deleted from the chain, which in essence provides records' integrity verification in a trustless environment. Hence, the provenance core security requirement for integrity is naturally supported with the immutability of blockchain records. While blockhain solutions at their core leverage cryptography and thus are a subset of cryptography-based secure provenance, we feel that with the popularity of blockchain-based provenance, these studies merit a separate category.

Threat provenance systems, in some sense, are secure provenance systems aimed to prevent the mishandling of data in untrusted environments. When these security measures are not in place or fail, provenance has shown a significant value in the identification and analysis of malicious activities and threats [56, 133, 138, 139]. The existing threat provenance studies can be broadly divided into the following categories:

- *Detection of individual attacks*: The detection of malicious activity can be realized through the analysis of provenance data. However, not all attacks are visible in the available provenance data. Hence, a significant number of studies have been proposed to collect and analyze provenance necessary for the detection of particular types of attacks.
- *Threat detection*: The use of provenance data for threat detection is becoming increasingly common. Provenance provides a unique and rich source of information that enables accurate detection and tracing of a variety of threats, e.g., from intrusive activity to violations of privacy policies.

3.3 Attacks on provenance

Over the years, a number of attacks on provenance have been explored. These attacks may aim to disrupt the normal provenance tracking and collection mechanisms or compromise provenance records after they have been collected. Regardless of the time of their occurrence, the following types of attacks were discussed in the reviewed literature:

- *Forging records*, a malicious user or multiple colluding users may forge provenance records. Forged data may then be added between legitimate provenance records or appended at the end of the existing provenance records, e.g., the end of the provenance chain. The latter attacks are commonly referred to as *append attacks*. Forging and adding records might be significantly simplified in the presence of multiple consecutive adversaries that may simply introduce forged provenance between them.
- *Modifying records*, an external adversary or a dishonest user may change or corrupt a provenance record before its verification and storage.
- *Record shuffling*, many systems rely on the chronological order of events; hence, provenance order information might be critical for some applications, such as forensic analysis and auditing. The record shuffling attacks manipulate provenance information to rearrange the order in which provenance is recorded.
- *Dropping records/entire provenance chain*, a malicious user or multiple colluding users may selectively drop provenance records or an entire provenance information captured on a system.
- *Bribe attack* introduced by Tosh et al. [125] assumes adversaries can bribe users to invalidate some message records. For example, in a blockchain-based provenance, users might be incentivized to append their blocks to the attacker's chain, resulting in the main chain being abandoned.
- *Ownership attack* aims to repudiate provenance ownership. Ownership can be modified during the provenance generation process, or after the provenance data are collected and stored. A variation of this attack is denial of the ownership of the produced data.
- *Inference attack* compromises provenance data privacy allowing an adversary to infer sensitive information about sources and process of provenance collection.

In addition to these provenance attacks, provenance can be compromised through physical attacks on devices that participate in provenance collection, storage or analysis, for instance, adversarial sensors, in the case of provenance collection in IoT environments. Physical attacks, however are rarely considered, and the majority of the existing provenance methods assume the trustworthiness of physical devices and platforms.

Table 1. The major provenance representations

Name	Domain	Security Protection	Description
OPM [91]	Generic	-	A universal provenance model where artifacts, processes, and agents are linked using causal relationships, representing their dependency.
OPM v1.01 [92]	Generic	-	Improved version for original OPM.
OPM v1.1 [90]	Generic	-	Improved version for OPM v1.01.
D-OPM [39]	Scientific workflow	-	D-OPM enables both prospective and retrospective provenance information access and exchange for scientific workflows.
PROV [89]	Generic	-	Application-agnostic model that is similarly to OPM links entities, activities, and agents through their relations.
PROV-Wf [38]	Scientific workflow	-	Runtime provenance can be provided and queried during the execution by PROV-Wf.
Wf4Ever [21]	Scientific workflow	-	A novel approach to the preservation of scientific workflows through the application of research objects—aggregations of data and metadata that enrich the workflow specifications.
ProvONE [33]	DataOne scientific community	-	DataOne scientific community, which is a large, federated data network for open, persistent, robust, and secure access to Earth observational data is supported.
Prov-IoT [68]	IoT	Security metadata to give evidence of necessary security controls	Unified model for IoT provenance data.
BDPM [51]	Big Data	Data security supervision strategies	Generic provenance representation suitable for data with multiple organization layers.
Provenance Chain [59, 61]	Generic	Integrity, authenticity	The provenance chain organized as a time-ordered chain of document modifications, where each user's modifications are encompassed in a provenance record.
SimP [67]	Generic	Access control policies regulate access to sensitive provenance data	A multi-granular provenance model that supports graph and relational database representations.

'-' indicates that the feature is not supported

4 SECURE PROVENANCE SOLUTIONS

Over the past decade, there has been a significant amount of research on the management aspects of provenance, for instance, efficient querying [106], provenance-aware storage systems (PASS [95] and PASSv2 [94]), provenance management system for scientific workflows [87], cross-platform distributed data provenance SPADE [53], and cloud management [78]. However, until recently, there have been only limited efforts to ensure the security and privacy of provenance information. Table 3 summarizes the reviewed secure provenance studies.

4.1 Cryptography-based secure provenance solutions

The protections laid out by secure provenance systems are often dictated by the security objectives within a specific application domain. Hence, we broadly categorize the existing systems into *generic* and *domain-specific* systems.

4.1.1 Generic provenance systems. The vast majority of cryptography-based systems target the confidentiality and integrity of provenance data. Many of these schemes rely on digital signatures as assurance of users' identity. If each user that contributes (modifies or creates) to provenance records is associated with a cryptographic key, the key can be used to sign the corresponding records, effectively providing non-repudiation guarantees. However, these assurances are not sufficient to guarantee the integrity of the provenance record structure. Indeed, the provenance is not a set of isolated records. Depending on provenance organization, the relationships between records also require integrity support.

In an early study, Hasan et al. introduced the idea of the *provenance chain* for securing provenance information for documents [59, 61]. The history of a document modification is organized as a time-ordered chain, where each modification of a document is represented in a provenance record. Sensitive fields of provenance records were secured with a cryptographic hash and sealed with signature-based checksums to verify the records' integrity. As an alternative to the all-or-nothing approach that would generally require encrypting all provenance records, the provenance chains embedded flexible broadcast and threshold encryption to avoid situations when a single session key to encrypt all sensitive fields is needed. This approach did not require an access control model to manage auditors and users who may later need access to the records.

This scheme resembled an onion-like structure where each provenance record's signature enclosed the signature of the previous record and therefore information of the complete preceding chain and hence was later referred to the *Onion scheme*. The proposed Onion scheme was extended in a more holistic cross-platform secure provenance system, SPROV [60]. Despite its ability to protect internal records in the chain, the Onion scheme suffered from several limitations, including the inability to detect owner history forgery or *selective provenance record dropping*, i.e., the lack of reference to the next record allows adversary to drop selective number records at the end of the chain, and re-sign with his own signature.

Regardless, the idea of the provenance chain and the Onion scheme has been further leveraged in numerous studies [6, 64, 105, 116, 117, 119, 135, 143, 145]. For example, Zhang et al. applied it in the context of databases [145]. Noting the complex structure of data objects commonly stored in databases and the non-linear nature of provenance records resulting from various operations on these compound objects, the researchers developed an advanced scheme for provenance integrity verification of compound objects (as opposed to atomic objects considered by [61]).

Syalim et al. [119] extended the proposed Onion scheme to a DAG representation of the provenance model. By signing the nodes and the relationships between nodes in the provenance graph, the integrity of the provenance was easily verified by checking the signatures. As opposed to the Onion scheme, Syalim et al. leveraged a multi-level access control model to create a separation between compartments, i.e., nodes that belong to different security levels.

Wang et al. [135] proposed a public-key linked chain (PKLC), which solved the Onion scheme deficiencies by linking public keys of the users of the provenance records. The PKLC structure provided better protection against selective provenance dropping but required the knowledge of the next record in the chain. This requirement was later removed through the use of *aggregated signatures*, i.e., a single signature that aggregates signatures verifying individual provenance records, in the follow-up study by Ahmed et al. [7].

Rangwala et al. proposed a signature-based mutual agreement scheme that incorporated three signatures into a single provenance record [105]. Similar to the Onion approach, the mutual agreement scheme maintained the signature of the previous record. Like the PKLC, it included a signature of the next record, hence allowing verification between any pair of provenance records, yet at the expense of higher performance overhead than both the PKLC and Onion schemes. The memory overhead, however, is a common side effect of provenance systems as the amount of collected provenance with metadata can be significant, which consequently creates challenges for practical data analysis (see, e.g., [19]).

These early studies eventually led to more comprehensive solutions to data provenance, often referred to as *whole system provenance*. A vision of distributed trustworthy provenance architecture was outlined in the End-to-End Provenance System (EEPS) [88]. Based on the notion of host-level provenance, EEPS relied on trusted monitors to instrument provenance collection and validation. The validation process provided a provenance record that identifies not only the inputs, involved systems and applications leading to a data item, but also evidence of the identity and validity of the recording instruments.

A more mature whole-system provenance solution, Hi-Fi, was developed by Pohly et al. [102] in 2012. This was the first attempt to introduce a practical provenance collection framework across applications for security purposes. Built on Linux Security Modules, Hi-Fi collected provenance information through mediated access to kernel-level objects, which effectively gave a trusted view of the whole system including communication with other potentially proven systems. The integrity of the collected provenance data was protected by a write-once read-many (WORM) like storage systems [113]. In fact, this approach is one of the few that explicitly assumes that provenance collection mechanisms at lower levels of a system are not trustworthy.

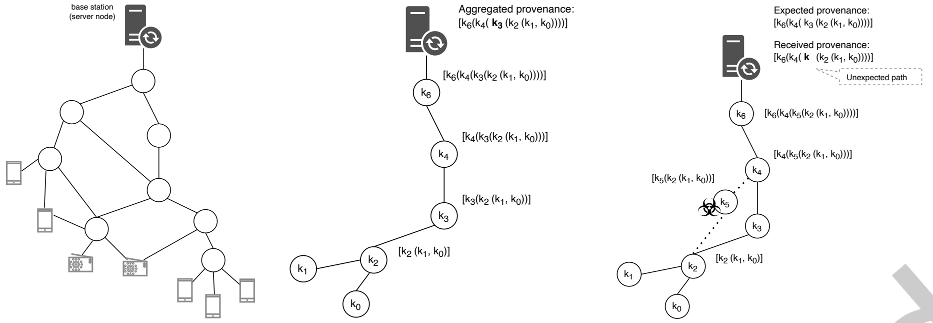
Bates et al. addressed this assumption in a framework called Linux provenance modules (LPM) [19]. LPM is a modular system that facilitates secure provenance collection at the kernel level. To provide strong provenance integrity guarantees, LPM leveraged a Trusted Platform Module (TPM) and the Linux Integrity Measurements Architecture (IMA) [107]. LPM created a trusted execution environment by monitoring and verifying operations on controlled objects (e.g., files, shared memory, sockets).

The architectural ideas that were implemented in LPM have also been seen in other whole-system provenance models (e.g., [95, 101, 120]). For instance, the CamFlow [101] provenance collection system, although it did not pursue security objectives, was leveraged by provenance applications to ensure the trusted collection of provenance data.

The other weakness that these earlier provenance systems exhibited is assuring security under relaxed threat models, often implicitly assuming honesty of users [18, 82]. This assumption significantly limits the practical application of these models. Indeed, malicious corruption of provenance records is challenging to avoid; hence, it has been taken more into consideration in follow-up studies.

4.1.2 Domain-specific provenance systems.

SECAP [143], a secure application provenance scheme, was introduced for cloud environments. It specifically addresses the presence of malicious users and assumes that a dishonest cloud provider can access and tamper with the provenance data. SECAP constructs provenance chains similar to the Onion scheme initially introduced by Hasan et al. [61]. The provenance chain and the corresponding proofs of provenance (i.e., that includes a signature of a cloud provider) are preserved in a Bloom filter structure, a space-efficient data structure based on hashing that allows to quickly check whether an item is present in the data structure or not. This approach essentially prevents an adversary from retrieving the original information.



(a) A network topology (b) An aggregated provenance graph (c) Provenance graph in a presence of malicious node

Fig. 8. An example of sensor or IoT network and the corresponding aggregated provenance graphs

SECAP is a tamper evident scheme that is realized through the use of cryptographic proofs calculated locally and published publicly (e.g., on an RSS feed) to prevent changes to the provenance after its calculation.

The issue of colluding malicious users was addressed in a series of studies by Ahmed et al. [5, 6, 69]. The majority of the solutions that are based on the idea of the provenance chain implicitly assume transitive trust and therefore are not able to handle multiple malicious users that may e.g., shuffle provenance records without being detected. To resolve this, Ahmed et al. [5, 6] extended the aggregated signatures introduced in their initial study [7] to prevent tampering with provenance records by multiple consecutive and non-consecutive users. Jamil et al. [69] proposed maintaining an authenticated Merkle tree in addition to a provenance chain to verify the integrity of provenance records. Although the scheme does not prevent tampering with records, it allows for the discovery of such activities. Similar to some of the previous works, these studies focused specifically on securing provenance collected within a single application (e.g., documents).

A few other secure provenance methods were proposed for specific purposes. Many of them focused on wireless sensor (WSN) and IoT networks. Collecting and transmitting provenance in these networks is typically associated with two challenges. Hostility of the deployment environment requires protection of information transmitted by the sensors. While a resource-constraint nature of sensor devices makes traditional encryption approaches not suitable for WSN and IoT networks. As a result, provenance schemes focused on compression and secure encoding strategies for provenance information. Most of the proposed schemes for wireless and IoT networks model provenance as a directed acyclic graph (DAG), where vertices correspond to sensor devices and edges represent a traversal path between them. The provenance is then collected at each node on the data flow path. To compress the amount of provenance data passed through network, nodes implement *aggregated provenance*, i.e., combine their own provenance with the provenance records received from previous nodes on the path. The provenance records are typically compressed and secured forming representation that in essence, resembles the provenance chain mechanism.

For example, Figure 8 shows a network topology and an example of aggregated provenance graph. Network data (e.g., packets) generated by the sensors k_0 and k_1 are aggregated with data from node k_2 and forwarded along the path to the base station (server node).

As opposed to the whole provenance solutions, the schemes for sensor networks limit the collected provenance to information describing a forwarding path of data since its generation, which commonly includes node identifiers. In the context of network hostility, this information is often sufficient to detect a presence of malicious nodes.

For example, presence of unexpected node provenance is visible at the base station when compared to the expected path (Figure 8c). However, more advanced detection requires additional information (see the overview in Section 5.1).

For example, to reduce the inevitably growing size of provenance information as packets travel through the network, Hussain et al. [64] used arithmetic encoding, a lossless data compression technique, to compress the collected provenance information. Since arithmetic encoding is dependant on the knowledge of nodes' occurrence probabilities, the approach requires a training phase to estimate occurrence probabilities of sensor nodes in a network. To prevent selective dropping of packets generated by benign nodes and to further bind nodes' identities to the generated provenance record, the scheme employed a distributed message digest based on the AM-FM sketch mechanism, a distributed aggregate computation to verify that the none of the captured results have been modified by adversarial aggregators [52].

DBNP [128] and CBP [141] schemes were similarly built on the idea of arithmetic encoding. DBNP leveraged Bayesian network to express the occurrence probabilities of edges in the packet path. This allowed to suppress redundant nodes, and reduce the size of the encoded provenance. CBP used layered clustering approach for incremental provenance encoding and consequently decoding. The primary benefit of such layered decoding is the rapid provenance trust evaluation, i.e., if a layer is not decoded correctly due to compromise, no further decoding is performed.

As opposed to Hussain et al. [64], DBNP and CBP focused strictly on provenance compression rate. Compression alone does not provide sufficient security guarantees. While provenance encoding to some extent masks transferred provenance information, an eavesdropping adversary similarly to the base station can obtain occurrence probabilities of nodes, hence compromising provenance. Similarly, the Probabilistic Provenance Flow (PPF) approach proposed by Alam et al. [65] offers efficiency rather than security. To constrain the size of transferred provenance data, PPF applies provenance probabilistically, i.e., each node embeds its identifier into the packet with a given probability, then, at the base station, the complete provenance path can be constructed using multiple partial paths analyzed in the past.

A few schemes offered some security guarantees alongside efficiency. Sultana et al. [117] and Shebaro [108] used in-packet Bloom filters (iBF) to encode provenance information. Each node in a data forwarding path encodes its id using an encryption key known to the base station and inserts the encrypted id in the iBF, which is then transmitted on the path to the base station. Since a Bloom filter has a fixed size, the structure size does not increase as a packet traverses the network collecting provenance.

Data provenance encoding scheme proposed by Wang et al. [129] encodes provenance using the node-based *packet path dictionary (PPD)*. Each node maintains records of path-related provenance information for all packets that passed through it in the PPD, reducing encoding and decoding process to lookup operations. This scheme further ensures integrity of provenance records by using distributed message digest mechanism to bind a packet and its provenance, the same mechanism used by Hussein et al. [64].

Similar to PPD, an index-based provenance compression algorithm (IBP) proposed by Liu et al. [81] encodes provenance record based on the path index maintained in a database at each node. Once encoded, only the compressed path index is transmitted between nodes on the path.

An alternative approach based on regulating access to provenance was proposed by Porkodi et al. [103] in the framework for IoT devices based on hybrid attribute based encryption (HABE). All users, devices and sensors in the IoT network are registered and authenticated, consequently allowing access control polices to be integrated within the framework to ensure restricted control to encrypted provenance data. With a focus on verification, the work leaves out details of provenance representation. Similar to [5, 6], the proposed scheme is also resistant to collusion attacks. A similar approach, although with more focus on the distributed nature of the IoT network, was considered by Siddiqui et al. [109].

Hasan et al. proposed a tamper-evident framework for ensuring the integrity and privacy of the location provenance records through endorsements of witnesses co-located within a mobile device's vicinity (e.g., wireless access points) [57, 58]. Although this prevents malicious users from forging their location records, the presence of necessary witnesses might not be feasible or appropriate in all scenarios. Hence, Wang et al. [136] relaxed this requirement and proposed a distributed spatial-temporal provenance proof architecture, STAMP.

Several studies have focused on securing provenance information of users' locations on mobile devices. The need for these solutions stems from numerous services that rely on users' location history. The digital proof of the user's presence in a given location is referred to as the *location proof*. Given the privacy-sensitive nature of location information, the ability to provide location proof to third parties while maintaining user privacy is critical.

An approach to secure provenance based on hardware technologies was initially conceptualized by Lyle et al. [83]. The proposed high-level design of the attestation-based provenance architecture leveraged TPM-supported attestation of executed code. Verification of provenance integrity based on TPM was also considered by Abbadi [2]. However, both frameworks remained theoretical, i.e., were not implemented and did not provide a formal security analysis.

Privacy-preserving provenance. While the vast majority of studies focus on security aspects of provenance, there are a few studies that offer *privacy-preserving provenance* solutions. The first attempt to explicitly address both security and privacy challenges of provenance management was offered by Bertino et al. [22]. The authors gave a high-level overview of challenges related to designing a secure privacy-preserving provenance management system.

As opposed to traditional systems that focus solely on the need to maintain privacy of individual data items, privacy-preserving provenance schemes require protection of their relationship and associated provenance. In this context, necessity to preserve privacy of collected provenance applies to the identity of processes and users involved in collection of such provenance. For example, a platform collecting online inquiries of financial loans should incorporate mechanisms to protect privacy of individual users sending the inquiries.

Protection of provenance privacy has been approached mostly from two perspectives: using cryptographic techniques (e.g., encryption) and access control mechanisms.

Cryptographic techniques: Lu et al. [82] was one of the few studies that explicitly stated the need for privacy preservation in any secure provenance scheme. Their proposed provenance approach incorporated a *conditional privacy* that required the real identity of users generating provenance to be disclosed only to a trusted authority. This conditional privacy was ensured through pairing-based cryptography. However, the approach did not consider provenance integrity protection. The latter was later addressed in the privacy-preserving data provenance scheme (PDP) proposed by Alharbi et al.[9]. PDP leveraged trusted servers to authenticate users and generate provenance data. All user requests were verified by the trusted servers, and the corresponding provenance information was signed and appended to the provenance chain (as in the Onion scheme [61]).

Sanchez et al. [32] investigated the privacy-preserving provenance for the IoT. The information that IoT sensors generate, the corresponding derivation history of data, and the owner of the data were protected with cryptographic pseudonyms that masked user credentials. The provenance data were signed using a non-interactive zero-knowledge proof (NI-ZKP) and allowed interactive de-anonymization.

Access control mechanisms: Access control is one of the main mechanisms for controlling access to sensitive provenance information. Several early studies envisioned access control as the vehicle for providing security and privacy in provenance systems [27, 96]. In their secure privacy-preserving provenance management system, Bertino et al. [22] also viewed access control as a main pillar for protecting provenance privacy. Noting the deficiencies of existing access control approaches, Bertino et al. posed the lack of provenance access control models, languages, and enforcement mechanisms for securing provenance as the main challenges in designing privacy-preserving provenance system. Some these challenges were later addressed in the follow-up studies that introduced access control-based provenance model, SimP [67], and the ProFact framework for evaluation of access control policy quality [3, 23].

Another approach to limiting access to sensitive information is *data sanitization* [24]. Sanitization approaches mask sensitive provenance information, thus preventing inference attacks. For example, when requested, sensitive provenance can be redacted, replaced or generalized concealing private information from those who do not have access to it.

The challenge in data sanitization is defining practical mechanisms for identifying and masking provenance information requiring sanitization. Several provenance sanitization mechanisms were considered [41–44, 137]. For example, ProPub [44] leveraged logic rules to derive a sanitized version of provenance graph based on user requests and the defined provenance policies. Since user requests can invalidate provenance policies, ProPub follows a set of logic rules to honour user requests that conforms to the provenance policies.

Many of the proposed methods focused on the privacy-preserving provenance of workflows often blurring the boundaries between data and workflow provenance [41–43]. For example, pointing to the limitations of the data privacy concept for workflow provenance, Davidson et al. [43] introduced *module privacy* and *structural privacy*, which refer to the privacy of internal modules that generate provenance and dependencies between them, which can be viewed in the data provenance realm, as *how* and *where* provenance. The authors [42] further explored provenance protection for module privacy posing what they called the *SecureView problem*, i.e., "What is the minimum cost of intermediate data that can be hidden while guaranteeing that individual modules are ϵ-private for some value of ϵ". The proposed solution developed a framework to provide a partial view of the module(s) given the user permissions allowed by an access control model. While these privacy-aware models do not offer a detailed design of the system, the provenance confidentiality is assumed to be protected. As an extension of this work, Wu et al. proposed GPPub, a privacy-preserving provenance method [137] generalizing the constraints of Davidson et al.'s [42, 43] model.

4.2 Blockchain-based secure provenance solutions

Cryptography-based solutions have evolved from a centralized architecture with a trusted authority to a distributed design. The latter was challenged by the presence of adversaries. Hence, the appearance of blockchain technologies offered methods that were logical and resilient to tampering platforms to support the integrity requirements of secure provenance schemes.

4.2.1 Generic provenance systems. ProvChain, a blockchain-based data provenance system was proposed by Liang et al. [79]. ProvChain maintains provenance records in a Merkle tree structure [1]. A Merkle tree is a tree of hashes that is used to authenticate a list of items. In ProvChain, provenance data are hashed and represented as a leaf of the Merkle tree. A non-leaf node's hash is calculated from the hashes of its children. The root of the Merkle tree then represents all hashes within the tree and can be used to verify provenance data items. A set of blockchain transactions form a block and after external verification (e.g., by an auditor) can be included in a chain. Hence, data provenance records are published globally on a blockchain, and any adversarial modification of a provenance data record after its verification requires an adversary to modify the transaction and the corresponding block on the chain. User privacy is preserved by hashing a user ID associated with a data provenance record.

Similar to ProvChain's approach, the LineageChain provenance system [106] stores provenance in a Merkle tree data structure. LineageChain is one of the few systems that offers flexible provenance capture and access mechanisms through smart contracts. A smart contract can define the exact provenance information to retain; upon execution, the message is automatically preserved on the blockchain. Similarly, smart contracts provide a way to access provenance information at runtime. However, access to historical information through contracts is limited and has to be explicitly tracked, which is a common limitation across the existing provenance schemes.

The SmartProvenance system proposed by Ramachandran et al. [104] addressed ProvChain limitations. As opposed to ProvChain, it implemented automated verification to prevent potential collusion between the external party and the users. Provenance records were represented using the OPM, i.e., the state of data before and after the change was recorded with the corresponding user information. SmartProvenance was implemented as a two-part system with an on-chain module and an off-chain module. The on-chain module represented an Ethereum blockchain that included smart contracts regulating users access control and provenance trails. The Ethereum blockchain platform is built on a variation of the Merkle tree structure called the Merkle Patricia tree[29]. The off-chain module was mainly used for the automated verification. The users' privacy was protected by design, i.e., SmartProvenance leveraged the Ethereum blockchain platform, which hides user's identity through the use of a public key. SmartProvenance was one of the few that explicitly provided availability of provenance data through the use of node replication.

Zhang et al. [147] combined the benefits of ProvChain and SmartProvenance in a secure data provenance scheme for cloud systems called ESP. Similar to SmartProvenance, ESP leveraged the Ethereum blockchain to integrate each provenance record into a transaction on the blockchain. ESP, however, outsourced provenance verification to an external trusted party, i.e., an authenticated server, that was responsible for automated user authorization, in the same fashion as ProvChain. The use of authenticated servers also allowed hidding user identities through digital pseudonyms, which guarantees the user's conditional privacy.

4.2.2 Domain-specific provenance systems. The development of later systems has mostly leveraged features of SmartProvenance and ProvChain and primarily focused on various domains. For example, Zeng et al. [144] proposed a blockchain-based compression free data provenance scheme (BCP) for wireless sensor networks (WSNs). Distributed and usually unattended deployment of wireless network sensors often requires collection and transmission of data to remote locations for analysis. Hence, tracking the provenance of the data origin and its traversal path is critical for establishing the trustworthiness of the data. Therefore, the provenance information is distributively stored on the nodes along the packet path. As opposed to the previous approaches, which mostly use a provenance chain, BCP stores the provenance records in the *provenance tables.* These tables are encoded and stored permanently on the Ethereum blockchain. Similar to earlier schemes, BCP implicitly assumes that sensor nodes are trusted, and even though the provenance data are stored on the blockchain in encrypted form, there is no verification of whether these data have been tampered with.

ProvNet blockchain [36] similarly focuses on tracking the provenance of distributed data. Designed for tracking access and sharing of data, ProvNet stores provenance records in a networked blockchain, a variant of the permissioned blockchain, which allows authentication of all participating users and hence assumes the presence of dishonest users. However, the verifiers that validate and grant user requests for data sharing and consequently store provenance on the blockchain are assumed to be trusted entities.

Recently, Mothukuri et al. proposed BlockHDFS, a blockchain-based secure provenance system for the Hadoop distributed file system (HDFS) [93]. HDFS, one of the most widely used file systems that deal with big data applications, is mainly used for batch processing of data. It is known for its high throughput data accessibility with low latency, and thus it is very popular. Since HDFS is designed for large volumes of data, robust security to facilitate file sharing in Hadoop is necessary. BlockHDFS stores provenance information for files in the blockchain, hence creating an immutable, tamper-proof set of logs for HDFS. Thus, even when HDFS storage is compromised, the corresponding provenance is preserved intact.

Several approaches were developed specifically for secure data provenance in the IoT domain [70, 110]. For example, Porkodi et al. introduced an approach based on HABE for IoT devices [103]. All users, devices and sensors in the IoT network are registered and authenticated, consequently allowing access control policies to be integrated within the framework to ensure restricted control to encrypted provenance data that are kept on a blockchain. Similar to [5, 6], the proposed scheme is also resistant to collusion attacks.

Griggs et al. [55] introduced a secure blockchain-based patient monitoring system. Tosh et al. [125] proposed BlockCloud, a secure data provenance in the cloud. To the best of our knowledge, the system was not implemented and remained an abstract model.

The AMP (authentication of media via provenance) system was introduced to preserve provenance information of authentic media [46]. AMP stores digitally signed media data with the corresponding provenance, including a full history of media publishing operations in a blockchain called the media provenance ledger. This ensures the authenticity of media items, their integrity, and auditability.

Only a few systems focused on privacy-aware blockchain-based provenance. Among them is Trac^{2}Chain, a provenance graph storage platform [124]. Provenance data naturally form a graph; however, the vast majority of systems record the provenance as a chain and store it linearly on the blockchain. Once stored on the blockchain, the data become public, threatening the privacy of users participating in provenance generation. Trac^{2}Chain addresses both concerns by protecting the data flow and dependencies, i.e., linkages between provenance graph nodes, against unauthorized users. The reconstruction of the provenance graph from blockchain transactions is based on an access control mechanism that restricts the capabilities of non-provenance participants, i.e., anyone outside the provenance system (including the blockchain peers).

The overview of blockchain-based data provenance systems clearly shows that blockchain undoubtedly provides significant benefits for ensuring data provenance integrity and confidentiality. However, the existing systems tend to offer niche advantages, often trading efficiency and various security and privacy guarantees (e.g., the presence of malicious users, dishonest blockchain peers and cloud owners).

Another major challenge is limited access to historical data across chains. Typically, historical context is available only to blockchain nodes that actively manage their history and cannot be used in a consensus protocol that requires all participating nodes to share the same data. Another challenge in this context is the lack of tampered evidence support for historical data. While the blockchain's tamper-resistant nature guarantees that the ledger is immutable, this cannot be ascertained for any data that is downloaded from a blockchain and maintained offline. For instance, a full blockchain archive node typically records all historical transactions data off-line in an unauthenticated data structure. This presents significant challenges for applications that are under regulatory pressure to ensure verifiable historical context.

5 THREAT PROVENANCE

Secure provenance schemes implement security measures to ensure secure and privacy-preserving collection and handling of provenance data. Threat provenance studies leverage provenance information to enhance system security. A few of the reviewed studies, however, combine the benefits of both approaches. The summary of the reviewed studies is given in Table 4.

5.1 Detection of individual attacks

5.1.1 Network-related attacks. Provenance technology has been widely used in database systems, distributed systems, and cloud networks. With the rapid adoption of IoT networks, several provenance studies have focused on the security and reliability of data transferred by IoT sensors. IoT networks and sensor networks in general are often deployed in untrusted environments where devices and the transmission of data are susceptible to attacks. The solutions typically involve path provenance information on the packets traversing networks; hence, this type of provenance is often referred to as *network data provenance*, and therefore, network-related attacks include packet dropping, data tampering and packet replay attacks.

One of the earlier studies by Sultana et al. [116] looked at a provenance-based mechanism for identification of malicious packet dropping by adversaries. This attack, also known as *selective forwarding attack*, involves a malicious node dropping some packets and selectively forwarding the remaining traffic to remain undetected.

The challenge in this context is the transient nature of sensor networks that can cause packets to be dropped due to various benign reasons, e.g., unavailability of nodes, communication failure, physical damage, etc. Analysis of packet delays can be indicative of the potential presence of an adversary. This, however, is not sufficient, as some nodes may aggregate sensor information. The detection scheme proposed by Sultana et al. [116] utilizes the data packets and inter-packet delays to encode provenance information. Manipulating inter-packet delays allows to embed provenance data in a manner resembling watermarking. As a result, manipulated inter-packet delays form a distinct distribution different from non-manipulated delays. Hence, an analysis of an estimated distribution allows to detect the malicious packet loss. Localization of adversarial nodes requires decoding of provenance information sent to the base station, which can then reconstruct and verify the provenance information. Similar to many other studies, provenance representation in this scheme resembles provenance chain, i.e., each node includes information of the last packet it received through this path.

The follow-up work by the same authors extended the secure provenance encoding scheme to relax some assumptions and offer an efficient handling of provenance [118]. However, both studies were evaluated in a simulated setting; hence, no analysis of the schemes' behavior in a real deployment environment was performed.

Alternatively, schemes proposed by Wang et al. [134], Lim [80], and Cho [37] constructed trust models to evaluate whether nodes transmitting provenance can be trusted. Although none of the network attacks are explicitly addressed, the schemes offer a way to assess credibility of network.

Suhail et al. [115] proposed a provenance-based packet path tracing (PPPT) scheme. The scheme is built on the RPL routing protocol for IoT networks that creates a destination-oriented DAG. The root node maintains the network topology information and is therefore aware of all nodes that generate data provenance and the potential path the data packets may take. The provenance information is maintained at the node level (to detect compromised nodes that are dropping packets) and at the system level (to have a complete view of the network and determine packet drops due to benign network failures). Individual nodes transmit the encoded data provenance information, which is decoded at the root node that can verify the data and determine the presence of malicious nodes. Compared to [118], apart from packet drop attacks, the PPPT system can detect packet replay attacks.

The enhanced index-based provenance compression algorithm (IBP) [81] can also detect replay attacks along with packet drop attacks. For each data packet, a node generates a sequence number that incorporates an encrypted packet counter information. The base station can decrypt the packet counter and compare it with the expected value to determine whether a packet was dropped or replayed on the path.

Numerous studies have explored protocol-based protections of data provenance in IoT, sensor, and wireless networks. The data provenance transmitted over these networks often includes location, data-device associations, and other sensitive contextual information that should be protected, yet the sensors are resource-constrained to use traditional cryptographic solutions. Research has focused on lightweight approaches to data provenance protection. Prior work in this area revolves around the protection of provenance largely from rogue sensors and man-in-the-middle (MiTM) attacks on communication channels.

Many of these studies leverage *wireless fingerprints* (also known as link fingerprints), unique characteristics of wireless channels (e.g., radio signal strength) that allow fingerprinting of the link between two nodes and associating it with the data that these nodes exchange. These fingerprints can be used to sign the provenance data or authenticate device links. A high-level idea of encoding paths between nodes based on Bloom filters was proposed by Shebaro et al. [108]. The scheme relies on trusted infrastructure and hence can identify malicious nodes, but does not protect against man-in-the-middle attacks. Fingerprints based on received signal strength indicator (RSSI) values were employed by Ali et al. [10]. The fingerprints are further encrypted and combined with a hash of transmitted data and digitally signed. The approach guards against MiTM attacks but suffers from high communication costs. Kamal et al. [71] leveraged this idea for advanced metering infrastructure.

Physical unclonable functions (PUFs) for wireless link fingerprint generation were explored by [11, 72]. PUFs offer a hardware-based challenge-response mechanism that produces a unique response for a given challenge.

PUFs are unique per device. This is a favorable quality since resource-constrained devices, e.g., IoT sensors, do not require storing any secret information (i.e., cryptographic keys).

The benefits of using PUFs for data provenance integrity verification were offered by Kanuparthi et al. [72]. A fully developed protocol for the protection of data provenance in IoT networks using PUFs was proposed by Aman et al. [11]. PUFs allow for the authentication of IoT devices that produce sensitive data without revealing identities, thereby preserving the anonymity of devices. A third-party trusted verifier can check the identities of the devices.

5.1.2 Attacks on ML algorithms. Since machine learning (ML) plays an important role in a wide range of applications, adversaries have an incentive to manipulate data or machine learning models (e.g., parameters, structure) to alter the outcome. These attacks are known as *poisoning attacks*. Handling poisoning attacks through tracking data provenance has been considered by a few studies.

Baracaldo et al. [17] introduced a provenance-based approach for detecting and filtering poisonous data collected to train an arbitrary supervised machine learning model. The approach segments all collected untrusted/partially trusted data into groups and evaluates them using trusted and immutable provenance records that contain data origin and lineage. A follow-up study extended this detection approach to IoT environments and provided a security analysis of the proposed protection [16].

As opposed to [16, 17], which focused on data poisoning, Stokes et al. [114] investigated model poisoning attacks. The proposed provenance system called VAMP aimed to prevent poisoning attacks on ML-based systems. VAMP used cryptographic authentication and provenance to protect both the data and the ML model. To the best of our knowledge, VAMP has not been implemented.

5.2 Threat Detection

The detection of anomalous events has been at the center of research for decades. One of its challenges is the accurate and timely differentiation between malicious activity and benign behaviour. In this context, data provenance can play a critical role providing a massive amount of information for analysis. Equipped with provenance information, further analysis can reveal suspicious dependencies and causalities.

A common approach taken by studies focused on threat detection is *provenance tracing*. The value of provenance tracing is in its ability to analyze input and analyze causalities across multiple sources (e.g., executed binary files, downloaded files, system calls). In this respect, provenance tracing has characteristics of whole-system provenance systems. However, as opposed to whole-system provenance systems, provenance tracing makes no provisions to secure collected provenance. Yet, similar to whole-system provenance mechanisms, provenance tracing relies on instrumentation to collect provenance of system-level events at different granularity levels. The core of provenance tracing is the ability to replay execution of events, which allows to reconstruct the flow of attack and to understand how it affects the system. For these purposes, most provenance tracing systems, in addition to events, collect causality dependencies related to the monitored instances (e.g., sensitive syscalls), which consequently incur a significant run-time and storage overhead. To be practical, one of the goals for provenance tracing systems is to balance the overhead and granularity of collected provenance information.

From application perspective, provenance tracing can be leveraged to reconstruct the specific instance of attack flow or to perform more general analysis to detect anomalous (e.g., rarely occurring) system behaviour. Depending on their application focus, we broadly categorized the provenance studies for threat detection into the following categories: generic provenance tracing, malware detection, intrusion detection, and analysis/detection of security faults.

5.2.1 Generic provenance tracing. Although typically applied in forensics for the analysis of security incidents, provenance tracing systems are not necessarily associated with anomaly detection mechanisms. The primary goal of a provenance tracing system is to facilitate attack investigation by collecting all causalities necessary to accurately disclose the root cause of the problem, trace the flow of the incident and assess attack damages. Thus, granularity of provenance tracing has a direct effect on quality of the following analysis.

The state-of-the art provenance tracing systems collect provenance at the unit level, i.e., semantically autonomous execution segments [77]. For example, the BEEP provenance tracing approach leverages the selective unit-level instrumentation of binaries [77]. BEEP partitions processes into autonomous units to determine causality relationships. Based on causality analysis, binaries are selectively instrumented to capture runtime provenance information at necessary program points. Attack investigation can be then performed by analyzing the causality between a root cause of an incident and its symptoms using backward and forward analysis. ProTracer improves BEEP and implements a dynamic coarse-grained provenance tainting, i.e., provenance propagation at the system call level [84].

More fine grained unit-based provenance tracing at the library and system call level is performed by LProv [131]. Partially built on ProTracer, LProv traces system calls at the kernel level and derives the corresponding path from the library perspective at the user level. This give LProv a more granular view of causality relationships compared to BEEP or ProTracer. Similar BEEP or ProTracer, LProv models the execution history of events, their dependencies and interactions as a provenance graph.

Although all three approaches view provenance tracing in a context of security incident analysis and detection, all make an explicit and difficult to achieve in practice assumption on trustworthiness of kernel and user-space daemons, and integrity of tracing components.

5.2.2 Malware detection. AMICO, the system developed by Vadrevu et al. [127] relies on *download provenance* to measure and detect malware downloads in live network traffic. In contrast to static blacklists, AMICO can dynamically and accurately detect malware samples based on the download behaviour of network users. Download provenance includes provenance characteristics of *who* downloaded files and *where* these downloads came from. Leveraging this download history, AMICO learns malware provenance models and detects malicious files through provenance classification.

Unlike [127], the majority of provenance schemes leverage the existing operating systems functionality. Upchurch et al. [126] introduced the Malware Provenance project. The Provenance project collects block-level code pieces reused among malware variants and calculates the *malware provenance signature*, a malware aggregated signature that effectively accounts for multiple malware family samples. In practice, however, the detection of code reuse is not sufficient to detect malware, e.g., in cases when no or an insufficient number of previous samples are available.

A more generic system for the detection of malware was proposed by Sze et al. [120]. SPIF, a secure provenance-based integrity fortification system, leverages the existing Windows mechanisms to protect system integrity from unknown malware. SPIF utilized sandboxing of untrusted code with discretionary access control (DAC) policies to limit information flow and consequently possible system modifications.

Similarly, a generic system was proposed by Wang et al. [132]. ProvDetector is a provenance-based system that aims to detect stealthy malware that hides the identity of the malware by impersonating known trusted benign processes. ProvDetector captures each process provenance at the kernel level and analyzes it to determine deviations from benign process provenance.

A kernel-level instrumentation is also adopted by HProve, the hypervisor-level provenance tracing system proposed by Wang et al. [130]. Designed for kernel malware, HProve allows replaying of the attack lineage to acquire provenance data. The approach instruments the kernel during the reply to acquire the execution traces. The backtracking technique is then applied to find function calls and reconstruct the object manipulation chain.

5.2.3 Intrusion Detection. Provenance information has been successfully employed more broadly for the detection of malicious activity irrespective of malicious software presence. The vast majority of provenance-based intrusion

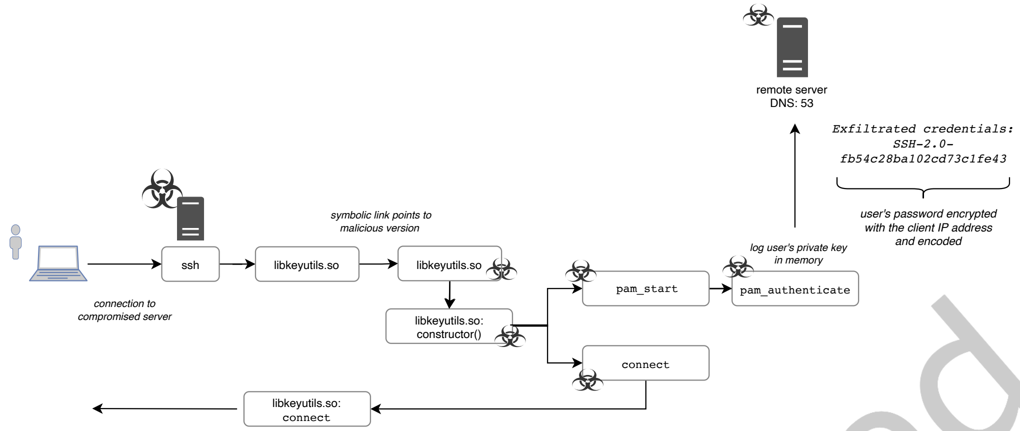


Fig. 9. Provenance graphs generated for detection of a Linux Ebury attack in a presence of a compromised SSH server

detection approaches use anomaly detection models, i.e., rely on models of benign behavior to detect anomalies based on deviations from the expected benign behavior.

For example, Figure 9 shows an example provenance graph illustrating Linux Ebury attack [47]. An attack replaces `libkeyutils.so`, a standard Linux library, usually called by ssh service, with a malicious version containing a backdoor `constractor()` that is tasked with exfiltrating hosts' passwords to a remote server. When a compromised library is loaded, it hooks standard functions, among which are `pam_start` and `pam_authenticate` that copy a user's password.

On the surface, this attack is difficult to detect due to its seemingly innocuous behavior. Without additional analysis, an security analyst may not be able to differentiate between a legitimate DNS request and a suspicious connection to an attacker controlled server over UDP protocol on port 53 that an attacker is using to avoid being blocked. Since the new method `constractor()` is not noticeable, it is challenging to understand the root cause of the problem. Provenance graph in this scenario gives a clear context allowing to see anomalous behavior and establish the causality with a suspicious connection through `libkeyutils.so` library.

PIDAS [139] and Pagoda [138] are examples of such provenance-based anomaly detection systems. PIDAS, a provenance-aware intrusion detection and analysis system, tracks file-level provenance captured with PASS [95] provenance collection system that records the lineage of various objects in a system (PIDAS focuses on files, processes, and sockets). The collected provenance is converted into a causality-based provenance graph and analyzed to reveal anomalous paths that deviate from normal system behavior. Pagoda [138] extends this approach by considering the anomaly degree of both a single path and the whole system provenance graph.

P-Gaussian, a provenance-based Gaussian distribution detection scheme, is a further extension of both schemes [140]. P-Gaussian leverages PASSv2 [94] to collect system calls and dependency relationships among objects (e.g., files, processes, and sockets), and to generate a provenance graph. To detect intrusive behavior, P-Gaussian calculates the similarity between paths known to correspond to a legitimate execution. Assuming that intrusive behavior is unstable, an intrusion path would have no exact matches with legitimate sequences.

threaTrace [133] is another provenance-based system capable of detecting intrusive behavior without prior knowledge of attack patterns. Similar to other approaches, threaTrace captures system-level provenance information in a provenance graph.

All these systems can be applied both for real-time detection and forensics analysis of intrusions.

Alongside intrusion detection, the detection of advanced persistent threats (APTs) has attracted a lot of research attention. Although APTs can be regarded as a type of intrusion behaviour, APTs generally exhibit long-term behaviour. Hence, some provenance-based systems specifically focus on the detection of APTs.

UNICORN, developed by Han et al. [56], is a real-time anomaly-based APT detection system. It constructs provenance graphs that expose causality relationships among system objects, considering the entirety of the graph by efficiently summarizing it as it streams into its analytic pipeline. UNICORN shares assumptions, architecture, provenance model, and limitations with other similar systems. ANUBIS similarly employs a provenance graph to detect APTs using machine learning classification [12]. TRACE [66] is an enterprise-wide provenance tracking system for APT detection, which leverages a static analysis technique for unit-based instrumentation, distributed causality tracking, and graph query analytics features.

Several provenance-based systems were proposed for attack detection and reconstruction specifically in the Android platform (Scippa [13], Quire [45]).

5.2.4 *Detection/analysis of security faults.*

Provenance has been broadly applied for the analysis and detection of various security concerns, e.g., data exfiltration [48, 49], security of industry control systems [8, 97], privacy policy violations [14, 15], and root cause analysis of security incidents [121].

PANDDE [49] was developed to detect data exfiltration attempts by inside users based on the analysis of provenance information. Provenance collected at the kernel level encompasses database user actions primarily related to write events. To detect anomalous activity, collected data are compared against the established profiles of users. A-PANDDE improves this approach, allowing the detection of advanced exfiltration attempts [48].

The ProvTalk [121] provenance analysis system guides the investigation of root causes analysis of security incidents in multi-tenant environments such as network function virtualisation. As opposed to the existing systems, ProvTalk captures provenance between different abstraction levels (e.g., virtual resources in the cloud and services on a host).

A provenance system proposed by Farooq et al. [8] focused on the detection of safety and security faults in programmable logic controllers (PLCs). The PLC system is used primarily for managing industrial processes, e.g., smart building management, power generation, water and wastewater management, and traffic control systems. The PLC typically collects inputs from a distributed set of sensors; hence, the provenance in this case is collected from execution traces and information received from sensors. The proposed PLC-PROV system detects violations in the safety and security policies of the PLC system by comparing them against the collected provenance. PLC-PROV has not been implemented and remains theoretical. A similar approach was taken by PROV-CPS, a trace-based data provenance system for cyber-physical systems [97].

Detection of privacy violations through provenance analysis were explored by Baeth et al. [14, 15]. User privacy policy modelled as custom rules can be compared against *social provenance data*, i.e., contextual information that describes the lifecycle of the social networking data, to detect privacy policy infringements.

The diversity of studies in this category emphasizes the value data provenance brings to the field.

6 THE CHALLENGES AND OPEN PROBLEMS OF PROVENANCE RESEARCH

Research on provenance in security has been rapidly evolving over the past decade. Several problems plaguing data provenance in security and privacy have been outlined. To resolve some of these problems, the research community has taken advantage of the recent advances in blockchain technology, machine learning, and information retrieval fields. Tables 3 and 4 summarize the reviewed provenance studies. Our review suggests several observations:

- *Trust assumptions.* Secure provenance is essential for trustworthy analysis. However, modern computational systems are complex and rarely free of vulnerabilities; hence, their security cannot be guaranteed. The existing provenance systems generally assume trustworthiness of provenance collection and storage mechanisms (e.g., [117, 144]) and often delegate trust to the trustworthy third party for provenance

Table 2. Provenance studies categorized by security properties

Provenance security properties	Security mechanisms	Research studies
Integrity	Hash	SPROV [60],[2, 7, 18, 32, 57, 61, 83, 88, 109],CamFlow [101], OTIT [73], PKLC [135], LPM [19], WORAL [58], SECAP [143], Mutual Agreement [105], BlockHDFS [93], AMP [46], Trac^{2}Chain [124], HABE [103]
	Digital signature	STAMP [136],[119]
	Blockchain-based	ProvChain [79],[55, 110], SmartProvenance [104], ESP [147], BlockPro [70], BlockCloud [125], LineageChain [106], [6], ProvNet [36]
	Other (e.g., MAC, checksums, AM-FM proof sketch [52])	Hi-Fi [102] , [64, 129]
Non-repudiation	Digital signature	SPROV [60], [6, 7, 18, 32, 57, 61, 88, 109, 119, 145], PKLC [135], STAMP [136], LPM [19], Progger [74], WORAL [58], SECAP [143], Mutual Agreement [105], ProvChain [79], ESP [147], BlockCloud [125], AMP [46]
Authenticity	Digital signature	SPROV [60],[7, 57, 61, 88, 119, 145], PKLC [135], STAMP [136], LPM [19], WORAL [58], SECAP [143], Mutual Agreement [105], ProvChain [79], ESP [147]
	Other (e.g., PUF)	BlockPro [70], [2, 6]
Confidentiality	Cryptographic encryption	SPROV [60], [6, 7, 58, 61, 82, 109, 119], PKLC [135], SECAP [143], Mutual Agreement [105], ProvChain [79], BCP [144]
	Access control-based	HABE [103], [18, 88, 119]
	Other (e.g., encoding)	[64]
Privacy	Cryptographic encryption	STAMP [136], [2, 32, 57, 82], WORAL [58], ProvChain [79], SmartProvenance [104], ESP [147]
	Access control-based	Trac^{2}Chain [124]
	Other	OTIT [73], GPPub [137], [41–43]
Availability	Replication/Decentralization	SmartProvenance [104], ProvChain [79], [110, 111], BlockHDFS [93], AMP [46]

verification [9, 32, 135, 136, 143]. These assumptions require strong assertions that these mechanisms and the overall environment are not compromised. This is unfeasible in practice, as a system can rarely provide full security protection.

One practical direction of research in this context is employing tamper-evident mechanisms that can demonstrate with high reliability that data have not been changed improperly. Tamper-resistant or tamper-proof mechanisms, i.e., structures that provide nearly complete protection from data tampering (e.g., blockchain technology), can further enhance integrity guarantees.

- *Resource-demanding provenance analysis.* The rate of provenance data creation is proportional to the number of objects monitored and tracked within a system. This may be exacerbated by the collected contextual information (e.g., dependencies, system characteristics). This massive amount of information leads to unavoidable complexity in analysis (e.g., long querying time). Hence, any provenance solution faces a trade-off between the completeness of the collected provenance information and practical computational constraints. In some domains, the lack of complete provenance undermines the capabilities of the systems. For instance, in threat provenance, compromises of provenance information collection are likely to make the detection of stealthy attacks more challenging.
- *Limited scope.* Initially, the main research focus was on ensuring some aspects of security within specific application domains. However, as the field matured, the scope of studies did not change or broaden, i.e., only a limited number of studies aimed to provide a comprehensive whole-system provenance solution. The overwhelming majority of research in secure provenance is application-specific, therefore, difficult to apply across multiple domains. For example, threat provenance studies are primarily designed to detect certain types of attacks and are not generally suitable for threat detection and analysis in a broader context.
- *Privacy-aware aspect of provenance.* The importance of privacy in data provenance has been repeatedly emphasized. Despite a significant amount of effort, research on privacy-preserving provenance is limited. Typically, these studies ignore security issues narrowing the focus on privacy issues and vice versa, and the majority of secure provenance studies almost exclusively focus on security aspects of data provenance. Developing privacy-preserving techniques for data provenance that do not compromise the security properties of the system is a necessary step to facilitate comprehensive secure data provenance.
- *Real life deployment.* While many secure provenance systems have been proposed, the vast majority of them are not maintained; hence, only a few systems offer a fully developed practical solution for secure provenance.
- *Lack of a unified secure provenance model.* Compared to the substantial amount of research on provenance storage and management, the techniques for securing provenance remain relatively unexplored. A major gap in this respect is the absence of a unified and secure provenance model that can provide adequate security and privacy guarantees. While a few studies have utilized generic data models such as OPM and PROV that are widely used in other fields, most rely on provenance chains. Provenance chains can ensure integrity and authenticity, however, they lack other necessary protections and therefore require adaptation to the specific application domain or study requirements.

To overcome the gaps in secure provenance, it is evident that there is a need for more efforts in developing a standardized framework. This framework should be rooted in a unified and secure provenance model that provides comprehensive security and privacy guarantees.

To be viable in practice, the framework must outline principles for collecting, storing, and analyzing provenance data across diverse application domains. Additionally, it should provide a standardized approach to provenance management.

Standardized formats and ontologies for representing and exchanging provenance data can enable interoperability among various systems and simplify querying and analyzing large data volumes. This will reduce the complexity of resource-demanding provenance analysis, which is one of the significant challenges in this field.

Table 3. The summary of the reviewed studies in secure provenance

Related Work	Year	Category	Application Domain	Provenance Model	Security properties					
					I	NR	Auth	C	Pri.	Avail
Hasan et al. [61]	2009	Cryptography based	File systems	Provenance chain	✓	✓	✓	✓	-	-
SPROV [60]	2009	Cryptography based	File system	Provenance chain	✓	✓	✓	✓	-	-
Zhang et al. [145]	2009	Cryptography based	Database	DAG	✓	✓	✓	✓	-	-
Syalim et al. [119]	2010	Cryptography based	Generic	DAG	✓	✓	✓	✓	-	-
Lyle et al. [83]	2010	Cryptography based	Generic	No description	✓	-	-	-	-	-
Lu et al. [82]	2010	Cryptography based	Cloud computing	No description	-	✓	✓	✓	✓	-
Davidson et al. [42]	2010	Cryptography based	Scientific workflow systems	DAG	-	-	-	-	✓	-
EEPS [88]	2010	Cryptography based	Whole-system provenance	Provenance chain, DAG	✓	✓	✓	✓	-	-
Davidson et al. [43]	2011	Cryptography based	Scientific workflow systems	DAG	-	-	-	*	✓	-
Davidson et al. [41]	2011	Cryptography based	Scientific workflow systems	DAG	-	-	-	*	✓	-
Hasan et al. [57]	2011	Cryptography based	Mobile devices	Provenance chain	✓	✓	✓	-	✓	-
ProPub [44]	2011	Cryptography based	Data management	OPM based	-	-	-	-	✓	-
PKLC [135]	2012	Cryptography based	Distributed networks	Provenance chain	✓	✓	✓	✓	-	*
Hi-Fi [102]	2012	Cryptography based	Whole-system provenance	OPM	✓	-	-	-	-	-
PDP [9]	2012	Cryptography based	Documents	Provenance chain	✓	-	✓	-	✓	-
Sultana et al. [117]	2012	Cryptography based	WSN	DAG	✓	✓	✓	✓	-	-
Abbadi et al. [2]	2013	Cryptography based	Cloud systems	No description	✓	-	✓	✓	-	-
STAMP [136]	2013	Cryptography based	Mobile devices	No description	✓	✓	✓	-	✓	-
Bates et al. [18]	2013	Cryptography based	Distributed cloud environment	PROV	✓	✓	-	✓	-	-
Bertino et al. [22]	2014	Cryptography based	Generic	DAG	✓	✓	✓	-	✓	-
Hussain et al. [64]	2014	Cryptography based	WSN	DAG	✓	-	-	✓	-	-
OTIT [73]	2014	Cryptography based	Mobile devices	Provenance chain	✓	-	-	-	✓	-
Progger [74]	2014	Cryptography based	Cloud	No description	✓	✓	✓	-	-	-
WORAL [58]	2015	Cryptography based	Mobile devices	Provenance chain	✓	✓	✓	-	✓	-
LPM [19]	2015	Cryptography based	Whole-system provenance	Compatible with PROV	✓	✓	✓	-	-	-
SECAP [143]	2016	Cryptography based	Cloud systems	Provenance chain	✓	✓	✓	✓	-	-
Mutual Agreement [105]	2016	Cryptography based	Generic	DAG	✓	✓	✓	✓	-	-
Ahmed et al. [7]	2016	Cryptography based	Generic	DAG	✓	✓	-	✓	-	*
CamFlow [101]	2017	Cryptography based	Whole-system provenance	PROV	✓	-	-	-	-	-
ProvChain [79]	2017	Blockchain based	Cloud	Merkle tree	✓	✓	✓	✓	✓	*
SmartProvenance [104]	2018	Blockchain based	Generic	OPM	✓	✓	✓	✓	✓	*
ESP [147]	2018	Blockchain based	File system	Provenance chain	✓	✓	✓	-	✓	-
BCP [144]	2018	Blockchain based	WSN	Provenance table	-	-	-	✓	-	-
Sanchez et al. [32]	2018	Cryptography based	IoT	No description	✓	✓	✓	-	✓	-
BlockPro [70]	2018	Blockchain based	IoT	Provenance chain	✓	✓	✓	-	-	*
Jamil et al. [69]	2018	Cryptography based	Documents	PROV, Provenance chain, DAG	✓	✓	✓	✓	-	-
GPPub [137]	2018	Cryptography based	Generic	DAG	-	-	-	-	✓	-
Griggs et al. [55]	2018	Blockchain based	IoT	Provenance chain			✓	-	✓	-
Siddiqui et al. [109]	2019	Cryptography based	IoT	No description	✓	✓	-	✓	-	-
BlockCloud [125]	2019	Blockchain based	Cloud	No description	✓	✓	-	-	-	
LineageChain [106]	2019	Blockchain based	Generic	Merkle DAG	✓	-	-	-	-	*
Ahmed et al. [5]	2019	Cryptography based	Distributed networks	PROV, provenance chain	✓	✓	-	✓	-	-
Sigwart et al. [110]	2019	Blockchain based	IoT	Provenance chain	✓	-	✓	-	-	✓
Sigwart et al. [111]	2020	Blockchain based	IoT	IoT model [99]	✓				✓	✓
Ahmed et al. [6]	2020	Cryptography based	Distributed networks	Provenance chain	✓	✓	✓	✓	-	*
ProvNet [36]	2020	Blockchain based	Data sharing	Blocknet (Similar to DAG)	✓	✓	✓	-	-	-
BlockHDFS [93]	2021	Blockchain based	HDFS	Provenance chain	✓	-	✓	-	-	✓
AMP [46]	2021	Blockchain based	Media Data	Merkle Tree	✓	✓	✓	-	-	✓
HABE [103]	2021	Blockchain based	IoT	No description	✓	-	✓	✓	-	*
Trac^{2}Chain [124]	2022	Blockchain based	Generic	DAG	✓	-	✓	*	✓	*

✓indicates that the feature is supported in the proposed approach
’-’ indicates that the feature is not supported
’*’ implied although not implemented or explicitly discussed/incorporated.
I stands for Integrity, NR stands for Non-repudiation, Auth. stands for Authentication, C stands for Confidentiality, Pri stands for Privacy, and Avail stands for Availability
Empty table cell corresponds to the case when the data about given approach feature is not explicitly discussed or mentioned in the research paper, or not applicable to the approach.

Table 4. The summary of the reviewed studies in threat provenance

Proposed solutions	Year	Category	Threats Detected	Application Domain
Sultana et al. [116]	2011	Network-related attacks	Packet drop attacks	Sensor networks
Quire [45]	2011	Malware detection	Confused deputy attacks	Android platform
Shebaro et al. [108]	2012	Network-related attacks	Packet drop attacks, malicious nodes	Sensor networks
AMICO [127]	2013	Malware detection	Download malware	Operating system
BEEP [77]	2013	Generic provenance tracing	Cyber attacks	Binary programs
Ali et al. [10]	2014	Network-related attacks	Man-in-the-middle attacks	Wearable devices
Scippa [13]	2014	Malware detection	Confused deputy, intent hijacking and intent spoofing attacks	Android platform
Sultana et al. [118]	2015	Network-related attacks	Packet drop attacks	Wireless sensor networks
SPIF [120]	2015	Malware detection	Malware	Microsoft Windows OS
PPD [129]	2016	Network-related attacks	Packet drop attacks, packet replay attacks	Wireless sensor networks
PIDAS [139]	2016	Intrusion detection	Intrusion detection	Servers
ProTracer [84]	2016	Generic provenance tracing	Advanced persistent threat	Operating system
PANDDE [49]	2016	Detection/Analysis of security faults	Anomalous actions	Database
Upchurch et al. [126]	2016	Malware detection	Malware (for code reuse)	Operating system
Baeth et al. [14]	2017	Detection/Analysis of security faults	Misinformation	Social networks
Aman et al. [11]	2017	Network-related attacks	Physical attacks	IoT
Baracaldo et al. [17]	2017	Attacks on ML algorithms	Poisoning attacks	Machine learning
Baracaldo et al. [16]	2018	Attacks on ML algorithms	Poisoning attacks	Machine learning
Baeth et al. [15]	2018	Detection/Analysis of security faults	Privacy policy violation	Social media
HProve [130]	2018	Malware detection	Kernel malware	Operating systems
PROV-CPS [97]	2018	Detection/Analysis of security faults	Anomalous actions	Cyber-Physical Systems
LProv [131]	2018	Generic provenance tracing	Threat analysis	Operating system
A-PANDDE [48]	2019	Detection/Analysis of security faults	Anomalous actions	Database
PLC-PROV [8]	2019	Detection/Analysis of security faults	Safety and security faults	Programmable logic controllers (PLC) systems
Unicorn [56]	2020	Intrusion detection	Advanced persistent threats	Servers
Pagoda [138]	2020	Intrusion detection	Intrusion detection	Servers
ProvDetector [132]	2020	Malware detection	Stealthy malware	Operating system
PPPT [115]	2020	Network-related attacks	Packet drop attacks, packet replay attacks	RPL-based IoT
IBP [81]	2020	Network-related attacks	Data tempering attacks, packet drop attacks, replay attacks, malicious nodes	Mulithop IoT
P-Gaussian [140]	2021	Intrusion detection	Intrusion (variants) detection	Servers
threaTrace [133]	2021	Intrusion detection	Host-based threats	Servers
Kamal et al. [71]	2021	Network-related attacks	Man-in-the-middle attacks	Vehicle to vehicle (V2V) communication
VAMP [114]	2021	Attacks on ML algorithms	Poisoning attacks	Machine learning
Irshad et al. [66]	2021	Intrusion detection	Advanced persistent threats	Servers (enterprise-wide)
Anjum et al. [12]	2022	Intrusion detection	Advanced persistent threats	Servers
ProvTalk [121]	2022	Detection/Analysis of security faults	Security incidents	Networking functions virtualization (NFV)

7 CONCLUSION

Data provenance presents a powerful platform for security and privacy analytics. As the interconnectivity of devices continues to grow rapidly, ensuring the authenticity, integrity, and confidentiality of data has become a critical concern. In this work, we give a comprehensive overview of the role of data provenance in security and privacy. We view provenance from two complementary perspectives: *secure provenance* that entails secure handling of provenance data collection and manipulation, and *threat provenance*, a term we use to refer to provenance mechanisms used for identification and analysis of malicious activities. We define basic concepts of data provenance, its properties, and models.

Our review of the state-of-the-art secure provenance solutions revealed some research gaps and limitations. Although existing studies have explored the potential of cryptography and blockchain to enhance the security properties of data provenance, there is still a significant gap between theoretical frameworks and their practical applications in this field.

Overcoming this challenge requires the removal of impractical assumptions (e.g., trustworthiness of provenance collection system, trusted third party verification) and the development of standardized frameworks, secure provenance models, and the corresponding formats to guide provenance interoperability among systems and facilitate applicability of the developed solutions in a broader context.

In summary, while data provenance offers a powerful platform for security and privacy analytics, there are still many opportunities for further research in this field. By highlighting the fundamental aspects of data provenance and outlining the essential challenges of provenance in security and privacy, we aim to provide a foundation for further research efforts in this domain. Our analysis serves as a guide through existing research in the field, providing insight into gaps in security and threat provenance.

8 ACKNOWLEDGMENTS

This work has been partially supported by the Mitacs Accelerate grant funded by The Mitacs-Ericsson Global Artificial Intelligence Accelerator (GAIA) partnership.

REFERENCES

[1] 2022. Merkle tree. https://en.wikipedia.org/wiki/Merkle_tree

[2] Imad M Abbadi. 2013. A framework for establishing trust in Cloud provenance. *International journal of information security* 12, 2 (2013), 111–128.

[3] Amani Abu Jabal, Maryam Davari, Elisa Bertino, Christian Makaya, Seraphin Calo, Dinesh Verma, and Christopher Williams. 2021. ProFact: A Provenance-Based Analytics Framework for Access Control Policies. *IEEE Transactions on Services Computing* 14, 6 (2021), 1914–1928.

[4] Umut Acar, Peter Buneman, James Cheney, Jan Van Den Bussche, Natalia Kwasnikowska, and Stijn Vansummeren. 2010. A Graph Model of Data and Workflow Provenance. 8.

[5] Idrees Ahmed, Abid Khan, Mansoor Ahmed, and Saif ur Rehman. 2019. Order preserving secure provenance scheme for distributed networks. *Computers & Security* 82 (2019), 99–117.

[6] Idrees Ahmed, Abid Khan, Adeel Anjum, Mansoor Ahmed, and Muhammad Asif Habib. 2020. A secure provenance scheme for detecting consecutive colluding users in distributed networks. *International Journal of Parallel Programming* 48, 2 (2020), 344–366.

[7] Idrees Ahmed, Abid Khan, Muhammad Saleem Khan, and Mansoor Ahmed. 2016. Aggregated Signatures for Chaining: A Secure Provenance Scheme. In *2016 IEEE Trustcom/BigDataSE/ISPA*. 2012–2017.

[8] Abdullah Al Farooq, Jessica Marquard, Kripa George, and Thomas Moyer. 2019. Detecting Safety and Security Faults in PLC Systems with Data Provenance. In *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*. 1–6.

[9] Khalid Alharbi and Xiaodong Lin. 2012. PDP: A Privacy-Preserving Data Provenance Scheme. In *Proceedings of the 2012 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW '12)*. IEEE Computer Society, USA, 500–505.

[10] Syed Taha Ali, Vijay Sivaraman, Diethelm Ostry, Gene Tsudik, and Sanjay Jha. 2014. Securing First-Hop Data Provenance for Bodyworn Devices Using Wireless Link Fingerprints. *IEEE Transactions on Information Forensics and Security* 9, 12 (2014), 2193–2204.

[11] Muhammad Naveed Aman, Kee Chaing Chua, and Biplab Sikdar. 2017. Secure Data Provenance for the Internet of Things. In *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security* (Abu Dhabi, United Arab Emirates) *(IoTPTS '17)*. Association

for Computing Machinery, New York, NY, USA, 11–14.

[12] Md. Monowar Anjum, Shahrear Iqbal, and Benoit Hamelin. 2022. ANUBIS: a provenance graph-based framework for advanced persistent threat detection. In *SAC '22: The 37th ACM/SIGAPP Symposium on Applied Computing*. ACM, 1684–1693.

[13] Michael Backes, Sven Bugiel, and Sebastian Gerling. 2014. Scippa: System-Centric IPC Provenance on Android. In *Proceedings of the 30th Annual Computer Security Applications Conference* (New Orleans, Louisiana, USA) *(ACSAC '14)*. Association for Computing Machinery, New York, NY, USA, 36–45.

[14] Mohamed Jehad Baeth and Mehmet S. Aktas. 2017. Detecting Misinformation in Social Networks Using Provenance Data. In *2017 13th International Conference on Semantics, Knowledge and Grids (SKG)*. 85–89.

[15] Mohamed Jehad Baeth and Mehmet S Aktas. 2018. An approach to custom privacy policy violation detection problems using big social provenance data. *Concurrency and Computation: Practice and Experience* 30, 21 (2018), e4690.

[16] Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Amir Safavi, and Rui Zhang. 2018. Detecting Poisoning Attacks on Machine Learning in IoT Environments. In *2018 IEEE International Congress on Internet of Things (ICIOT)*. 57–64.

[17] Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, and Jaehoon Amir Safavi. 2017. *Mitigating Poisoning Attacks on Machine Learning Models: A Data Provenance Based Approach.* Association for Computing Machinery, New York, NY, USA, 103–110.

[18] Adam Bates, Ben Mood, Masoud Valafar, and Kevin Butler. 2013. Towards Secure Provenance-Based Access Control in Cloud Environments. In *Proceedings of the Third ACM Conference on Data and Application Security and Privacy* (San Antonio, Texas, USA) *(CODASPY '13)*. Association for Computing Machinery, New York, NY, USA, 277–284.

[19] Adam Bates, Dave (Jing) Tian, Kevin R.B. Butler, and Thomas Moyer. 2015. Trustworthy Whole-System Provenance for the Linux Kernel. In *24th USENIX Security Symposium (USENIX Security 15)*. USENIX Association, Washington, D.C., 319–334.

[20] Richard A. Becker and John M. Chambers. 1988. Auditing of Data Analyses. *SIAM J. Sci. Statist. Comput.* 9, 4 (1988), 747–760.

[21] Khalid Belhajjame, Jun Zhao, Daniel Garijo, Matthew Gamble, Kristina Hettne, Raul Palma, Eleni Mina, Oscar Corcho, José Manuel Gómez-Pérez, Sean Bechhofer, Graham Klyne, and Carole Goble. 2015. Using a suite of ontologies for preserving workflow-centric research objects. *Journal of Web Semantics* 32 (2015), 16–42. https://doi.org/10.1016/j.websem.2015.01.003

[22] Elisa Bertino, Gabriel Ghinita, Murat Kantarcioglu, Dang Nguyen, Jae Park, Ravi Sandhu, Salmin Sultana, Bhavani Thuraisingham, and Shouhuai Xu. 2014. A Roadmap for Privacy-Enhanced Secure Data Provenance. *J. Intell. Inf. Syst.* 43, 3 (dec 2014), 481–501. https://doi.org/10.1007/s10844-014-0322-7

[23] Elisa Bertino, Amani Abu Jabal, Seraphin Calo, Christian Makaya, Maroun Touma, Dinesh Verma, and Christopher Williams. 2017. Provenance-Based Analytics Services for Access Control Policies. In *2017 IEEE World Congress on Services (SERVICES)*. 94–101.

[24] Matt Bishop, Justin Cummins, Sean Peisert, Anhad Singh, Bhume Bhumiratana, Deborah A. Agarwal, Deborah A. Frincke, and Michael A. Hogarth. 2010. Relationships and data sanitization: a study in scarlet. In *Proceedings of the 2010 Workshop on New Security Paradigms, Concord, MA, USA, September 21-23, 2010*, Angelos D. Keromytis, Sean Peisert, Richard Ford, and Carrie Gates (Eds.). ACM, 151–164.

[25] Rajendra Bose and James Frew. 2005. Lineage Retrieval for Scientific Data Processing: A Survey. *ACM Comput. Surv.* 37, 1 (mar 2005), 1–28.

[26] Shawn Bowers. 2012. Scientific workflow, provenance, and data modeling challenges and approaches. , 19–30 pages.

[27] Uri Jacob Braun, Avraham Shinnar, and Margo I Seltzer. 2008. Securing provenance. In *Proceedings of the 3rd USENIX Workshop on Hot Topics in Security (HotSec'08)*. Usenix Association.

[28] Peter Buneman, Sanjeev Khanna, and Tan Wang-Chiew. 2001. Why and Where: A Characterization of Data Provenance. In *International Conference on Database Theory (ICDT)*. 316–330.

[29] Vitalik Buterin et al. 2014. Ethereum: A next-generation smart contract and decentralized application platform.

[30] Anila Sahar Butt and Peter Fitch. 2020. Provone+: a provenance model for scientific workflows. In *International Conference on Web Information Systems Engineering*. Springer, 431–444.

[31] Anila Sahar Butt and Peter Fitch. 2021. A provenance model for control-flow driven scientific workflows. *Data & Knowledge Engineering* 131-132 (2021), 101877.

[32] Jose Luis Canovas Sanchez, Jorge Bernal Bernabe, and Antonio F. Skarmeta. 2018. Towards privacy preserving data provenance for the Internet of Things. In *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. 41–46.

[33] Yang Cao, Christopher Jones, V Cuevas-Vicenttín, Matthew B Jones, Bertram Ludäscher, Timothy McPhillips, Paolo Missier, Christopher Schwalm, Peter Slaughter, Dave Vieglais, et al. 2016. ProvONE: extending PROV to support the DataONE scientific community. (2016).

[34] James Cheney, Laura Chiticariu, and Wang-Chiew Tan. 2009. Provenance in Databases: Why, How, and Where. *Found. Trends Databases* 1, 4 (April 2009), 379–474.

[35] James Cheney, Laura Chiticariu, and Wang-Chiew Tan. 2009. Provenance in Databases: Why, How, and Where. *Found. Trends Databases* 1, 4 (apr 2009), 379–474.

[36] Changhao Chenli and Taeho Jung. 2020. ProvNet: Networked Blockchain for Decentralized Secure Provenance. In *Blockchain – ICBC 2020*, Zhixiong Chen, Laizhong Cui, Balaji Palanisamy, and Liang-Jie Zhang (Eds.). Springer International Publishing, Cham, 76–93.

[37] Jin-Hee Cho and Ing-Ray Chen. 2018. PROVEST: Provenance-Based Trust Model for Delay Tolerant Networks. *IEEE Transactions on Dependable and Secure Computing* 15, 1 (2018), 151–165.

[38] Flavio Costa, Vítor Silva, Daniel de Oliveira, Kary Ocaña, Eduardo Ogasawara, Jonas Dias, and Marta Mattoso. 2013. Capturing and Querying Workflow Runtime Provenance with PROV: A Practical Approach. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops* (Genoa, Italy) *(EDBT '13)*. Association for Computing Machinery, New York, NY, USA, 282–289. https://doi.org/10.1145/2457317.2457365

[39] Víctor Cuevas-Vicenttín, Saumen Dey, Michael Li Yuan Wang, Tianhong Song, and Bertram Ludäscher. 2012. Modeling and Querying Scientific Workflow Provenance in the D-OPM. In *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*. 119–128.

[40] Yingwei Cui, Jennifer Widom, and Janet L. Wiener. 2000. Tracing the Lineage of View Data in a Warehousing Environment. 25, 2 (jun 2000), 179–227.

[41] Susan Davidson, Zhuowei Bao, and Sudeepa Roy. 2011. Hiding Data and Structure in Workflow Provenance. In *Databases in Networked Information Systems*, Shinji Kikuchi, Aastha Madaan, Shelly Sachdeva, and Subhash Bhalla (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 41–48.

[42] Susan B Davidson, Sanjeev Khanna, Debmalya Panigrahi, and Sudeepa Roy. 2010. Preserving module privacy in workflow provenance. (2010).

[43] Susan B. Davidson, Sanjeev Khanna, Sudeepa Roy, Julia Stoyanovich, Val Tannen, and Yi Chen. 2011. On Provenance and Privacy. In *Proceedings of the 14th International Conference on Database Theory* (Uppsala, Sweden) *(ICDT '11)*. Association for Computing Machinery, New York, NY, USA, 3–10.

[44] Saumen C. Dey, Daniel Zinn, and Bertram Ludäscher. 2011. ProPub: Towards a Declarative Approach for Publishing Customized, Policy-Aware Provenance. In *Scientific and Statistical Database Management*, Judith Bayard Cushing, James French, and Shawn Bowers (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 225–243.

[45] Michael Dietz, Shashi Shekhar, Yuliy Pisetsky, Anhei Shu, and Dan S. Wallach. 2011. Quire: Lightweight Provenance for Smart Phone Operating Systems. In *Proceedings of the 20th USENIX Conference on Security* (San Francisco, CA) *(SEC'11)*. USENIX Association, USA, 23.

[46] Paul England, Henrique S. Malvar, Eric Horvitz, Jack W. Stokes, Cédric Fournet, Rebecca Burke-Aguero, Amaury Chamayou, Sylvan Clebsch, Manuel Costa, John Deutscher, Shabnam Erfani, Matt Gaylor, Andrew Jenks, Kevin Kane, Elissa M. Redmiles, Alex Shamis, Isha Sharma, John C. Simmons, Sam Wenker, and Anika Zaman. 2021. AMP: Authentication of Media via Provenance. In *Proceedings of the 12th ACM Multimedia Systems Conference* (Istanbul, Turkey) *(MMSys '21)*. Association for Computing Machinery, New York, NY, USA, 108–121.

[47] ESET. 2014. An In-depth Analysis of Linux/Ebury.

[48] Daren Fadolalkarim and Elisa Bertino. 2019. A-PANDDE: Advanced Provenance-based ANomaly Detection of Data Exfiltration. *Computers & Security* 84 (2019), 276–287.

[49] Daren Fadolalkarim, Asmaa Sallam, and Elisa Bertino. 2016. PANDDE: Provenance-Based ANomaly Detection of Data Exfiltration. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy* (New Orleans, Louisiana, USA) *(CODASPY '16)*. Association for Computing Machinery, New York, NY, USA, 267–276.

[50] Juliana Freire, David Koop, Emanuele Santos, and Cláudio T. Silva. 2008. Provenance for Computational Tasks: A Survey. *Computing in Science Engineering* 10, 3 (2008), 11–21.

[51] Yuanzhao Gao, Xingyuan Chen, and Xuehui Du. 2020. A Big Data Provenance Model for Data Security Supervision Based on PROV-DM Model. *IEEE Access* 8 (2020), 38742–38752.

[52] Minos Garofalakis, Joseph M. Hellerstein, and Petros Maniatis. 2007. Proof Sketches: Verifiable In-Network Aggregation. In *2007 IEEE 23rd International Conference on Data Engineering*. 996–1005.

[53] Ashish Gehani and Dawood Tariq. 2012. SPADE: Support for Provenance Auditing in Distributed Environments. In *Proceedings of the 13th International Middleware Conference* (ontreal, Quebec, Canada) *(Middleware '12)*. Springer-Verlag, Berlin, Heidelberg, 101–120.

[54] Todd J. Green, Grigoris Karvounarakis, and Val Tannen. 2007. Provenance Semirings. In *ACM Symposium on Principles of Database Systems (PODS)*. 31–40.

[55] Kristen N. Griggs, Olya Ossipova, Christopher P. Kohlios, Alessandro N. Baccarini, Emily A. Howson, and Thaier Hayajneh. 2018. Healthcare Blockchain System Using Smart Contracts for Secure Automated Remote Patient Monitoring. *J. Med. Syst.* 42, 7 (jul 2018), 1–7.

[56] Xueyuan Han, Thomas Pasquier, Adam Bates, James Mickens, and Margo Seltzer. 2020. Unicorn: Runtime Provenance-Based Detector for Advanced Persistent Threats. *Proceedings 2020 Network and Distributed System Security Symposium* (2020).

[57] Ragib Hasan and Randal C. Burns. 2011. Where Have You Been? Secure Location Provenance for Mobile Devices. *CoRR* abs/1107.1821 (2011).

[58] Ragib Hasan, Rasib Khan, Shams Zawoad, and Md Munirul Haque. 2016. WORAL: A Witness Oriented Secure Location Provenance Framework for Mobile Devices. *IEEE Transactions on Emerging Topics in Computing* 4, 1 (2016), 128–141.

[59] Ragib Hasan, Radu Sion, and Marianne Winslett. 2007. Introducing Secure Provenance: Problems and Challenges. In *Proceedings of the 2007 ACM Workshop on Storage Security and Survivability* (Alexandria, Virginia, USA) *(StorageSS '07)*. Association for Computing Machinery, New York, NY, USA, 13–18.

[60] Ragib Hasan, Radu Sion, and Marianne Winslett. 2009. Preventing History Forgery with Secure Provenance. *ACM Trans. Storage* 5, 4, Article 12 (dec 2009), 43 pages.

[61] Ragib Hasan, Radu Sion, and Marianne Winslett. 2009. The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance. In *Proceedings of the 7th Conference on File and Storage Technologies* (San Francisco, California) *(FAST '09)*. USENIX Association, USA, 1–14.

[62] Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. 2017. A Survey on Provenance: What for? What Form? What From? *The VLDB Journal* 26, 6 (dec 2017), 881–906.

[63] Melanie Herschel and Marcel Hlawatsch. 2016. Provenance: On and Behind the Screens. In *Proceedings of the 2016 International Conference on Management of Data* (San Francisco, California, USA) *(SIGMOD '16)*. Association for Computing Machinery, New York, NY, USA, 2213–2217.

[64] Syed Rafiul Hussain, Changda Wang, Salmin Sultana, and Elisa Bertino. 2014. Secure data provenance compression using arithmetic coding in wireless sensor networks. In *2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC)*. 1–10.

[65] S. M. Iftekharul Alam and Sonia Fahmy. 2011. Energy-efficient provenance transmission in large-scale wireless sensor networks. In *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. 1–6.

[66] Hassaan Irshad, Gabriela Ciocarlie, Ashish Gehani, Vinod Yegneswaran, Kyu Hyung Lee, Jignesh Patel, Somesh Jha, Yonghwi Kwon, Dongyan Xu, and Xiangyu Zhang. 2021. TRACE: Enterprise-wide provenance tracking for real-time apt detection. *IEEE Transactions on Information Forensics and Security* 16 (2021), 4363–4376.

[67] Amani Abu Jabal and Elisa Bertino. 2016. SimP: Secure interoperable multi-granular provenance framework. In *2016 IEEE 12th International Conference on e-Science (e-Science)*. 270–275.

[68] Fariha Tasmin Jaigirdar, Carsten Rudolph, and Chris Bain. 2020. Prov-IoT: A Security-Aware IoT Provenance Model. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. 1360–1367. https://doi.org/10.1109/TrustCom50675.2020.00183

[69] Fuzel Jamil, Abid Khan, Adeel Anjum, Mansoor Ahmed, Farhana Jabeen, and Nadeem Javaid. 2018. Secure provenance using an authenticated data structure approach. *Computers & Security* 73 (2018), 34–56.

[70] Uzair Javaid, Muhammad Naveed Aman, and Biplab Sikdar. 2018. BlockPro: Blockchain Based Data Provenance and Integrity for Secure IoT Environments. In *Proceedings of the 1st Workshop on Blockchain-Enabled Networked Sensor Systems* (Shenzhen, China) *(BlockSys'18)*. Association for Computing Machinery, New York, NY, USA, 13–18.

[71] Mohsin Kamal, Gautam Srivastava, and Muhammad Tariq. 2021. Blockchain-Based Lightweight and Secured V2V Communication in the Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems* 22, 7 (2021), 3997–4004.

[72] Arun Kanuparthi, Ramesh Karri, and Sateesh Addepalli. 2013. Hardware and Embedded Security in the Context of Internet of Things. In *Proceedings of the 2013 ACM Workshop on Security, Privacy & Dependability for Cyber Vehicles* (Berlin, Germany) *(CyCAR '13)*. Association for Computing Machinery, New York, NY, USA, 61–64.

[73] Rasib Khan, Shams Zawoad, Md Munirul Haque, and Ragib Hasan. 2014. OTIT: Towards Secure Provenance Modeling for Location Proofs. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security* (Kyoto, Japan) *(ASIA CCS '14)*. Association for Computing Machinery, New York, NY, USA, 87–98.

[74] Ryan K.L. Ko and Mark A. Will. 2014. Progger: An Efficient, Tamper-Evident Kernel-Space Logger for Cloud Data Provenance Tracking. In *2014 IEEE 7th International Conference on Cloud Computing*. 881–889.

[75] Natalia Kwasnikowska, Luc Moreau, and Jan Van Den Bussche. 2015. A Formal Account of the Open Provenance Model. *ACM Trans. Web* 9, 2, Article 10 (may 2015), 44 pages.

[76] Brian Lee, Abir Awad, and Mirna Awad. 2015. Towards Secure Provenance in the Cloud: A Survey. In *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*. 577–582.

[77] Kyu Hyung Lee, Xiangyu Zhang, and Dongyan Xu. 2013. High Accuracy Attack Provenance via Binary-based Execution Partition. In *20th Annual Network and Distributed System Security Symposium, NDSS 2013, San Diego, California, USA, February 24-27, 2013*. The Internet Society.

[78] Tao Li, Ling Liu, Xiaolong Zhang, Kai Xu, and Chao Yang. 2014. ProvenanceLens: Service provenance management in the cloud. In *10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*. 275–284.

[79] Xueping Liang, Sachin Shetty, Deepak Tosh, Charles Kamhoua, Kevin Kwiat, and Laurent Njilla. 2017. ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. 468–477.

[80] Hyo-Sang Lim, Yang-Sae Moon, and Elisa Bertino. 2010. Provenance-Based Trustworthiness Assessment in Sensor Networks. In *Proceedings of the Seventh International Workshop on Data Management for Sensor Networks* (Singapore) *(DMSN '10)*. Association for Computing Machinery, New York, NY, USA, 2–7.

[81] Zhe Liu and Yuting Wu. 2020. An Index-Based Provenance Compression Scheme for Identifying Malicious Nodes in Multihop IoT Network. *IEEE Internet of Things Journal* 7, 5 (2020), 4061–4071. https://doi.org/10.1109/JIOT.2019.2961431

[82] Rongxing Lu, Xiaodong Lin, Xiaohui Liang, and Xuemin (Sherman) Shen. 2010. Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security* (Beijing, China) *(ASIACCS '10)*. Association for Computing Machinery, New York, NY, USA, 282–292.

[83] John Lyle and Andrew Martin. 2010. Trusted Computing and Provenance: Better Together. In *Proceedings of the 2nd Conference on Theory and Practice of Provenance* (San Jose, California) *(TAPP'10)*. USENIX Association, USA, 1.

[84] Shiqing Ma, Xiangyu Zhang, and Dongyan Xu. 2016. ProTracer: Towards Practical Provenance Tracing by Alternating Between Logging and Tainting. In *23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*. The Internet Society.

[85] Sidra Malik, Volkan Dedeoglu, Salil S. Kanhere, and Raja Jurdak. 2021. PrivChain: Provenance and Privacy Preservation in Blockchain enabled Supply Chains. *CoRR* abs/2104.13964 (2021).

[86] Sidra Malik, Salil S Kanhere, and Raja Jurdak. 2018. Productchain: Scalable blockchain framework to support provenance in supply chains. In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. IEEE, 1–10.

[87] Anderson Marinho, Leonardo Murta, Cláudia Werner, Vanessa Braganholo, Sérgio Manuel Serra da Cruz, Eduardo Ogasawara, and Marta Mattoso. 2012. ProvManager: a provenance management system for scientific workflows. *Concurrency and Computation: Practice and Experience* 24, 13 (2012), 1513–1530.

[88] Patrick McDaniel, Kevin Butler, Stephen McLaughlin, Radu Sion, Erez Zadok, and Marianne Winslett. 2010. Towards a Secure and Efficient System for End-to-End Provenance. In *2nd USENIX Workshop on the Theory and Practice of Provenance (TaPP 10)*. USENIX Association, San Jose, CA.

[89] Paolo Missier, Khalid Belhajjame, and James Cheney. 2013. The W3C PROV Family of Specifications for Modelling Provenance Metadata *(EDBT '13)*. Association for Computing Machinery, New York, NY, USA, 773–776.

[90] Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, Beth Plale, Yogesh Simmhan, Eric Stephan, and Jan Van den Bussche. 2011. The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems* 27, 6 (2011), 743–756. https://doi.org/10.1016/j.future.2010.07.005

[91] Luc Moreau, Juliana Freire, Joe Futrelle, Robert E McGrath, Jim Myers, and Patrick Paulson. 2008. The open provenance model: An overview. In *International provenance and annotation workshop*. Springer, 323–326.

[92] Luc Moreau, Beth Plale, Simon Miles, Carole Goble, Paolo Missier, Roger Barga, Yogesh Simmhan, Joe Futrelle, Robert E McGrath, Jim Myers, et al. 2008. The open provenance model (v1. 01). *Technical Report 16148, Electronics and Computer Science* (2008).

[93] Viraaji Mothukuri, Sai S. Cheerla, Reza M. Parizi, Qi Zhang, and Kim-Kwang Raymond Choo. 2021. BlockHDFS: Blockchain-integrated Hadoop distributed file system for secure provenance traceability. *Blockchain: Research and Applications* 2, 4 (2021), 100032.

[94] Kiran-Kumar Muniswamy-Reddy, Uri Braun, David A. Holland, Peter Macko, Diana Maclean, Daniel Margo, Margo Seltzer, and Robin Smogor. 2009. Layering in Provenance Systems. In *Proceedings of the 2009 Conference on USENIX Annual Technical Conference* (San Diego, California) *(USENIX'09)*. USENIX Association, USA, 10.

[95] Kiran-Kumar Muniswamy-Reddy, David A. Holland, Uri Braun, and Margo Seltzer. 2006. Provenance-Aware Storage Systems. In *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference* (Boston, MA) *(ATEC '06)*. USENIX Association, USA, 4.

[96] Qun Ni, Shouhuai Xu, Elisa Bertino, Ravi Sandhu, and Weili Han. 2009. An Access Control Language for a General Provenance Model. In *Secure Data Management*, Willem Jonker and Milan Petković (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 68–88.

[97] Ebelechukwu Nwafor. 2018. *Trace-Based Data Provenance For Cyber-Physical Systems*. Ph. D. Dissertation. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-05-19.

[98] Wellington Oliveira, Daniel De Oliveira, and Vanessa Braganholo. 2018. Provenance Analytics for Workflow-Based Computational Experiments: A Survey. *ACM Comput. Surv.* 51, 3, Article 53 (may 2018), 25 pages.

[99] Habeeb Olufowobi, Robert Engel, Nathalie Baracaldo, Luis Angel D. Bathen, Samir Tata, and Heiko Ludwig. 2017. Data Provenance Model for Internet of Things (IoT) Systems. In *Service-Oriented Computing – ICSOC 2016 Workshops*, Khalil Drira, Hongbing Wang, Qi Yu, Yan Wang, Yuhong Yan, François Charoy, Jan Mendling, Mohamed Mohamed, Zhongjie Wang, and Sami Bhiri (Eds.). Springer International Publishing, Cham, 85–91.

[100] Jaehong Park, Dang Nguyen, and Ravi Sandhu. 2012. A provenance-based access control model. In *Tenth Annual International Conference on Privacy, Security and Trust (PST 12)*. 137–144.

[101] Thomas Pasquier, Xueyuan Han, Mark Goldstein, Thomas Moyer, David Eyers, Margo Seltzer, and Jean Bacon. 2017. Practical Whole-System Provenance Capture. In *Proceedings of the 2017 Symposium on Cloud Computing* (Santa Clara, California) *(SoCC '17)*. Association for Computing Machinery, New York, NY, USA, 405–418.

[102] Devin J. Pohly, Stephen McLaughlin, Patrick McDaniel, and Kevin Butler. 2012. Hi-Fi: Collecting High-Fidelity Whole-System Provenance. In *Proceedings of the 28th Annual Computer Security Applications Conference* (Orlando, Florida, USA) *(ACSAC '12)*. Association for Computing Machinery, New York, NY, USA, 259–268.

[103] S Porkodi and D Kesavaraja. 2021. Secure Data Provenance in Internet of Things using Hybrid Attribute based Crypt Technique. *Wireless Personal Communications* 118, 4 (2021), 2821–2842.

[104] Aravind Ramachandran and Murat Kantarcioglu. 2018. SmartProvenance: A Distributed, Blockchain Based Data Provenance System. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy* (Tempe, AZ, USA) *(CODASPY '18)*. Association for Computing Machinery, New York, NY, USA, 35–42.

[105] Mohammed Rangwala, Zhengli Liang, Wei Peng, Xukai Zou, and Feng Li. 2016. A mutual agreement signature scheme for secure data provenance. *environments* 13, 14 (2016), 726–733.

[106] Pingcheng Ruan, Gang Chen, Tien Tuan Anh Dinh, Qian Lin, Beng Chin Ooi, and Meihui Zhang. 2019. Fine-Grained, Secure and Efficient Data Provenance on Blockchain Systems. *Proc. VLDB Endow.* 12, 9 (May 2019), 975–988.

[107] Reiner Sailer, Xiaolan Zhang, Trent Jaeger, and Leendert van Doorn. 2004. Design and Implementation of a TCG-Based Integrity Measurement Architecture. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13* (San Diego, CA) *(SSYM'04)*. USENIX Association, USA, 16.

[108] Bilal Shebaro, Salmin Sultana, Shakthidhar Reddy Gopavaram, and Elisa Bertino. 2012. Demonstrating a Lightweight Data Provenance for Sensor Networks. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (Raleigh, North Carolina, USA) *(CCS '12)*. Association for Computing Machinery, New York, NY, USA, 1022–1024.

[109] Muhammad Shoaib Siddiqui, Atiqur Rahman, and Adnan Nadeem. 2019. Secure Data Provenance in IoT Network using Bloom Filters. *Procedia Computer Science* 163 (2019), 190–197. 16th Learning and Technology Conference 2019Artificial Intelligence and Machine Learning: Embedding the Intelligence.

[110] Marten Sigwart, Michael Borkowski, Marco Peise, Stefan Schulte, and Stefan Tai. 2019. Blockchain-Based Data Provenance for the Internet of Things. In *Proceedings of the 9th International Conference on the Internet of Things* (Bilbao, Spain) *(IoT 2019)*. Association for Computing Machinery, New York, NY, USA, Article 15, 8 pages.

[111] Marten Sigwart, Michael Borkowski, Marco Peise, Stefan Schulte, and Stefan Tai. 2020. A secure and extensible blockchain-based data provenance framework for the Internet of Things. *Personal and Ubiquitous Computing* (06 2020).

[112] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. 2005. A Survey of Data Provenance in E-Science. *SIGMOD Rec.* 34, 3 (sep 2005), 31–36.

[113] Radu Sion. 2008. Strong WORM. In *Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems (ICDCS '08)*. IEEE Computer Society, USA, 69–76.

[114] Jack W. Stokes, Paul England, and Kevin Kane. 2021. Preventing Machine Learning Poisoning Attacks Using Authentication and Provenance. arXiv:2105.10051 [cs.CR]

[115] Sabah Suhail, Rasheed Hussain, Mohammad Abdellatif, Shashi Raj Pandey, Abid Khan, and Choong Seon Hong. 2020. Provenance-enabled packet path tracing in the RPL-based Internet of Things. *Computer Networks* 173 (2020), 107189.

[116] Salmin Sultana, Elisa Bertino, and Mohamed Shehab. 2011. A Provenance Based Mechanism to Identify Malicious Packet Dropping Adversaries in Sensor Networks. In *2011 31st International Conference on Distributed Computing Systems Workshops*. 332–338.

[117] Salmin Sultana, Gabriel Ghinita, Elisa Bertino, and Mohamed Shehab. 2012. A Lightweight Secure Provenance Scheme for Wireless Sensor Networks. In *2012 IEEE 18th International Conference on Parallel and Distributed Systems*. 101–108.

[118] Salmin Sultana, Gabriel Ghinita, Elisa Bertino, and Mohamed Shehab. 2015. A Lightweight Secure Scheme for Detecting Provenance Forgery and Packet Drop Attacks in Wireless Sensor Networks. *IEEE Transactions on Dependable and Secure Computing* 12, 3 (2015), 256–269.

[119] Amril Syalim, Takashi Nishide, and Kouichi Sakurai. 2010. Preserving Integrity and Confidentiality of a Directed Acyclic Graph Model of Provenance. In *Data and Applications Security and Privacy XXIV*, Sara Foresti and Sushil Jajodia (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 311–318.

[120] Wai Kit Sze and R. Sekar. 2015. Provenance-Based Integrity Protection for Windows *(ACSAC 2015)*. Association for Computing Machinery, New York, NY, USA, 211–220.

[121] Azadeh Tabiban, Heyang Zhao, Yosr Jarraya, Makan Pourzandi, Mengyuan Zhang, and Lingyu Wang. 2022. ProvTalk: Towards Interpretable Multi-level Provenance Analysis in Networking Functions Virtualization (NFV). In *Network and Distributed System Security Symposium (NDSS)*.

[122] Wang Chiew Tan et al. 2007. Provenance in databases: Past, current, and future. *IEEE Data Eng. Bull.* 30, 4 (2007), 3–12.

[123] Yu Shyang Tan, Ryan K.L. Ko, and Geoff Holmes. 2013. Security and Data Accountability in Distributed Systems: A Provenance Survey. In *2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing*. 1571–1578.

[124] Wenyi Tang, Changhao Chenli, Chanyang Ju, and Taeho Jung. 2022. Trac2Chain: Trackability and Traceability of Graph Data in Blockchain with Linkage Privacy. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing* (Virtual Event) *(SAC '22)*. Association for Computing Machinery, New York, NY, USA, 272–281.

[125] Deepak Tosh, Sachin Shetty, Xueping Liang, Charles Kamhoua, and Laurent L. Njilla. 2019. Data Provenance in the Cloud: A Blockchain-Based Approach. *IEEE Consumer Electronics Magazine* 8, 4 (2019), 38–44.

[126] Jason Upchurch and Xiaobo Zhou. 2016. Malware provenance: code reuse detection in malicious software at scale. In *2016 11th International Conference on Malicious and Unwanted Software (MALWARE)*. 1–9.

[127] Phani Vadrevu, Babak Rahbarinia, Roberto Perdisci, Kang Li, and Manos Antonakakis. 2013. Measuring and Detecting Malware Downloads in Live Network Traffic. In *Computer Security – ESORICS 2013*, Jason Crampton, Sushil Jajodia, and Keith Mayes (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 556–573.

[128] Changda Wang and Elisa Bertino. 2017. Sensor Network Provenance Compression Using Dynamic Bayesian Networks. *ACM Trans. Sen. Netw.* 13, 1, Article 5 (jan 2017), 32 pages.

[129] Changda Wang, Syed Rafiul Hussain, and Elisa Bertino. 2016. Dictionary Based Secure Provenance Compression for Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems* 27, 2 (2016), 405–418.

[130] Chonghua Wang, Shiqing Ma, Xiangyu Zhang, Junghwan Rhee, Xiaochun Yun, and Zhiyu Hao. 2018. A Hypervisor Level Provenance System to Reconstruct Attack Story Caused by Kernel Malware. In *Security and Privacy in Communication Networks*. Springer International Publishing, Cham, 778–792.

[131] Fei Wang, Yonghwi Kwon, Shiqing Ma, Xiangyu Zhang, and Dongyan Xu. 2018. Lprov: Practical Library-Aware Provenance Tracing. In *Proceedings of the 34th Annual Computer Security Applications Conference* (San Juan, PR, USA) *(ACSAC '18)*. Association for Computing Machinery, New York, NY, USA, 605–617.

[132] Qi Wang, Wajih Ul Hassan, Ding Li, Kangkook Jee, Xiao Yu, Kexuan Zou, Junghwan Rhee, Zhengzhang Chen, Wei Cheng, Carl A Gunter, et al. 2020. You Are What You Do: Hunting Stealthy Malware via Data Provenance Analysis. In *NDSS*.

[133] Su Wang, Zhiliang Wang, Tao Zhou, Xia Yin, Dongqi Han, Han Zhang, Hongbin Sun, Xingang Shi, and Jiahai Yang. 2021. threaTrace: Detecting and Tracing Host-based Threats in Node Level Through Provenance Graph Learning. *CoRR* abs/2111.04333 (2021).

[134] Xinlei Wang, Kannan Govindan, and Prasant Mohapatra. 2010. Provenance-Based Information Trustworthiness Evaluation in Multi-Hop Networks. In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. 1–5.

[135] Xinlei Wang, Kai Zeng, Kannan Govindan, and Prasant Mohapatra. 2012. Chaining for securing data provenance in distributed information networks. In *2012 IEEE Military Communications Conference (MILCOM 2012)*. 1–6.

[136] Xinlei Wang, Jindan Zhu, Amit Pande, Arun Raghuramu, Prasant Mohapatra, Tarek Abdelzaher, and Raghu Ganti. 2013. STAMP: Ad hoc spatial-temporal provenance assurance for mobile users. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*. 1–10.

[137] Jian Wu, Weiwei Ni, and Sen Zhang. 2018. Generalization Based Privacy-Preserving Provenance Publishing. In *Web Information Systems and Applications*, Xiaofeng Meng, Ruixuan Li, Kanliang Wang, Baoning Niu, Xin Wang, and Gansen Zhao (Eds.). Springer International Publishing, Cham, 287–299.

[138] Yulai Xie, Dan Feng, Yuchong Hu, Yan Li, Staunton Sample, and Darrell Long. 2020. Pagoda: A Hybrid Approach to Enable Efficient Real-Time Provenance Based Intrusion Detection in Big Data Environments. *IEEE Transactions on Dependable and Secure Computing* 17, 6 (2020), 1283–1296. https://doi.org/10.1109/TDSC.2018.2867595

[139] Yulai Xie, Dan Feng, Zhipeng Tan, and Junzhe Zhou. 2016. Unifying intrusion detection and forensic analysis via provenance awareness. *Future Generation Computer Systems* 61 (2016), 26–36.

[140] Yulai Xie, Yafeng Wu, Dan Feng, and Darrell Long. 2021. P-Gaussian: Provenance-Based Gaussian Distribution for Detecting Intrusion Behavior Variants Using High Efficient and Real Time Memory Databases. *IEEE Transactions on Dependable and Secure Computing* 18, 6 (2021), 2658–2674.

[141] Qinbao Xu, Rizwan Akhtar, Xing Zhang, Changda Wang, and Kim-Kwang Raymond Choo. 2018. Cluster-Based Arithmetic Coding for Data Provenance Compression in Wireless Sensor Networks. *Wirel. Commun. Mob. Comput.* 2018 (jan 2018), 15 pages.

[142] Faheem Zafar, Abid Khan, Saba Suhail, Idrees Ahmed, Khizar Hameed, Hayat Mohammad Khan, Farhana Jabeen, and Adeel Anjum. 2017. Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes. *Journal of Network and Computer Applications* 94 (2017), 50–68.

[143] Shams Zawoad and Ragib Hasan. 2016. SECAP: Towards Securing Application Provenance in the Cloud. In *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*. 900–903.

[144] Yu Zeng, Xing Zhang, Rizwan Akhtar, and Changda Wang. 2018. A Blockchain-Based Scheme for Secure Data Provenance in Wireless Sensor Networks. In *2018 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*. 13–18.

[145] Jing Zhang, Adriane Chapman, and Kristen LeFevre. 2009. Do You Know Where Your Data's Been? – Tamper-Evident Database Provenance. In *Secure Data Management*, Willem Jonker and Milan Petković (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 17–32.

[146] Olive Qing Zhang, Markus Kirchberg, Ryan K.L. Ko, and Bu Sung Lee. 2011. How to Track Your Data: The Case for Cloud Computing Provenance. In *2011 IEEE Third International Conference on Cloud Computing Technology and Science*. 446–453.

[147] Yuan Zhang, Xiaodong Lin, and Chunxiang Xu. 2018. Blockchain-Based Secure Data Provenance for Cloud Storage. In *Information and Communications Security*, David Naccache, Shouhuai Xu, Sihan Qing, Pierangela Samarati, Gregory Blanc, Rongxing Lu, Zonghua Zhang, and Ahmed Meddahi (Eds.). Springer International Publishing, Cham, 3–19.

[148] Yuankai Zhang, Adam O'Neill, Micah Sherr, and Wenchao Zhou. 2017. Privacy-Preserving Network Provenance. *Proc. VLDB Endow.* 10, 11 (aug 2017), 1550–1561.

[149] Michael Zipperle, Florian Gottwalt, Elizabeth Chang, and Tharam Dillon. 2022. Provenance-Based Intrusion Detection Systems: A Survey. *ACM Comput. Surv.* 55, 7, Article 135 (dec 2022), 36 pages.