A Framework for Blockchain-Based Verification of Integrity and Authenticity

Anirban Basu, Theo Dimitrakos, Yuto Nakano, Shinsaku Kiyomoto

A framework for blockchain-based verification of integrity and authenticity

Anirban Basu[1⋆], Theo Dimitrakos[2,3⋆⋆], Yuto Nakano[1], Shinsaku Kiyomoto[1]

[1] KDDI Research, Inc., Japan
{basu, yuto, kiyomoto}@kddi-research.jp
[2] CSPL, Huawei Technologies Dusseldorf GmbH, Germany
theo.dimitrakos@huawei.com
[3] University of Kent, UK
t.dimitrakos@kent.ac.uk

Abstract. In many application scenarios, such as cloud computing and network function virtualisation, entities from different domains or their interactions are short-lived. Yet, it is often necessary to ensure accountability of events recorded by such entities about their application-specific interactions. The distributed and multi-domain nature of this problem makes a decentralised architecture imperative, particularly in the context of key management and trust. This architecture also needs to address challenges in terms of cross-domain privacy and confidentiality of shared data. For concreteness and without loss of generality, we consider the use case of firewalls as virtual network functions (VNFs) across multiple domains where short-lived firewall VNF instances spin up and down, logging events (e.g., security incidents) during their life spans. Such event logs need to exist, for purposes of accountability, beyond the life-cycles of their generating entities. In this position paper, we present a dual blockchain framework that facilitates the verification of integrity as well as authenticity of events while supporting privacy and confidentiality of data shared across multiple domains.

Keywords: integrity, authenticity, confidentiality, decentralised verification, trust

1 Introduction

The emergence of the cloud, network functional virtualisation (NFV), edge computing and IoT paradigms has necessitated the accountable collection of distributed logs and audit information over multiple administrative domains or trust realms and across service provision paths in complex ICT supply networks. There is an ever increasing need to develop scalable technologies that

⋆ This work was done while Anirban was with KDDI Research. He currently works for Hitachi R&D within Hitachi Ltd. He is also a Visiting Research Fellow with the University of Sussex and is reachable at a.basu@sussex.ac.uk.

⋆⋆ This work was partly done while Theo was visiting KDDI Research as a NICT-supported invited researcher.

ensure the integrity of such critical information (log, audit data, etc.) to enforce accountability and non-repudiation while taking into account the scope of use of the corresponding services, network functions or devices.

Blockchains and other forms of distributed ledgers (the underlying technologies of the Bitcoin [1] cryptocurrency, Ethereum [2] and other applications including cryptocurrencies [3–11]) offer cryptographic irreversibility of recorded data agreed upon by consensus amongst a set of decentralised entities. This property is useful for reliably logging event information generated by virtualised functions with short-lived, stateless and on-demand instances that reside on cloud infrastructures. Industry verticals, such as financial services, telecommunications, energy and smart vehicles – to name a few – are looking into distributed ledger technologies (including but not restricted to blockchains) for improving the integrity and availability of their services, their cross-service data flow and secure information sharing.

1.1 Objectives

In this paper, we consider the application scenario of a firewall as a virtual network function (VNF) and demonstrate how blockchains can be utilised to design and implement the distributed event log architecture. Our main objectives in this context are:

(1) to protect the integrity and confidentiality of important information of VNFs and IoT gateways such as events, configurations, policies, credentials, and so on;
(2) to strengthen the authenticity, accountability and integrity of security policies, security capabilities and VNF and IoT gateways;
(3) to reduce the risks of impersonation and privileged access abuse;
(4) to reduce the difficulties of cryptographic key management, revocation and trust management complexities; and
(5) to assess suitability and potential benefit of leveraging emerging technologies for multi-ledger and smart-contracts.

In this work-in-progress short paper, we consider the application scenario of a firewall as a VNF and demonstrate how a dual blockchain framework can be utilised to design and implement the distributed event log architecture. In general terms, we develop a multi-ledger model that protects the integrity of key information in a large-scale distributed computing systems and assures authenticity and accountability of modifications.

2 Proposed dual blockchain framework

Use-case scenario overview A virtualised network function (VNF) is a software code, an instance of which can run inside a virtual machine, on top of actual physical hardware. A distributed stateless firewall abstracted as a firewall VNF can have multiple, possibly geographically dispersed, instances that can be spun

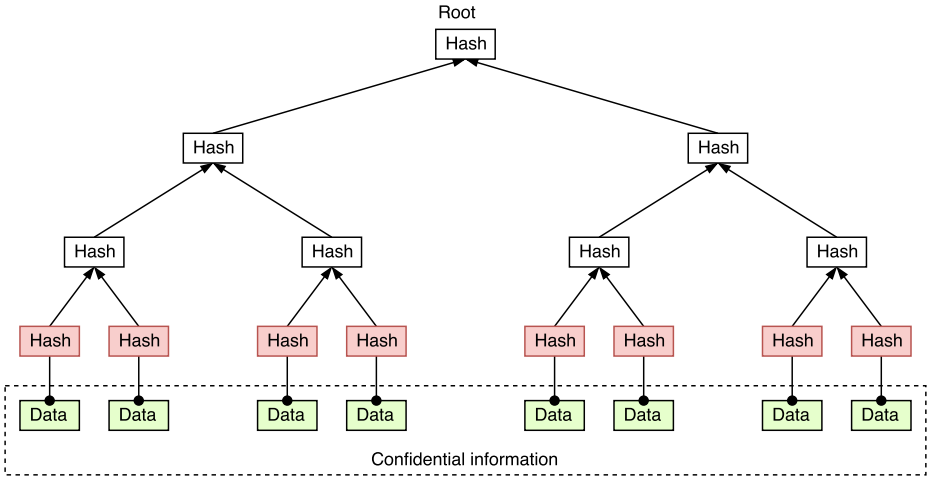

Fig. 1. Event log report structure.

up on demand. Each such instance logs events and incidents that it "sees". These logs need to be cryptographically signed to preserve their integrity. However, due to the short-lived and volatile lifespans of such VNF instances, maintaining and sharing signing keys between all instances of the same VNF is challenging even when there are separate keys per domain.

In [12], we proposed a preliminary direction for accountability and integrity of data management making use of blockchains. In this paper, we describe an architecture for VNF logs that are verifiable in terms of integrity and authenticity. We propose the use of two blockchains for two separate purposes. The first blockchain is a permissioned blockchain used to verify the integrity of the data logged by individual firewall instances (called the **i-Ledger** from now on) while the second public blockchain helps with the verification of the authenticity of the logged data (called the **a-Ledger** hereafter). Due to the public nature of the second blockchain, the actual event log data is either not stored in it or stored with some confidentiality-preserving transformation (e.g., keyed hash, encryption). These two blockchains are not necessarily linked in terms of actual connectivity between nodes, but are semantically 'linked' during the data verification process since data on one blockchain needs to be cross-checked with the information recorded in the other to help verify consistency.

2.1 Event log reports

Central to the idea of the blockchain for integrity is the way a VNF instance generates an event log report. The events considered in this log are typically security incidents, but we use the general terminology – events – throughout this paper. The individual events are added as leaf nodes to a hash tree, e.g., a Merkle Tree [13]. The level of granularity of the events is configurable, i.e., a VNF may wish to combine multiple events together instead of writing one event as one leaf of the Merkle Tree. While for the rest of the paper we stick to Merkle

Trees for the property of independent verification of sub-trees, any generalised hash tree satisfying the same property will suffice. The root of the Merkle Tree along with the entire tree structure is what constitutes a complete event log report, as illustrated in Figure 1. The actual event data on the leaf nodes can be privacy sensitive. Information from hash trees can be trimmed, similar to the delete operation in our existing work – VIGraph [14], which uses generalised hash trees for selective disclosure of information.

2.2 System overview

Figure 2 describes the overall system using a multi-domain scenario for the two blockchains involving three domains as well as an external notary. The blue lines indicate the topology of the i-Ledger whereas the red lines represent that of the a-Ledger. The domain on the left illustrates some of the actors in one organisational domain, such as the **entities** (VNFs in this case) that generate events and event log reports; the **Life-cycle Event Manager (LEM)** which generates events related to the life-cycle of entities; the **log manager** in charge of maintaining the **i-Ledger** and the **notary** in charge of maintaining the **a-Ledger**. The *Domain security manager* is responsible for controlling confidential data sharing policies and agreements, which we discuss later. The *Auditor* is responsible for cross-verifying the integrity and authenticity of events across the two blockchains.

2.3 Blockchain for integrity – the i-Ledger

The purpose of verifying the integrity of a data log is to ensure the signature on the log is valid, and that the signing entity is an authorised entity, i.e., an authorised firewall instance, in our running use-case. With the traditional certificate authority (CA) based keys, a CA signs the public key of an entity. However, the traditional CA style architecture requires the presence of a centralised and trusted certificate authority. It also assumes the existence of long-lived public-private key pairs. Neither of these hold true in our architecture of the distributed VNFs spanning across organisations.

A VNF instance generates an ephemeral private key (could remain stored only in volatile memory), which is removed when the VNF instance spins down gracefully or crashes. The corresponding public key, however, lives on. In our architecture, each separate administrative domain has its own **key manager** and **log manager**. Either the log manager or the life-cycle event manager may provide the key management functionality, and thus we do not specify a key manager separately. This key-pair can be generated based on a Hardware Security Module or a Trusted Platform Module (HSM/TPM) backed seed, and the public key is registered with the key manager. A VNF instance collates its logs in a report (Figure 1), adds a monotonically increasing numeric identifier and signs the Merkle Tree root and the numeric ID. This constitutes the main information for the block of varying size in this blockchain, illustrated in Figure 3, with the event data in green signifying privacy sensitive information.

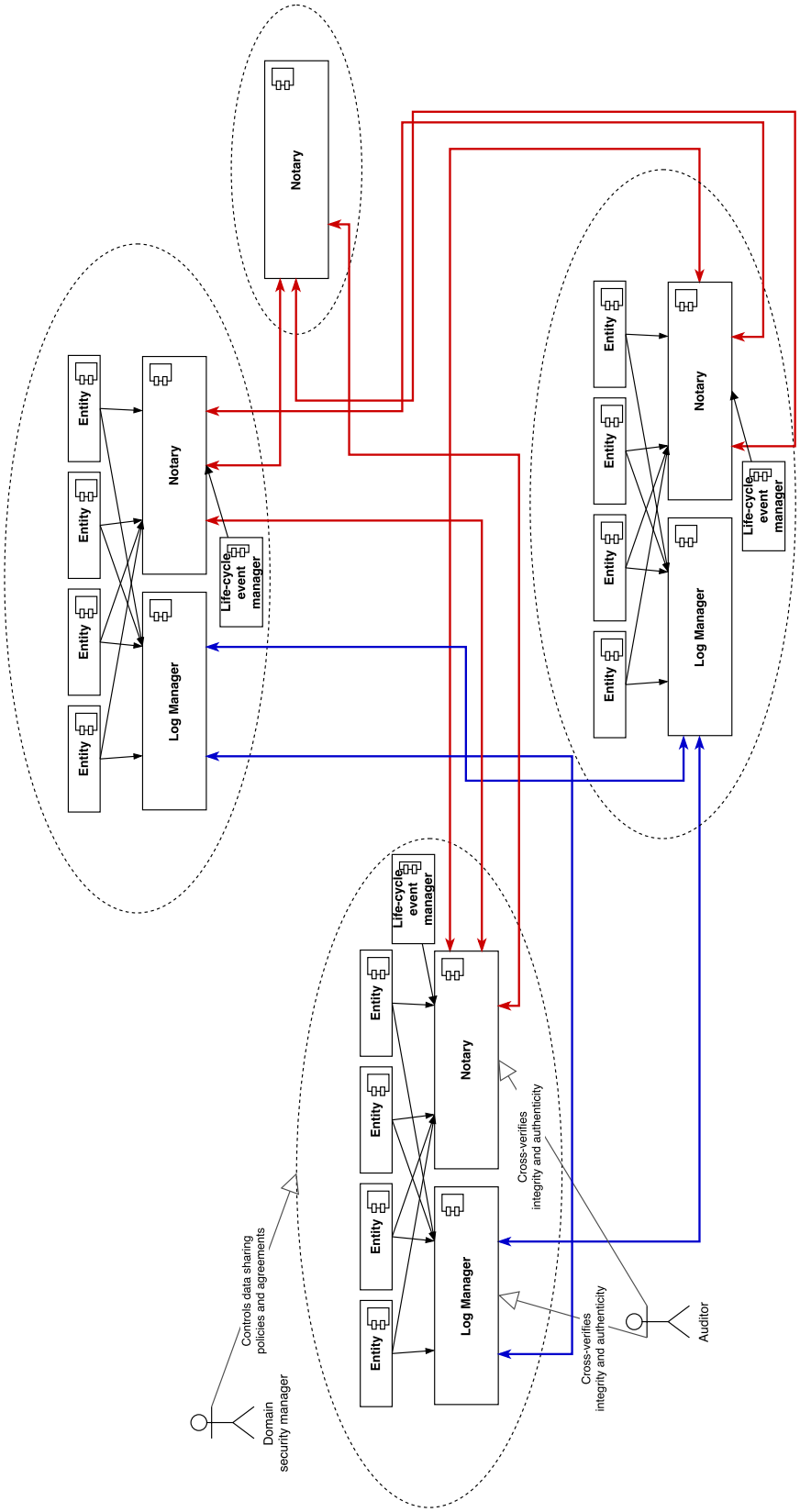

Fig. 2. A multi-domain system overview.

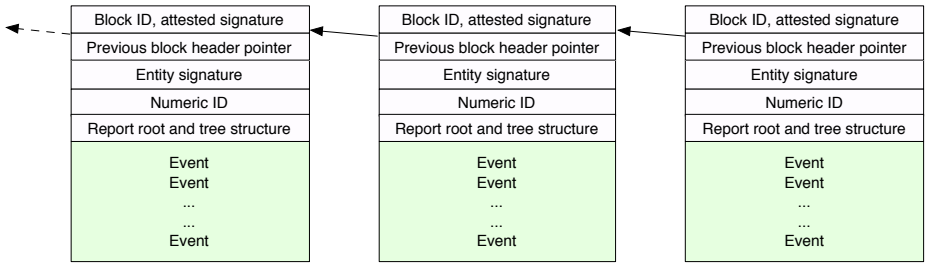

Fig. 3. Block structure for event log reports in i-Ledger.

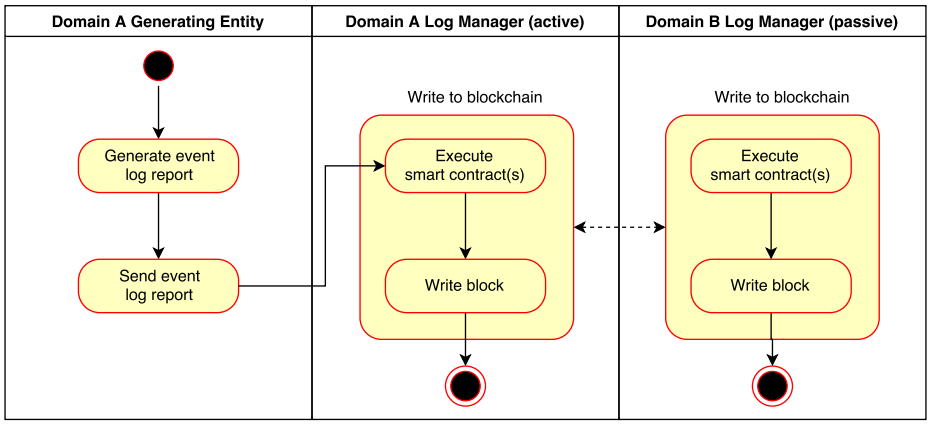

Fig. 4. Operation of the i-Ledger.

Due to the short-lived and resource-constrained nature of the VNF, it does not participate directly in reading from or writing to the blockchain. That task is delegated to the log manager of the domain. The pointer to the previous block and the block ID are, in turn, signed by the log manager that acts as a domain attester. The log managers across all the domains maintain the i-Ledger. The entire operation of the i-Ledger is shown in Figure 4, involving three actors:

(a) generating entity in the active domain (Domain A);
(b) the active domain log manager; and
(c) any other domain (Domain B) log manager.

A log manager runs a smart contract on the event log report before it accepts the block. Each log manager in each domain knows the identity (i.e., public key) of every other participating log manager from every other domain. Thus, the smart contract running on the log managers conceptually looks like the one illustrated in Figure 5, encapsulated as the "Execute smart contracts" state in Figure 4. The active domain contains the entity (i.e., VNF instance) that is attempting to write the report to the blockchain while the passive domain contains the log manager that accepts the event log report based on the acceptance of the valid identity of the active domain log manager.

Confidential information sharing The event data in the event log report could be considered privacy sensitive across different domains. Hence, if such an

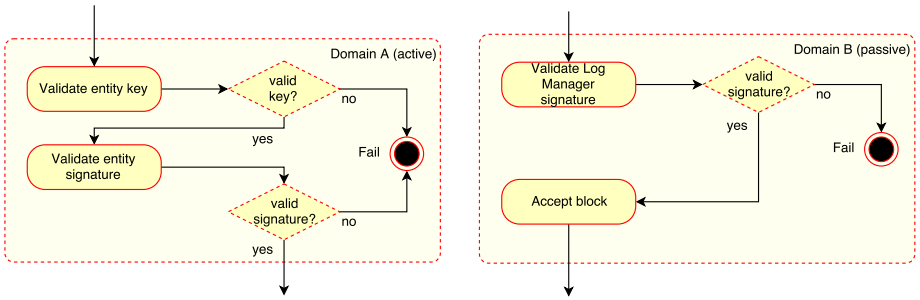

Fig. 5. Smart contracts in the i-Ledger.

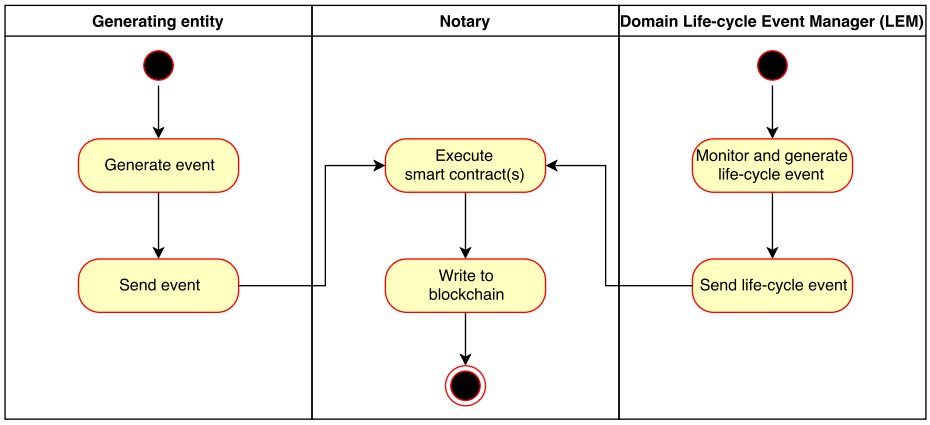

Fig. 6. Operation of the a-Ledger.

event log report is to be shared across domains, the leaf nodes are either removed from the tree structure while keeping their hashed parents intact; or the leaf nodes go through some confidentiality preserving transformation, e.g., symmetric key encryption where the relevant key is shared with authorised entities; or it could be attribute based encryption (ABE) where relevant entities have their access control policies defined in the ABE structure, in the keys (KP-APE) or in the ciphertexts (CP-ABE). This type of confidential information sharing policy is controlled the domain security manager as illustrated in Figure 2.

2.4 Blockchain for authenticity – the a-Ledger

The purpose of verifying the authenticity of the data log is to ensure that the publicly recorded log of the data (without details of the actual data to preserve confidentiality) corresponds to an actually recorded log in the i-Ledger that verifies it integrity. It also ensures that logs made by the same entity can be linked and their partial orders validated. Furthermore, with the records of life-cycle events, the a-Ledger allows a verifier to check that a specific log made by a specific VNF instance happened while it was active.

The entities maintaining this public a-Ledger are **notaries** that can exist in the aforementioned domains, but also in other unrelated domains, as shown

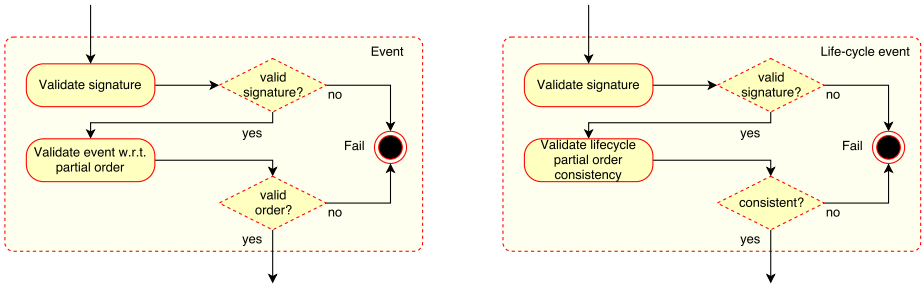

Fig. 7. Smart contracts in the a-Ledger.

in Figure 2. For instance, the VNF use-case could exist in domains such as telecommunications carriers while notaries could exist in other external domains such as financial and legal institutions. Unlike the permissioned i-Ledger, the block in this a-Ledger is not equivalent to a single event log report. Instead, blocks in this blockchain contain single events from the event log report as well as entity life-cycle events. Any such event is recorded as a *transaction* and many such transactions form a block in this blockchain. In order to facilitate life-cycle event reporting, we have the **life-cycle manager** per domain, which is typically a virtual machine monitor that knows when a VNF instance is up or down.

The transaction for an event (i.e.., VNF generated event) is a tuple consisting of:

(a) a specific event, E;
(b) the monotonically increasing numeric identifier for the event specific to the generating entity, n_t at time t; and
(c) the signature from the generating entity, sig_{entity}.

The transaction for a life-cycle log is a tuple consisting of:

(a) the life-cycle event (LE) to be recorded, e.g., LE_{start}, LE_{end} and so on;
(b) the public key of the entity whose life-cycle event is being recorded, i.e., $pubK_{entity}$; and
(c) the signature from the life-cycle manager, i.e., sig_{LEM}.

The entire operation of the a-Ledger is shown in Figure 6, involving three actors:

(a) generating entity;
(b) a notary in any domain; and
(c) the life-cycle manager in the same domain as the generating entity.

The notaries run two smart contracts depending on the type of event being added, as shown in Figure 7, encapsulated as the "Execute smart contracts" state in Figure 6. To check the validity of the signature of a life-cycle manager, it is imperative that notaries know and store the identities of each life-cycle manager for every domain.

3 Related Work

Bozic et al. [15] present an on-going work on a blockchain-based mechanism to protect cloud and NFV orchestration operations, specifically the authentication of orchestration commands in the lifecycles of cloud services. The scheme proposed in [16] helps ensure the necessary integrity and confidentiality properties application provenance in a cloud environment. Rübsamen et al. presented, in [17], a system that uses distributed software agents for secure evidence collection to enable automated evaluation during cloud accountability audits. In [18], Redfield and Date proposed a system where data is signed on the device that generates it, transmitted from multiple sources to a server using a signature scheme, and stored with its signature on a database running a protocol for long-term archival systems that maintains the data integrity of the signature even over the course of changing cryptographic practices. Sanz et al. [19] proposed a framework for automatic performance evaluation of service function chaining in network function virtualisation. The paper in [20] described the idea of securing drone data collection and communication with a public blockchain for provisioning data integrity and cloud auditing. In [21], authors proposed an architecture to secure federated cloud networks by enforcing a global security policy on all network segments of a federation, and local security policies on each network of the federation. In the context of the IEC 61499 standard [22] for distributed control systems, the work in [23] proposed an ongoing research on the implementation of function blocks as smart contracts executed by a blockchain as well as the integration with the edge nodes that are responsible for process control. The work in [24] adopted blockchains to address the lack-of-trust problem by mapping a business process onto a peer-to-peer execution infrastructure that stores transactions in a blockchain. Amongst the various benefits of their approach, an audit trail for the complete collaborative business processes, for which payments, escrow, and conflict resolution can be enforced automatically. The authors presented the idea of using blockchain as a service for IoT and evaluates the performance of a cloud and edge hosted blockchain implementation in [25]. In [26], the authors proposed a blockchain-based architecture for the secure configuration management of virtualised network functions (VNFs), which provides immutability, non-repudiation, and auditability of the configuration update history as well as integrity and consistency of stored information; and the anonymity of VNFs, tenants, and configuration information. Authors in [27] presented an architecture of a collaborative mechanism using smart contracts to investigate the possibility of mitigating a DDoS attack in a fully decentralized manner whereby the service providers can not only signal the occurrence of attacks but also share detection and mitigation mechanisms. Xu et al. [28] proposed a blockchain-based solution for trust in virtual machine images to reduce the risk of DoS attacks and at the same time provide a signature verification service for Docker images. Kouzinopoulos et al. discussed, in [29], the benefits of using blockchains to strengthen the security of IoT networks through a resilient, decentralized mechanism for connected home use-case that enhances the network self-defense by safeguarding critical security-related data.

Kataoka et al. presented [30] a 'trust list', which describes the distribution of trust among IoT-related stakeholders and provides autonomous enforcement of IoT traffic management at the edge networks by integrating blockchains and Software-Defined Networking (SDN). This, according to the authors, helps automating the process of doubting, verifying, and trusting IoT services and devices to effectively prevent attacks and abuses.

4 Conclusions and future work

The work-in-progress short paper presented a preliminary concept regarding the use of blockchains to provide accountability of events. Our running example use-case has been virtual firewalls instances as VNFs, but this architecture can be extended to other use cases with similar short-lived entities, e.g., various IoT and connect car scenarios. A number of avenues for future work exist, which include but are not limited to:

(i) validating the proposed architecture with a proof-of-concept on blockchains with an abstraction for the virtual network functions;
(ii) fully adopting a thorough security controls architecture utilising controls from the CSA Cloud Controls Matrix (e.g., CCM v3.0)[4] in order to build an actual working example of cloud-based virtual network functions with the proposed event logging architecture implemented on the two blockchains; and
(iii) investigating the use of post-quantum signature schemes, e.g., Lamport or Merkle in the event log reports because their hash-tree like structures lend themselves well to such signatures.

5 Acknowledgement

This work was partially done during Theo's visit to KDDI Research, Inc., Japan under the Japan Trust International Cooperation programme funded by the National Institute of Information and Communication Technology (NICT) in Japan.

References

1. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
2. Wood, G., et al.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper **151** (2014) 1–32
3. Wilkinson, S., Boshevski, T., Brandoff, J., Buterin, V.: Storj: a peer-to-peer cloud storage network (2014)
4. Vorick, D., Champine, L.: Sia: Simple decentralized storage. White paper available at `https://sia.tech/sia.pdf` (2014)

[4] `https://cloudsecurityalliance.org/working-groups/cloud-controls-matrix/`.

5. Danezis, G., Meiklejohn, S.: Centrally banked cryptocurrencies. arXiv preprint arXiv:1505.06895 (2015)
6. Koning, J.P.: Fedcoin: a central bank-issued cryptocurrency. R3 Report **15** (2016)
7. Hopwood, D., Bowe, S., Hornby, T., Wilcox, N.: Zcash protocol specification. Technical report, Zerocoin Electric Coin Company (2016)
8. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N.: Algorand: Scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th Symposium on Operating Systems Principles, ACM (2017) 51–68
9. Al-Bassam, M., Sonnino, A., Bano, S., Hrycyszyn, D., Danezis, G.: Chainspace: A sharded smart contracts platform. arXiv preprint arXiv:1708.03778 (2017)
10. Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Ford, B.: Omniledger: A secure, scale-out, decentralized ledger. IACR Cryptology ePrint Archive **2017** (2017) 406
11. Zamani, M., Movahedi, M., Raykova, M.: Rapidchain: Scaling blockchain via full sharding. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, ACM (2018) 931–948
12. Basu, A., Daniel, J.J., Ruj, S., Rahman, M.S., Dimitrakos, T., Kiyomoto, S.: Accountability and integrity for data management using blockchains. In: Poster at the 21st International Conference on Financial Cryptography and Data Security (FC), Malta (2017)
13. Merkle, R.C.: Method of providing digital signatures. `https://www.google.com/patents/US4309569` (1979)
14. Basu, A., Rahman, M.S., Xu, R., Fukushima, K., Kiyomoto, S.: VIGraph – A Framework for Verifiable Information. In: Proceedings of the IFIP WG 11.11 International Conference on Trust Management (IFIPTM), Göteborg, Sweden (2017) 12–20
15. Bozic, N., Pujolle, G., Secci, S.: Securing virtual machine orchestration with blockchains. In: Cyber Security in Networking Conference (CSNet), IEEE (2017) 1–8
16. Zawoad, S., Hasan, R.: Secap: Towards securing application provenance in the cloud. In: 9th International Conference on Cloud Computing (CLOUD), IEEE (2016) 900–903
17. Rübsamen, T., Pulls, T., Reich, C.: Security and privacy preservation of evidence in cloud accountability audits. In: International Conference on Cloud Computing and Services Science, Springer (2015) 95–114
18. Redfield, C.M., Date, H.: Gringotts: securing data for digital evidence. In: IEEE Security and Privacy Workshops, IEEE (2014) 10–17
19. Sanz, I.J., Mattos, D.M.F., Duarte, O.C.M.B.: Sfcperf: An automatic performance evaluation framework for service function chaining. In: IEEE/IFIP Network Operations and Management Symposium (NOMS), IEEE (2018) 1–9
20. Liang, X., Zhao, J., Shetty, S., Li, D.: Towards data assurance and resilience in iot using blockchain. In: IEEE Military Communications Conference (MILCOM), IEEE (2017) 261–266
21. Massonet, P., Dupont, S., Michot, A., Levin, A., Villari, M.: An architecture for securing federated cloud networks with service function chaining. In: IEEE Symposium on Computers and Communication (ISCC), IEEE (2016) 38–43
22. Vyatkin, V.: The IEC 61499 standard and its semantics. IEEE Industrial Electronics Magazine **3**(4) (2009)
23. Stanciu, A.: Blockchain based distributed control system for edge computing. In: 21st International Conference on Control Systems and Computer Science (CSCS), IEEE (2017) 667–671

24. Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., Mendling, J.: Untrusted business process monitoring and execution using blockchain. In: International Conference on Business Process Management, Springer (2016) 329–347
25. Samaniego, M., Deters, R.: Blockchain as a service for iot. In: IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE (2016) 433–436
26. Alvarenga, I.D., Rebello, G.A., Duarte, O.C.M.: Securing configuration management and migration of virtual network functions using blockchain. In: IEEE/IFIP Network Operations and Management Symposium (NOMS), IEEE (2018)
27. Rodrigues, B., Bocek, T., Lareida, A., Hausheer, D., Rafati, S., Stiller, B.: A Blockchain-Based Architecture for Collaborative DDoS Mitigation with Smart Contracts. In: IFIP International Conference on Autonomous Infrastructure, Management and Security, Springer (2017) 16–29
28. Xu, Q., Jin, C., Rasid, M.F.B.M., Veeravalli, B., Aung, K.M.M.: Blockchain-based decentralized content trust for Docker images. Multimedia Tools and Applications **77**(14) (2018) 18223–18248
29. Kouzinopoulos, C.S., Spathoulas, G., Giannoutakis, K.M., Votis, K., Pandey, P., Tzovaras, D., Katsikas, S.K., Collen, A., Nijdam, N.A.: Using blockchains to strengthen the security of internet of things. In: International Security Workshop (ISCIS), Springer (2018) 90–100
30. Kataoka, K., Gangwar, S., Podili, P.: Trust list: Internet-wide and distributed iot traffic management using blockchain and sdn. In: 4th World Forum on Internet of Things (WF-IoT), IEEE (2018) 296–301