Visualizing and Reasoning about Presentable Digital Forensic Evidence with Knowledge Graphs

Weifeng Xu
School of Criminal Justice
University of Baltimore)
Baltimore, USA
wxu@ubalt.edu

Dianxiang Xu
School of Computing and Engineering
University of Missouri-Kansas City
Kansas City, USA
dxu@umkc.edu

***Abstract*—Making digital evidence presentable is hard due to the intangible and complex nature of digital evidence and the variety of targeted audiences. In this paper, we present Digital Forensic Knowledge Graph (DFKG) for visualizing and reasoning about digital forensic evidence. We first describe the criteria of presentable evidence to ensure authenticity, integrity, validity, credibility, and relevance of evidence. Then we specify DFKG to capture presentable forensic evidence from three perspectives: (1) the background of a criminal case, (2) the reconstructed timeline of a criminal case, and (3) the verifiable digital evidence related to the criminal activity timeline. We also present a case study to illustrate the DFKG-based approach.**

***Index Terms*—digital forensic, formalization, knowledge graph, presentable digital forensic evidence**

I. Introduction

As our daily life increasingly depends on information technology, digital forensic evidence cannot be overstated because it is often the key to proving someone is guilty or innocent of the actions they have been charged. When testifying in court as an expert, digital forensic professionals also need to present and interpret evidence to various audiences. Their testimony is usually in the form of verbal presentations with visual aids, such as images, screenshots of file systems, and printed log files. The limitations of verbal presentation in a court are well-illustrated by Watt [1] in the following scenario: *It is Friday afternoon in week three of a trial. The jury has returned from lunch, the well-presented computer forensic examiner, though very expert and professional, commences their verbal evidence of the facts and their opinions of what occurred. The jury looks at the books containing the extracted data, emails, log files, or other data. The expert provides in-depth technical knowledge of the intricate workings of the computer and explains in detail how the information presented was likely to have occurred. If half of the jury is still listening after five minutes, it would be considered good going; if two of them understand what is being said, this would be an amazing achievement.* Eriksson also argued that "no forensic method has been rigorously shown to have the capacity to consistently, and with a high degree of certainty, demonstrate a connection between evidence and a specific individual or source." [2]

Making the evidence presentable is a challenging problem that digital forensic professionals face – whether in a judge-only trial or where there is a jury present. They need to answer the following questions:

- Is the evidence trustable? Digital forensics has become increasingly complex with the growing usage of various digital devices, operating systems, and computing applications. Judges and juries often have no digital forensic knowledge. It is not easy to convince them whether or not a piece of digital evidence is valid.
- How is evidence related to a criminal case? It is critical to understand and verify the extent to which evidence is related to a case and how evidence is used to reconstruct crime scenes.
- How to interpret evidence? The interpretation affects the final verdict. It is important to ensure the interpretation of the evidence is consistent regardless of which expert presents the evidence and who the audiences are.

To address the above issues, this paper presents Digital Forensic Knowledge Graph (DFKG) for representing digital forensic evidence. DFKG meets the needs of **Pre**sentable **D**igital forens**I**c **E**vidence (PreDIE) for delivering testimony in courts. It puts evidence information in context via linking and the semantic metadata of evidence. The contributions of this paper include:

- Identifying the criteria for cross-jurisdictional acceptance of digital evidence.
- Representing comprehensive intelligence of criminal cases in DFKG.
- Demonstrating that the acceptance of evidence specified in DFKG meets the criteria.
- Analyzing presentable evidence with DFKG.

The rest of the paper is organized as follows: Section II introduces presentable evidence with DFKG. Sections III, IV, and V describe three different types of nodes in DFKG, respectively, for characterizing cybercrime case background, forensic observable artifacts, and reconstructed crime scenes. Section VI presents an empirical study. Section VII summarizes the related works. Finally, Section VIII concludes this paper.

The project is funded by National Science Foundation Grant No. 2039289, "EAGER: SaTC-EDU: Exploring Visualized and Explainable Artificial Intelligence to Improve Students' Learning Experience in Digital Forensics Education".

II. Presentable evidence with Knowledge Graphs

A. Digital Forensic Artifact and Evidence

A digital forensic artifact is any type of item produced by digital devices, stored in an electronic form, and used for forensic investigations. It is a by-product of a suspect's activity of using digital devices. Common digital artifacts include Word documents, pictures, software applications, network traffic logs, and system logs. An artifact has metadata, i.e., descriptive properties, to depict what the artifact is without any context. Formally, we define an artifact A has a list of properties $P_A = \{p_1, \ldots, p_n : n \in \mathbb{N}\}$. For example, an image may have two properties $\{name, size\}$.

Digital forensic evidence is a digital forensic artifact presentable to prove a crime. Like other transitional non-digital evidence, digital forensic evidence must be admissible [3] at court. In general, admissible evidence [4] is an artifact that the trial judge finds useful in helping the trier of fact (a jury if there is a jury, otherwise the judge), and which cannot be objected to on the basis that it is irrelevant, immaterial, or violates the rules against hearsay and other objections.

Cybercrime refers to the use of computing devices as a means to conduct illegal behaviors, including attacking information technology infrastructure, infringing intellectual property, committing financial fraud, spreading child pornography, stealing identities, or violating privacy. Digital forensic investigation applies a systematic approach to solving cybercrime cases. It is an investigative procedure for uncovering and analyzing digital evidence acquired from computing devices. After applying derived and proven methods toward the identification, collection, preservation, validation, and analysis of the evidence, digital forensic experts can identify past events and reconstruct a crime scene [5]. Ultimately, the evidence and interpretation of the evidence need to be presented by the court of law.

B. Presentable Digital Forensic Evidence

We consider two principles for formulating the criteria of Presentable Digital Forensic Evidence (PreDIE): (1) PreDIE must comply with the admissibility requirements of traditional non-digital evidence because it is a special type of evidence. (2) PreDIE needs to extend the admissibility requirements of traditional non-digital evidence for the field of digital forensic investigations. In this paper, the criteria of PreDIE are as follows:

- Authenticity: A piece of digital forensic evidence is proven to be genuine and not a forgery. The authenticity of the evidence is proved by demonstrating its provenance. Authenticity also provides the traceability of evidence. For example, if an image is authenticated evidence, the source needs to be provided - either the server's IP address if it was downloaded from a server or a disk image containing it.
- Integrity: A piece of digital forensic evidence should not have been tampered. Unlike physical evidence, such as a gun used to commit a crime, digital evidence is intangible and consists of a sequence of binary numbers. Intangible evidence always is exposed to the risk of tampering, such as a person altering, concealing, falsifying, or destroying evidence with the intent to interfere with an investigation by a law-enforcement, governmental, or regulatory authority [6].
- Validity: The validity of evidence means that a tool used in forensic examination meets standards. It avoids the risk of producing incorrect analysis results. For example, a log parser may produce inaccurate log interpretation. A reference to a parser tool allows us to validate the evidence.
- Credibility: The credibility of evidence demonstrates a forensic examiner who will be responsible for the discovery or interpretation of artifacts. Digital evidence with credibility is considered truthfully constructed and defined.
- Relevance: Evidence is relevant if it has any tendency to make a fact more or less probable than it would be without the evidence and the fact is of consequence in determining the action [7]. In other words, it requires a piece of evidence containing the necessary information to explain when, why, what, and how a piece of evidence was related to a criminal case, or more specifically, an activity performed by a suspect.

C. Digital Forensic Knowledge graphs

Since named in 1982 [8], knowledge graph, e.g., Google Knowledge Graph [9], is proven to be a promising way to store data, organize information, and answer questions in various domains.

DFKG is a knowledge graph representing a collection of entities and relationships among these entities. Entities are often real-world objects or abstract concepts presented in cybercrimes and digital forensic investigations. For example, a forensic investigator is a real-world entity. Digital forensic artifacts, evidence, and investigation software tools are concepts that do not have a physical form. Relationships interconnect entities with proper reasonings. Figure 1 shows the simplest DFKG with two rectangular nodes representing a forensic investigator and a criminal case under the investigation. There are two edges (i.e.., binary directional edge) connect them with labels *investigates* and *investigated-by*. The labels on the edges capture the meaning of the relationships. Ellipse nodes that the *investigator* object points to are the properties of the object. These are default *"has"* relationships between properties and the entity they belong to.

Definition 1. Digital Forensic Knowledge Graph: Given a set of nodes N and a set of relationships R between nodes, a DFKG is a subset of the cross product $N \times R \times N$.

Each node in N represents an object with metadata. All objects fall into three categories: Cyber Case Descriptive Objects (CCDO), Cyber Forensic Observable Objects (CDOO), and Cyber Forensic Domain Objects (CFDO). Formally, $N = N_{CCDO} \cup N_{CDOO} \cup N_{CFDO}$. They will be described in Sections III, IV, and V, respectively.

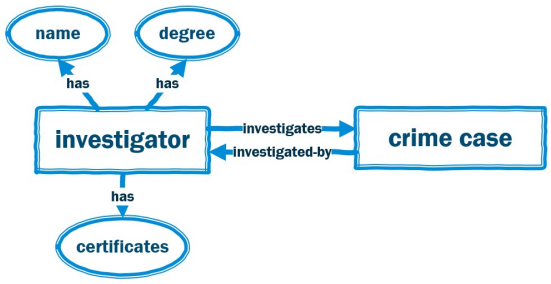

Fig. 1. *investigates* and *investigated-by* relationships between a digital forensic investigator entity and a crime case entity.

Each relation $< x, r, y >$ $(x \in N, y \in N, x \neq y, r \in R)$ forms a simple sentence with a subject (x), a predicate (r), and an object (y). Objects add extra information to subjects. We also denote the sentence by $r(x, y)$, and use r as a label in the DFKG. For example, the semantics of Figure 1 is written as *investigates(investigator, crime case)* and the corresponding label is *investigates*.

Definition 2. Implied inverse relationship: $r'(y, x)$ is an implied inverse relationship of $r(x, y)$ if and only if r' is the inverse of r.

For example, *investigates(investigator, crime case)* is an inverse relationship of *investigated-by(crime case, investigator)*. We refer to the original relationship as a forward relationship.

Definition 3. PreDIE Reasoning graph: A PreDIE reasoning graph (PRG) is a subgraph of DFKG for reasoning about a specific PreDIE criterion.

The simplest PRG contains only one relation $r(x, y)$. For example, a PRG has the relationship $investigated-by(crimecase, investigator)$. The relationship supports the credibility of evidence by showing that a criminal case has been assigned to an investigator. We express such a relationship with the following predicate:

$$\exists \mathbf{r} \in R, \mathbf{r(x,\ y)} \tag{1}$$

It can be simply written as $\exists$**r(x, y)**. To improve the readability and specify which object that a relationship refers to, we can explicitly attach identity information (e.g., ID or name) to entities as shown below.

$$\exists \mathbf{r.id(x.id,\ y.id)} \tag{2}$$

For a relationship with three variables r, x, and y and a given DFKG, if any two variables are bound to concrete information, the unknown variable can be solved in the DFKG. The following formulas represent such cases, where the ? symbol refers to an unknown variable.

$$\exists \mathbf{r.id(?x.id,\ y.id)} \tag{3}$$

$$\exists \mathbf{r.id(x.id,\ ?y.id)} \tag{4}$$

$$\exists ? \mathbf{r.id(x.id,\ y.id)} \tag{5}$$

If **x** and **y** refer to the same object, i.e., **x = y**, then the relationship supports the reasoning about a relation between the object and its property. For example, $\exists ?has(investigator, investigator.name)$ reasons about the name of the investigator. We omit **id** of the first $investigator$ for the purpose of simplicity. All **id**s are omitted in the rest of the paper.

III. Cyber Case Descriptive Objects

A. The Design

CCDOs capture a criminal case's background information, such as the suspects, victims, investigators, and seized raw data [10]. They serve three purposes: (1) showing how each piece of digital forensic evidence is relevant to the criminal case. (2) providing a meaningful and comprehensible context for all stakeholders, including investigators, juries, attorneys, and judges to make informed decisions. (3) providing a guideline and template to describe the context in detail.

CCDOs should provide essential knowledge to answer the following questions:

1) What is the nature of a cybercrime?
2) Who are the suspects?
3) who are victims?
4) Who is assigned to the case?
5) Are there any collected digital forensic artifacts, such as disk images and networking traffic raw data?
6) Where are these raw images acquired from?

The following are sample CCDOs:

- Crime Case Object: A start point of a cybercrime background description. It connects other CCDOs to provide the context of cyber investigations.
- Suspect Object: A person thought to be guilty of cybercrime or offense without certain proof.
- Victim Object: A person attacked, harmed, injured, or killed as a result of a crime.
- Investigator Object: A digital forensic analyst to collect, store, and analyze digital evidence for reconstructing a crime scene
- Computer Object: A computing device that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming. A computer object can be used to present PCs as well as mobile and IoT devices.
- RAM Object: A random access memory, primary storage, that is used to store information for immediate use in a computer or related computer hardware device.
- Secondary Storage Object: A non-volatile and long-term storage, such as a hard disk drive, tape drive, floppy disk, optical disc, or USB flash drive.
- Image Object: A computer file containing the raw contents and structure of a RAM or a secondary storage object.

B. Relationships among CCDOs

We use a set of relationships to describe how CCDOs are connected. Figure 2 shows an example background of a criminal case depicted by CCDOs (shown in solid rectangles)

and their relationships (marked in purple). Note that CCDOs may connect to cyber intelligence. For example, A disk *image* CCDO is a connecting CCDO that may contains multiple files. A file object with a set of predefined metadata has been well-defined by Structured Threat Information Expression (STIX) [11]. A dashed rectangle in the figure indicates an object (e..g, file) that a CCDO connecting to and it is not CCDO.

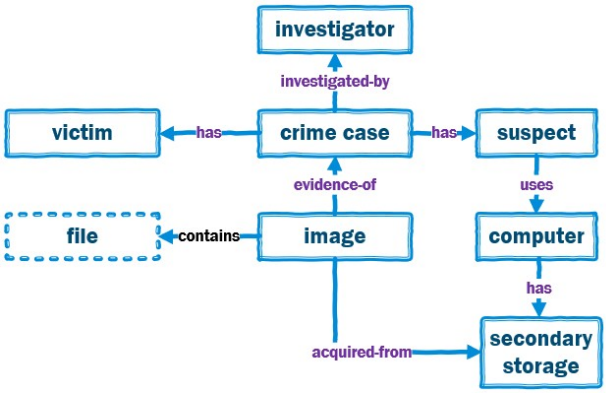

Fig. 2. An example background of a criminal case

The relationships are interpreted as follows:

- *has(crime case, suspect)*: A crime case has a suspect.
- *has(crime case, victim)*: A crime case has a victim.
- *uses(suspect, compute)*: The suspect uses a computer.
- *has(computer, secondary storage)*: The compute has a secondary storage.
- *acquired-from(image, secondary storage)*: A raw image is acquired from the secondary storage device.
- *evidence-of(image, crime case)*: The raw image is a piece of evidence of the criminal case.

Relationships between two CCDOs are utilized to answer questions related to the background of cybercrime cases.

Definition 4. Simple Question:: A simple question is either (a) to retrieve the value of a metadata (i.e., a property p_i) of an object x or (b) to check the existence of a property of an object.

When retrieving metadata information, a simple question is expressed by the predicate:

$$\exists\_(x, ?p_i),\ p_i \in P_x \tag{6}$$

The ? symbol is added to the front of each property to indicate the corresponding value of interest. The _ symbol represents a general *has_a* relationship.

For example, the predicate logic statement $\exists\_(investigator, ?name)$ indicates a question is to ask the name of an investigator for a given $investigator$ object. Without the ? symbol, the statement indicates an investigator has a name. The question to check the existence of a property p_i of an object x is expressed as: $\exists?\_(x, p_i)$.

C. Support for PreDIE

The relationship $investigates$ shown in Figure 2 supports the credibility of evidence with the simplest PRG $\{\exists investigates(investigator, crime\ case)\}$. In addition, the background information is used to prove whether or not a file artifact (shown in a dashed rectangular) is relevant to the criminal case. Specifically, the relationship *contains(image, file)* connects the file of interests to the criminal case. It proves that the file artifact is relevant to the case. Verifying whether an observable object, such as a file, is relevant to a criminal case is equivalent to finding all paths between these two nodes in a DFKG. Algorithm 1 provides a general solution to calculate all paths with Depth First Search (DFS). Each path can be viewed as is a PRG.

Algorithm 1: Find All Paths Between Two Nodes in a DFKG

Input: G: A DFKG
u: A starting node
v: A ending node
Output: allPaths: A list of paths from u to v and each path is a set of relationships

for $r \in R$ **do**
  /* compute inverse relationships in G */
  $r \leftarrow inverse(r)$
/* compute all paths utilizing depth first search */
$allPaths \leftarrow DFS(G, u, v)$

IV. Cyber Forensic Observable Objects

A. The Design

CFOOs are the smallest identifiable and comprehensible artifacts generated from Information Technology (IT) infrastructures and devices using the infrastructures. Figure 3 shows a typical IT infrastructure and exemplary devices that generate CFOOs. IT infrastructures and devices often include computers, servers, network routers, switches, smartphones, and IoT devices. Artifacts generated from these devices include files, directories, emails, IP addresses, MAC addresses, URLs, etc.

CFOOs are categorized based on two perspectives of cyber investigations: cyber threat intelligence group and digital forensics intelligence group. Observable objects in cyber threat intelligence have been included in STIX. Observable objects in digital forensic intelligence focus on describing new intelligence in the field of forensic investigations.

The new proposed observable objects in the digital forensics intelligence group are described as follows:

- Disk Partition Object: Refers to as the creation of one or more regions on secondary storage, so that each region can be managed separately. A Disk Partition object specifies the properties that are associated with the disk segment.
- Plug and Play (PnP) Event Object: An event recorded by Windows Kernel-Mode Plug (pnp) and Play Manager. PnP manager is a combination of hardware technology

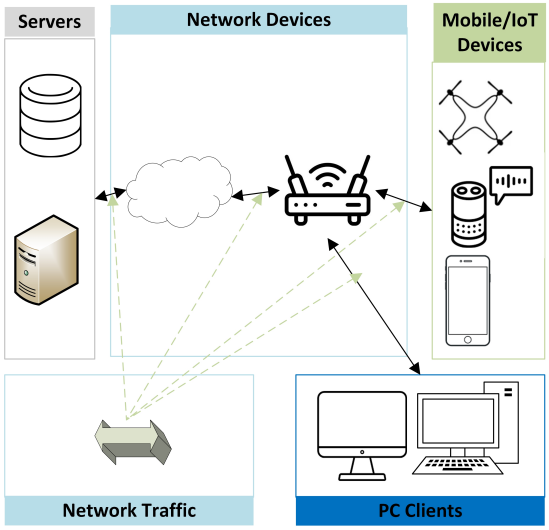

Fig. 3. A typical IT infrastructure and its components.

and software techniques that enables a PC to recognize when a device is added to the system.

- Windows Event Object: Properties of an event, which is recorded by Windows Operating System. Windows Event is triggered by user or software behaviors. It excludes PnP Event Object.
- Webpage Object: A webpage that was opened in a browser.
- Webpage Visit Object: A visit to a webpage.
- Cloud Storage Object: A cloud space to store data.
- File Visit Object: A file or directory visit performed by operating systems or applications. The basic operation of the visit to the file can be read, write, modify, update, execute, and delete. The visit may be saved in different forms, e.g., file, cache, Windows registry, etc. Note that one user's action may trigger one or multiple basic operations.

B. The Importance of File Visit Object

Both Unix-based and modern Windows-based operating systems implement "everything is a file" philosophy, to some extent [12]. From the forensic perspective, anything that happens to a file, e.g., read and write, will leave something somewhere in the file system.

The design philosophy certainly makes *File Visit Object* is the most important CFOO. *File Visit Object* records forensic intelligence that could assist investigators to reconstruct a suspect's behaviors in detail, including when, why, how a suspect visited a file. For example, *File Visit Object* contains *File Visit Common Name Vocabulary* to classify digital forensic artifacts based on different purposes of visits. The list of *File Visit Common Name Vocabulary* is described as follows:

- userassist: Refers to as digital forensic artifacts that track every GUI-based program launched from the desktop, Artifacts are saved in the userassist registry key.
- shimcache: It can be used for identifying applications' execution status. Originally, it was created to identify application compatibility issues.
- recentfilecache: Only contains references to programs that recently executed,
- prefetch: Traces most frequently used software because these software needs to be preloaded into memory to improve system performance.
- muicache: Traces recently execute software using the multiple languages supporting feature of Windows OS.
- usnjournal: Traces files changes, including file addition, deletion, modification, etc.
- shellbag: Traces folder's visiting information. Originally, shellbag was used to store user preferences for GUI folder display within Windows Explorer.
- mru: Traces most recently used files.
- mft: Master file table for file management. Traces file Creation, modification, deletion, moving, etc.

C. Relationships

Figure 4 shows relationships among three CFOOs and one relationship that connects CFOO and CCDOs. The relationships (marked in purple in the figure) are described below:

- *file_visited(file visit, file)*: A file visit object contains a reference to a visited file. A relationship label contains a "_" symbol indicates that it is an embedded relationship.
- *contains(disk partition, file)*: A disk partition contains the file. Note that there is an inverse relationship (indicated by a dashed arrow) that is labeled as *part-of*. A relationship labeled with a " – " symbol indicates it is a relationship object.
- *has(file visit, reason)*, *has(file visit, common name)*, *has(file visit, operation)*: Four example proprieties of a file visit object, including the reason that a file was visited, the type of the artifacts commonly referred to by digital forensics communities, and what operation was applied to the file.

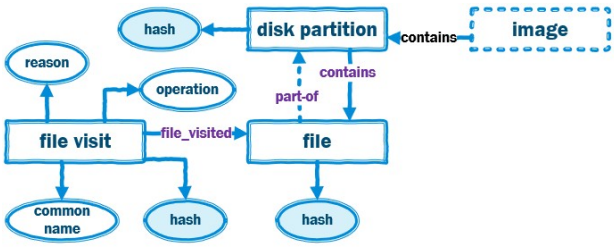

Fig. 4. Relationships between three CFOOs and one relationship that connects a CFOO and a CCDO

The relationship *contains(image, disk partition)* connects a CFOO and a CCDO. It indicates that a raw disk image contains a disk partition.

D. Support for PreDIE

The DFKG in Figure 4 supports integrity and authenticity of PreDIE. The integrity of digital forensic evidence is "the

property whereby digital data has not been altered in an unauthorized manner" [13]. A hashcode of a CFOO proves the integrity of evidence. Each CFOO in Figure 4 has a hashcode marked as a solid ellipse. The reasoning about the integrity can be expressed as the predicate $\exists\_(x, x.hashcode)$.

To prove the authenticity of an observable object, we need to trace the source of the object. Specifically, we can (1) reuse the algorithm 1 to compute paths between a CFOO and a raw image. (2) demonstrate that a collection of relationships in each path must be a generalized *part-of* relationship, including *has, contains, extracted-from*, etc. For example, The *file* object in Figure 4 has been authenticated because there is path from the *file* to a raw disk image and all relationships in the path is a type of *part-of* relation.

V. Cyber Forensic Domain Objects

A. The Design

CFDOs focus on investigation procedures and specify how criminal activities are reconstructed, what evidence is associated with these activities, and how evidence is observed and identified. Similar to CFOOs, CFDOs fall into two groups: cyber threat intelligence group and digital forensics intelligence group.

CFDOs in the cyber intelligence group, called STIX domain objects (SDOs), have been well-defined by STIX [11]. SDOs "can create and share broad and comprehensive cyber threat intelligence". For example, SDOs include *indicator object* to define a pattern that can be used to detect suspicious or malicious cyber activity; *observed object* conveys information about cybersecurity-related entities such as files, systems, and networks.

CFDOs in the digital forensics intelligence group to capture possible actions that a suspect has performed. Actions are determined by PreDIE and organized in a timeline. CFDOs in the group is described as follows:

- Timeline Object: It provides a chronological-based reporting format to describe how reconstructed criminal activities are organized. Formally, a timeline $T =< o_1, \ldots, o_n : n \in \mathbb{N} >$, where o_i is an illegal operation. For example, a sequence of illegal operations in terms of a timeline can be depicted as $< search\_for\_an\_illegal\_image, download\_the\_image >$.
- Action Object: One cybercriminal activity performed by threat actors. An action object captures a meaningful ACID (Atomicity, Consistency, Isolation, Durability) activity applied to file systems or hardware components. The name of the action is a verb chosen from a predefined but open-to-extend verb list, called *Action Name Open Vocabulary*, In the aforementioned illegal image possession case, two action objects are *search* and *download*. With the definition of action object, the $operation$ defined earlier can be further refined as a 2-tuple $< download, file >$, i.e., an action object and the artifact that the action applies to. Formally, an operation $o \in \{< c, a >: c \in C, t \in A\}$, where C is the *Action Name Open Vocabulary* and A is collection of artifacts.
- Investigation Tool Object: Refers to as software or a hardware tool that is used by investigators to perform digital forensic investigations on artifacts, files, etc.

B. Relationships

Figure 5 demonstrates how CFDOs (shown in solid rectangles) collaborate with CCDOs and CFOOs (shown in dashed rectangles) to describe a reconstructed timeline with two actions. The semantics of the graph is expressed in the following relationships:

- *reconstructed-from(timeline, crime case)*: A timeline is reconstructed from a criminal case.
- *action_ref(timeline, actionX)*: A timeline contains an *actionX*. *actionX* represents either *action1* or *action2*. Note that the predicate *action_ref* is an embedded relationship that references to the *objectX action*.
- *targets(action1, file)*: The *action1* targets on an observable object, i.e., a file object. Note that other relationships and objects that reference to *action2* are omitted and marked as a symbol ...for simplicity.
- *indicated-by(indicator, action1)*: An indicator object contains a pattern that is used to indicate *action1* has been observed.
- *based-on(indicator, observed data)*: The pattern in the indicator applies to observed data to prove that an action has been performed.
- *object(observed data, file visit)*: The observed data reference to observable objects, which is a *file visit* object in the example.
- *object(observed data, file)*: The observed data reference to observable objects, which is a *file* object in the example.
- *input(investigation tool, file)*: The *file* object is an input of an investigation tool.
- *output(investigation tool, file visit)*: The *file visit* object is an output of an investigation tool.
- *file_visited(file visit, file)*: The *file visit* object contains a *file_visited* reference to the visited file.

C. Reasoning about PreDIE

The reconstructed timeline in Figure 2 is a DFKG that supports validity and relevance of PreDIE.

Since *observed data* object only contains references to observable data, we need to approve that each CFOO to that an observed data reference is valid. For example, the *file visit* object is one of the observed data. To prove the validity of the *file visit* object, the DFKG includes an investigation tool that is used for producing the *file visit* object. The *investigation tool* object specifies various properties, including name, version, inputs, and outputs, to ensure its validity. Only valid tools can produce valid results. Formally, the reasoning about the validity of the *file visit* object is expressed as the predicate $\exists output(?investigation\ tool, file\ visit)$.

For each CFOO that is associated with *observed data*, we need to prove its relevance to crime activities. For example, to prove a *file* object is relevance to crime activities, we can

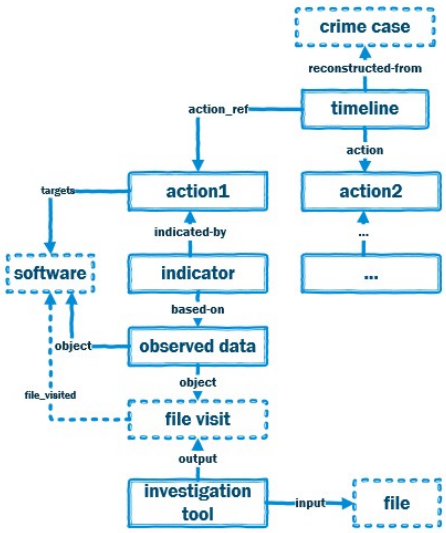

Fig. 5. A reconstructed timeline with two actions.

adapt the algorithm 1 with the input *(G=Figure 5 , u=file, v=action)* to verify the relevance of the *file* evidence.

One of the most important applications of a DFKG is to generate a reconstructed timeline of a crime scene for different types of clients. It can be achieved by retrieving a PRG of the DFKG. For example, after a timeline of the crime scene was recreated by digital forensic investigators, judges and juries may only want to review a sequence of actions performed by a suspect in chronological order. Let's assume Figure 5 describes a partial crime case scenario in which a suspect downloaded (i.e., *action1*) and installed (i.e.,*action2*) anti-forensics software, i.e., ccleaner. The corresponding timeline, that is indicated by the relationship *reconstructed-from(timeline, crime case)*, is shown in Table I. The two actions, download and install, are the names of the two actions. The timestamp shows when an action was performed and the description summarizes each activity of the suspect to improve readability. The name, timestamp, and description are attributes of the action object. A DFKG with PreDIE requires each action has a target on which the action applies. The *targets* relationship retrieves the target, which is the ccleaner software. Note that a timeline table is a different representation of a PRG. Algorithm 2 describes how a PRG is retrieved from a DFKG to generate a timeline.

TABLE I
A RECONSTRUCTED TIMELINE OF INSTALLING ANTI-FORENSICS SOFTWARE IN TABLE

id	Action	Target	Timestamp	Description
1	download	ccleaner	2016-01-20T12:31:00Z	download ccleaner software.
2	install	ccleaner	2016-01-20T12:40:00Z	install ccleaner software.

Algorithm 2: Retrieving a PRG from a DFKG to generate a timeline

Input: G: A DFKG
Output: A PRG for generating timeline report

```
/* find a timeline object for a given
   crime case                                  */
t← timeline(?timeline, crimecase)
/* find a list of action objects for
   a timeline t                                */
actionList ← action_ref(t, ?action)
/* for each action a, print
   action-related information                  */
for a ∈ actionList do
    print a.name
    /* find the action target, which is
       a software object s                     */
    s ← targets(a, ?target)
    print s.name
    print a.timestamp
    print a.description
```

VI. A CASE STUDY

The case study demonstrates how DFKG can be applied to capture the reconstructed scenario and make evidence presentable in courts.

A. Case Background

The illegal image possession case was contributed by Dr. Golden G. Richard III, and was originally used in the Digital Forensic Research Workshop 2005 RODEO Challenge [14]. The scenario is summarized as follows: *The city of New Orleans passed a law in 2004 making possession of nine or more unique rhinoceros images a serious crime. The network administrator at the University of New Orleans recently alerted police when his instance of RHINOVORE flagged illegal rhino traffic. Evidence in the case includes (1) Three network traces (log files) and (2) a USB key seized from one of the University's labs (DD image).*

B. Reconstructed Case Scenario in DFKG

We have investigated the case in a Kali virtual machine [15], There are complex processes and various open-source tools used to extract potential evidence. The major investigation activities include file recovering, stenography, password cracking, and FTP/HTTP traffic analysis. We have taken snapshots of the complete investigation processes and published them online for verification purpose[1].

Figure 6 shows the reconstructed case in DFKG. We outline the semantics of the DFKG from three different perspectives, including case background, a reconstructed timeline, and the supporting evidence of the timeline:

[1]https://github.com/frankwxu/digital-forensics-lab/tree/main/Illegal_Possession_Images

- The object *x-crime-case* is the center of any DFKG. Note that we add the letter **x** to the name of an object to distinguish digital forensic intelligence and cyber threat intelligence. In addition, we use the symbol **-** instead of spaces to improve readability.
- A set of CCDOs along with *x-crime-case* describe the background of the criminal case, includes four log files *rhino.log*, *rhino2.log*, *rhino3.log*, *Rhino Hunt.pdf* and one USB image acquired from the crime scene.
- A set of CFDOs indicates the reconstructed timeline, expressed by *x-timeline*, consists of four actions performed by a suspect, such as download, hiding, and deleting files/images.
- A set of CFOOs describes the objects of actions and how these objects are observed and related to the timeline and case. For example, there are four images (grouped and shown in the up-left corner of the DFKG) are deleted from a USB drive. including *f0106393.jpg*, *f0106409.jpg*, *f0106865.jpg*, and *f0106889.jpg*.

C. Support for PreDIE

The DFKG meets PreDIE as follows:

- Authenticity: Figure 6 shows that four photos have been deleted from the given USB image. To demonstrate the provenance of the four deleted photos, we show the reconstructed PRG from DFKG as followings:
  PRG={*output(file.“f0106393.jpg”, x-investigation-tool.“PhotoRec7.1”), output(file.“f0106409.jpg”, x-investigation-tool.“PhotoRec7.1”), output(file.“f0106865.jpg”, x-investigation-tool.“PhotoRec7.1”), output(file.“f010889”, x-investigation-tool.“PhotoRec7.1”), inputf(x-investigation-tool.“PhotoRec7.1”), x-image.“usb”*}.
- Integrity: The DFKG is implemented in a JSON file [2]. Each CFOO in the DFKG contains a hashcode to prove the integrity of evidence as we have discussed earlier. For example, the predicate $\exists\_(file.\text{“}f0106865.jpg\text{”}, file.hash.\text{“}7a74\text{”})$ shows the file has a hashcode.
- Validity: Each deleted image is associated with a photo recovering tool named *PhoteRec7.*, e.g., $\exists output(file.\text{“}f0106865.jpg\text{”}, x - investigation - tool.\text{“}PhotoRec7.1\text{”})$.
- Credibility: An PRG is reconstructed from DFKG to prove the credibility of evidence. The following path, i.e., a set of predicates, indicates that a recovered photo evidence *file=“f0106865.jpg* is recovered and verified by an investigator.
  {*object(observed-data.“deleted images”, file.“f0106865.jpg”, based-on(indicator“delete image indicator”, observed-data.“deleted images”), indicated-by(x-action.“delete images”, indicator.“delete image indicator”, object(x-timeline.“illegal possession”, x-action.“delete images”), reconstructed-by(x-timeline.“illegal possession”, x-investigator.“Frank Xu”))*}.
- Relevance: A DFKG supports the relevance of observed data by requiring each action referencing to a target. For example, $\exists\ target(x - action.\text{“}delete\ images\text{”}, file.\text{“}f0106865.jpg\text{”})$.

D. Question Answering

Because of the support for PreDIE, a DFKG can be utilized to answer questions related to the criteria of PreDIE. For example, to verify the integrity of a file, the simple question $\exists\_(file.\text{“}f0106393.jpg\text{”}, ?file.hashcode)$ asks “what is the hashcode of file *f0106865.jpg*”.

A complex question consists of multiple simple questions and therefore may involve multiple objects. We first need to decompose a complex question into multiple simple questions and then express each simple question as predicates. For example, one may ask “Does the recovered image *f0106865.jpg* have any credibility?” It really asks a few simple questions based on the corresponding PRG extracted from a given DFKG. The list of simple questions that the PRG described includes:

- Is the evidence a target of action?
- Is the action a part of the reconstructed timeline?
- Is there anyone working on the timeline?
- Who did reconstruct the timeline?

The complexity question is formalized as the simply question set: $\{\exists?target(x - action, file.\text{“}f0106865.jpg\text{”}\text{”}) \wedge \exists?action(x - timeline, x - action) \wedge \exists?reconstructed - by(x - timeline, x - investigator) \wedge \exists has(x - investigator, ?x - investigator.name)\}$.

Note that there are possible different ways to decompose the complex question because there are different paths from the *file* object to the *investigator* object for a given DFKG.

VII. Related Works

The formalization of digital forensic intelligence, as a subfield of data and behavioral modeling, has drawn attention in the cybersecurity communities. Various approaches and models have been proposed to formalize evidence and represent digital forensic knowledge. [16] and [17] have proposed the concept of model formalization for analyzing and constructing forensic procedures at a high level, showing the advantages of formalization. Their model procedure consists of attack, operating system, and forensic action. [18] has proposed a systematic approach to reconstructing attack scenes based on a forensic evidence acquisition model. The model they have built is a type of special attack tree from which digital forensic examiners can collect corresponding forensic evidence based on the leaf nodes. The tree-like model is formalized but the evidence itself. [19] and [20] have proposed a finite state machine approach to digital event reconstruction. While the former generalizes evidence and events as an abstract concept for modeling, the latter proposes a layered structure to organize

[2]https://github.com/frankwxu/digital-forensics-lab/blob/main/STIX_for_digital_forensics/Illegal_Possession_Images/illegal_possession_image.json

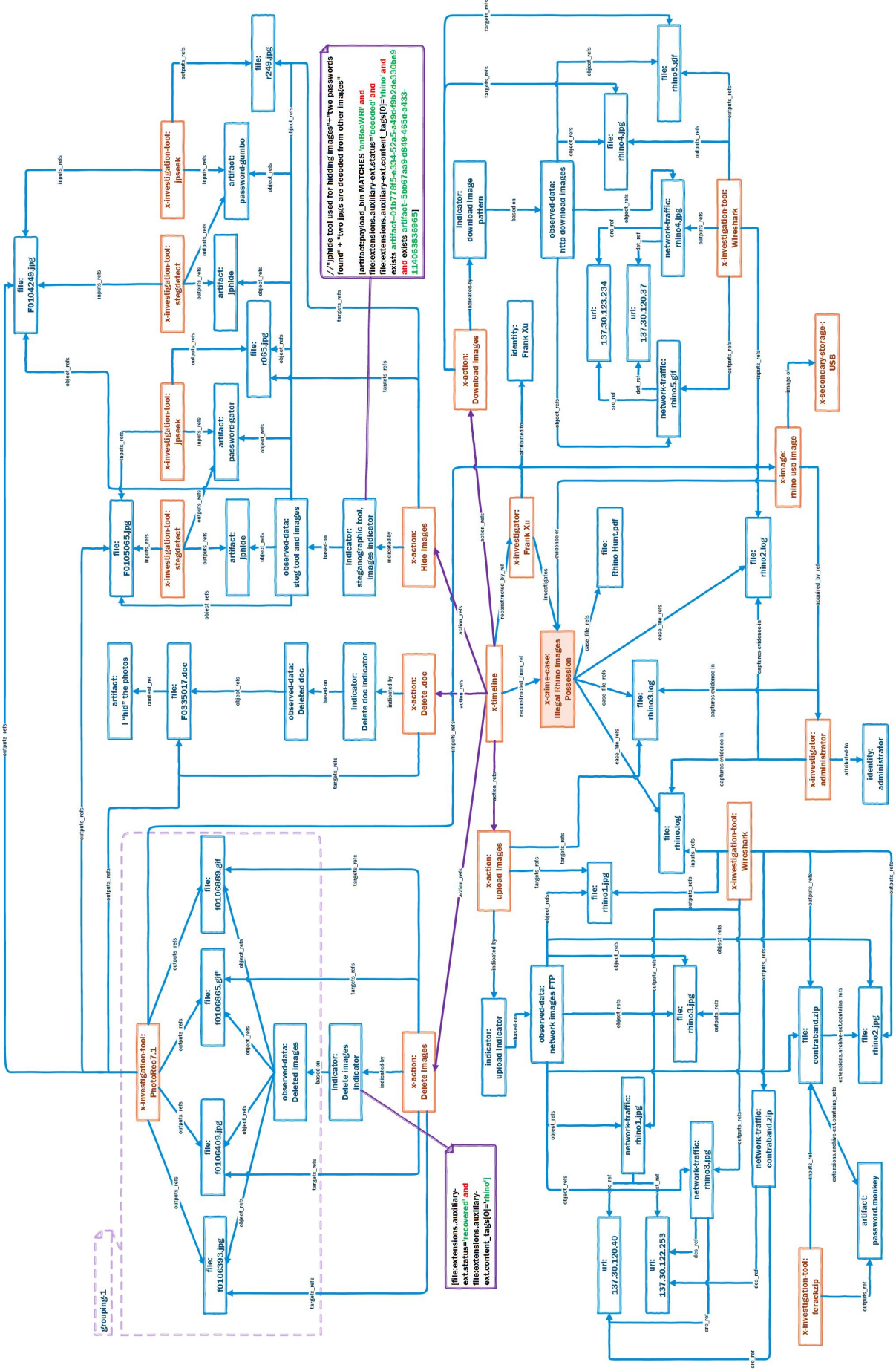

Fig. 6. The reconstructed illegal image possession crime case expressed in DFKG.

forensic evidence objects, including a cybercrimes layer, a forensic evidence generator layer, an evidence category layer, and an evidence object layer. [21] has proposed a formalized knowledge representation model for digital forensics timeline analysis. The knowledge model depicts cyber incidents with subject, object, event, and footprint. The model monitors the object creation, modification, suppression, and utilization.

Besides these efforts, several efforts to formalize evidence to promote evidence sharing. [22] is one of the early theoretical works that has proposed a forensic integration architecture framework for evidence sharing and integration. It abstracts the evidence source and storage format information from digital evidence and explores the concept of integrating evidence information from multiple sources. [11] [23] is the state-of-art framework designed for formalize and share cyber intelligence. However, it mainly focuses on sharing cyber threat intelligence, not forensic intelligence.

It is worth mentioning there is a general framework for representing interconnected data on the web, Resource Description Framework (RDF) [24] [25]. RDF statements, along with Web Ontology Language (OWL) [26], are used for describing and exchanging metadata in a standardized way. Entities and relations in RDF can be used to construct an RDF graph that shows how those entities are related. RDF fits for capturing static information on the web and, however, it lacks the infrastructure to support the description of activities and processes, such as capturing criminal activities and investigation of processes information. In other words, RDF is only a general version of CFOO. Digital forensic communities need a domain-specific framework with predefined objects to present digital forensic evidence.

VIII. Conclusions

We have presented DFKG for formalizing digital forensic intelligence to address the challenge of making digital evidence presentable. The graph structure allows digital forensic professionals to put evidence information in context via linking and semantic metadata of evidence for better visualization. Semantic metadata of evidence and relationships among evidence support the presentable criteria of digital forensic evidence, including authenticity, integrity, validity, credibility, and relevance. Nodes (i.e., objects) in DFKG are grouped into three categories to help various audiences to comprehend semantics and improve the usability of DFKG: CCDOs for describing background information of a given case, CFOOs for capturing metadata of observable objects, and CCDOs tell a story in terms of a timeline from the perspective of digital investigations. PRG provides a general approach for reasoning digital forensics intelligence and answering forensic-related questions.

Future works will focus on discovering new relationships among CFOOs with Artificial Intelligence (AI), e.g., graph neural networks. It is also interesting to leverage AI to verify the consistency of evidence and predicting missing evidence.

References

[1] A. Watt, "The challenges of interpreting digital evidence in the courtroom," *Precedent (Sydney, NSW)*, no. 139, pp. 43–46, 2017.

[2] A. Eriksson, "Presenting evidence in court–some fundamental problems to be considered," in *Comunicación presentada en el Congreso de la International Association for Forensic Phonetics and Acoustics*, 2011.

[3] M. McCormick, "Scientific evidence: Defining a new approach to admissibility," *Iowa L. Rev.*, vol. 67, p. 879, 1981.

[4] Law, "Legal dictionary - law.com," https://dictionary.law.com/Default.aspx?selected=2339 , 2022, accessed: 20 January 2022.

[5] G. Palmer *et al.*, "A road map for digital forensic research," in *First digital forensic research workshop, utica, new york*, 2001, pp. 27–30.

[6] C. W. Sanchirico, "Evidence tampering," *Duke LJ*, vol. 53, p. 1215, 2003.

[7] U. Courts, "Federal rules of evidence," https://www.uscourts.gov/sites/default/files/Rules%20of%20Evidence., 2007, accessed: 26 January 2022.

[8] E. W. Schneider, "Course modularization applied: The interface system and its implications for sequence control and data analysis." 1973.

[9] A. Singhal, "Introducing the knowledge graph: things, not strings," https://blog.google/products/search/introducing-knowledge-graph-things-not/ , 2012, accessed: 21 January 2022.

[10] R. Gehl and D. Plecas, *Introduction to criminal investigation: processes, practices and thinking*. Justice Institute of British Columbia, 2016.

[11] Oasis, "Stix version 2.1," https://docs.oasis-open.org/cti/stix/v2.1/os/stix-v2.1-os.html, 2021, accessed: 26 January 2022.

[12] L. Torvalds, "The everything-is-a-file principle," https://yarchive.net/comp/linux/everything_is_file.html, 2007, accessed: 25 January 2022.

[13] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, "Handbook of applied cryptography crc press," *Boca Raton*, 1997.

[14] G. Richard, "Rhino hunt," https://cfreds-archive.nist.gov/dfrws/Rhino_Hunt.html, 2005, accessed: 2022-01-02.

[15] Kali, "Kali linux," https://https://www.kali.org/, 2022, accessed: 2022-02-10.

[16] P. Stephenson, "Using a formalized approach to digital investigation," *Computer Fraud & Security*, vol. 7, pp. 17–20, 2003.

[17] R. Leigland and A. W. Krings, "A formalization of digital forensics," *International Journal of Digital Evidence*, vol. 3, no. 2, pp. 1–32, 2004.

[18] W. Xu, J. Yan, and H. Chi, "A forensic evidence acquisition model for data leakage attacks," in *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2019, pp. 53–58.

[19] P. Gladyshev and A. Patel, "Finite state machine approach to digital event reconstruction," *Digital Investigation*, vol. 1, no. 2, pp. 130–149, 2004.

[20] W. Xu, J. Yan, and D. Stone, "A collaborative forensic framework for detecting advanced persistent threats," in *The 33th International Conference on Software Engineering and Knowledge Engineering*, 2021, pp. 67–73.

[21] Y. Chabot, A. Bertaux, C. Nicolle, and T. Kechadi, "A Complete Formalized Knowledge Representation Model for Advanced Digital Forensics Timeline Analysis," in *Fourteenth Annual DFRWS Conference*, ser. Digital Investigation, vol. 11, no. 2. Denver, United States: Elsevier, Aug. 2014, pp. S95–S105. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01199449

[22] S. Raghavan, A. Clark, and G. Mohay, "Fia: an open forensic integration architecture for composing digital evidence," in *International Conference on Forensics in Telecommunications, Information, and Multimedia*. Springer, 2009, pp. 83–94.

[23] Oasis, "Taxii version 2.1," https://https://docs.oasis-open.org/cti/taxii/v2.1/os/taxii-v2.1-os.html, 2021, accessed: 12 February 2022.

[24] E. Miller, "An introduction to the resource description framework." *D-lib Magazine*, 1998.

[25] O. Lassila, R. R. Swick *et al.*, "Resource description framework (rdf) model and syntax specification," 1998.

[26] D. L. McGuinness, F. Van Harmelen *et al.*, "Owl web ontology language overview," *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.