International Journal of Information Security (2025) 24:81
https://doi.org/10.1007/s10207-025-00991-8

REGULAR CONTRIBUTION

WEFT: a consistent and tamper-proof methodology for acquisition of automatically verifiable forensic web evidence

Alessandro Cantelli-Forti[1] · Giacomo Longo[2] · Francesco Lupia[3] · Enrico Russo[2]

Abstract

The pervasiveness of web-based services has further complicated forensic operations, as traditional acquisition techniques do not fit with the volatile nature of online evidence. The current best practice often suffer from procedural shortcomings and are prone to tampering, which can lead to the dismissal of evidence in legal contexts. This paper introduces an acquisition methodology aimed at improving the integrity and admissibility of digital evidence acquired from live web environments. Our proposed approach addresses these issues and adheres to the requirement of international standards by establishing a unified format as a single source of truth, secure timestamping, and enabling automatic verification of integrity and its content, thereby offering more transparency to the involved parties. An extensive evaluation with live acquisition of the top 100 most popular websites indicates that the methodology produces an artifact comparable to state-of-the-art tools with added benefits.

Keywords Forensic web evidence · Digital evidence integrity · Tamper-proof acquisition · Secure timestamping · Automatic verification · Web forensics methodology

1 Introduction

Digital Forensics (DF) has evolved from a relatively niche skill to a relevant study area within the current research landscape [5]. The DF market is expected to grow significantly, with projections estimating a market size of approximately $42.52 billion within the next ten years, at a Compound Annual Growth Rate (CAGR) of 15.9% [1]. However, this growth comes with significant technological and legal challenges [18], which are anticipated to gradually bring an end to the *golden age* of DF. A prominent complexity is due to the growing trend in digital life to shift towards web-based services, online communications, and e-commerce transactions, making web-collected evidence increasingly decisive in legal proceedings and contexts where DF is applied.

Evidence collection generally relies on widely recognized standards and best practices established within the legal-tech field, where technology intersects with legal procedures. Experts developed these procedures during an era when the primary sources of evidence included hard drives and removable media such as USB drives, CD-ROMs, tapes, and floppy disks [23]. Data stored on these devices can be physically handled, and they are seized to be used as evidence. The process, known as "post-mortem" forensic copying, involves duplicating data bit-by-bit from powered-off devices - effectively "dead" - in a controlled environment. The result is a single, unaltered artifact that keeps the precise state of the data at the time of acquisition. This procedure ensures the preservation of date, time, and data *integrity*, documented in a detailed and legally admissible report that includes the hash of the artifact. The artifact is further safeguarded through hash calculations that allow *reproducibility*, enabling verification at a later stage.

Given that the acquisition of web evidence typically occurs without direct access to the servers hosting it and happens as "live" forensic copying, it is evident that traditional post-mortem techniques are inadequate to guarantee the same principles of integrity and reproducibility.

✉ Giacomo Longo
giacomo.longo@dibris.unige.it

Alessandro Cantelli-Forti
alessandro.cantelli.forti@cnit.it

Francesco Lupia
francesco.lupia@unical.it

Enrico Russo
enrico.russo@unige.it

[1] RaSS National Laboratory, CNIT, Pisa, Italy

[2] DIBRIS, University of Genova, Genova, Italy

[3] DIMES, University of Calabria, Rende, Italy

Published online: 18 February 2025

Web sources are composed of diverse elements, including static resources, dynamic user interactions, and rendered content, all of which contribute to a complex and interconnected environment. By its nature, web-based content is highly dynamic, with evidence capable of rapidly changing or disappearing. Ensuring integrity involves safeguarding artifacts against tampering throughout the live acquisition process, including providing reliable timestamps and maintaining a Single Source of Truth (SSOT).. Furthermore, ensuring reproducibility necessitates the implementation of robust mechanisms for self-verification and correlation of diverse evidence sources, which must address the intricate dependencies between the diverse elements.

These complexities are well-documented in the literature [11], and several studies have highlighted the limitations of traditional acquisition techniques [6, 7, 31, 45]. Such shortcomings can result in the disqualification of crucial evidence, even when not directly related to the central issues at hand.

To address these challenges, this paper proposes a methodology designed to produce a single, tamper-resistant, and verifiable artifact that overcomes the inherent limitations of current approaches. It focuses on refining DF acquisition procedures to account for the dynamic nature of web environments and the complexities associated with accurate timestamping. By ensuring the reliability and trustworthiness of digital evidence from live web sources, it guarantees integrity throughout the acquisition process. Additionally, it enhances reproducibility by introducing automated mechanisms to assess artifacts, improving process transparency and enabling the opposing party to verify the evidence independently.

The main contributions of the paper can be summarized as follows.

- We discuss the limitations of the current best practice approach against the requirements of international standards, emphasizing the inherent challenges in ensuring integrity and reproducibility.
- We propose WEb Forensics Toolkit (WEFT), a solution to forensically acquire online web sources, producing a single, tamper-resistant, and verifiable evidence capable of overcoming the above challenges.
- We extensively assess our approach by acquiring multiple real and highly visited websites, followed by a comparative analysis with a leading state-of-the-art tool.

A preliminary version of this work appeared in [3]. The present paper extends the previous proposal in several ways. We conduct a more comprehensive analysis of the limitations inherent in current web forensic acquisition methods through a detailed comparison with the requirements of international standards. We provide a complete presentation of our methodology, including a novel improvement that ensures greater security and integrity of the acquisition timeline through a keepalive generator component. Finally, we propose an enhanced experimental comparison with the current state-of-the-art, which has provided us with deeper insights into the effectiveness and applicability of our methodology.

Paper Structure. The paper is structured as follows. Section 2 describes the current best practice approach for live web forensics acquisition, the limitations, and inherent challenges. Section 3 introduces our methodology, which addresses these limitations and takes the open challenges into consideration Sect. 4 illustrates experimental results on live acquisitions, including a comparison with a state-of-the-art tool. Finally, in Sect. 5, we compare with related work, and Sect. 6 draws the conclusions.

2 Live Web forensics acquisition

In this section, we introduce the best practice approach for live web forensics acquisition. Then, we compare this approach against international standards and discuss the current limitations and inherent challenges.

2.1 Overview

Conducting a live web forensics acquisition involves a specific scenario of collecting Live Network Evidence (LNE),, as outlined in the general definition by Nikkel [31] and further refined characterization by Castiglione et al. [7].

Unlike traditional digital evidence, LNEs are not physically stored on a device accessible to the investigator but can be accessed through a computer network. The access method follows the *client–server model*: a server stores the evidence, and an investigator, like users, interacts with the server and gathers the evidence using a compatible client and a standard protocol. Web forensics specifically relates to web servers, web browsers as clients, and HTTP as the standard protocol. The core evidence is web pages and all the artifacts to which they refer, e.g., images, videos, or scripts, but more generally, any object that can be retrieved using HTTP.

As web forensics deals with LNE, collected artifacts are inherently *dynamic*. This dynamism means that web content, whether generated or static, can change over time—dynamic pages vary with each visit, e.g., depending on the client configuration or user parameters, and even static pages can be updated or modified on the server. This dynamism is analogous to the concept of *volatility* in memory acquisition, where consistency problems can arise due to the ephemeral nature of acquired data [33, 34]. Proper handling of temporal information during acquisition becomes crucial, as it enables reliability assessment of collected data and helps identify potential inconsistencies in artifacts captured across different time points [35]. Therefore, timely and accurate capture is essential for forensic integrity.

Despite the ubiquity of the World Wide Web and the rising incidents requiring forensic activities, there is no established procedure universally recognized by courts for conducting web forensic investigations. Digital investigators currently rely on best practices that leverage state-of-the-art tools and techniques.

Regarding web forensics and the aforementioned best practices, this paper considers the following conditions inherent to the investigation process.

- We assume investigators' access to the web server is limited to HTTP,[1] and they proceed with acquisition rather than collection, considering prevailing constraints and investigative trade-offs. Specifically, they are unable to physically access it, e.g., to obtain logs or access the sources of dynamic web pages.
- We assume that the preparation and planning phase to determine the scope and objective of the investigation has already been conducted. It is worth noting that preparation and planning could be complicated by potential client-side dependencies or by explicitly rogue websites and anti-forensics techniques [24]. While overcoming these complexities remains a research challenge [29], our solution focuses on the subsequent phases of the investigation and, therefore, does not address this aspect. As a result, investigators have identified the target website(s) and know how to obtain the specific data of interest (e.g., web pages or user interactions).

In the following, we detail the current approach investigators adopt for live web forensics acquisitions and the challenges and inherent limitations it presents.

2.2 Best-practice approach

As previously introduced, live web forensics acquisition follows a best-practice approach. This approach has been consolidated through scientific research [7], professional settings [13], and within both open-source [53] and commercial tools [4, 28]. It relies on the guidelines for specific activities in handling potential digital evidence provided by the International Standard ISO 27037:2012 [23].

In particular, the standard focuses on four activities: (i) identification, (ii) collection, (iii) acquisition, and (iv) preservation. It is important to note that the collection phase primarily involves gathering physical items containing potential digital evidence. As specified in Sect. 2.1, investigators can rely only on the HTTP protocol to collect the digital content without being able to access any related physical resources, making the collection activity impracticable in this context.

Below, we delineate how the best-practice approach aligns with the guidelines for the three remaining activities.

2.2.1 Identification

According to ISO 27037:2012, *identification* involves recognizing and labeling potential digital evidence that might be relevant to the investigation. This critical step establishes a foundation for subsequent forensic activities by guaranteeing all potential evidence is accounted for and prepared for further processing.

Investigators execute the activities established in the preparation and planning phase using a dedicated solution for live acquisitions. This solution provides of a preconfigured environment, namely the Acquisition Environment (AE), with at least (i) an operating system, (ii) a web browser configured to export the keys, namely *SSLKEYLOGFILE* [46], used for encrypting TLS/SSL sessions, (iii) software for network traffic capture, and (iv) tools to record the audio and video of remote desktop activities. A typical configuration consists of a virtual machine hosting all the aforementioned facilities and accessible via remote desktop access, such as Remote Desktop Protocol (RDP) or Virtual Network Computing (VNC). If a web acquisition is required, deploying a new virtual machine instance enables investigators to perform it in a clean, ready-to-use environment, minimizing contamination risks.

It is worth noting that the AE ensures that all content displayed and interacted with during the use of the web browser is identifiable and ready for acquisition, thereby adhering to the best practices of the standard.

2.2.2 Acquisition

ISO 27037:2012 defines acquisition as the process of producing a digital evidence copy and documenting the methods used and activities performed. When it is not feasible or permissible to create a digital evidence copy of an evidence source, i.e., a physical acquisition, the standard admits to performing a *logical acquisition*, which targets only *specific data types, directories, or locations*.

HTTP-based acquisition, as assumed in our methodology, naturally falls under the category of logical acquisition. The scope of this process is inherently constrained by what the system is configured or permitted to expose through the HTTP protocol, limiting the investigator to the accessible data and resources at the time of acquisition.

In the best-practice approach, the acquisition process involves accessing target websites using the investigator's designated web browser provided by the AE. Simultaneously,

[1] Throughout the article, "HTTP" refers to the application-level protocol, no matter the underlying transport or version.

the AE records the audio-video stream, network traffic, and user interactions (keystrokes, mouse events), in accordance with the standard's requirement to *document the methods used and activities performed* during the acquisition process. A list of files and collected artifacts typically produced at the end of the acquisition process is shown below.

- **System state**. This file captures detailed information about the current AE, such as the operating system type and version, network and routing configuration, and software versions. This record helps validate and track the environment in which the acquisition was performed during the successive forensic analysis.
- **Web artifacts**. This collection includes web page sources and all associated objects retrieved using HTTP (images, videos, scripts, style sheets).
- **Video recording**. This file provides a continuous visual record of the web session, capturing what was exactly displayed on the screen (including dynamic interactions and transitions) that static images cannot capture.
  It serves as evidence of the investigator's observations and interactions.
- **Input events**. This file records all user inputs during the session, such as keystrokes, mouse clicks, and form submissions. This data is critical for reconstructing user actions and understanding the sequence of events that led to specific outcomes on the website.
- **Traffic data**. This data consists of all network traffic captured in a standard format, namely Packet Capture (pcap) [22], exchanged between the AE and servers. This traffic provides a comprehensive overview of the exchanges, including requests and responses, which are essential for investigating any data transmitted during acquisition.

2.2.3 Preservation

The standard identifies *preservation* as the phase that ensures digital evidence remains intact and unaltered. This phase involves documenting the evidence to establish its integrity and authenticity from the time of collection until its presentation in a legal or investigative context. Digital investigations center around the creation of an electronic version of a document that acts as the Chain of Custody (CoC) [38].

To create the CoC, the AE logs the entire process by saving the list of (i) all files it outputs with the corresponding hash and (ii) all artifacts it collects with their hash and acquisition Timestamp (TS).. At the end of the acquisition, a certified TS Authority (TSA), acting as a Trusted Third Party (TTP), signs and dates the CoC. The TTP ensures that the CoC cannot be altered, securing the integrity of everything listed and associated with a hash.

Moreover, the TTP adds a precise timestamp to the acquisition. As this timestamp exclusively certifies the end of the process, it is conventional to prove the beginning by visiting a website capable of providing evidence of the start time and date, e.g., by querying Google for the current time. However, it is important to note that while this practice can aid in correlating with the timestamps of downloaded artifacts to verify their consistency, it is inherently weaker than the timestamp provided by the TTP.

2.3 Limitations and challenges

Although the approach described above is commonly implemented by mainstream software and is considered the most effective compromise for live web forensics acquisition, it still has notable limitations. We identify several open challenges that must be addressed to comply with ISO 27037:2012 specifications and requirements, as detailed below.

C_1 **Evidence tampering.** This challenge refers to the preservation process within the standard framework, underlining the need to effectively safeguard the integrity of digital evidence from collection through presentation in legal contexts. All produced artifacts, including traffic captures, are susceptible to tampering and spoliation. Even encrypted traffic can be compromised by forging a new handshake, transit (symmetric) keys, and altering content. This latter capability is nearly unique to the forensic acquisition of web evidence, as it depends on an SSLKEYLOGFILE containing all the secrets that would otherwise be unavailable in traffic captures.
The current approach mitigates data tampering risks by hashing artifacts and including them in the CoC, signed by a TTP upon acquisition completion. Any subsequent changes to the artifacts would cause a hash mismatch, detectable through the CoC. However, signing the CoC only at the end of the live web forensics process presents a critical limitation, allowing the opportunity to alter data before this point. For example, malicious actors can modify web artifacts, video recordings, input events, and traffic data before generating and signing the CoC, as the final signature does not guarantee that the data was not tampered with during this window.

C_2 **Timestamp tampering.** This challenge involves providing a reliable and accurate time reference, which is crucial for establishing a clear and trustworthy timeline of events. The current method relies on a TTP to provide a timestamp at the end of the acquisition process. The limitation of this approach is that it certifies only a single instant—conventionally the end of the acquisition—through the CoC signature, not an actual timeline of the entire process. Visiting a site that provides the current date or

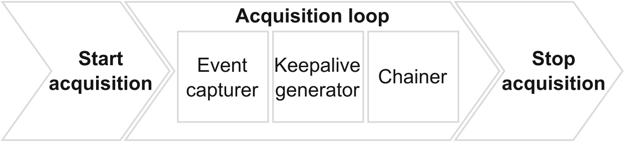


Fig. 1 Phases and main tasks of the acquisition process

relying on timestamps in HTTP headers does not ensure a trustworthy timeline because, as seen above, this data can be altered.

C_3 Single Source of Truth. This challenge emphasizes the need to ensure authenticity and integrity when multiple pieces of evidence are correlated by having one authoritative source, i.e., a SSOT, that avoids inconsistencies. For example, a forensic copy of mass storage creates a single image that allows investigators to correlate the included artifacts without misalignment and false inferences. However, the current approach produces different, separate files without a single source of truth to guarantee correlation. This limitation is evident, for instance, in the absence of a consistent method to align video recordings with the captured network traffic.

C_4 **Self verification.** This challenge addresses the requirement of the standard for a proven verification function to verify digital evidence. In the current approach, validation is typically accomplished only by checking the signature and timestamp of the TTP on the CoC and the validity of the recorded hashes. However, ensuring the integrity of each artifact necessitates information from different logs, which may include user input, its accurate transmission across the protocol, and its correct rendering in the browser. Given the extensive user interaction in modern web pages and the multitude of artifacts generated during browsing, the lack of a proven function to assist verification correlating the different sources can severely limit the transparency of digital evidence.

3 Methodology

In this section, we describe WEFT, our proposed methodology for live web forensic acquisition that addresses the limitations and challenges identified earlier (see Sect. 2.3).

3.1 Acquisition process

The acquisition process integrates and refines the traditional approach outlined in Sect. 2.2 and maintains the AE as the core element of its architecture (see Sect. 2.2.1).

Figure 1 provides an overview of the sequence of its three phases and main tasks. The *Start acquisition* phase marks the beginning of the process, capturing initial metadata and securing a trusted timestamp to establish a verified starting point. The *Acquisition Loop* phase uses the *Event Capturer* to record data events (i.e., network traffic, video frames, audio samples, input events, HTTP requests). The *Keepalive Generator* ensures continuity in the acquisition timeline, and the *Chainer* securely links all outputs in sequence. The *Stop acquisition* phase finalizes the process by securing the final state with a certified timestamp and completing the sequence of recorded evidence.

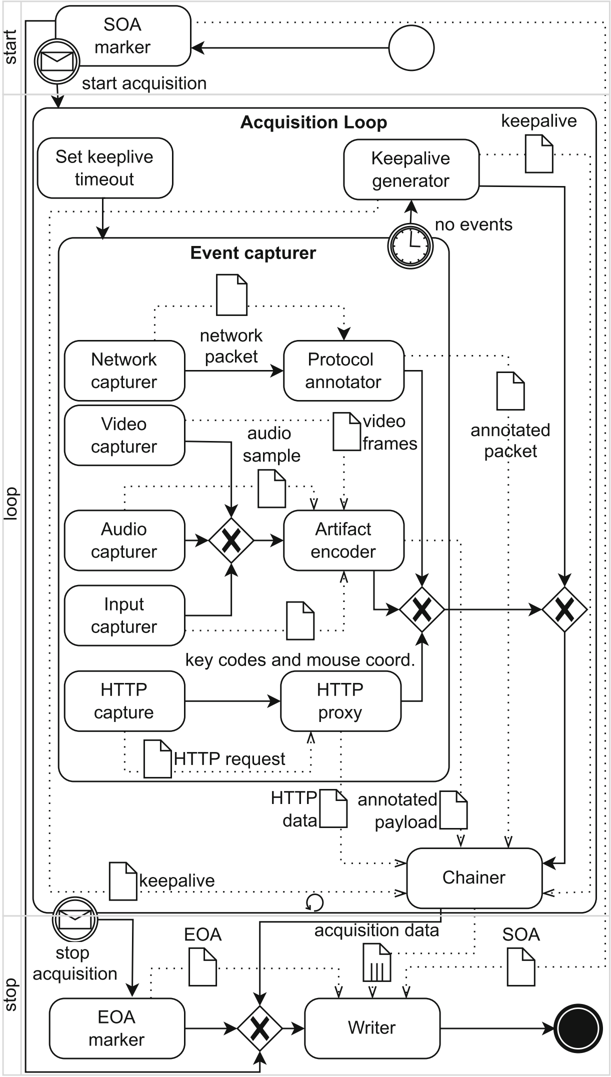


Fig. 2 Workflow of the acquisition process

In the following, we introduce the building blocks of the acquisition output and outline the phases described above, as represented in the detailed workflow diagram (see Fig. 2).

3.1.1 Building blocks of acquisition output

The acquisition process outputs a single artifact composed of a sequence of concatenated *blocks*. We classify these blocks into three types: Start of Acquisition (SOA), End of Acquisition (EOA), and Intermediate Block (IB).

SOA and EOA are of variable length, positioned at the beginning and end of the acquisition output, respectively. Their format is specified below.

Type	Length	Payload	Signature	Hash

Briefly, the *Type* identifies whether it is an SOA or an EOA, while *Length* holds the size of the block. The *Payload* contains data, and the *Signature* is generated by an external TTP to authenticate the payload. The *Hash* is a locally computed hash and is only present in the SOA.

The format of IBs blocks differs, as outlined below.

Type	Sequence No.	Nonce	Payload	Hash

The *Type* field specifies the type of payload contained within the block. For example, it can indicate whether the payload is a video sample, an input hook, or a network packet. The *Sequence No.* ensures the acquisition order and helps to verify that no blocks are missing. The *Nonce* is a unique value used to ensure different hash values for each block. Finally, the *Payload* and *Hash* are the same as described for the previous block type.

A special and noteworthy IB is the *Keepalive* block. The AE ensures that the acquisition output consistently contains blocks at a specified frequency and uses the keepalives when no inputs are generated directly by the acquisition source; we explain its purpose in Sect. 3.2.

3.1.2 Start acquisition

Once the process has started, the *SOA marker* task generates the SOA block. The block contains comprehensive metadata about the acquisition process, including the version of the AE software, the browser version, and the hardware specifications of the host machine, all of which are encapsulated within its payload. A hash of the payload is then stored in the corresponding field of the block. Moreover, the AE obtains an RFC3161-certified timestamp of the hash from a TSA that the block stores in the signature field. As the timestamp response cannot be forged without access to the private key of the TSA, this solution ensures that the block provides a tamper-proof record of the exact time the acquisition began and the specific conditions recorded in the payload.

This phase ends by starting the acquisition loop (in the figure, see the *start acquisition* event) and transmitting the SOA block to the task in charge of writing the final output, namely the *Writer* task.

3.1.3 Acquisition loop and stop acquisition

The *Acquisition loop* process gathers all inputs from the acquisition activity, capturing them as single and asynchronous events. Each iteration is managed by an *Event capturer* process that receives inputs from network, video, audio, input, and HTTP sources as events fired by dedicated captures.

The Event capture is initialized by a *Set keepalive timeout* process that configures the time limit an iteration can wait to receive an event. A *no events* condition indicates that the time limit has expired, triggering the execution of the *Keepalive generator* task. This task ends the iteration by creating a Keepalive payload containing the current timestamp.

If there is no timeout, a specific task processes the output from the capturer that generated the event.

The *Network capturer* outputs raw data of a *network packet* and simultaneously inputs the *Protocol annotator* task. This task applies techniques to annotate the raw data with additional information to enhance the trustworthiness of the acquired evidence. Currently, two techniques are implemented, although others can be introduced as further extensions. The first technique prompts for any new IP or domain address a DNS-over-HTTPS (DoH) request to verify the integrity of the name resolvers used by the AE. The second technique triggers an Online Certificate Status Protocol (OCSP) request or certificate revocation list download for any encrypted connections to certify the identity of the connected server. It is important to note that the work performed by the protocol annotator is asynchronous, meaning its results are not immediately included with the input that triggered it due to potential delays in data acquisition. Instead, the results create a separate *packet metadata* output once the associated query has been completed. Moreover, the annotation procedure can be recursively undertaken, for instance, triggering an OCSP request in response to a DoH request.

The *Video*, *Audio*, and *Input capturers* generate raw data corresponding to a *video frame*, an *audio sample*, and a *key code or mouse coordinate*, respectively. The *Artifact encoder* receives such generated data to transform them into an *encoded format* suitable for long-term storage. For example, it outputs opus for audio, VP8 video segments, and the integer keycode corresponding to the pressed keyboard key.

Finally, the *Web Browser* feeds the *HTTP proxy* with each *HTTP request* received during the acquisition. The HTTP proxy works similarly to the Protocol annotator by enriching the request with all the information that can be gathered during the client–server interaction. In particular, it outputs *HTTP data* containing the request and response bodies, headers, request method, response status code, and request URL. For TLS connections, *HTTP data* includes the certificate chain presented by the server and TLS secrets in SSLKEYLOG-compatible format. For WebSocket [30] traffic, each non-control frame generates *HTTP data* populated with its contents.

All the above outputs are collected by the *Chainer* task. This task aims to create a valid IB B_i for each output and chain it in the correct sequence with the other blocks. The Chainer sets the block type according to the event capturer from which it received data, assigns the next sequence number, generates a nonce, and uses the output as the payload. Finally, B_i hash is replaced by $\text{hmac}(B_i, H_{i-1})$, where $\text{hmac}(d, k)$ denotes a keyed-hash message authentication code function that operates on data d with key k, and H_{i-1} denotes the hash content found in block B_{i-1}. For the first block, H_{i-1} is the hash found in the SOA.

The Acquisition loop ends after it receives a *stop acquisition* event. The final output is the chained sequence of blocks from the Chainer, denoted as the *acquisition data*.

Finally, an EOA marker block is created. Within the EOA Marker, the process also encodes a timestamp response relative to H_{N-1}, the last hash associated with the acquisition. The process ends once the writer outputs the final artifacts merging SOA, acquisition data, and EOA.

3.2 Verification

In accordance with challenge $\mathbf{C_4}$, we also provide a unified, automatic, and reliable verification procedure, which enables the counterpart to easily validate the forensic evidence produced.

The verification procedure has multiple objectives: (i) ensuring the integrity of the evidence, (ii) certifying the timeline, and (iii) confirming the content – specifically, verifying that the evidence includes everything listed in the CoC with an accurate timestamp (see Sect. 2.2.3).

Ensuring the integrity of the evidence is straightforward, as all the evidence is contained within a single file, making it easy to verify by simply calculating its hash.

Certifying the timeline works by validating the certificate of the initial timestamp response, recalculating the hash for each data block, and checking the final timestamped hash.

Confirming the content operates as follows. All images found within the network recording, specifically those labeled with an HTTP content-type header starting with *image*, are searched for within the video frames. This method allows automatic correlation between the video evidence and image artifacts. Moreover, Optical Character Recognition (OCR) is applied to the video and correlated with text content found in the pages. This cross-referencing allows the counterpart to easily match up the evidence sources.

For encrypted traffic, further steps are taken to incorporate additional timestamps between the certified start and end moments. For TLS 1.2 [41] and earlier [14] connections, the timestamp from the client Client Hello message is used. For TLS 1.3 [40] connections, the search focuses on the *date* header sent by the server. These additional timestamps help to finely tune the timing of the evidence, which is crucial for establishing a precise timeline of events. To account for clock skews [43], we calculate the offset for each *raw* timestamp against the time certified by the TSA.

Finally, the rate of events recorded within the file is computed. The presence of keepalive blocks ensures a minimum block rate in every SSOT. By validating that blocks are present at the expected frequency, the procedure can confirm that the acquisition process consistently captured events without any gaps or interruptions, assuring that the SSOT represents a continuous and uninterrupted record between SOA and EOA.

4 Implementation and results

In this section, we detail the implementation of the methodology outlined in Sect. 3. We then evaluate the implementation by acquiring data from the top 100 most popular websites and comparing the results with those obtained using a leading state-of-the-art tool. Finally, we provide a discussion of our findings.

4.1 Implementation details

To evaluate and validate the approach introduced in Sect. 3, we implemented it within a bespoke program developed in Rust. The implementation leverages GStreamer [16], PipeWire [37], the Opus [50] and VP8 [51] codecs, and the XDG Screencast protocol [17] for audio and video recording.

SHA256 was used as our hash and HMAC function, and OpenSSL as our TSA implementation. Packet capturing was performed instead using Linux AF_PACKET [27] sockets. All HTTP requests were routed through an in-process HTTP and HTTPS proxy, handling the generation of SSLKEYLOGs and extraction of all request and response data contained within HTTP exchanges. On the user side, we used an unmodified Firefox 129 web browser as our acquisition platform, configured to use the aforementioned proxy. For very popular websites, Firefox could be replaced by a robotized browser such as Selenium to achieve reproducibility also in the actions performed during acquisition. Our implementation supports this scenario, as it does not modify the browser in any way.

To produce a SSOT with tamper-resistant and verifiable properties, we chose to utilize the well-recognized and industry-standard PCAP Next Generation (pcapng) format [48] as the one resulting artifact of an acquisition.

This format has two main features: (i) it allows the addition of custom binary payloads, known as *options*, to the packets, and (ii) the ability to add entirely new blocks to the pcap contents. By using these features, we can take advan-

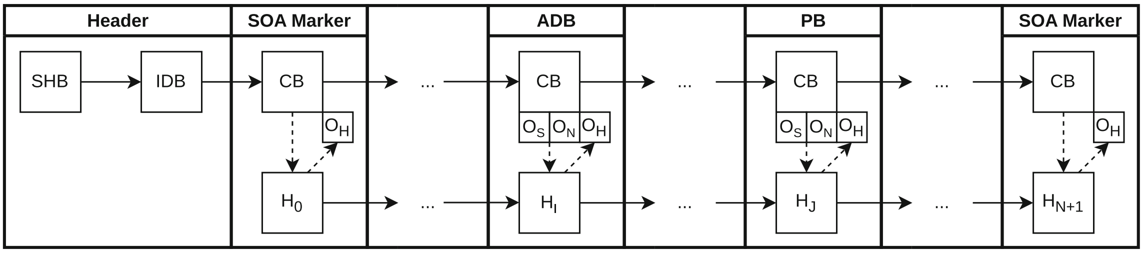


Fig. 3 Layout of the SSOT pcapng file

tage of mainstream tools like Wireshark [15] to access the original packet capture while also benefiting from archiving all the other required information and metadata in one single file.

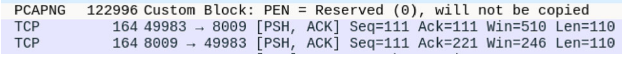
In relation to the concepts from Sect. 3.1.1, every block (SOA, EOA, IB) is modeled within the pcap file as Custom Block (CB), where the CB payload is composed of the type flag followed by the IB payload. For SOA and EOA blocks, the signature is included as part of the block. The length field is inherently included within the pcap file format specification, so there is no need to specify it separately in our model. To incorporate the nonce, payload, and hash fields, we leverage the pcap custom option mechanism. Each option is characterized by the vendor code (or PEN in the standard) `0x57424652` (`WBFR`).

As the hash content is contained as part of the block within the pcap structure, the determination of its payload follows a procedure similar to network protocols. The payload used for the hash computation includes the to-be-appended block with the hash part of the option set to all zeroes. This ensures that the hash value is calculated over the entire block, including the nonce and payload, while excluding the actual hash field itself.

Figure 3 depicts the layout of the produced forensic artifact. Dashed lines indicate computed values, while solid lines illustrate the flow of original data through the blocks, showing how data is captured, stored, and annotated.

Each capture begins with the standard header, including a Section Header Block (SHB) and an Interface Description Block (IDB), as required by the format specification.

As soon as the acquisition starts, a CB containing the SOA is appended to the file. Following this, all IBs are appended to the file by encoding them using the previously described method in structures we refer to as Acquisition Data Blocks (ADB). An ADB is composed of a CB containing vendor options: O_S for the sequence number, O_N for the nonce, and O_H for the HMAC of the block and options contents keyed by H_{i-1}, i.e., the contents of O_H found in the preceding block. An exception to this rule applies to IBs containing network packets. We record these using pcapng standard Enhanced Packet Block (EPB) blocks with associated options O_S, O_N, and O_H, calculated as described earlier. This configuration



```
PCAPNG   122996 Custom Block: PEN = Reserved (0), will not be copied
TCP         164 49983 → 8009 [PSH, ACK] Seq=111 Ack=111 Win=510 Len=110
TCP         164 8009 → 49983 [PSH, ACK] Seq=111 Ack=221 Win=246 Len=110
```

Fig. 4 An ADB, as shown within Wireshark

is called an Annotated Packet Block (APB). The contents of APBs, besides the semantics associated with their options, remain accessible to any other program parsing pcap files.

Figure 4 illustrates an ADB followed by two APBs in the popular pcap analyzer Wireshark. In this figure, while the content of the ADB is completely opaque to the tool, the two subsequent APBs are correctly decoded by Wireshark as network packets.

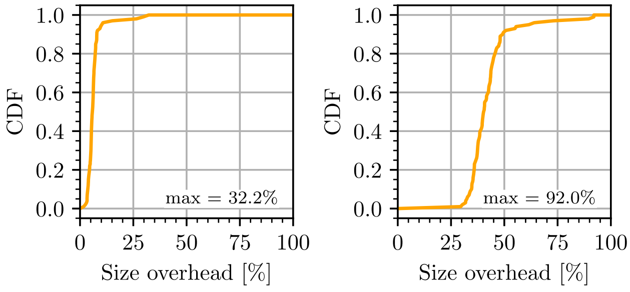
Finally, the last block in the SSOT is always a CB containing the EOA, signaling the completion of the acquisition process.

4.2 Experimental settings

For comparison, we selected the Freezing Internet Tool (FIT) [53] as the current open-source state-of-the-art (SOTA) solution, which aligns with the best-practice approach described in Sect. 2.2. FIT is a software that allows spawning a specially modified browser and performing the acquisition of a single web page, together with a desktop video capture of the acquisition. On stop of the recording, FIT finalizes the acquisition by writing its logs, scraping from the currently loaded page all media and saving it into a local archive, and hashing all of the generated files.

Two identical virtual machines (VMs), each equipped with 8 CPU virtual cores and 32 GB RAM, were provisioned on the same Proxmox VE 7 hypervisor equipped with AMD EPYC 7413 CPUs. The first VM ran Fedora Workstation 40 and our bespoke implementation, while the second VM executed Windows 11 and FIT version 1.2 release candidate.

Each experiment consisted of acquiring web browsing activity for at least 10 s towards the top 100 most popular websites, as measured by Google Chrome usage data [42]. This 10-second interval without user interaction represents the reference use case where an investigator directly accesses a specific webpage for acquisition. By avoiding explicit interactions with page elements during this period, we eliminated



(a) Only packets (b) Overall

Fig. 5 Size overhead of the SSOT

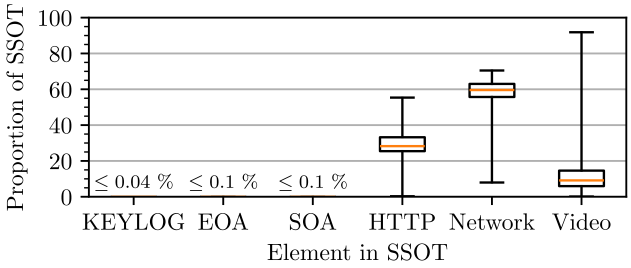


Fig. 6 Composition of the SSOT

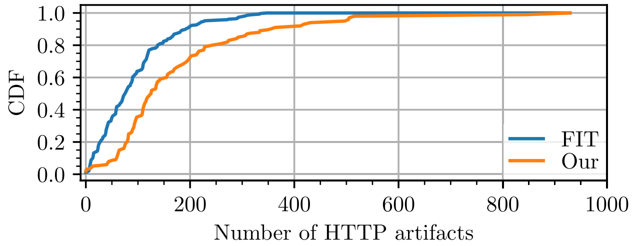


Fig. 7 Distribution of extracted HTTP artifacts

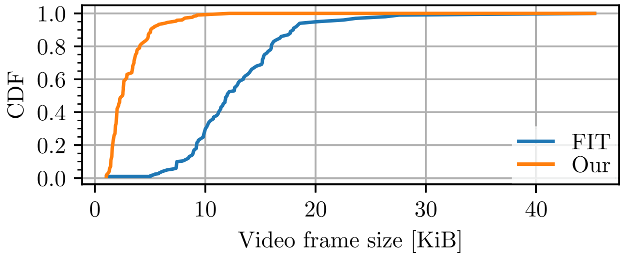


Fig. 8 Distribution of video frame sizes

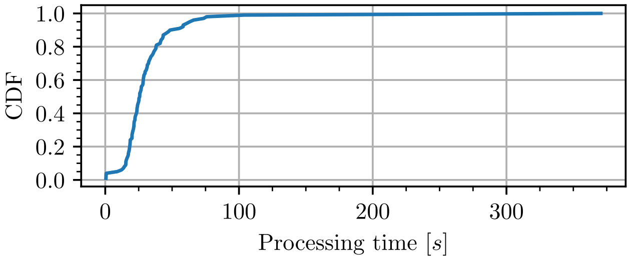


Fig. 9 Distribution of successful FIT finalization timings

potential bias that could arise from arbitrary choices in user behavior. After this, we allow each tool to finalize its artifact files within a 10-minute timeframe. If an approach failed to produce its artifacts within this time limit, we deemed the acquisition attempt unsuccessful.

4.3 Results

Figure 5 illustrates the storage overhead introduced by our approach compared to merely recording network traffic. The first subgraph (Fig. 5a) depicts the cumulative distribution function (CDF) of size overheads introduced by including validation information, such as chained hash values, in the SSOT file. This graph shows a median overhead of approximately 5.8 percent, with an average of 6.6 percent and a maximum overhead of 32.2 percent. The second subgraph (Fig. 5b) presents the CDF of size overhead when all possible forensic information - including audio, video, keystroke data, SSLKEYLOG, and HTTP artifacts (i.e. the contents of *HTTP data* as described in Sect. 3.1.3) - is included in the SSOT file. Here, the median overhead is 40.4 percent, with an average of 42.7 percent, ranging from 29.5 to 92 percent.

Figure 6 details the composition of elements found in SSOT files built by our method. The SSLKEYLOG, EOA, and SOA elements contribute negligibly to the overall size, with contributions of less than 0.04, 0.1, and 0.1 percent, respectively. HTTP artifacts comprise between 0.2 and 55.3 percent of the file, with a median value of 28 percent and an average of 29.3 percent. Network traffic itself ranges from 7.9 to 70.5 percent of the SSOT, with a median of 59.6 percent and an average of 57.3 percent. Video content shows the highest variability, ranging between 0.2 and 91.8 percent, with a median size of 9.1 percent and an average of 13.9 percent.

Figure 7 illustrates the cumulative distribution of extracted HTTP artifacts for both methods. The SOTA approach extracts an average of 93 artifacts per acquisition, with a median of 76. Our method extracts more artifacts, averaging 177 per acquisition with a median of 125.

Figure 8 compares the sizes of video frames between our method and the SOTA approach, normalized by recording length to ensure a fair comparison. Our approach consistently produces smaller video frame sizes, with a median size of 2.5 KiB and a range from 1 KiB to under 10 KiB per frame. In contrast, the SOTA approach exhibits larger video frame sizes, ranging from approximately 5 KiB to 46 KiB, with a median size of 12 KiB.

Figure 9 presents the CDF of finalization times for the SOTA approach across all trials. Since our approach consistently finishes in less than 1 s throughout all trials, it is not included in this graph. The median processing time for the

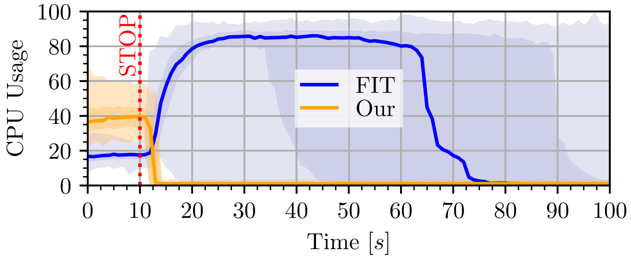
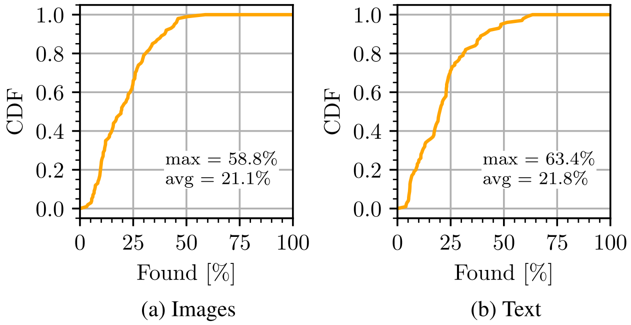


Fig. 10 CPU usage envelope over time



(a) Images

(b) Text

Fig. 11 Detection rate of automated content validation

SOTA approach is 25.8 s, with processing times ranging from 0.5 to 370.6 s. Notably, in nine cases, the finalization process was not completed within the allowed 10-minute timeout, resulting in a failed acquisition.

Figure 10 displays the CPU usage over time for both approaches, with lines marking the median and shaded areas indicating the zeroth (minimum), first, third, and fourth (maximum) quartiles. Our approach exhibits an average CPU usage of 37.5 percent during acquisition, with values ranging from 23 to 65 percent. During finalization, the CPU usage for our approach averages 1.7 percent, with a minimum of 0.2 percent and a maximum of 3.1 percent. In comparison, the SOTA approach demonstrates an average CPU usage of 16.8 percent during acquisition, with values ranging from 10.4 to 28.8 percent. During finalization, the CPU usage of SOTA approach ranges from 0.6 to 97.5 percent, with an average of 82.0 percent. Although our approach exhibits higher CPU usage during acquisition, it consumes less CPU time overall due to its faster termination.

Figure 11 shows the detection rates for images and text during the validation, i.e., how many images or text snippets recorded as HTTP artifacts are recognized in the SSOT video. For images (Fig. 10a) the rates vary from a minimum of 3.2 percent to a maximum of 58.8 percent, with an average detection rate of 21.1 percent. Text (Fig. 10b) detection rates range from 4.1 percent to 63.4 percent, with an average rate of 21.8 percent.

4.4 Discussion

While considering only the overhead added by validation information, our method exhibits higher disk usage compared to network recording alone. However, as disk space is often one of the least valuable resources, this limitation likely has a negligible impact on the viability of our approach, except during the transfer phase of the evidence itself, where the increased size could require additional effort.

Observing the composition of the SSOT files, network data, media, and HTTP artifacts are the largest contributors to the file sizes, as expected. Network recording is a superset of the HTTP artifacts, which is also an anticipated result. Notably, video media shows a highly variable size distribution, reflecting the variability in compressibility of web page contents, with static pages allowing better compression by codecs compared to video streaming sites.

Our approach allows for recovering a greater quantity of content from the same website compared to SOTA methods that rely on export functionality provided by browsers. While these snapshot-based approaches capture only the assets required to render the page at the moment the snapshot was taken, our approach records all assets as they are loaded from the network. This enables the extraction of comparatively more artifacts as many websites leverage ephemeral media with animations or features like prefetching, requesting network information that may be used later but does not actually belong to the assets used in rendering the pages themselves. Furthermore, our method is capable of extracting messages from persistent connections, e.g., web sockets, with each message as separate artifact. However, since our approach does not modify the browser, artifacts already present in the cache are not captured in our recording – a limitation that could be addressed by resetting the browser state before each acquisition.

The video frame size statistics show that incorporating media files and similar content into evidence packages can minimize the dimensions of associated media files by fully leveraging state-of-the-art codecs. This highlights the importance of carefully choosing such codecs to minimize the overhead associated with including media in the evidence. Over time, better codecs are expected to further increase achievable compression ratios.

Finalization time results demonstrate the benefits of our decision to distribute actions typically performed during the finalization phase (e.g., video encoding and hashing) throughout the acquisition process. This provides advantages in terms of integrity, as the SOTA method allows potential attackers an ample time window to compromise the generated archive. It also improves reliability, as for some websites, performing these actions caused the SOTA approach to fail to finalize an artifact within the allotted time. The reduced overall CPU usage suggests that conducting finalization operations in a streaming manner could enable increased efficiency compared to SOTA approaches. However, this requires managing concerns such as ensuring the video encoder can process data in real time without lagging behind the acquisition and supporting continuous streaming, e.g., not including expected video length within headers. VP8, the codec used in our implementation, supports these features.

Integrity validation is a matter of implementing code that follows through the hash chain and checks the validity of the TSA signatures. Content validation, on the other hand, has shown an overall low detection rate of content found in both the network traffic and the video recording. This low detection rate is expected and can be attributed to the fact that during the acquisition process, the browser loaded many assets that were never rendered to any screen during video recording. These undetected artifacts were still transferred at some point in the acquisition but are not guaranteed to have been found in two sources within the SSOT.

Finally, regarding challenge $\mathbf{C_1}$, the chained hash structure of the SSOT file and the presence of all artifacts multiplexed together within the SSOT provide the file with a cryptographic proof of its integrity. Furthermore, modification of the produced artifact files now requires a significantly more capable actor, raising the barrier needed to alter the evidence integrity. As for $\mathbf{C_2}$, since integrity verification links both the start and end of the chain with the certified timestamp of a TSA and any alteration to the file would be caught by the invalidation of the hash chain, the system ensures that for a given valid SSOT, its start and end of acquisition instants are precisely known and associated with the artifacts found within it. Moreover, the SSOT structure ensures that such an association also preserves the order in which those artifacts have been recorded. In terms of challenge $\mathbf{C_3}$, the produced file is a single, atomic, easily shareable file, fully satisfying its requirements. Lastly, to address challenge $\mathbf{C_4}$, the method allows for fully automated integrity verification of any given SSOT and partially automated verification of the contents found within it, enabling counterparts to assess both the integrity and whether specific content is present in the evidence in a unified way.

5 Related work

Forensic investigations have evolved into sub-fields relating to the nature of the evidence involved. DF focuses on recovering and analyzing material found in digital environments. Network Forensics (NF) is one of the sub-branches of DF, where the data being analyzed is the network traffic going to and from the system under investigation. The scientific literature covers both DF and NF, including incident remediation and techniques for timelining [10, 21], and anti-tampering acquisition of local (persistent or volatile) memory [2, 9, 26]. NF is commonly described as the discipline concerned with the capture, recording, and analysis of network traffic for detecting and investigating intrusions as an extended phase of network security [12, 25, 39, 44] and can also be extended to the acquisition of web pages [36]. Ghaleb et al. [19] address website traffic analysis as a method for cybercrime investigation, particularly in the context of encrypted traffic. Their study employs fingerprinting rather than focusing on forensic copying of web page content. Similarly, [32] describes the use of specialized hardware for packet capturing to gather evidence from websites or other online services. An investigation model specifically targeting online social networks is presented in [52].

Since web forensics is considered a sub-discipline [49] of NF, the previous research above can serve as a reference for identifying effective methodologies in live web forensics acquisition. However, none of these studies take a comprehensive approach to the legal acquisition of web-based evidence for disputes involving published content, nor do they propose innovations that ensure consistency, tamper resistance, and automatic verification.

Other works provide perspectives that align with our objectives, and we compare them to our approach in terms of the challenges $\mathbf{C_1}$, $\mathbf{C_2}$, $\mathbf{C_3}$, and $\mathbf{C_4}$ described in Sect. 2.3. The results of this comparison are summarized in Table 1.

Comandini's work [11] addresses the problem of proving that data existed before a specific point in time, which is crucial in various forensic acquisition scenarios. The study explores methods to enhance the security of timestamps by using systems that achieve distributed consensus without relying on a trusted third party, such as Bitcoin. The use of blockchain technology inherently provides a form of anti-tampering, addressing challenge $\mathbf{C_1}$ and partially satisfying $\mathbf{C_2}$. The work focus is on OpenTimestamps, a protocol that sets a standard for creating timestamps and explores a proposed improvement involving elliptic curve commitments. While Comandini's work provides a comprehensive review of this scheme, ultimately leading to a practical application, his method has limitations in dynamic web environments and highlights the need for cost-effective, *trustless* timestamping. Our approach, goes beyond these limitations. Indeed, unlike Comandini's method, which only verifies that an acquisition occurred before a specific date, our method provides a timestamp that spans the entire acquisition period, thereby fully satisfying challenge $\mathbf{C_2}$. Our approach also supports scenarios where an acquisition can be paused and later resumed without compromising the authenticity of the evidence. In contrast, all other methods would compromise the continuity of evidence by shifting the entire timeline forward and aligning it with the end of the acquisition. Although blockchain technology does provide a degree of anti-tampering, grouping the data as a single source entity,

Table 1 Comparison between our and other literature works

	[11]	[7]	[31]	[20]	[47]	Our work
C_1 (Evidence tampering)	●	●	●	●	●	●
C_2 (TS tampering)	◑	◑	◑	◑	◑	●
C_3 (SSOT)	◑	○	○	◑	○	●
C_4 (Self verification)	○	○	○	◑	○	●

this aspect is not directly addressed in the work, nor is there any mention of an autonomous verification system. Castiglione et al. [7] introduce another method for evidence collection, with a particular emphasis on gathering data from online services such as web pages, chats, documents, photos, and videos. This approach is designed to be user-friendly, providing guidance throughout the entire acquisition process. It automatically collects information from the remote source during acquisition, capturing not only network packets but also any data generated by the client, such as video and audio. To ensure the integrity of the evidence and the acquisition process, a trusted third party, acting as a digital notary, certifies both the acquired information and the actions performed by the analysts. However, this only partially addresses the challenge $\mathbf{C_2}$ related to anti-tampering timelines, especially in cases where the acquisition is paused and then restarted. The challenges $\mathbf{C_3}$, $\mathbf{C_4}$ are not addressed.

Nikkel [31] introduces a portable device for network forensic evidence collection, built using COTS hardware and open-source software. This device uses promiscuous packet capturing to improve the collection of evidence from remote network sources, such as websites and other online services. It operates at the link layer, allowing for transparent insertion inline between a network node and the rest of the network. Nikkel provides a detailed overview of the device's architecture, construction, and operation. Challenge $\mathbf{C_1}$ is addressed through the application of hashes, which are considered an integral part of evidence preservation. However, the complete timeline robustness required for web evidence acquisition $\mathbf{C_2}$ is not fully addressed, and the challenges $\mathbf{C_3}$, $\mathbf{C_4}$ are not considered.

Han et al. [20] presented a digital evidence container based on a Merkle tree, capable of storing media images, bit streams transmitted over networks, and files in the cloud. The use of Merkle trees ensures the satisfaction of challenge $\mathbf{C_1}$. However, the timestamping is not continuous, which only guarantees that the acquisition occurred before a specific date, rather than providing the exact timing of the entire stream, meaning that challenge $\mathbf{C_2}$ is only partially met. Han only partially meets challenge $\mathbf{C_3}$, as the output of their system is described as consisting of several items without specifying a unified, monolithic format. Similarly, challenge $\mathbf{C_4}$ is partially addressed at the methodological level, with no explicitly developed tool for this purpose.

The work of Tian et al.[47] examines the transition of the Internet from a host-centric to a content-centric structure, where content has become the central focus. We compare our approach with this blockchain-based study because it specifically addresses the acquisition of web pages. This study particularly focuses on securing digital evidence against file tampering, addressing challenge $\mathbf{C_1}$ through the use of blockchain technology. However, it only partially covers challenge $\mathbf{C_2}$, and challenges $\mathbf{C_4}$ and $\mathbf{C_3}$ are not specifically addressed by the work.

Our work does not leverage blockchain technology; however, we compared it with studies focused on web forensics that utilize blockchain. While there are other approaches that adopt blockchain for forensic purposes, they are not specifically aimed at the forensic acquisition of web pages, such as those discussed in [2, 8, 9, 26]. We do not further explore blockchain-based approaches due to several critical limitations, which are listed below:

i) Cost and Speed: Blockchain transactions can be costly and slow, affected by network traffic.
ii) Scalability: High transaction volumes can lead to scalability issues, increasing delays and costs.
iii) Privacy: Permanent data on the blockchain could inadvertently reveal sensitive information or behavioral patterns.
iv) Overuse: Blockchain may not be strictly necessary for timestamping.
v) Regulations and Legality: The legal validity of blockchain timestamps can vary by jurisdiction.
vi) Long-Term Data Retrieval: Verifying hashes after a long time can require downloading substantial portions of blockchain history, which is resource-intensive.

6 Conclusions

This paper highlights the limitations present in the current best practices for acquiring web forensic evidence in live environments. Furthermore, it addresses the unique challenges associated with the presentation of results from live web acquisitions as forensically defensible evidence in legal contexts, in accordance with international standards. Based on the above considerations, we propose WEFT, a novel methodology for such acquisitions that is able to produce a single and tamper-resistant artifact. Our implementation shows that this artifact can be automatically verified, which not only guarantees its reliability but also enhances transparency, aiding the counterpart in their defense. An extensive evaluation with live acquisition of the top 100 most popular websites, along with a comparison to an open-source tool implementing the current best practice approach, demonstrates both its effectiveness and advancement over the existing solutions. It is our hope that this approach will serve to establish a foundation for more secure and transparent web forensic practices, ultimately improving the reliability of digital evidence and ensuring greater fairness in legal proceedings.

Acknowledgements This work was carried out while Giacomo Longo was enrolled in the Italian National Doctorate on Artificial Intelligence run by the Sapienza University of Rome in collaboration with the University of Genoa.

Funding Open access funding provided by Università degli Studi di Genova within the CRUI-CARE Agreement. This work was partially funded by the NextGenerationEU project "Security and Rights in CyberSpace" (SERICS).

Data Availability The dataset in COTS format for Sect. 4 is available on request.

Declarations

Conflict of interest The authors declare no Conflict of interest, financial or otherwise.

References

1. Digital forensics market size & report analysis | 2032, (2023) Accessed: 2023-06-06
2. Akter, Omi, Arnisha Akther, Md., Uddin, Ashraf, Manowarul, M., Islam.: Cloud forensics: Challenges and blockchain based solutions. International Journal of Wireless and Microwave Technologies **10**(5), 1–12 (2020)
3. Cantelli-Forti, Alessandro, Longo, Giacomo, Russo, Enrico: Work-in-progress: Consistent and tamper-proof acquisition of automatically verifiable forensic web evidence. In *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&P)*, page 726–730. IEEE, (2024)
4. Casadio, Emanuele: Kopjra Web Forensics. https://webforensics.kopjra.com,
5. Casey, Eoghan: *Handbook of digital forensics and investigation*. Academic Press, (2009)
6. Casino, Fran, Dasaklis, Thomas K., Spathoulas, Georgios P., Anagnostopoulos, Marios, Ghosal, Amrita, Borocz, Istvan, Solanas, Agusti, Conti, Mauro, Patsakis, Constantinos: Research trends, challenges, and emerging topics in digital forensics: A review of reviews. IEEE Access **10**, 25464–25493 (2022)
7. Castiglione, Aniello, Cattaneo, Giuseppe, De Maio, Giancarlo, De Santis, Alfredo, Roscigno, Gianluca: A Novel Methodology to Acquire Live Big Data Evidence from the Cloud. IEEE Trans. on Big Data **5**(4), 425–438 (2019)
8. Chen, Shijie, Zhao, Chengqiang, Huang, Lingling, Yuan, Jing, Liu, Mingzhe: Study and implementation on the application of blockchain in electronic evidence generation. Forensic Science International: Digital Investigation **35**, 301001 (2020)
9. Chopade, Mrunali, Khan, Sana, Shaikh, Uzma, Pawar, Renuka: Digital forensics: Maintaining chain of custody using blockchain. In *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 744–747. IEEE, (2019)
10. Ciardhuáin, Séamus. Ó.: An extended model of cybercrime investigations. International Journal of Digital Evidence **3**(1), 1–22 (2004)
11. Comandini, Leonardo: Sign-to-contract: How to achieve trustless digital timestamping with zero marginal cost. Master's thesis, Politecnico di Milano, (2017)
12. Corey, Vicka, Peterman, Charles, Shearin, Sybil: Michael S Greenberg, and James Van Bokkelen. Network forensics analysis. IEEE Internet computing **6**(6), 60–66 (2002)
13. Dal Checco, Paolo: Forensic Acquisition of Websites and Webpages - OSDFCON 2021. https://www.dalchecco.it/forensic-acquisition-of-websites-and-webpages-osdfcon-2021/. Accessed on December (2024)
14. Dierks, Tim, Rescorla, Eric: The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346, April (2006)
15. Wireshark Foundation. Wireshark. https://www.wireshark.org/
16. freedesktop.org. Gstreamer. https://gstreamer.freedesktop.org, (2024)
17. freedesktop.org. org.freedesktop.portal.screencast, (2024)
18. Garfinkel, Simson L.: Digital forensics research: The next 10 years. *digital investigation*, 7:S64–S73, (2010)
19. Ghaleb, Taher Ahmed: Website fingerprinting as a cybercrime investigation model: Role and challenges. In *2015 First International Conference on Anti-Cybercrime (ICACC)*, pages 1–5. IEEE, (2015)
20. Han, Jaehyeok, Han, Mee Lan, Lee, Sangjin, Park, Jungheum: Ecobag: An elastic container based on merkle tree as a universal digital evidence bag. Forensic Sci. Int.: Digital Investigation **49**, 301725 (2024)
21. Hargreaves, Christopher, Patterson, Jonathan: An automated timeline reconstruction approach for digital forensic investigations. Digital Investigation **9**, S69–S79 (2012)
22. Harris, Guy, Richardson, Michael: PCAP Capture File Format. Internet-Draft draft-ietf-opsawg-pcap-04, Internet Engineering Task Force, August (2024). Work in Progress
23. ISO/IEC. ISO/IEC 27037:2012 Information technology–Security techniques –Guidelines for identification, collection, acquisition and preservation of digital evidence, (2012)
24. Kessler, Gary: Anti-forensics and the digital investigator. *5th Australian Digital Forensics Conference*, Edith Cowan University:December 3rd 2007, (2007)
25. Khan, Suleman, Gani, Abdullah, Wahab, Ainuddin Wahid Abdul., Shiraz, Muhammad, Ahmad, Iftikhar: Network forensics: Review, taxonomy, and open challenges. J. Network and Computer Appl. **66**, 214–235 (2016)

26. Li, Shancang, Qin, Tao, Min, Geyong: Blockchain-based digital forensics investigation framework in the internet of things and social systems. IEEE Trans. on Computational Social Systems **6**(6), 1433–1441 (2019)
27. Linux man-pages maintainers. *packet(7) - Linux manual page*
28. Envolve Forensics LTD. FAW - Forensics Acquisition of Websites. https://www.fawproject.com
29. Marshall, A.M.: An improved protocol for the examination of rogue www sites. Science & Justice **43**(4), 237–248 (2003)
30. Melnikov, Alexey, Fette, Ian: The WebSocket Protocol. RFC 6455, December (2011)
31. Nikkel, Bruce J.: Generalizing sources of live network evidence. Digital Investigation **2**(3), 193–200 (2005)
32. Nikkel, Bruce J.: A portable network forensic evidence collector. *digital investigation*, 3(3):127–135, (2006)
33. Ottmann, Jenny, Breitinger, Frank, Freiling, Felix: Defining Atomicity (and Integrity) for Snapshots of Storage in Forensic Computing. March (2022)
34. Ottmann, Jenny, Breitinger, Frank, Freiling, Felix: An Experimental Assessment of Inconsistencies in Memory Forensics. *ACM Trans. Priv. Secur.*, 27(1):2:1–2:29, (2023)
35. Pagani, Fabio, Fedorov, Oleksii, Balzarotti, Davide: Introducing the Temporal Dimension to Memory Forensics. *ACM Trans. Priv. Secur.*, 22(2):9:1–9:21, (2019)
36. Pilli, Emmanuel S., Joshi, Ramesh C., Niyogi, Rajdeep: Network forensic frameworks: Survey and research challenges. *digital investigation*, 7(1-2):14–27 (2010)
37. PipeWire. Pipewire project. https://pipewire.org/, (2024)
38. Prayudi, Yudi, Azhari, S.N.: Digital chain of custody: State of the art. Int. J. Computer Appl. **114**(5), 1–9 (2015)
39. Rasmi, Mohammad, Jantan, Aman, Al-Mimi, Hani: A new approach for resolving cyber crime in network forensics based on generic process model. In *The 6th International Conference on Information Technology (ICIT 2013)*. Citeseer, (2013)
40. Rescorla, Eric: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, August (2018)
41. Rescorla, Eric, Dierks, Tim: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, August (2008)
42. Ruth, Kimberly, Fass, Aurore, Azose, Jonathan, Pearson, Mark, Thomas, Emma, Sadowski, Caitlin, Durumeric, Zakir: A world wide view of browsing the world wide web. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22. ACM, October (2022)
43. Schatz, Bradley, Mohay, George, Clark, Andrew: A correlation method for establishing provenance of timestamps in digital evidence. *digital investigation*, 3:98–107 (2006)
44. Sikos, Leslie F.: Packet analysis for network forensics: A comprehensive survey. Forensic Sci. Int.: Digital Investigation **32**, 200892 (2020)
45. Stoykova, Radina: Encrochat: The hacker with a warrant and fair trials? Forensic Science International: Digital Investigation **46**, 301602 (2023)
46. Thomson, Martin: The SSLKEYLOGFILE Format for TLS. Internet-Draft draft-ietf-tls-keylogfile-01, Internet Engineering Task Force, April (2024). Work in Progress
47. Tian, Zhihong, Li, Mohan, Qiu, Meikang, Sun, Yanbin, Shen, Su.: Block-def: A secure digital evidence framework using blockchain. Information Sciences **491**, 151–165 (2019)
48. Tüxen, Michael, Risso, Fulvio, Bongertz, Jasper, Combs, Gerald, Harris, Guy, Chaudron, Eelco, Richardson, Michael: PCAP Next Generation (pcapng) Capture File Format. Internet-Draft draft-ietf-opsawg-pcapng-02, Internet Engineering Task Force, August (2024). Work in Progress
49. Vaghela, Rajdipsinh, Gowda, V Dankan, Taj, Mohammad, Arudra, Annepu, Chopra, Manoj: Digital evidence collection and preservation in computer network forensics. In *Handbook of Research on Innovative Approaches to Information Technology in Library and Information Science*, pages 42–62. IGI Global, (2024)
50. Valin, Jean-Marc, Vos, Koen, Terriberry, Timothy B.: Definition of the Opus Audio Codec. RFC 6716, September (2012)
51. Westin, Patrik, Lundin, Henrik, Glover, Michael, Uberti, Justin, Galligan, Frank: RTP Payload Format for VP8 Video. RFC 7741, March (2016)
52. Zainudin, Norulzahrah Mohd, Merabti, Madjid, Llewellyn-Jones, David: A digital forensic investigation model and tool for online social networks. In *12th Annual Postgraduate Symposium on Convergence of Telecommunications, Networking and Broadcasting (PGNet 2011), Liverpool, UK*, pages 27–28, (2011)
53. Zito, Fabio: Freezing internet tool. https://github.com/fit-project/fit, (2024)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.