Dismissing Poisoned Digital Evidence from Blockchain of Custody

David Billard

University of Applied Sciences in Geneva
HES-SO
Geneva, Switzerland
Email: David.Billard@hesge.ch

***Abstract*—This paper presents a solution to dismiss a digital evidence from a permissioned blockchain-based legal system, serving as evidence chain of custody. When challenged into court, a digital evidence can be entirely dismissed, as well as all the procedural acts originating from this evidence, including personal gathered data. Since a blockchain, by design, cannot be altered, this paper proposes an alternative solution based on an access control to the blockchain. This solution relies on an additional structure, linked to the blockchain, representing the history and current legal state of the case. Access to the blockchain is controlled by first interrogating this additional structure in order to serve only legally accepted evidence. Therefore, an evidence stored into the blockchain is not destroyed, but is no longer visible nor accessible. Furthermore, evidence data is separated from the blockchain transaction's payload, that holds only metadata, and this separation reinforces privacy protection. The solution presented in this paper is explainable to all parties to a court trial.**

Keywords–Digital Evidence; Blockchain; Chain of custody; Privacy.

I. INTRODUCTION

This paper focuses on an often-forgotten aspect of digital evidence handling, when a court dismisses an evidence from a trial. Multiple reasons can lead to dismiss an evidence: it can be challenged by a party during an investigation or in front of the court, have an expired delay if it is time-bounded, or simply dropped by the prosecutor.

Of course, different countries apply different laws, but let's take a simple example, quite universal. Bob is suspected to hold illegal child pornography material. A warrant is issued and a police search is conducted at Bob's house. During the search, a hard drive is seized and following police procedure, the drive is registered. Since this police body is a modern one, a chain of custody is initiated into the blockchain-based evidence inventory software.

Digital forensics experts examine the drive and find connections with Alice, who seems deeply involved in child pornography. A police search is therefore triggered on Alice and a USB stick with a lot of inculpatory evidence is found at Alice's home. As required by the procedure, the USB stick is registered into the same blockchain-based software.

Much later in the investigation, a defense lawyer raises the legality of the first police search on serious grounds. The court follows the motion and the first police search is dismissed. Since the second police search is a direct offspring of the first, it is also dismissed from the case.

Now let's have a look at how to implement the dismissing of evidence when it is stored into a blockchain, since the blockchain does not allow for alteration, deletion or cancellation. Having a unique structure at hand, there exist at least two possible options in order to dismiss transactions.

The first one is to delete the whole blockchain and to issue a new blockchain, without the dismissed evidence. In practice, it means to start from the root block and re-issue all the subsequent transactions (excepted transactions linked to the dismissed evidence of course). Although it is theoretically doable, it means a huge effort of transaction and block validation, involving voting algorithms, and keeping track of all the blockchain intra references. This option is studied further in this work, but the reader can already notice that the computational complexity is quite significant.

The second option is to issue undo-transactions whose purpose is to indicate that the referenced transaction is void and cannot be used anymore. It means that the blockchain contains two categories of transactions:

- transactions for registering evidence;
- undo-transactions for dismissing evidence.

This technique of using undo-transactions is widely used in DataBase Management Systems (DBMS) for recovery or rollback purposes. Unfortunately, while it is well suited for DBMS, it brings some issues in blockchain-based systems.

The major issue concerns the verification of transaction validation. For a user to check if a transaction is valid, the user will have to verify if the chain of hashes and signatures has not been broken since a particular point in time (usually the begin of the blockchain). This check means that the transaction has been correctly entered into the system and has been validated following the rules.

But this check does not prove that the transaction is valid from a legal point of view: the evidence linked to the transaction may have been dismissed later. Therefore, the check process must continue until either: (1) it finds the undo-transaction, in which case the transaction is not legally valid or (2) it reaches the end of the blockchain, in which case the transaction is legally valid. The reader will notice that the computational complexity of this check is significantly higher than the single transaction verification protocol usually observed in blockchain.

There exists another perspective for solving this problem, with manageable complexity, relying on an additional structure recording the invalidated transactions, and a controlling structure granting or denying access to the blockchain.

When a transaction is invalidated, an undo-transaction is inserted into a *new distinct blockchain structure*, that holds all undo-transactions. The system is then composed of the

 ISBN: 978-1-61208-818-1

evidence blockchain and the undo-transactions blockchain.

In order to verify the validity of a transaction, the system first look into the undo-transaction blockchain if an undo-transaction exists for this particular transaction. If it exists, then the system returns an error and exits. If it does not exist, the system proceeds with the verification in the evidence blockchain. Checking *a priori* the undo-transaction blockchain has a lower overhead, directly connected to the number of invalidated transactions.

Solving the invalidation of transactions related to dismissed evidence is still not complete, since transactions' payload may contain sensitive data, which is considered as a privacy issue. This paper advocates that the blockchain storing the evidence should know only signatures, hashes and metadata about a case. All the content should be taken out from the transaction payload and kept in distinct, encrypted and secured structures. Thus, when a transaction is invalidated, its content can be safely erased without compromising the blockchain structure.

This paper is organized as follows: section II introduces some related works about blockchain-based systems designed for digital forensics. Then, in section III, the notion of tainted evidence, and what it implies, is presented. Further, section IV presents several solutions for dealing with dismissed evidence and their degree of workability. Section V exposes a solution based on two blockchains and an access control and, in section VI, the identification of tainted transactions is specified. Then, in section VII, the paper studies the privacy protection for this solution before concluding, in section VIII, with future works.

II. Related work

The idea of storing the chain of evidence into a blockchain has recently sparked a lot of attention from the digital forensics community. The blockchain is ideally fitted for legal evidence because the properties attached to legal evidence are embedded into the blockchain properties. In [1], the author lists the desirable properties of a blockchain transaction:

- *Immutability*. The blockchain cannot be tampered with, otherwise the tampering is detected. Although in [2] the authors are cautiously advising that immutability can only hold up to the cryptographic strength of the hash function used, it is still one of the major blockchain properties.
- *Provenance*. The assets embedded into a transaction have a provenance and any authorized reader can know where the asset comes from and how its ownership has changed over time. Data provenance is the representation of the origin of data, and its subsequent alterations.
- *Finality*. The blockchain holds all the references to an asset, its ownership, its validity.
- *Consensus*. When digital evidence is added to the blockchain, as a transaction, it is validated by the users of the blockchain. At that peculiar moment, all (or a vast majority) of voters agreed on the transaction outcome.

These properties adhere well to the concept of "chain of custody". The NIST, in [3], defines the *chain of custody* as "*A process that tracks the movement of evidence through its collection, safeguarding, and analysis lifecycle by documenting each person who handled the evidence, the date/time it was collected or transferred, and the purpose for any transfers.*"

Therefore, many authors tried to propose blockchain mechanisms in order to capture the chain of custody / evidence and to offer associated services.

In [4]–[7], the authors use blockchain in the context of Internet of Things forensics, in particular aboard intelligent cars. The blockchain purpose is to record data about navigation and provide evidence should accident occurs.

In [8]–[10] the authors propose architectures based on blockchain and smartcontrats in order to store evidence, or evidence metadata, into the blockchain.

In [11] the authors advocate for a loose coupling structure in which the evidence reference and its content are maintained separately. Only the evidence reference is stored into the blockchain, and the evidence data is stored on a trusted storage platform. this paper thrives for the same separation, in order to avoid privacy issues when facing deletion of evidence.

In [12], the author describes an architecture where evidence is stored in a Digital Evidence Inventory blockchain, and additional structures provide a global timeline to order evidences and a tentative of evidence rating. Each transaction is expressed as a CASE object [13] or an XML token [14].

Many researches [8], [10], [15]–[17] propose blockchain for holding evidence. However, none of these papers address simultaneously two important specificities of digital forensics: 1) evidence can be dismissed by court order and 2) evidence cannot be inserted or viewed by everyone.

III. Dismissing tainted evidence

An investigation or a trial is not a straightforward process and dismissing of evidence can be triggered by several causes, for instance a procedural issue like a 4th amendment violation for the USA, which in short states that any evidence illegally obtained should be excluded from a case.

Therefore, an evidence can be tainted by a breach of rights, and derivative evidence have to be dismissed, since it becomes "tainted" too. Some jurists refer to that situation as the "*fruit of the poisonous tree*".

A famous example is the *Mapp v. Ohio, 367 U.S. 643* [18] in 1961. In this case, Dollree Mapp's house was searched because it was assumed that a bombing suspect was hiding there. During the search, the police found a small number of pornographic books and pictures. Ms Mapp was arrested, prosecuted for possession of the books and found guilty (sentenced to one to seven years in prison). She appealed to the U.S. Supreme Court because the warrant was concerning the hiding of the bombing suspect, not the possession of pornographic books. The Court overturned the conviction, and five Justices held that the states were bound to exclude evidence seized in violation of the Fourth Amendment.

This case can easily translate into modern days with possession of pedo-pornographic material in a digital form. But besides the case itself, it's the impact of such decision on computerized systems, and especially when cases are large ones, that interests this paper.

With the current technology, the blockchain records evidence that is dismissed, which is not correct. The transactions related to the dismissed evidence must be deleted or made non-reachable. Some work, at Interpol in 2018 [19], but most notably in 2019 [20], devised a schema in a permissionless blockchain, like the bitcoin's one, in order to alter a block.

 ISBN: 978-1-61208-818-1

However, this scheme cannot apply easily in a permissioned blockchain because alterations have to be recorded and not all the transactions can be altered by anyone. An authorization mechanism must exist, thus the use of permissioned blockchain, which unfortunately prevent the use of the mechanism depicted in [20]. In [21], the authors review some ways of modifying the blockchain structure to allow mutability for GDPR (General Data Protection Regulation) constraints, alas destroying the alteration information.

The reader can imagine an Enron-like investigation put into a blockchain. The number of evidence items is staggering, and the number of people having access to the evidence is also very high. But the blockchain is precisely designed to hold a large number of evidence, as well as many users at the same time.

But how to prevent tainted evidence to be used by one party or another when the information of which evidence is tainted dissolves into the sheer number of evidence to process? How to prevent names and private data to be used when included into a tainted evidence?

The answer to those questions is a system that *controls the distribution of evidence data with respect to its legitimacy*.

IV. BLOCHAIN STRUCTURE V/S BLOCKCHAIN ACCESS

The blockchain, by definition, is immutable. Immutable roughly means that validated transactions cannot be modified without the alteration being detected. Therefore, how to proceed to undo a transaction, or a set of transactions?

This paper presents three scenarios that are feasible, at different costs: (1) rewriting the whole blockchain, (2) issuing undo transactions, (3) working on the blockchain access, not on its structure.

A. Rewriting the blockchain

Although ludicrous it seems at first, this option might be exploited in some blockchain implementations.

The validation of a transaction is done by consensus, more rarely by proof-of-work in the case of evidence blockchains. Consensus property originally means that more than a sufficient percentage of certified voters agreed on the outcome of a transaction. It's the turning point when a transaction, or more precisely the block containing the transaction, is validated. When a block is added to the system, it is unmovable.

Of course, in case of proof-of-work, with enough computing power and cryptographic effort, a majority of the voters can twist the system and prevent a block from being validated. It's a common threat in the crypto-currency world, and it is a real danger. But this attack is more an idle-threat in the case of blockchain used in digital forensics. As a matter of fact, legal systems rely on permissioned blockchains with voting algorithms and certificates, and no on mining and proof of work.

However, the problem at hand is not to modify the future chain, but to rewrite history, which means to re-validate every transaction block that was entered into the system since the block containing the transactions associated to the tainted evidence. That means to force the certified voters to vote again the same transactions which is doable if a blockchain is devoted to only one case and the voters are still the same and available. Which is not the usual setup seen in several related works, since a blockchain may contain information from several cases.

However, if doable, the cost of this operation is a one-time $O(n)$ where n is the length of the blockchain since the first transaction related to the tainted evidence. It means also that the blockchain is unavailable for use during this cleaning operation and the duration of the cleaning process might be long, depending on the validation schema used for transactions and blocks. It also means that the decision of justice to dissmiss an evidence is lost.

B. Issuing undo transaction

In DBMS systems, where ACID transactions are a central part, a committed transaction can be undone only by issuing a new transaction voiding the effects of the committed transaction. Undoing a committed transaction is far from trivial and leads to interesting problems, especially when failure occurs.

In the case of a blockchain holding evidence, one solution is to consider a "dismiss evidence" transaction or undo-transaction, in order to remove the evidence from the case.

Alas, it means that when a user wants to access an evidence, the system has to parse all the subsequent transactions in order to detect if a "dismiss evidence" transaction has been issued for this transaction. Practically speaking, it means that for each transaction T that is searched, or for validating a new transaction that references T, one need to parse the whole blockchain in order to eventually find if T is valid. The cost of this search is $O(n)$ where n is the number of transactions in the system. And this additional cost will occur whether there are, or not, invalidated transactions in the blockchain. It also means that if a user wants to have access to each transaction of the system, it will cost $O(n^2)$ in terms of verification.

C. Controlling the blockchain access

Instead of modifying the structure of the blockchain or its purpose, another way is to prevent a user to access tainted evidence. Let's name the evidence blockchain *InventoryTX*.

This paper proposes to add an additional structure, *InvalidatedTX*, that records the invalidated transactions, and a controlling structure *AccessTX* which is the access point to *InventoryTX*.

In order to access a transaction from the *InventoryTX* blockchain, the request goes through the *AccessTX* access point that first parses the *InvalidatedTX* blockchain. The cost for parsing *InvalidatedTX* is $O(m)$ where m is the number of invalidated transactions. In usual cases, m will be close to zero, thus the search overhead will be insignificant.

When a transaction is returned from the *InventoryTX* blockchain, it has the properties inherited from being in a blockchain, and the additional property that the transaction is legally sound and has not been voided.

V. THE ACCESS-BASED SOLUTION

This solution works with a majority of blockchain implementation because it does not modify the blockchain structure.

The payload of every transaction in *InvalidatedTX* contains the transaction ID related to a tainted evidence. It is recommended that each transaction in *InvalidatedTX* is signed by the jurisdiction issuing the removal of the tainted evidence.

The validation of each invalidating transaction is processed as in a normal blockchain, since the root of *InvalidatedTX*. Only the nature of the invalidating transaction differentiates it from a traditional blockchain.

 ISBN: 978-1-61208-818-1

An example might be the best way to illustrate the different components of the proposed solution. In this fictious case, the police searches Ms Marple's home. This woman is suspected to host a suspected man running from the police. Three evidence items are found at her home:

- Agent *Poirot* found a USB key with the searched man identity documents and 1000 bitcoins;
- Agent *Ness* found a notebook with pornographic contents and a hyperlink to a web server;
- Agent *Loch* found a love letter from the suspected man to Ms Marple.

Later, the web site is investigated by agent *Chris* and it contains drug recipes.

The *InventoryTX* blockchain is built and has the look of Figure 1. The reader will notice that it is a generic representation of a blockchain and that different authors in the literature may have additional features.

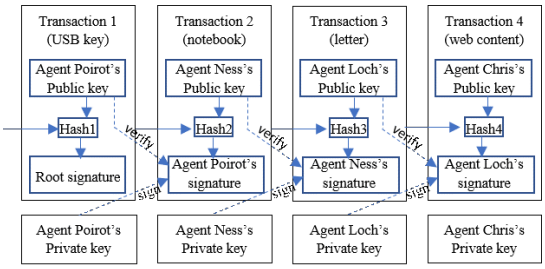

Figure 1. InventoryTX for the Marple case

In this fictious example, the defense argues that pornographic materials and drug recipes are not the subject of the search and should be dismissed. The court follows this request and judges *Roy* and *Prince* update the *InvalidatedTX* blockchain which is depicted in Figure 2.

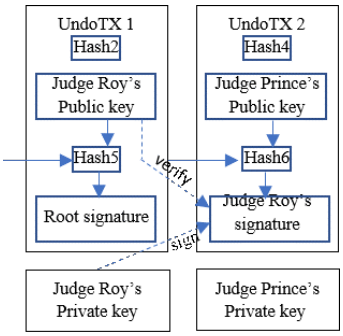

Figure 2. InvalidatedTX for the Marple case

When parties will access the evidence stored into the *InventoryTX*, the system will first look into the *InvalidatedTX* to verify if the transaction concerning the evidence is legally sound. Three scenarios are then possible:

- If the transaction hash is absent from *InvalidatedTX*, and present in *InventoryTX* then the system will serve the transaction payload, which is usually a reference to a safe storage entity holding the content, or a description, of the evidence.
- If the transaction hash is absent from *InvalidatedTX*, and also absent from *InventoryTX* then the system will raise a "Transaction not found" exception.
- If the transaction hash is present in *InvalidatedTX* then the system will raise a "Transaction invalidated by court order #xxx" exception.

This system possesses the advantage of being very lightweight. In the absence of dismissed evidence, the cost for the lookup is $O(1)$, since *InvalidatedTX* is empty. In the presence of dismissed evidence, the cost for the lookup is in $O(m)$ with m the total number of dismissed evidence records. A broader overview of the system is depicted in Figure 3.

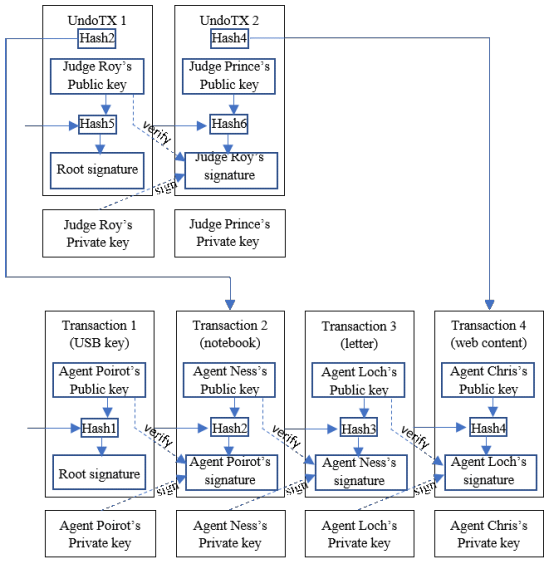

Figure 3. Overview of the two structures

The algorithm used to access a transaction of the blockchain can be summarized as in Figure 4. T references the transaction to be accessed, $hash(T)$ represents the transaction hash (its ID) and $payload(T)$ is the transaction's payload.

if (hash(T) $\notin$ InvalidatedTX) **then**
  if (hash(T) $\in$ InventoryTX) **then**
    return payload(T)
  else
    return "Transaction not found"
  end if
else
  return "Transaction invalidated by court order #xxx"
end if

Figure 4. AccessTX: Controlling access to a transaction

This algorithm, which is simple enough and explainable to parties concerned by a trial, should help in the adoption of blockchain solutions by providing more flexibility in the evidence management.

VI. IDENTIFYING DISMISSED TRANSACTIONS

Works related to blockchain use in digital forensics are different in many respects. But usually, the transaction payload refers to the evidence. For instance, in [9], the author expressed each transaction as a CASE object [14] using XML. Therefore, mentioning the evidence into the transaction payload can be achieved quite simply with an XML tag.

An example can illustrate a simplified blockchain, using the CASE format. Table I is the result of examining the USBSTOR

Windows registry hive of the suspect computer *AliceComputer*. This table shows that three USB devices have been connected to the computer at different times.

It is worth noting that XML allows for missing or partial element. For instance, the first entry from the USBSTOR hive has no registered user and no first connection date.

TABLE I. CONTAIN OF USBSTOR

Serial #	Name	User	Last conn.	First conn.
42014287	S3300		04.11.2016 08:52:50	
7299803F	Kingston Data-Traveler 2.0 USB Device	BadGuy	08.11.2016 12:30:11	2016.05.17 12:45:57
182127000	USB Flash Memory USB Device	BadGuy	18.07.2016 12:15:16	2016.07.18 08:39:50

Table II is a simplified version of a transaction representing the USBSTOR in the blockchain. In this example, the transaction payload contains a reference to *AliceComputer*.

TABLE II. TRANSACTION FOR THE USBSTOR IN INVENTORYTX

```
<Transaction>
<TransactionID>0001</TransactionID>
<EvidenceID>AliceComputer</EvidenceID>
<USBSTOR>
<Holder>\\SecureServer\AliceCase\usbstor</Holder>
<Access key>Decyphering Element</Access key>
<Element hash>0x123423e234fdaa5787e</Element hash>
</USBSTOR>
</Transaction>
```

The important element is that the reference to the digital evidence is present in the payload. Here, <EvidenceID> will be used to parse the blockchain for transactions to dismiss.

By using the CASE format, parsing for the transactions issued from a tainted evidence EvidenceID is straightforward: all the transactions are checked and the transactions referring to the tainted transaction are added to InvalidatedTX.

VII. PRIVACY PROTECTION

Privacy protection has gained momentum in the public and in particular in the processing of evidence or police files. When investigating digital evidence, scores of names are retrieved and recorded. Some names will lead to persons that will be investigated, but other names will be cleared. This puts forward how personal data is stored and managed in investigations.

Some research works, like [9], advocate for information to be stored inside the blockchain, in the transaction payload. Unfortunately, if the personal information is recorded into a blockchain, it will stay in the blockchain forever. And if a transaction needs to be dismissed because it is linked to a tainted evidence, then its payload needs to be deleted.

So, this paper advocates for a model where evidence contents is stored inside an encrypted and secured vault. The blockchain transaction payload will store only the evidence hash, or series of hashes, in addition to the location information and the deciphering key. In case of a transaction being voided via a legal order, the evidence content can be safely deleted, without any modification to the transaction.

An example of such a transaction is depicted in Table II, where the transaction data (the USBSTOR content) is stored at: `\\SecureServer\AliceCase\usbstor` and the hash of USBSTOR is: `0x123423e234fdaa5787e` .

An example of the undo-transaction added to the *InvalidatedTX* blockchain is provided in Table III, where <OrigTransactionID> is the ID of the original transaction.

TABLE III. UNDO-TRANSACTION FOR USBSTOR IN INVALIDATEDTX

```
<Transaction>
<TransactionID>00034</TransactionID>
<OrigTransactionID>0001</OrigTransactionID>
<EvidenceID>AliceComputer</EvidenceID>
</Transaction>
```

Therefore, the system offers a double privacy protection:

- The access control provided by *AccessTX* that will prevent the transaction payload to be disclosed;
- In case *AccessTX* is bypassed by a malevolent user, the information from the payload will lead to nowhere.

To summarize, when an evidence is dismissed from a case, following a court order, or a procedural decision, the following process is followed:

- Parsing of the *InventoryTX* transactions in order to identify the transactions linked to the tainted evidence;
- For each of these transactions, atomically execute:
  - issue undo transaction into *InvalidatedTX*,
  - delete the content referred by transaction payload.

This scheme ensures that information which is outside the scope of a case is definitely erased from the case and cannot be accessed anymore by the parties. The algorithm to dismiss transactions is summarized in Figure 5.

```
for all transaction T do
  if EvidenceID(T) = EvidenceID then
    Add a new transaction to InvalidatedTX
    Delete referenced content
  end if
end for
```

Figure 5. Dismissing transaction from a tainted evidence

VIII. CONCLUSION AND FUTURE WORKS

This paper presents a cost-effective solution for obliterating blockchain transactions from a case, in the presence of tainted evidence. The algorithms are simple enough to be explainable to all parties concerned by a trial, and should help in the adoption of blockchain solutions by providing more flexibility in the evidence management.

The presented solution for dismissing tainted evidence does not erase the fact that the evidence was once part of the procedure, but it will prevent the use of this evidence by the parties.

When a transaction is added to a case, its payload includes at least a reference to the evidence, a reference to the storage location of the evidence data, as well as its hash value. The payload does not contain evidence data.

When a court rules that a digital evidence has to be dismissed, our solution proceeds in three steps:

1) The transactions originated from tainted evidence are detected via the reference included in their payload.
2) Each time a transaction is positively checked:

 ISBN: 978-1-61208-818-1

a) an undo-transaction is added to an *InvalidatedTX* blockchain, holding all the undo-transactions
b) the evidence content referred by the transaction is erased from its secure storage.

Steps 2a and 2b need to be executed atomically in order to guarantee that when a transaction is erased, all its content is erased as well.

When a transaction is requested by a party, a component *AccessTX* does a first lookup in the *InvalidatedTX* blockchain in order to verify if the transaction has been previously dismissed. If the transaction is absent from *InvalidatedTX*, its payload is served to the party, otherwise an exception is raised, mentioning that the evidence was dismissed by court order.

As a matter of fact, the system will not serve a transaction which is linked to a tainted evidence, and in the case of a malevolent bypassing of the controlling mechanism, the digital evidence content is unavailable since 1) the transaction payload is only a reference to evidence data and 2) the evidence data has been erased from storage.

In short, this solution helps in the management of tainted digital evidence by removing the dismissed transactions while providing privacy protection over personal data that may appear in criminal investigations.

The solution presented in this paper can be improved in many ways. For instance, it does not take into account the cascading nature of the dismissal. As a matter of fact, the dismissal of a legal evidence should automatically lead to the dismissal of all the legal evidences which are an offspring. Unfortunately, to determinate if an evidence is an offspring of exactly one and only one evidence is not trivial: two distinct procedure acts may lead to obtain the same evidence. In this paper, it is assumed that the list of dismissed evidence is provided by the court. The automatization of the dismissed evidence list is the subject of a future work.

This work is now being considered for implementation, by using the IBM blockchain framework [22] on top of Hyperledger Fabric developed by Linux Foundation [23], which offers an extensive framework for permissioned blockchain.

REFERENCES

[1] M. Gupta, *Blockchain for Dummies, vol. 51*. John Wiley & Sons, Inc., ibm limited edition ed., 2018.

[2] Conte de Leon Daniel, "Blockchain: properties and misconceptions," *Asia Pacific Journal of Innovation and Entrepreneurship*, vol. 11, pp. 286–300, Jan. 2017.

[3] R. Ayers, S. Brothers, and W. Jansen, "Guidelines on Mobile Device Forensics," NIST Pubs 800-101 Rev 1, May 2014.

[4] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, and S. Uluagac, "Block4Forensic: An Integrated Lightweight Blockchain Framework for Forensics Applications of Connected Vehicles," *IEEE Communications Magazine*, vol. 56, pp. 50–57, Oct. 2018.

[5] K. Decoster and D. Billard, "HACIT: a privacy preserving and low cost solution for dynamic navigation and Forensics in VANET," *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018)*, pp. 454–461, 2018.

[6] C. Oham, S. S. Kanhere, R. Jurdak, and S. Jha, "A Blockchain Based Liability Attribution Framework for Autonomous Vehicles," *arXiv:abs/1802.05050*, 2018.

[7] H. F. Atlam, A. Alenezi, M. O. Alassafi, and G. Wills, "Blockchain with Internet of Things: Benefits, challenges, and future directions," *International Journal of Intelligent Systems and Applications*, vol. 10, pp. 40–48, June 2018.

[8] A. H. Lone and R. N. Mir, "Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer," *Digital Investigation*, vol. 28, pp. 44–55, Mar. 2019.

[9] D. Billard, "Weighted forensics evidence using blockchain," *International Conference on Computing and Data Engineering*, pp. 57–61, May 2018.

[10] S. Brotsis, N. Kolokotronis, K. Limniotis, S. Shiaeles, D. Kavallieros, E. Bellini, and C. Pavue, *Blockchain Solutions for Forensic Evidence Preservation in IoT Environments*. Mar. 2019.

[11] Z. Tian, M. Li, M. Qiu, Y. Sun, and S. Su, "Block-DEF: A secure digital evidence framework using blockchain," *Information Sciences*, vol. 491, pp. 151 – 165, 2019.

[12] D. Billard, "Blockchain-Based Digital Evidence Inventory," *Journal of Advances in Information Technology*, vol. 10, pp. 41–47, May 2019.

[13] E. Casey, G. Back, and S. Barnum, "Leveraging CybOX^{TM} to standardize representation and exchange of digital forensic information," *Digital Investigation*, vol. 12, pp. S102–S110, 2015.

[14] E. Casey, S. Barnum, R. Griffith, J. Snyder, H. v. Beek, and A. Nelson, "Advancing coordinated cyber-investigations and tool interoperability using a community developed specification language," *Digit. Investig.*, vol. 22, no. C, pp. 14–45, 2017.

[15] G.S.Harihara, S. S. Akila, Ashmithashree, Gayathri, and A. Jebin, "Digital Forensics Using Blockchain," *International Journal of Recent Technology and Engineering (IJRTE)*, pp. 182–184, Sept. 2019.

[16] S. Bonomi, M. Casini, and C. Ciccotelli, "B-CoC: A Blockchain-based Chain of Custody for Evidences Management in Digital Forensics," *ArXiv*, 2018.

[17] H. Al-Khateeb, G. Epiphaniou, and H. Daly, "Blockchain for Modern Digital Forensics: The Chain-of-Custody as a Distributed Ledger," Apr. 2019.

[18] "Mapp v. Ohio, 367 U.S. 643 (1961)."

[19] G. Tziakouris, "Cryptocurrencies—A Forensic Challenge or Opportunity for Law Enforcement? An INTERPOL Perspective," *IEEE Security Privacy*, vol. 16, pp. 92–94, July 2018. Conference Name: IEEE Security Privacy.

[20] D. Deuber, B. Magri, and S. A. K. Thyagarajan, "Redactable Blockchain in the Permissionless Setting," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 124–138, May 2019. ISSN: 2375-1207.

[21] E. Politou, F. Casino, E. Alepis, and C. Patsakis, "Blockchain Mutability: Challenges and Proposed Solutions," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2020. Publisher: Institute of Electrical and Electronics Engineers (IEEE).

[22] IBM, "IBM Blockchain Platform.," *https://ibm-blockchain.github.io/develop/*, 2017. Retrieved: 09-2020.

[23] L. Foundation, "HyperLedger Fabric docs," *https://hyperledger-fabric.readthedocs.io/en/release/*, 2016. Retrieved: 09-2020.

 ISBN: 978-1-61208-818-1