Providing Tamper-Resistant Audit Trails for Cloud Forensics with Distributed Ledger based Solutions

Mats Neovius
Department of Business Management and Analytics
Arcada University of Applied Sciences
Jan-Magnus Janssons plats 1, 00550 Helsinki, Finland
mats.neovius@arcada.fi

Magnus Westerlund
Department of Business Management and Analytics
Arcada University of Applied Sciences
Jan-Magnus Janssons plats 1, 00550 Helsinki, Finland
magnus.westerlund@arcada.fi

Jonny Karlsson
Department of Business Management and Analytics
Arcada University of Applied Sciences
Jan-Magnus Janssons plats 1, 00550 Helsinki, Finland
jonny.karlsson@arcada.fi

Göran Pulkkis
Department of Business Management and Analytics
Arcada University of Applied Sciences
Jan-Magnus Janssons plats 1, 00550 Helsinki, Finland
goran.pulkkis@arcada.fi

***Abstract*—Network and information security are often more challenging in cloud computing than in onsite computing. Cloud computing resources are publicly accessible and thereby through this availability increase the risk of intrusion. The increase in the processing of sensitive data on cloud resources makes security challenges more noteworthy, particularly in light of legal issues around cross-border transfers and data protection. Technologies preventing intrusion are effective, yet not perfect. Once a system is compromised, the intruder frequently starts to delete and to modify audit trails and system log files for covering-up the intrusion. Complete and untampered audit trails and log files are essential for the legitimate owner of the cloud instance or service to estimate the losses, to reconstruct the data, to detect the origin of the intrusion attack, and eventually in a court of law be able to prosecute the attacker. Due to this, improved methods for performing forensics in the cloud domain are desperately needed. The baseline for any forensic investigation is assured data availability and integrity. In this position paper, we outline how the availability and integrity of this forensic data can be assured by applying distributed ledger based solutions for securely storing audit trails and log files in immutable databases. Given this approach, an attacker can neither delete, nor modify past trails or logs but merely stop generating new data into log files. The position presented here is novel, yet light enough for practical use.**

Keywords-forensics; cloud computing; distributed ledger; blockchain; security; privacy.

I. INTRODUCTION

The last decade has entailed a transition from onsite to cloud computing. Cloud computing provides access to a pool of interconnected resources enabled by the Internet. It abstracts the hardware from the client and has a “pay-per-use” business model. In cloud computing, the resources are elastically provisioned with storage space, service, computing platforms as virtual machines [1], and networking infrastructures obtained upon request [2] [3]. Hence, cloud computing is “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 effort or service provider interaction” [2]. Three basic cloud computing service models are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Contemporary cloud-based software engineering directs towards Cloud Native Applications (CNA). A CNA is a service specifically designed to run in the cloud. CNAs are often deployed as self-contained units (containers) that are designed to scale horizontally. A CNA is often implemented as micro-services [4]. The technicalities are described in detail in [5]. In addition, the availability of cloud computing resources is augmented by the Intercloud initiative [6], envisioned as the “cloud of clouds”. Hence, the Intercloud then provides virtually unlimited resources to any connected device. Connected devices include mobile devices, giving rise to the term Mobile Cloud Computing [7], and Internet-of-Things (IoT) devices [8]. Consequently, the end user’s device running an application that utilizes cloud resources may be seen as the mere portal to the cloud relying on the service provider in administering the security and privacy of the data.

Academic research in network and computer forensics has a long history. Schneier and Kelsey [9] suggests a solution for keeping an audit log on insecure servers by offering a tamper-proof forensic scheme that stored and maintained log entries. However, with the shift to cloud computing the complexity and importance of keeping an audit trail has increased drastically. Cloud forensics has been defined as “the application of digital forensics in cloud computing as a subset of network forensics” [10] and as “to reconstruct past cloud computing events through identification, collection, preservation, examination, interpretation and reporting of digital evidence” [11]. As the former definition suggests forensics to be restricted to the

ISBN: 978-1-61208-607-1

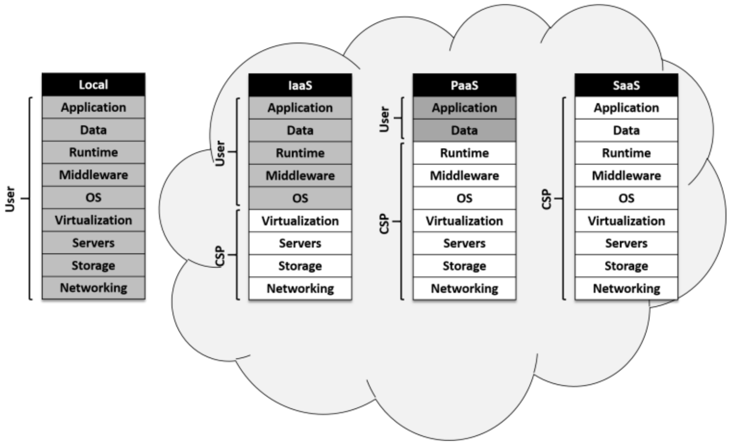

Figure 1. Access control to basic cloud service models in comparison to a local system.

network access, the latter definition includes the audit trail as a means to reconstruct events, as well as interpretation and reporting of evidence. Cloud forensics, therefore, requires audit trails to be stored in a manner with assured availability and integrity where no changes may occur.

A reasonable first choice for storage of audit trails for cloud forensics is an append-only (immutable) conventional database installation where read rights are assigned only to carefully selected set of agents. Existing implementations of immutable databases include configured conventional ones. In its most secure installation, it is hosted in-house with no means of external access and restricted physical access. Every access point (let these be logical or physical) weaken assurance of integrity. In-house installations are, however, not pragmatic for a cloud computing environments; nor are the cloud remote installations. On this challenge, purpose-built databases and filesystems are being developed, e.g., Datomic [12]. Implementation details of an immutable database for cloud audit trail are reported by Duncan and Whittington in [13]. Another attempt is the InterPlanetary File System (IFPS) [14]. The IPFS is fundamentally a protocol inspired by the Bitcoin blockchain protocol. It tries to make the web a digital resemblance to printed paper in documenting data, i.e., something that is permanent, unalterable and controllable.

Regardless of the technology, a distributed and replicated append-only storage provides stronger tamper resistance to a centralized one, specifically in relevance to nation-state-sponsored cyber-attacks. A distributed ledger is a replicated database, which is shared by nodes in a peer-to-peer network. Consensus algorithms are required to ensure replication and insertion across network nodes. In a truly distributed ledger, there is no central administrative node or centralized data storage. Thus, a distributed ledger storage for audit trails has stronger tamper resistance than any centralized immutable database implementation [15] [16].

This paper is a position paper that outlines an approach for storing the audit trail data using blockchain solutions. In the next Section, we discuss the current status of digital forensics in the cloud. In Section III, a tamper-resistant distributed ledger of the blockchain type that is based on protected storage of audit trails is presented. Finally, conclusions and proposals for future work are presented in Section IV. The distributed ledger technology is briefly described in an Appendix with the emphasis on the blockchain.

II. Cloud Forensics And Audit Trails

Audit trails for cloud forensics consist of collected log data of network traffic and data processing activities of computing devices. A generator of such data is the Intrusion Detection System (IDS) that extracts features from collected log data and analyzes these. The cloud service provider (CSP) is responsible for generating this IDS data. However, depending on the service model, the point of responsibility deviates.

Log data for audit trails can be scattered and stored in different locations due to the characteristics of the cloud. In the cloud, the level of access is divided between the cloud service user and the CSP. The level of access in the basic cloud service models is shown in Figure 1. This significantly complicates the data acquisition process. For example in the SaaS and PaaS models, only application related logs can be accessed by the cloud service user. Though in PaaS, a cloud service user can develop an application to be able to get some additional forensics data whereas, in SaaS, this is not possible. In the IaaS model, cloud service users can move to the operating system layer for acquiring forensic data. In all service models, the forensic investigators are dependent on the CSP to ensure that needed audit trail data has been collected. This is currently thus a trust issue since the availability and integrity of the data that may be affected are not transparent. Only when both parties are fully contributing to an immutable audit trail can it provide the required transparency needed for continued investigation and legal measures.

Verifiable audit trails are essential in forensic investigations to reconstruct and rigorously examine intrusions in the cloud. The reconstruction is central to find out what damage the intrusion has caused and discover sources and origins of intrusion attacks. When an attack has occurred, the cloud service user must engage a cloud forensics investigation to analyze the audit trail related to the attacked service in order to find forensic evidence. For this, the audit trail is fundamental in meeting with the EU General Data Protection Regulation (GDPR) [17], requiring enterprises to report security breaches within 72 hours after detection. Moreover, it should be possible for a CSP to present evidence on its own behalf that the source of the intrusion was external.

Traditionally, in digital forensics investigators take control of the affected physical device and perform forensic investigations on these by searching for evidence of malicious activity. For cloud computing being inherently dynamic, the methods traditionally used in digital forensics render themselves impractical [18]. Different cloud service users may virtually share physical resources through the hypervisor and thus, isolate the scene for forensics is next to impossible. This leads to issues that have to be addressed by the forensic investigation, namely, it must be proven that any data extracted is not mixed with some other customer’s data and that the availability, privacy, and integrity of the other user's data must be maintained.

Cloud forensics challenges are mostly related to architectural, data collection, and legal issues [11] [19], as well as in composing provenance data. Provenance data is the “metadata that provides details of the origins (history) of a data object” [20]. That is, provenance data is metadata tracing the history of data objects starting from original source data [21]. Complete provenance of all data stored in the cloud, all distributed computations, all data exchanges, and all transactions would enable identification of exact sources of cloud intrusion attacks and detect insider attacks in forensic investigations [22].

III. PROTECTION SOLUTIONS FOR AUDIT TRAIL DATA

Audit trail data for cloud forensics requires secure protection since it is vulnerable to corruption by accidental faults and malicious forgery [23]. Protection must repel accidental corruption and all malicious anti-forensics attacks by ensuring both integrity and availability of the data. This Section discusses requirements for distributed ledger based protection solutions for audit trails in the cloud and presents some blockchain based solution proposals. Distributed ledger technology with the focus on blockchain technology is described in an Appendix.

A. Requirements for Distributed Ledger based Solutions

Usage of a distributed ledger for protection of cloud forensics data is possible only if three fundamental requirements are fulfilled. First, a sufficiently large network of nodes must be available for storing replicated copies of the distributed ledger. Secondly, each network node must have sufficient storage and processing resources for management of a distributed ledger replication. Thirdly, it must be possible to extend the distributed ledger with new data produced at the data rate needed (i.e. throughput).

B. Existing Blockchain Based Solutions

Applying the blockchain and distributed ledger technologies in various domains is currently a hot research and business development topic. These technologies have been proposed for many financial technology solutions with extensions assuring programmatical smart contracts, to preserve (and control) privacy and personal data, provide transparency on transactions, and in the industrial IoT to keep track of logistic chains. These are all very intriguing applications, but we concentrate on ones that are directly relevant to the distributed audit trail data. Further, we focus on forensic data in the cloud computing environment as we find this area to be among the most challenging problems for distributed ledgers.

The integrity of cloud forensics data can be ensured by Public Key Infrastructure (PKI) signatures which depend on a certificate authority. This is not a feasible solution in the cloud infrastructure which is inherently decentralized. An alternative to PKI signatures is keyless signatures implemented by a blockchain based distributed Keyless Signature Infrastructure [24] [25].

A blockchain based data provenance architecture, the ProvChain, is described and evaluated in [26]. ProvChain has been designed for collection and verification of cloud computing users’ provenance data. ProvChain can use the global Bitcoin blockchain since the collected provenance data is restricted to metadata records of cloud service users’ operations on data files stored in the cloud. Recorded metadata attributes are RecordID, Date and Time, UserID, Filename, AffectedUser, and FileOperation. A FileOperation is file creation, file modification, file copy, file share, or file delete. UserID attributes are hashed to protect cloud users’ privacy. Provenance auditors can, therefore, access cloud users’ provenance metadata but cannot correlate the metadata to users owning the metadata. Only the Cloud Service Provider (CSP) can relate provenance data to cloud service users owning the data. Provenance metadata records are published in blocks of a blockchain implemented by a blockchain network consisting of globally participating nodes. Several metadata records can be stored in one blockchain transaction. Each metadata record is extended with a hash and a Merkle hash tree [27], is constructed for the metadata records in a block. The Merkle root is stored as a block header attribute. ProvChain is built on the top of the open source cloud computing application ownCloud [28]. The Tierion Data API [29], is used to publish provenance metadata records in the blockchain. Tierion generates for each transaction a blockchain receipt based on the Chainpoint standard [30]. The Merkle hash tree included in this blockchain receipt proves that the provenance metadata records were recorded at a specific time. A provenance auditor can request a blockchain receipt via Tierion Data API, access the related blockchain block with Blockchain Explorer [31], and validate the provenance metadata records in the block with the Merkle hash tree in the receipt. Measured ProvChain overhead for retrieval of provenance metadata of one file operation is about 0.7…0.8 s in an ownCloud test application [26].

Blockchain-based tamper-resistant registration of provenance data related to accessing medical data records in cloud storage is outlined in [32] [33]. The provenance data stored in the blockchain is available for auditing and in forensic investigations to detect privacy violations of medical data record owners. The outlined solution for protection of provenance data is applicable also to other types of personal data records.

C. Proposed Distributed Ledger based Solutions

An ideal solution would be a global network of nodes fulfilling all three requirements in Section III A. The global Bitcoin blockchain fulfils the two first requirements, but this blockchain cannot be extended with new blocks at a rate needed. Computationally it is not possible that even for a small cloud computing environment all the audit trail data for forensic investigations would be stored in the Bitcoin blockchain. The reason is the current blockchain size in combination with the throughput constrained Proof-of-Work (PoW) consensus algorithm.

However, other possible solutions may be engineered that circumvent this issue. One possible solution is a network of distributed ledger nodes, for example, blockchain nodes maintained by a CSP or preferably by several cooperating CSPs. As of the second requirement in Section III A, all cloud computing users cannot be nodes in a distributed ledger network since also resource-constrained mobile devices and IoT devices can use cloud computing services. Moreover, a faster consensus algorithm than PoW must be implemented for the used distributed ledger.

Hashgraph is a distributed ledger technology with a Byzantine consensus algorithm using a gossip protocol [34] [35]. While Bitcoins PoW implementation limits the throughput 7 transaction/s, the Hashgraph consensus algorithm can process even tens of thousands transactions/s [36]. The Archive Database proposed in [13] to be used as an immutable database for cloud audit trails could be implemented by a network of Hashgraph nodes maintained by a CSP or several cooperating CSPs. Each time when the database audit trail plugin stores log data the same data is transmitted to a preferably randomly chosen Hashgraph node. Reception of the log data creates a signed time-stamped event including a transaction storing the log data. An immutable record of all stored events is - due to the high event processing rate of a Hashgraph network – almost immediately available in each Hashgraph node. The Hashgraph fulfils all requirements in Section III A. However, at the time of writing it is deployed in permissioned environments and is, therefore, a permissioned distributed ledger technology. Still, a federated decentralized installation maintained by several cooperating CSP or other service providers may offer an alternative to a public distributed ledger.

There are also other proposals that address the need for high throughput distributed ledgers. Off-chain state agreement solutions commonly referred to as state channel technology, have been developed for handling many small transactions. A use case for the development of state channel technology has been to handle micro-transactions, which in addition to needing a high throughput also require a minuscule transaction cost for the clearance of each transaction. [37]

IV. CONCLUSIONS

This is a position paper outlining novel ideas on applying distributed ledger based solutions for storing audit trails in the cloud and more specifically, for micro-service deployments. The security features of the distributed ledger assure the integrity of the audit trails which is essential for trustable cloud forensics. The challenge is timely as the EU GDPR becomes enforced from May 2018. Moreover, the recent advancements in distributed ledgers, blockchains (cryptocurrencies) and their various spinoffs set the scene for applying this new technology by novel means. This paper lay the ground for distributed ledger technology in terms of cloud forensics.

APPENDIX

A. Distributed Ledger Technology

The most deployed distributed ledger type is a blockchain, which extends the shared database with a sequence of blocks storing transactional data. Blocks are chronologically and cryptographically linked to each another. Other distributed ledger types are the Tangle Network and Hashgraph. For the Tangle network, a Directed Acyclic graph-based network is used instead of a replicated linked chain of blocks in blockchain network nodes [38].

A Hashgraph network consists of nodes, which create context dependent events and communicate with each other using a gossip protocol. An event is a timestamped and digitally signed data structure consisting of one or several transactions and two hashes. One hash is extracted from the latest event on the node from which the latest gossip was received and the other hash is extracted from the preceding event created on the same node. A created event is sent as gossip to another randomly selected Hashgraph node together with all events still not known by the selected node. As event creation and gossip transmission continue in all Hashgraph nodes, all created events are immutably stored in each Hashgraph node. A Byzantine consensus on the order of events is achieved with probability 1 using a virtual voting procedure if more than 2n/3 nodes are uncorrupt where n is the number of nodes in the Hashgraph network. The details of the gossip protocol, the virtual voting, and the Byzantine consensus algorithm are presented in [39] and [35].

The blockchain technology is at the time of writing the best-known solution for implementing distributed ledgers and we, therefore, choose to focus on it. Findings concerning distributed ledgers, in general, should be transferable to other solutions such as the hashgraph and the Tangle network, once they become widely validated as secure.

Nakamoto introduced in 2008 blockchain technology as the Bitcoin cryptocurrency platform [40]. A blockchain implements a distributed database in which a list of records called blocks is stored. New blocks can always be appended to the list but stored blocks are neither removed nor changed. The distributed database is replicated in nodes of a peer-to-

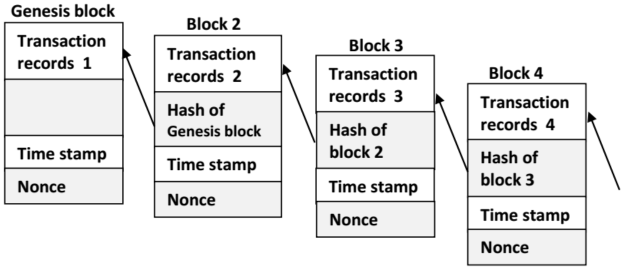

Figure 2. Basic blockchain structure.

peer blockchain network. A complete database copy is therefore stored in each network node. The blockchain topology is a chain, since after the first block each additional block contains a hash link to the preceding block, see Figure 2. The first block is called Genesis Block. Each block is also time stamped, however not necessarily to a universal time server.

A blockchain network node is owned by a blockchain user for execution of blockchain operations. A unique key pair of public key cryptography must also be owned by a blockchain user. The public key represents the identity of a blockchain user. A blockchain user executes a blockchain operation by initiating a transaction, which transfers some asset, for example, a cryptocurrency amount or a data object, to another blockchain user. A transaction creates a record, which is signed by the initiator of the transaction and transmitted to all nodes in the blockchain network. Each blockchain network node tries to validate a received transaction record with the transaction initiator's public key. A transaction record, which does not become validated by all blockchain network nodes, is discarded as invalid. Validated transaction records are collected by so-called mining nodes in the blockchain network and stored as lists in candidate blocks, which are time stamped. Each mining node executes a computation called mining on its candidate block. The candidate block of the mining node which first achieves a predefined mining goal is linked to the blockchain and all other mining nodes' candidate blocks are discarded. Several mining implementations for blockchains exist. Bitcoin blockchain mining uses PoW, where each mining node repeats hashing the concatenation of the last block in the blockchain and a new randomly chosen value. The mining goal is to create a hash of required difficulty.

There are public, permissioned, and private blockchains. A public blockchain, for example, Bitcoin, can be used by anyone. A public blockchain user copies the entire blockchain and installs the blockchain software on a personal node, which joins the blockchain network. Any blockchain user can also install the mining software on their own blockchain network node. Only a public blockchain can be trusted to fulfil the distributed ledger definition, as permission and private blockchains often maintain a centralized control node.

Recent blockchain implementations with extended functionality are denoted as Blockchain 2.0 for which an interesting feature is the smart contract introduced in [41]. A smart contract is a software component encompassing contractual terms and conditions enabling the verification, negotiation, or enforcement of a contract. A blockchain platform supporting smart contracts is Ethereum [42].

Blockchain security relies on the hash links between successive blocks combined with the replication of the entire blockchain to all blockchain network nodes. A public blockchain is therefore practically tamper-proof because a block cannot be changed without changing all the subsequent blocks and participation of all blockchain network nodes to validate and register the change. As the public blockchain is not managed by any centralized authority that could be a target of attacks it is less sensitive to some attack types such as DOS attacks, because full blockchain replicas are stored in many blockchain network nodes. However, an intrusion into a sufficient number of blockchain network nodes including some mining nodes can cause data losses and/or insertion of corrupt data in the attacked blockchain [43].

The tamper resistance of a blockchain does not exclude security vulnerabilities. Security attacks against blockchains are described and evaluated in [44] [45] [46] [47].

REFERENCES

[1] D. A. B. Fernandes, L. F. B. Soares, J. V. Gomes, M. M. Freire, and P. R. M. Inácio, "Security issues in cloud environments: a survey," International Journal of Information Security, vol. 13, iss. 2, pp. 113-170, Apr. 2014.

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing,“ Special Publication 800-145, National Institute of Standards and Technology, U.S. Dept. Commerce, 2011.

[3] J. Köhler, K. Jünemann, and H. Hartenstein, "Confidential database-as-a-service approaches: taxonomy and survey," J. Cloud Computing: Advances, Systems and Applications, vol. 4, no. 1, 2015. doi:10.1186/s13677-014-0025-1

[4] N. Dragoni et al., "Microservices: yesterday, today, and tomorrow," April 2017. [Online]. Available from: https://arxiv.org/pdf/1606.04036.pdf

[5] N. Kratzke and P.-C. Quint, "Understanding cloud-native applications after 10 years of cloud computing - A systematic mapping study," J. Systems and Software, vol. 126, pp. 1-16, April 2017, https://doi.org/10.1016/j.jss.2017.01.001

[6] D. Bernstein, E. Ludvigson, K. Sankar, S Diamond, and M. Morrow, "Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability," Proc. Fourth International Conference on Internet and Web Applications and Services (ICIW'09), IEEE Press, 2009, pp.328-336.

[7] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," IEEE Communications Surveys and Tutorials, vol. 15, no. 3, pp. 1294–1313, 2013.

[8] L. Jiang et al., "An IoT-Oriented Data Storage Framework in Cloud Computing Platform," IEEE Transactions on Industrial Informatics, Vol. 10, No. 2, pp. 1443-1451, May 2014.

[9] B. Schneier, and J. Kelsey, "Secure audit logs to support computer forensics," ACM Transactions on Information and System Security, vol. 1, no. 3, pp. 159-176, 1999.

[10] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie, "Cloud Forensics: An Overview," in Advances in Digital Forensics VII, pp. 35–46, 2011. [Online]. Available from: http://cloudforensicsresearch.org/publication/Cloud_Forensics_An_Overview_7th_IFIP.pdf

[11] P. Mell and T. Grance, “Nist cloud computing forensic science challenges,” Draft NISTIR 8006, National Institute of Standards and Technology, U.S. Department of Commerce, June 2014. [Online]. Available from: https://csrc.nist.gov/csrc/media/publications/nistir/8006/draft/documents/draft_nistir_8006.pdf

[12] Cognitect, Inc. Datomic The fully transactional, cloud-ready, distributed database, 2016. [Online]. Available from: http://www.datomic.com/

[13] B. Duncan and M. Whittington, “Creating an Immutable Database for Secure Cloud Audit Trail and System Logging,” Eighth International Conference on Cloud Computing, GRIDs, and Virtualization, Athens: IARIA, 2017, pp. 54–59.

[14] J. Benet, “IPFS – Content Addressed, Versioned, P2P File System (DRAFT 3)”, 2017 [Online]. Available from: https://github.com/ipfs/ipfs/blob/master/papers/ipfs-cap2pfs/ipfs-p2p-file-system.pdf

[15] Distributed Ledger Technology: beyond blockchain, 2016. [Online]. Available from: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf

[16] D. Mills et al., “Distributed ledger technology in payments, clearing, and settlement,” Finance and Economics Discussion Series 2016-095, 2016.

[17] EUR-Lex Regulation [EU] 2016/679. General Data Protection Regulation (GDPR). [Online]. Available from: http://eur-lex.europa.eu/eli/reg/2016/679/oj

[18] V. M. Katilu, V. N. L. Franqueira, and O. Angelopoulou, “Challenges of Data Provenance for Cloud Forensic Investigations,” Proc. 10th Int. Conf. on Availability, Reliability and Security, IEEE Press, 2015, pp. 312-317.

[19] M. E. Alex and R. Kishore, “Forensics Framework for Cloud computing,” J. Computers and Electrical Engineering, vol. 60, iss. C, pp. 193-205, May 2017.

[20] K.-K. Muniswamy-Reddy and M. Seltzer, “Provenance as first class cloud data,” ACM SIGOPS Operating Systems Review, vol. 43, no. 4, pp. 11-16, Jan. 2009, doi:10.1145/1713254.1713258

[21] Y. L. Simmhan, B. Plale, and D. Gannon, “A survey of data provenance in e-science,” ACM Sigmod Record, vol. 34, no. 3, pp. 31–36, 2005.

[22] D. K. Tosh et al., “Security Implications of Blockchain Cloud with Analysis of Block Withholding Attack,” Proc. 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE Press, 2017, pp. 458-467.

[23] B. Lee, A. Awad, and M. Awad, “Towards secure provenance in the cloud: A survey,” Proc. 8th International Conference on Utility and Cloud Computing (UCC), IEEE Press, 2015, pp. 577–582.

[24] A. Buldas, A. Kroonmaa, R. Laanoja, “Keyless signatures infrastructure: How to build global distributed hash-trees,” Nordic Conference on Secure IT Systems, Springer, 2013, pp. 313–320.

[25] Guardtime. Cloud Assurance with Blockchains, 2017. [Online]. Available from: https://guardtime.com/solutions/cloud

[26] X. Liang, et al., “ProvChain: A Blockchain-based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability,” Proc. 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE Press, 2017, pp. 468-477.

[27] R. C. Merkle, "A Digital Signature Based on a Conventional Encryption Function," in Advances in Cryptology - CRYPTO '87, LNCS 293, Springer, 1988.

[28] ownCloud, 2017. [Online]. Available from: https://owncloud.org/

[29] Tierion Documentation, 2017. [Online]. Available from: https://tierion.com/docs

[30] Chainpoint, 2017. [Online]. Available from: https://chainpoint.org/

[31] BTC.com, 2017. [Online]. Available from: https://btc.com/

[32] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, ‘‘BBDS: Blockchain-based data sharing for electronic medical records in cloud environments,’’ Information 2017, vol. 8, iss. 2, Apr. 2017, doi:10.3390/info8020044

[33] Q. Xia et al., “MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain,” IEEE Access, vol 5, pp. 14757-14767, July 2017.

[34] G. Kingslay, “Hashgraph vs. Blockchain Is the end of Bitcoin and Ethereum near?” [Online]. Available from: https://coincodex.com/article/1151/hashgraph-vs-blockchain-is-the-end-of-bitcoin-and-ethereum-near/

[35] L. Baird, The Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance, Swirlds Tech Report Swirlds-TR-2016-01, May 31, 2016. [Online]. Available from: http://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf

[36] Blockchain Technologies Feature Analysis, 2016. [Online]. Available from: https://lists.w3.org/Archives/Public/public-blockchain/2016Oct/att-0004/BlockchainTechnologiesFeatureAnalysis.html

[37] Z. Hess, Y. Malahov, and J. Pettersson, “Æternity blockchain”, 2017. [Online]. Available from: https://aeternity.com/aeternity-blockchain-whitepaper.pdf

[38] S. Popov, “The Tangle,” White Paper, 2017. [Online]. Available from: https://iota.org/IOTA_Whitepaper.pdf

[39] L. Baird, “Hashgraph Consensus: Detailed Examples,” Swirlds Tech Report Swirlds-TR-2016-02, Dec 11, 2016. [Online]. Available from: http://www.swirlds.com/downloads/SWIRLDS-TR-2016-02.pdf

[40] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. [Online]. Available from: https://bitcoin.org/bitcoin.pdf

[41] N. Szabo, “The Idea of Smart Contracts,” 1997. [Online]. Available from: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html

[42] Ethereum Blockchain App Platform. [Online]. Available from: https://www.ethereum.org/

[43] M. Conoscenti, A. Vetro, J. C. de Martin, “Blockchain for the Internet of Things: a Systematic Literature Review,” Proc. 13th International Conference on Computer Systems and Applications (AICCSA), IEEE Press, 2016, pp. 1-6.

[44] Eyal, I and E. G. Sirer, “Majority is not Enough: Bitcoin Mining is Vulnerable,” arXiv:1311.0243v5 [cs.CR], Nov. 2013. [Online]. Available from: https://arxiv.org/pdf/1311.0243v5.pdf

[45] S. Barber, X. Boyen, E. Shi, and E. Uzun, “Bitter to Better – How to Make Bitcoin a Better Currency,” in LNCS 7397, Switzerland: Springer, pp. 399-414, 2012.

[46] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg., “Eclipse Attacks on Bitcoin’s Peer-to-Peer Network,” Proc. 24th USENIX Security Symposium, 2015, pp. 129-144.

[47] K. Nayak, S. Kumar, A. Miller, and E. Shi, ”Stubborn mining: Generalizing selfish mining and combining with an eclipse attack,” Proc. IEEE European Symposium on Security and Privacy (EuroS&P), IEEE Press, 2016, pp. 305-320.