Journal of Engineering Sciences and Information Technology
Volume (5), Issue (3) : 30 Sep 2021
P: 92 - 111

ISSN: 2522-3321

مجلة العلوم الهندسية وتكنولوجيا المعلومات
المجلد (5)، العدد (3) : 30 سبتمبر 2021م
ص: 92 - 111

# Proposed Digital Evidence Sharing and Preserving Mechanism in Cloud Computing Environments

**Waseem Mohammad Maamoun Al-Sbaiti**
**Maher Abdulrahman Abbas**
**Mohssen Aziz Abboud**
College of Informatics || Al-Baath University || Syria

**Abstract:** Investigations and digital evidence have become an important and critical discipline that has made many researchers devote vigorous efforts to developing digital surveillance and investigation mechanisms, especially after the great expansion of the technical infrastructure on cloud computing platforms, which added more challenges to digital investigation. So far, no robust model has been found for preserving and exchanging digital evidence between clouds and users without this model causing a breach of user privacy or affecting performance. Most of the current studies on digital evidence exchange mechanisms rely at one stage of the exchange or evidence formation process on the CSP, which allows the cloud provider (or a malicious employee within the cloud provider) to manipulate the evidence or data. This research will present a proposal for a mechanism for sharing and preserving digital evidence between the cloud parties, taking into account the performance in the major cloud computing models (IaaS, PaaS, SaaS), and how this model can achieve evidence integrity and a less level of interference in the privacy of the user as well as the cloud service provider considering that may be more than one party accused as forgery. To achieve this, we have selected some digital evidence that digital investigators can rely on as digital forensic evidence in cases related to information crimes as a sample that can be exchanged and verified that none of them has tampered with this evidence, especially since cloud environments may go beyond having a single cloud that performs the service and thus there are several clouds involved in forming evidence, then we tested this mechanism by applying the SHA-2 Hashing process to digital evidence, then encrypting the output with the Elliptic Curve Cryptography algorithm and measuring the time needed to exchange and verify the evidence. We will compare the proposed model with models in previous studies to illustrate how the proposed model overcame the problem of relying on one party to form the evidence with the least impact for all parties on the level of performance or privacy, and how distributed SHA-2 hashing values proved its effectiveness in the inability of any party to deny the evidence or tamer it.

**Keywords:** cloud computing, virtualization technology, digital investigation, cloud forensics, digital evidence.

# اقتراح آلية لتبادل الأدلة الرقمية والحفاظ عليها في بيئات الحوسبة السحابية

**وسيم محمد مأمون السبيتي**
**ماهر عبد الرحمن عباس**
**محسن عزيز عبود**
كلية هندسة المعلوماتية || جامعة البعث || سوريا

**المستخلص:** غدت التحقيقات والأدلّة الجنائية الرقميّة تخصّص مهم وحرج جعل العديد من الباحثين يسخِّرون جهود حثيثة لتطوير آليات المراقبة والتحقيق الرقمي، خصوصاً بعد التوسُّع الكبير الذي شهدته البنى التقنية على منصّات الحوسبة السحابية، الأمر الذي أضاف تحدّيات أكثر على التحقيق الرّقمي. وإلى الآن لم يتم إيجاد نموذج عمل متين لحفظ الأدلة الرقمية وتبادلها بين السحابات والمستخدمين دون أن يسبب هذا النموذج خرقاً لخصوصية المستخدم أو تأثير على الأداء. وإن معظم الدراسات الحالية المتعلقة بآليات تبادل الأدلة الرقمية تعتمد في مرحلة من مراحل عملية التبادل أو تشكيل الدليل على مزود الخدمة السحابية CSP الأمر الذي يتيح للمزود السحابي (أو لموظف خبيث ضمن المزود السحابي) التلاعب بالدليل أو البيانات. سيقدم هذا البحث مقترح آلية لتبادل وحفظ الأدلة الرقمية بين الأطراف السحابية مراعياً الأداء في نماذج الحوسبة السحابية الرئيسية (IaaS, PaaS, SaaS)، وكيف يمكن لهذا النموذج أن يحقق تكاملية الدليل ومستوى تدخل أقل في خصوصية المستخدم ومزود الخدمة السحابي باعتبار أن الأطراف التي قد تكون متّهمة بالتزوير قد تتعدى أن يكون طرف واحد مُزوِّر. ولتحقيق ذلك قمنا بانتخاب بعض الأدلة الرقمية التي من الممكن أن يستخدمها المحققون الرقميون كأدلة جنائية رقمية في القضايا المتعلقة بالجرائم المعلوماتية كعينة يمكن تبادلها والتوثق من صحتها وعدم تلاعب أي طرف منهم في هذا الدليل خصوصاً أن البيئات السحابية قد تتعدى أن تكون هناك سحابة واحدة تؤدي الخدمة وبالتالي هناك عدة سحابات تشترك في تكوين الدليل، ثم قمنا باختبار هذه الآلية بتطبيق عملية التقطيع SHA-2 على الأدلة الرقمية ثم تشفير الناتج بخوارزمية المنحنيات الإهليلجية Elliptic Curve Cryptography وقياس الزمن اللازم لتبادل الأدلة والتأكد من صحتها. وسنقارن النموذج المقترح مع نماذج في دراسات سابقة لتوضيح كيف تجاوز النموذج المقترح مشكلة الاعتماد على طرف واحد في تشكيل الدليل مع أقل تأثير لجميع الأطراف على مستوى الأداء أو الخصوصية، وكيف أثبتت طريقة توزيع تابع التقطيع SHA-2 فعاليتها في عدم قدرة أي طرف على إنكار الدليل أو التعديل فيه.

**الكلمات المفتاحية:** الحوسبة السحابية، التقنيات الافتراضية، التحقيق الرقمي، الطب الشرعي السحابي، الأدلّة الرقمية.

## 1. Introduction

Information technology has recently seen great dependencies on the architecture of cloud computing to provide its requirements at the level of individuals and organizations, which have increased the challenge for cloud service providers in providing the changing and rapid requirements of customers in several respects, the most important of which are information security and privacy protection. according to a study (E. Johns, 2020) which indicated that 46% of businesses (small and medium-sized companies) have been subjected to many types of cyber-attacks. The biggest fear among the technologists in the organizations was the protection and security of information. While the data are located on remote servers the investigators are encountering difficulties to guarantee data privacy and security, which in turn led to the development of a new trend in the field of information security and digital evidence, namely cloud forensics. Cloud forensics investigation is a recent science, and NIST defined this science as the intersection of cloud computing science with digital forensic science (K. Ruan, 2012).

In the model that we proposed, we assumed the most difficult cases, which include eliminating the total dependence on the cloud provider in digital investigation processes without neglecting the legal and regulatory aspect of the work taking into account cost and privacy, which led to a model that confirms the validity of the evidence with the least possible consumption of resources.

### 1.1 Research Objectives

The main objective of this research is to reach a digital evidence exchange mechanism between the involved parties in the cloud, which guarantees access to an integrated investigation at the lowest possible costs and with the least violation of user privacy, service provider privacy, and the related local law on information crime. The proposed model will provide a method for sharing and ensuring evidence, taking into consideration important previous studies in the field of digital evidence on the cloud.

### 1.2 Research Problem

Having a powerful framework or mechanism in a high dynamic environment such as cloud computing environments while maintaining high performance and high privacy is an urgent need for investigators, users and CSPs. Also, having digital evidence is considered a challenge in cloud computing fields especially after the very high expansion of cloud infrastructure as a response of the high demand on cloud services, this high demand enforces in a way or in another CSPs to have a cooperation with other CSPs in order to meet cloud user requirements, this cooperation made another challenge with cloud forensics that cloud investigator must deal with it.

Because multiplicity of parties that are related to cloud service without having a powerful mechanism to preserve the digital evidence without high impact on performance and privacy among clouds and users that digital investigator can depend on to track evidence. I embarked on this research to find a mechanism to exchange digital artifacts as an evidence with the participation of all the involved parties in the formation of the evidence making use of SHA-2 hashing algorithm and ECC encryption algorithm to protect the integrity, privacy and confidentiality. Based on the above the main research question are:

1. How digital evidence can be exchanged and preserved in cloud computing environments?
2. What is the impact of exchanging the evidence on performance and privacy?

   To answer these questions, we have to answer another important question which is:

1. What are the artifacts that cloud providers can exchange to form evidences? And what is the impact of collecting these artifacts?

### 1.3 Research materials and method

While researches on cloud forensics field are rare, we depend on these researches and other previous studies in digital forensics field to form the proposed mechanism. In this paper we divided the efforts to two sections, the first one is about selecting the digital evidence from some previous studies that tackle this issue and implementing some monitoring and digital forensics tools like Splunk, FTK Imager to acquire artifacts that could be implemented in our mechanism as an evidence for testing purposes, this is done by dividing this stage to three scenarios, where every scenario tackle different types of artifacts. Then a deep look has been made on previous studies like Progger, Flogger, OCF and others to build more

comprehensive mechanism that cover the main weakness point in these mechanisms which is depending on CSP to form the evidence, then we implemented this proposal using C# on Windows environment to test the efficacy of the solution, then we compared the, then we compared the results of the mechanism with the most powerful forensics mechanisms.

**1.4 Research structure:**

This research will pass through out different sections to reach the proposed mechanism which are:

1st: background: define the key terms of this study.

2nd: previous studies: summarize the most common studies in the field of cloud forensics. It will demonstrate 4 studies which provide different ways to preserve evidence.

3rd: demonstrate the proposed mechanism from a theoretical aspect, and it covers:

- The assumptions and attacking possibilities that the study relies on.
- Choosing artifacts to monitor and exchange
- How the proposed mechanism overcome the full dependency on a single side evidence provisioning?
- Mechanism scheme.

4th: practical implementation, and it covers:

- Three scenarios so we can deal with digital artifacts as digital evidence.
- The consumed time of mechanism implementation.
- comparison between cloud forensics mechanisms

5th: results, conclusions and recommendations

## 2. Previous Studies

Many previous studies have presented several models that contributed to the development of mechanisms for preserving digital evidence on the cloud, but most of them have focused on developing a forensic model which provides the evidence by one side, which is the cloud service provider.

Zhang presented a forensic tool "Flogger" (O, Zhang, 2012) that can collect evidence over multi-layers at the same time on cloud environment (application, layer, virtual machine layer, physical machine layer and cloud provider), this tool also depends totally on the CSP to provide the evidence.

Zawoad and Hasan (S. Zawoad, 2013) have presented a powerful mechanism that assumes any involved part as a dishonest part and can provide a manipulated evidence to the court, but at some point the model depends on the CSP to provide the proof of evidence which may be tampered by the CSP itself.

K. Ryan et al have presented "Progger" (K. Ryan, 2014) which is a monitoring tool that tracks every single operation (open, close, read and write), this tool can be placed at the VM as well as PM taking

into consideration that the security that the PM (host) can provide and the availability of the evidence are better than the single VM, which means that the tool can provide the evidence from two sides. On the other hand, the impact on performance is big, this is because every operation must be signed with a signature before executing it which means system latency in carrying out system calls and operations.

Zawoad and Hasan also presented another cloud model under title (Open Cloud Forensics Model for reliable Digital Forensics), the study also assumed that the CSP is honest and can be trusted (S. Zawoad, 2015), which means dependency on the CSP in providing the evidence.

## 3. Background

**3.1 Virtualization Technology** is the simulation of the software and/or hardware upon which other software runs (K. Scarfone, 2011). This simulated environment is called a virtual machine (VM).

**3.2 Cloud Computing** defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management efforts (P.Mell, 2011).

**Cloud Computing Service Models** Cloud computing services can be presented in many types. The three main models are Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

**3.3 Digital Evidence and Digital Forensics** The application of science to the identification, collection, examination, and analysis of data while preserving the integrity of the information and maintaining a strict chain of custody for the data (NIST, K. Kent, 2006), (M. Hewling, 2013).

**3.4 Cloud Forensics** Cloud forensics is a cross-discipline of cloud computing and digital forensics. So, cloud forensics is the application of scientific and technological concepts and the derivation of appropriate methods to reconstruct the event occurring in the cloud environment through the implementation of digital investigation processes (M. Herman, 2014).

### 4. The Proposed Mechanism

We will demonstrate the assumptions that we depended on to develop our mechanism then we demonstrated the general scheme to reach the practical implementation in order to test the validity and the time consumed to preserve and share evidence.

**4.1 Attacking Possibilities:** the attacks vary according to the provided cloud service model as well as the institution's business model on the cloud. For instance, the evidence that must be monitored in a SaaS service model for a cloud application such as Google Documents will differ according to the nature of the evidence that must be monitored for a SaaS cloud application such as Evernote and will differ

significantly from Microsoft Azure's virtual machine rental service. Consequently, the tools and methods used to capture this evidence will be different.

we assumed the following cases:

- An attacker accessed a file stored on an Azure cloud and modified it.
- An attacker modified the configuration file of a server hosting PaaS software.
- An attacker can access the registry files and make some modifications to them.

Note that all parties are not trusted, and any party can be suspected. Thus, any party can present to the court forged evidence that cannot be relied upon in any case.

**4.2 General Perspective**: The proposed model is based on selecting the most important evidence that can lead investigators to an integrated cloud investigation, then implementing an exchange mechanism for this evidence and confirming its validity. Therefore, there will be amendments in the service agreement between the customer and the cloud provider to give the provider access to some operations for monitoring purposes (such as access to History records) as well as the user's ability to access some of their evidence on the cloud. The evidence is selected to ensure good performance and high privacy (for example: in the case of network monitoring, the packet header will be monitored and not the entire data) at the same time, this process may take up a lot of space, so IPsrc, IPdes, Portsrc, Portdest, Time, can be satisfied.

**4.2 Selecting the artifacts:** There is a wide variety of types of evidence that can be collected, from the smallest unit such as a log containing the port address to imaging a full copy of a hard disk, so we have filtered the most important evidence that can be collected according to the cloud model. In Appendix-A, we reviewed cloud evidence that could help investigators to pull out a copy of the virtual device (the important movements) that was running or was used to commit the crime.

**4.3 Publishing the proof of evidence:** According to the assumption that all parties are suspects, the best way to prove the validity of the evidence is through the parties' participation in forming evidence or proof of evidence, for example, If a cloud service provider wants to form evidence that one of its customers has reached another cloud service provider in a federation (or another CSP), then they will form the evidence with the customer and to ensure that the file has not tampered the evidence, the SHA-2 hashing method will be applied to the evidence on the customer, the first provider, and the second provider. Next, in case of a hash match, the evidence will be considered valid, and the evidence will be stored and inserted in a separate database, then the external storage database will generate the hash value for the newly inserted record.

**4.4 Accessing the evidence**: for each device, there is a log file in which all the processes that must be monitored are recorded to ensure any subsequent investigation. A copy of this evidence can be at the customer's device and it can remain with the service provider, but in both cases upon completion of the work (or during a period of a specified time) the evidence must be sent with the hash value to the

customer, the customer calculates the hash value of the received data from the provider, then both the cloud provider and the customer send the hash value to a third party, "Amazon S3" as an example, which provides us with a hash-value for the newly inserted record. Amazon sends the resulting hash-value with some attributes of each record table-1 to the customer and the CSP to match the resulting hash value.

note: to enhance security we can use elliptic curve cryptography algorithm (ECC) because it achieved better results compared to RSA as what it is mentioned in (M. Pourvahab, 2019) and (R. Sinha, 2013).

A copy is sent to a third party, the third-party stores the hash values in records in the database as follows:

**Table (1) inserted record into a third-party database.**

| R.id | CSP.1 | HValue | T.CSP.1 | Port | USR.ID | HValue | T.USR.7 | Port | MD5 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | IP/address | #Sv23efd | 10:2:2 | 44 | IP/ID | #Sv23efd | 10:2:3 | 70 | #554ge |

The final step is that the CSP and the customer send each other the calculated hash value of the data received from the third-party (figure-1). Thus, the validity check process of the evidence has finished (table-2).

**Table (2) forming evidence operations.**

| Letter | Operation |
|---|---|
| A | by the cloud service provider, the evidence is encrypted using the shared key between the client and the CSP using elliptic curve cryptography.<br>The hash value is calculated using SHA-256.<br>A copy of the output is sent to the storage server. |
| B | by the user, the evidence is encrypted using the shared key between the client and the CSP using elliptic curve cryptography.<br>The hash value is calculated using SHA-256.<br>A copy of the output is sent to the storage server. |
| C | The data received by the cloud service provider and the customer is inserted according to Table1.<br>Calculate the hash value and send the results to both the provider and the customer with the data. |
| D | The user sends a copy of the hash to the service provider for the matching process. |
| E | The cloud service provider sends a copy of the hash to the user for the matching process. |
| F | In the case of a match, the user sends to the service provider that the data is correct and the evidence is valid, and the process can take place according to the application. |

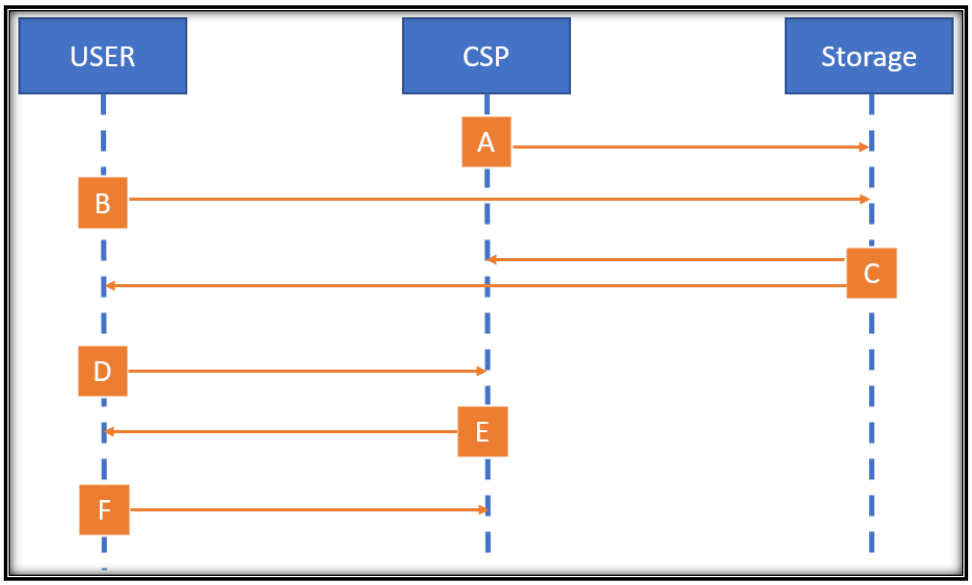



**Figure (1) the proposed model steps**

We can notice that the same process can be performed to the log itself, not this is because the evidence can be formed with the least possible consumption of resources in cooperation with the cloud user, such as the evidence mentioned in Appendix A.

It must be noted, the same process can be done for the evidence itself and not just for the value of the evidence's exfoliation, because the directory can be shaped with the lowest possible resource consumption in collaboration with the cloud user, such as the evidence in Appendix-A. also The same process can be done for the evidence itself and not just for the value of the evidence's exfoliation, because the directory can be shaped with the lowest possible resource consumption in collaboration with the cloud user, such as the evidence in Appendix a, the other thing is that the directory is configured with the cloud service provider

## 5. Practical Implementation

**5.1 testing tools:** we choose the monitoring tools that can give us the outputs as raw data as an input to another tests(table-3)

**Table (3) practical tools**

| Purpose | Tools |
|---|---|
| Measure performance and consumption | Zabbix 4.4.6 – Zabbix Appliance |
| Memory and Processes Monitoring | Splunk |
| Hosting Environment | VMware Workstation 15.5 |
| Database Management System | SQLserver2014 |
| Encrypting and Decrypting Data | C# Application |

| Purpose | Tools |
|---|---|
| Imaging RAM | Memory Dump |
| Imaging Hard Disk | FTKImager |

**5.2 Scenarios Practical implementation:** Because the variety of digital evidence, we tested many possible evidences and measured the consumed resources in order to compare the proposed mechanism with other mechanisms, taking into consideration that the evidence is may be different between the mechanisms. So, we divided the practical implementation to three scenarios as we can see from table-4.

**Table (4) the goals from the suggested scenarios**

| Scenario | Goal |
|---|---|
| Scenario 1 | This scenario aimed to measure the consumed resources of having the hash value as a digital evidence for some files and the consumed resources for sending this evidence to cloud service providers. Also, the scenario aimed to keep the high privacy of having the evidence without having the original file, which means better privacy protection among several cloud service providers.<br>this scenario can be used between several cloud providers as a method to fight the illegal content on the internet, this can be achieved by sharing the hash values between CSPs. |
| Scenario 2 | This scenario aimed to reduce the consumed resources in monitoring RAM such as the traditional RAM memory dump tools. This is done by replacing memory dump operation with monitoring the critical ram operation the user may ask for.<br>In case of IaaS: This scenario can be used to monitor the mapping between VM on the same virtualization host to track VM acquisitioned blocks<br>In case of SaaS or PaaS: This scenario can be used to monitor a specific process instead of imaging the entire RAM memory |
| Scenario 3 | This scenario aimed to reduce the required hard disk to save evidence by choosing the minimum required evidence from files properties. Also, the scenario clarify how can user and CSP implement the proposed mechanisms to save evidence. |

**scenario (1)** Hashing several files and inserting them into a third-party database then performing the matching process. we performed SHA-256 hash on 23 different files (pictures, videos, books) and stored them in a local database, and then sent them to third-party storage. figures 2-3-4-5 demonstrate that the resources consumed by both user and CSP (this hash value can be used as a digital evidence instead of the original file).

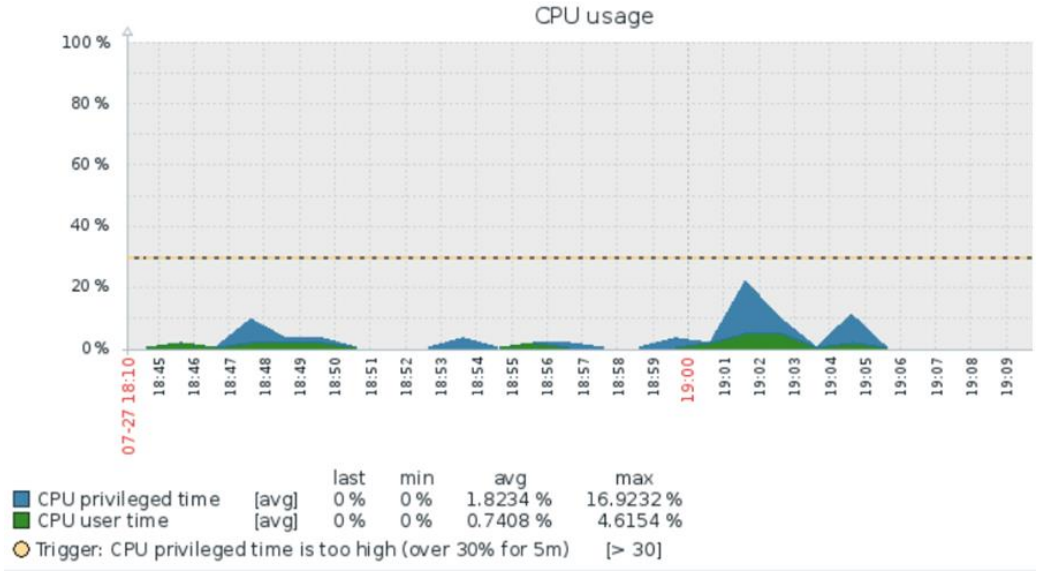
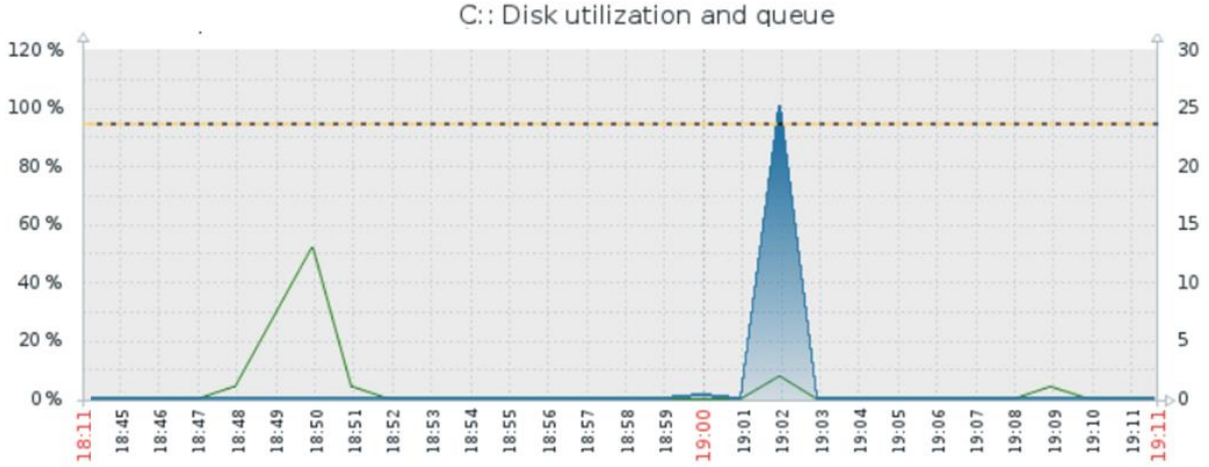



**Figure (2) CPU Usage for scenario-3**


**Figure (3) Memory Utilization for scenario-3**




**figure(4) Disk utilization and queue**

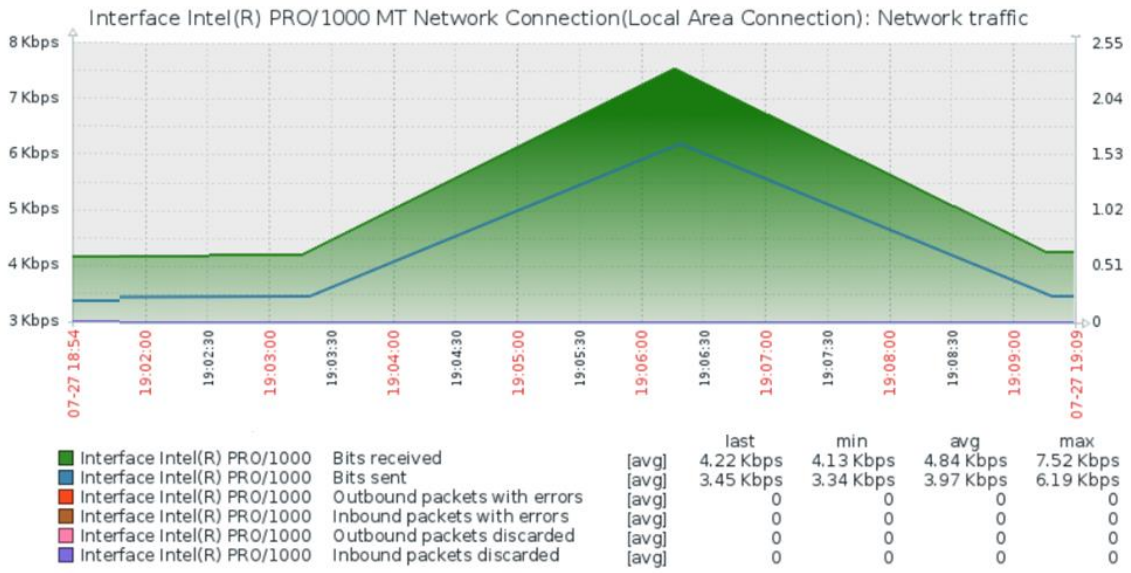



**Figure (5) Network traffic**

we can see that the hashing operation did not take a long time, this is because the hash operations performed on the required data for this service but not everything.

We can notice from figures 2-3-4-5 that the consumed resources are very low compared to other methods. According to (Dykstra, 2012) study, the process of making a copy for the entire hard disk or the entire contents of the random memory will consume a lot of time also, it leads to taking over more space on the hard drive as well as the amount of processing.

**Table (5) resource consuming comparison between traditional method and the proposed mechanism**

| Resource | Traditional methods | Proposed mechanism |
|---|---|---|
| CPU | More than 50% during imaging operations | Additional 1.8% consumption |
| Memory Utilization | no additional load mentioned | Additional 7% to run the database and hashing program |
| Hard Disk allocated Space | 82186 KB (the entire data) | 2.51 KB |
| Bandwidth / network traffic | 682.08 Mbit | 0.019609 M.bit<br>peak of sending 7.52 Kbit/sec<br>peak of receiving 6.19 Kbit/sec |

As we can see from table-5, the consumed resources are too low in general, this is because the minimizing of the artifacts to reach a full investigation without having the original files. Next step is to measure the consumed time (table-6) of store these hash values in a third-party database to act as a

central database that receive and send the hash values of the illegal content (23-files) to the known CSPs (synchronization between CSPs).

**Table (6) Consumed time to synchronize hash values**

| Operation | Time Consumed |
|---|---|
| Consumed time to sync CSPs with third party | 63.47 ms |
| Consumed time to sync third-party with other CSPs | 35.98 ms |

**scenario 2:** monitoring critical processes to avoid saving the entire memory. Using Splunk figure-6, Windows Explorer processes were monitored for 33 minutes, so that the output was 25080 KB, in which the tool recorded 52384 records. In contrast, if we want to export a copy of a Memory-Dump on the same device, the result will be 24-GB (depending on the size of the device's memory), (Dykstra, 2012).

The output of such monitoring operation will help investigators to reach the evidence of service (PaaS or SaaS) over the cloud provider by tracking the RAM for a specific operation without ever having the entire image of RAM memory. This is because tools such as Splunk can give the investigator the required evidence from memory for a critical process, which means reduction with resources and high flexibility, also this way can avoid investigators privacy violation for the service provider because the outputted log file for this method is just related to the involved process and not any other process that works on the CSP system.

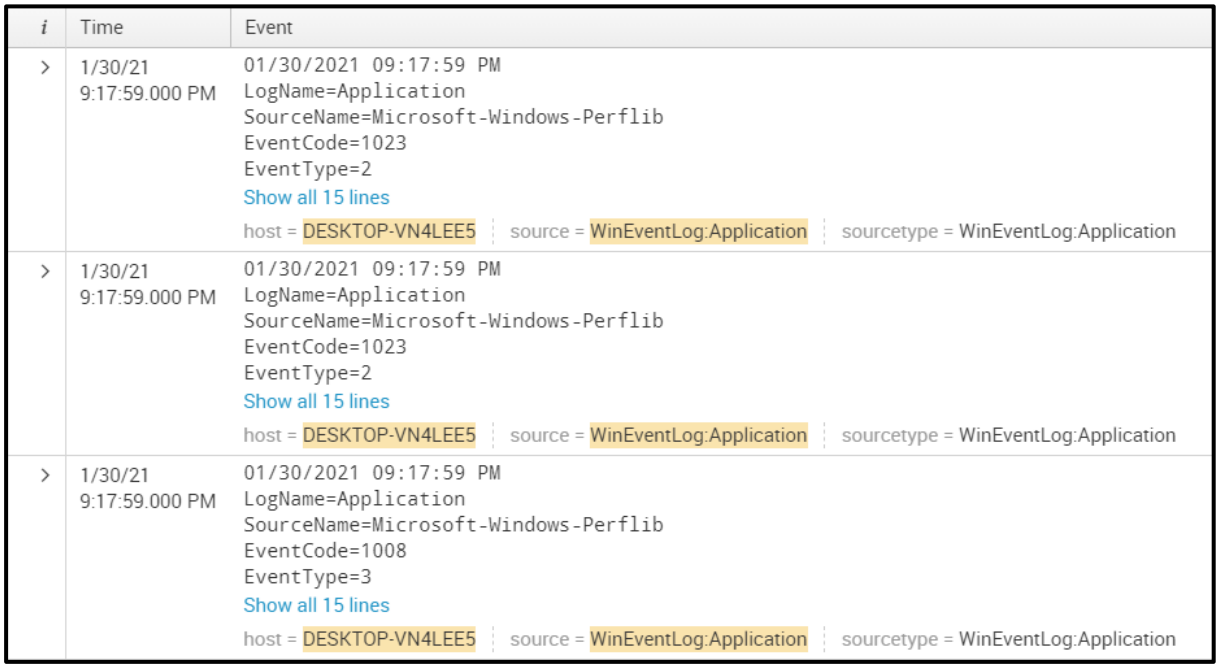



**Figure (6) Splunk memory results for Windows Explorer**

**scenario 3:** Using the PowerShell tool, we can extract the properties of a file located on the cloud figure-7.

```
PS C:\Users\Waseem> Get-ItemProperty test-log-file-on-cloud |  Format-list -Property * -Force
```

**Figure(7) PowerShell command to extract file properties on the cloud**

It is possible to extract the properties of a file located on the cloud, where investigators can use these properties in order to reach the required evidence without having the original file. This what saves user privacy from unwanted violations during investigations. These properties can be last modified, the file is read-only, the number of bytes in the file, owner of the file, filename, file path, etc. In most cases, the volume of the resulting file will not exceed 2 KB. On the other hand, the entire file must be copied, or even an image can be taken from the hard disk to acquire the evidence figure-8.

```
Name              : test-log-file-on-cloud
Length            : 11711
DirectoryName     : Z:\
Directory         : Z:\
IsReadOnly        : False
Exists            : True
FullName          : Z:\test-log-file-on-cloud
Extension         :
CreationTime      : 7/27/2020 7:39:52 PM
CreationTimeUtc   : 7/27/2020 4:39:52 PM
LastAccessTime    : 7/27/2020 7:39:52 PM
LastAccessTimeUtc : 7/27/2020 4:39:52 PM
LastWriteTime     : 7/27/2020 8:17:01 PM
LastWriteTimeUtc  : 7/27/2020 5:17:01 PM
Attributes        : Archive
```

```
Name              : test-log-file-on-cloud
Length            : 11699
DirectoryName     : Z:\
Directory         : Z:\
IsReadOnly        : False
Exists            : True
FullName          : Z:\test-log-file-on-cloud
Extension         :
CreationTime      : 7/27/2020 7:39:52 PM
CreationTimeUtc   : 7/27/2020 4:39:52 PM
LastAccessTime    : 7/27/2020 7:39:52 PM
LastAccessTimeUtc : 7/27/2020 4:39:52 PM
LastWriteTime     : 7/27/2020 7:36:44 PM
LastWriteTimeUtc  : 7/27/2020 4:36:44 PM
Attributes        : Archive
```

**Figure (8) the output of PowerShell command**

To have the best practice of this scenario, we hosted a text file on a 5-tera storage from an external organization that depends on Microsoft Azure Cloud services, then we hosted Function-as-a-Service on the cloud to calculate the hash value for the properties of the file at the same user logging in time and user signing out time, then the function sends the hash value to the user, the user is already calculated the hash value of the same properties when logging in and when signing out. In case of mismatching, user application will inform the user that there are some changes within the file, which means file manipulating or unauthorized access has been detected. In this scenario we asked an employee with superuser permissions at the external organization to access the file and edit it after user singing out. When the original user logs in again, the user's hash function will calculate the hash value, at this point the user will find out the hash value mismatching.

Where the CSP can deny or evade this mismatching, our proposed mechanism can avoid this evading from both user or provider, this can be done by involving a third party that keeps hash value signed with other hashing keys in order to save the evidence integrity.

## 6. Results

By making use of the outputted data from previous scenarios, we have tested the proposed model in order to reach a clear view to the ability and how successful is this model. So, using C# we have built an application to calculate hash values then encrypts the output using ECC algorithm

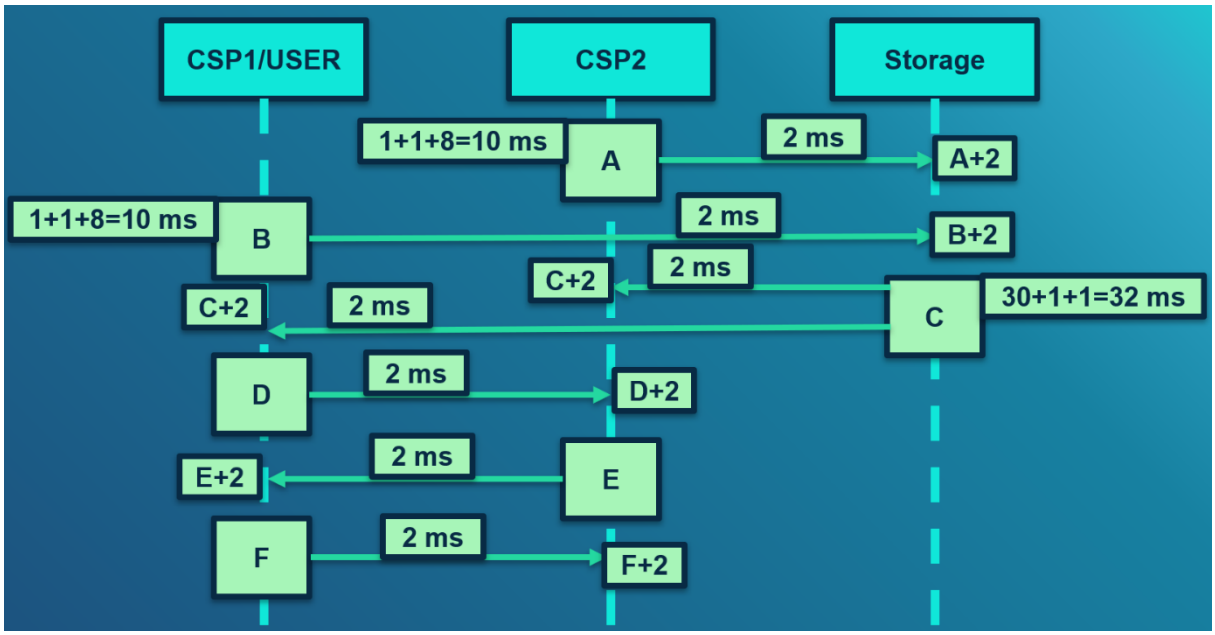



**Figure (9) the consumed time (for every single operation) to exchange the evidence**

The most significant change that can be touched in this mechanism is the lightness forming and exchanging the evidence using SHA-2 and ECC, table **(7)**

| Littler | Operation | Time consumed |
|---|---|---|
| A, B | Hashing the evidence + Encrypt the result + save the result | 10 ms + 2 ms (sending the data over 100-mb bandwidth) |
| C | Decrypting the result + hashing the inputs + save the result | 32 ms + 2 ms (sending the data over 100-mb bandwidth) |
| D, E, F | Matching evidence | 2 ms (sending the data over 100-mb bandwidth) |

As we can see from the theoretical view of the proposed mechanism, the usage of distributed SHA-2 hash values with ECC encryption algorithm that involves every stakeholder in the process to form the evidence and the proof of the evidence have reached a robust mechanism that can save the evidence and share it with customers or other CSPs with minimum resource consumption and least privacy violation (table-8)

Table (8) comparison between cloud forensics mechanisms

| Property | OCF | SecLaaS | Flogger | Progger | Our model |
|---|---|---|---|---|---|
| Provenance | Provided by CSP | Provided by CSP | Provided by CSP | Provided by client or CSP | Provided by client and CSP |
| Privacy | Low | High – encryption is implemented | High – encryption is implemented | Low when depending on CSP | High – encryption is implemented and only what the client accepts to monitor |
| Resource Consuming | Low (depends on synchronization intervals) | Low | Low | Low | Low (just the accepted evidence are involved) |
| Performance | Maintain high performance | Maintain high performance by depending on separated software | Maintain high performance by depending on separated software | High impact on system operations | Maintain high performance by depending on separated software |
| Evidence integrity | Achieved | Achieved | Not achieved | Achieved by signing every operation | Achieved |
| Evidence Availability | Available by Read-only APIs | Available by Read-only APIs | Available by Read-only APIs | Available in case of Progger works on client VM | Available by Read-only APIs |
| Cloud Layers | IaaS, PaaS, SaaS | IaaS, PaaS, SaaS | IaaS, PaaS, SaaS | PaaS (or System Calls) | IaaS, PaaS, SaaS |
| Timing | NTP | NTP | Local between PM and VM | NTP | NTP |

## 7. Results Discussions:

**7.1 Consumed time:** as we can see from figure-9 the consumed times in general are very little compared to other mechanisms, this is because our mechanisms depend on the hash values that can keep the evidence safe from any tampering, also we choose the least artifacts that can provide the required evidence instead of having the entire physical machine or stopping the machine to have the evidence.

**7.2 Considering the properties of our model**

**Provenance:** as we can see that all the previous studies depend on a single side provenance which is in most cases the CSP, while in our model the CSP is a part of the operation and the evidence

cannot be admitted without user (or other CSPs) participation, also the user will have the proof of the evidence as well as the CSP to guarantee that no one side can tamper the evidence.

**Privacy:** in most of previous studies the encryption is implemented in a way or another, but what distinguishes our study is the that the monitor tool records just the required artifacts instead of every artifact, also using the hash value as an evidence without having the original file will keep user privacy in a safe zone.

**Resource Consuming**: as we can see from the above three scenarios, the resources that are required to have the evidence is too small especially if we dealt with it as raw data, also it is clear from figure-9 that implementing hashing and encryption did not take a long time on the implementation environment.

**Performance:** the monitoring tools that depend on in this mechanism can rely on virtual machine within the CSP infrastructure without causing performance drawbacks on other virtual machine and also without the user incurring any additional cost except hashing

**Evidence integrity:** we achieved the evidence integrity by implementing the hash operation over two stages, the first one to insure that the evidence is true and valid, and the second one is to prove that the proof of the evidence is safe and cannot be tampered without user or CSP attention.

**Evidence Availability:** our study and the most of previous studies offered or suggest a read-only APIs as method to reach the evidence ether for user, CSP or investigator.

**Cloud Layers:** most of the previous studies suggested tools that works on the main three cloud models except Progger.

**Timing:** because timing is a critical field in crimes, we depend in our model on standard protocol to synchronize and record times on NTP protocol as a well-known way and easy implemented protocol.

## 8. Conclusion

It is not possible to reach a complete mechanism that ensures a fully integrated digital investigation, and there is no one framework can be applied to all types of digital businesses, but every digital business model has an investigation framework that fits according to the cloud model, local law, international and international agreements regarding information technology. The application of traditional investigation models will consume a lot of resources, both for the customer and the cloud service provider, thus a heavy burden in sharing and processing digital evidence. Also, most of the tools are only available from the service provider without the customer intervention, meaning that the evidence formation is only by the provider, which makes validating the evidence very difficult and unreliable on the part of the cloud provider. Our study provided a mechanism that focuses on electing and gathering the most important digital evidence which is necessary to perform an integrated digital investigation, then

share these evidences in a robust method that save this evidence from tampering and also guarantee the integrity and the availability of the evidence.

## 9. Recommendations

- The use of Hash-Value distributed over more than one cloud can be helpful to ensure the integrity among the involved parts.
- electing just the required artifacts that may be involved with the crime scene formation, not every artifact on the device.
- Because logging the right information at the right time is more critical than extracting them, we recommend using ECC encryption mechanisms as a fast, strong and lightweight method to encrypt evidence.
- Electing just the most critical evidence that investigator may use to track the crime instead of having the entire crime environment will help in reducing the required resources to monitor, exchange or investigate artifacts.

## 10. References


- E. Johns, "Cyber Security Breaches Survey 2020", Department for Digital, Culture, Media and Sport, 2020.
- Josiah Dykstra, Alan T. Sherman, 2012, acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques International Cyber Defense Lab, University of Maryland.
- K. Kent, S. Chevalier, T. Grance, and H. Dang, 2006, Guide to integrating forensic techniques into incident response, National Institute of Standards and Technology (NIST).
- K. Ruan ,J Carthy,T Kechadi, 2012, M Crosbie, Cybercrime and Cloud Forensics: Applications-ch.3, IGI Global.
- K. Scarfone, M. Souppaya, P. Hoffman, 2011, Guide to Security for Full Virtualization Technologies, National Institute of Standards and Technology (NIST).
- M. Herman, M. Iorga, 2014, Cloud Forensics Challenges, National Institute of Standards and Technology (NIST).
- M. Hewling, 2013, Digital forensics: an integrated approach for the investigation of cyber/computer related crimes, University of Bedfordshire.
- M. Pourvahab, G. Ekbatanifrad, 2019, Digital Forensics Architecture for Evidence Collection and Provenance Preservation and in IaaS Cloud Environment Using SDN Blockchain Technology, Department of Computer Engineering, Islam Azad University.

- O. Zhang, M. Kirchberg, 2012, How to Track Your Data: The Case for Cloud Computing, HP Laboratories.
- P. Mell, T, Grance, 2011, The NIST Definition of Cloud Computing, National Institute of Standards and Technology (NIST).
- R. Sinha, H. Srivastava, 2013, Performance Based Comparison Study of RSA and Elliptic Curve Cryptography, International Journal of Scientific & Engineering Research.
- Ryan k., M. A. Will, Progger: An Efficient, 2014, Tamper-Evident Kernel-Space Logger for Cloud Data Provenance Tracking, The University of Waikato.
- S. Zawoad A. Kumar R. Hasan, 2013, SecLaaS: Secure Logging-as-a-Service for Cloud Forensics", University of Alabama.
- S. Zawoad, R. Hasan, 2015, An Open Cloud Forensics Model for Reliable Digital Forensics, Department of Computer and Information Sciences, University of Alabama.


## Appendix-A

We formed this table using several previous studies that tackled digital evidences and digital forensics tools then we added some critical evidence that we faced during our practical implementation.

| Artifact / Evidence | | purpose | The cloud model that can provide | | | |
|---|---|---|---|---|---|---|
| item | Data type | | SaaS | PaaS | STaaS | IaaS |
| User ID / Credentials / Requester / authorized user | Integer/String | Canonical user id of the requester. User tracking among several servers or CSPs | X | - | - | - |
| Timestamp | Time/date | To prevent Synchronization attacks, replay attack | X | X | X | X |
| Digital Signature / Cryptography Keys | Long integer / hash | In case of unauthorized access | X | - | - | - |
| Operation ID | Integer / Char | Operation tracking / Such as SOAP.operation, and REST.HTTP_method | X | X | X | X |
| Operation Type | char | Like Create, Read, Edit or Remove GET_Object | X | X | X | X |
| Hash Value for the Data / document / file (just like the case of Syrian servers and) evidence hash | Hash value / long-Integer / string | integrity check offline or online like we present in section | X | X | X | X |
| Port | Short / integer | to track the service, in case of many services on the same application / | X | X | - | X |

| Artifact / Evidence | | purpose | The cloud model that can provide | | | |
|---|---|---|---|---|---|---|
| | | software / platform | | | | |
| Session ID | String | Link every operation with sessions | X | X | - | X |
| live time | Time | It can help using deep learning according to previous information that we can use in behavior study and intrusion detection systems | X | X | X | X |
| Location in the document / file | String | Determine the location of document editing | X | X | - | - |
| External URLs | String | To determine any external connection with the document | X | X | - | X |
| Permissions | Char | | X | X | X | X |
| Web Browser | Char | Digital Security Certificates | X | X | - | - |
| Used OS | Char | | X | X | - | - |
| Config Files | XML / Text | In case of main config manipulation | - | X | - | X |
| Log files / Access log files | XML / Text | Tracking | X | X | X | X |
| Hash of Access Log files | Hash | Integrity check in case of manipulation | X | X | X | X |
| Security Certificate | Char | Verify the | X | X | - | - |
| Request ID | Integer | The request ID is a string generated by Amazon S3 to uniquely identify each request | X | X | X | X |
| HTTP status | Integer | HTTP status code in response message | X | X | - | - |
| Error log files | XML | Tracking any unusual activity | X | X | X | - |
| Bytes Sent | Integer | Number of bytes sent in response message | - | - | - | X |
| Object Size | Integer | Total size of the object requested | X | - | - | - |
| Total Time | integer | Number of milliseconds the request was in flight from the server's perspective | X | X | X | X |
| Turn Around Time | integer | Number of milliseconds that Amazon S3 spent processing your request | - | - | - | X |
| Last Access | Date | Check out when last update has been done | X | X | X | X |
| Is Read Only | Boolean | | X | X | - | - |
| time_micros | Integer | Time taken by a request to complete in microseconds | X | X | X | X |
| c_ip_type | Integer | The version of IP used i.e. either IPv4 | - | - | - | X |

| Artifact / Evidence | | purpose | The cloud model that can provide | | | |
|---|---|---|---|---|---|---|
| | | or IPv6 | | | | |
| Virtual Switch | integer | Tracking the attacker among several virtual switches in the same CSP. | - | - | - | X |
| Hyper-Visor ID | Integer | Tracking the attacker among several hyper visors in the same CSP. | - | - | - | X |
| VM-Hypervisor ID | Integer | To build Evidence map in order to track evidence among several hypervisors | - | - | - | X |
| Cloud Path | Strings | This field lists the full path of the file within the Cloud Drive. | - | - | X | X |
| File Name | Strings | This field lists the name, including extension, of the file stored on the Cloud Drive. | - | - | X | X |
| Packet Header | XML | Tracking application | - | - | - | X |
| Mac Address / Device ID | String | Tracking device internally | - | - | - | X |