Towards an attestation architecture for blockchain networks

World Wide Web
https://doi.org/10.1007/s11280-021-00869-4

Towards an attestation architecture for blockchain networks

Thomas Hardjono[1] · **Ned Smith**[2]

Received: 13 May 2020 / Revised: 8 December 2020 / Accepted: 16 February 2021 /
Published online: 11 March 2021

Abstract

If blockchain networks are to become the building blocks of the infrastructure for the future digital economy, then several challenges related to the resiliency and survivability of blockchain networks need to be addressed. The survivability of a blockchain network is influenced by the diversity of its nodes. Trustworthy device-level *attestations* permits nodes in a blockchain network to provide truthful evidence regarding their current configuration, operational state, keying material and other system attributes. In the current work we review the recent developments towards a standard attestation architecture and evidence conveyance protocols. We explore the applicability and benefits of a standard attestation architecture to blockchain networks. Finally, we discuss a number of open challenges related to node attestations that has arisen due to changing model of blockchain network deployments, such as the use of virtualization and containerization technologies for nodes in cloud infrastructures.

Keywords Blockchains · Trusted computing · Attestations · Virtual assets

1 Introduction

We believe there is a crucial role for trusted computing technologies, and more specifically attestations technologies, within the nascent area of blockchain networks. As blockchain networks play an increasing role in the future digital economy [57, 75] – such as becoming the underlying infrastructure for future crypto-currencies and virtual assets exchange

This article belongs to the Topical Collection: *Special Issue on Emerging Blockchain Applications and Technology*
Guest Editors: Rui Zhang, C. Mohan, and Ermyas Abebe

✉ Thomas Hardjono
hardjono@mit.edu

Ned Smith
ned.smith@intel.com

[1] Massachusetts Institute of Technology, Cambridge, MA, USA

[2] Intel Corporation, Santa Clara, CA, USA

networks – the security, resiliency and survivability of blockchain systems becomes crucial to their business value-proposition. Since the dawn of the computer age and the development of networked computer systems and the Internet, there has been the need for operators of computing equipment to obtain correct and truthful insights into the state of computing devices as part of managing the security of these devices. Given the proliferation of malware and viruses in the past decade, there has been a need for networked devices to have the capability to report its configuration, internal state and other parameters in a truthful and unforgeable manner. The technical term used to describe this process is *attestations*, which we will review in Section 2.

If blockchains and distributed ledger technology (DLT) are to become the foundation for critical infrastructures of the future (e.g. trade and finance, etc.) then the question of "network health" becomes pertinent also to these types of networks of nodes. Typically, an owner of a computer system (i.e. node) would seek to deploy the system as it is configured and programmed, without any authorized modifications. However, with the increasing prevalence of malware and computer viruses, the owner of a node currently has no visibility into the true state of the software and firmware of the node. There is little value in a consensus-outcome made by a collection of decentralized nodes when the nodes have been unknowingly compromised by malware and/or viruses. The situation is somewhat dire when there are no means to assess the degree to which the network of nodes have been affected (e.g. how many nodes compromised).

The goal of the current work is threefold. The first is to report and review the current development towards a standard attestation architecture in the computer and network industry (Sections 3 to 4). Secondly, to explore the applicability and benefits of the attestations architecture to nodes in a blockchain network (Section 5). Thirdly, we discuss some of the current challenges in attestations that has arisen due to changing model of blockchain networks, such as the use virtualization technologies for nodes in cloud infrastructures (Section 6). We discuss a number of areas for innovation in this space (Section 7), before concluding.

The subject of attestations is at least two decades old – stemming from the industry efforts around the Trusted Platform Module (TPM hardware) [73] – and numerous research papers have been devoted to this subject. Because the area of trusted computing has been heavily influenced by the design of the TPM hardware, much of the discourse in the broader research literature has been focused on one feature or another of the TPM hardware (e.g. the functions of its PCR registers, its identity keys, the Quote protocol, sealing, the agility of ciphers, and so on).

In the current work instead of providing a TPM-centric technical discussion around attestations and the required infrastructure to support TPM-based attestations, our goal is to discuss the notion of attestations in an accessible and meaningful manner. As such, we will strive to abstract-up from the various design features of the TPM and focus on the intent of some of these features, narrowing our interest on those features that support attestation and its potential use in blockchain networks. We direct readers to the excellent works of [3, 11, 59] for a deeper treatment of the TPM and its features.

2 Attestations of blockchain nodes: motivations

There are a number of possible benefits derived from the use of device-level attestations in the context of blockchain networks generally. The ability for a node to provide a truthful, complete and unforgeable report regarding its configuration, computational state, keying material and other system attributes provides a foundation to building trust (technical trust) in the network as a whole.

- *Node device identification*: In some blockchain deployments (e.g. permissioned blockchain networks) the ability to identify a node, authenticate and obtained signed assertions (e.g. reports) from the node provides a crucial feature for the manageability of the network.

  Attestation evidence must be source-authentic from the device. This means that attestation evidence or assertions must be signed by a private-key that is *bound to the hardware* (or a private-key that is derived from a hardware-bound key). A key that is hardware-bound means that it cannot be removed from the hardware (i.e. removal attack is uneconomical). Furthermore, a hardware-bound key may even be inaccessible (invisible) to the user's application. A user's application would instead use keys derived from this hardware-bound key. An example of this is the certified-keys in the TPM hardware (e.g. AIK-certified keys) [28].
- *Node device configuration reporting*: The ability for nodes in a blockchain network to truthfully report its device configuration (i.e. hardware, firmware, software) allows the questions related to fairness in the distribution of computing power (e.g. hashing rate) and other computing capabilities to begin to be addressed [20, 25].

  Detailed reporting of hardware configurations can be achieved using composite attestations approach (see Section 4.4). For example, the ability for a node to provide truthful and unforgeable evidence regarding the number of GPU cards (for hash-power increase) based on composite attestations allows a third-party verifier to glean as to the actual hash-power available at that node. This in turn allows a community of node-owners in a permissioned blockchain network, for example, to obtain accurate estimations regarding the hash-powers available at each of the members. Such estimations allows fairness to be more readily achieved.
- *Diversity of nodes and survivability of blockchain networks*: The ability for nodes to truthfully generate unforgeable evidence of its device configuration permits the question of network *diversity* – and therefore network *survivability* [33, 35] – to begin to be addressed.

  The question of the diversity of nodes (i.e. diversity of software stack and hardware) was also touched upon by NIST in their report (NISTIR 8202, Section 8): ... "A blockchain network is only as strong as the aggregate of all the existing nodes participating in the network. If all the nodes share similar hardware, software, geographic location, and messaging schema then there exists a certain amount of risk associated with the possibility of undiscovered security vulnerabilities" [77]. If there is one lesson learned from the past two decades of viruses in PC computers, it is that a relatively homogeneous network of systems (e.g. Windows only) is less resilient than one consisting of a diverse set of operating systems (e.g. mix of Windows, Linux, MacOS, AIX, etc).

  As blockchain networks carry increasingly valuable virtual assets [22], the question of blockchain resiliency and survivability becomes crucial to the business value proposition of the blockchain network.
- *Consensus protocol input*: The state of a node/device can be used as an additional input parameter into the consensus-making protocol used in the blockchain network. The idea is that the node that mines or forges new blocks should be in a compliant or "healthy" state. Here "compliance" means that the node is deemed satisfactory as evaluated against the appraisal policies driving the network. Thus, for example, the compliance

status of a node could be a factor in (input parameter into) the consensus-algorithm of a given blockchain network.

A survey of consensus protocols is beyond the focus of current work, but using the specific example of Ethereum and its Proof-of-Stake [9], the compliance status of nodes could be an additional factor in selecting the the PoS validator node – in addition to the current selection factors (e.g. staking-age, randomization, node's stake amount, lowest hash value, etc.).

- *Confidence in Smart Contract Platforms*: Smart contracts have been a major feature of attraction for blockchain platforms such as Ethereum. However, smart contracts themselves may introduce various unforeseen weaknesses and vulnerabilities [2].

  For example, one concern pertains to the accurate execution and the correct reporting of the outcome of the smart contract (at the application level). One key issue here is that even if a smart contract (visibly readable on a node) was digitally signed by its author at the application level, there is no guarantee when the contract binary was being loaded into memory to be executed (e.g. by the CPU of the node) that the code has not been modified or contaminated by malware. Here, the availability of Trusted Execution Environments (TEE) capability within the node's hardware can mitigate this problem to a large extent (e.g. Intel's SGX [15, 50]). However, attestation evidence must be conveyed by the node that: (i) the node possesses true TEE functionality, and (ii) that the contract code was indeed executed by (inside of) the TEE. Other challenges related to the type of consensus protocol used in a blockchain system and their suitability for TEEs have been identified in [6].
- *System-evidence for gateway nodes*: The need for interoperability across blockchain systems represents one of the key challenges today in many blockchain deployments. One approach currently being developed is one based on the use of *gateway nodes* that act on behalf of its blockchain system [36, 39]. Gateway nodes are tasked to perform cross-chain asset transfer transactions in an atomic manner (i.e. the ACID properties of atomicity, consistency, isolation and durability [17, 27, 40, 67]). The ability for gateway nodes to include their respective attestations as part of their session negotiation (e.g. TLS session establishment) can provide a policy decision point for a gateway node as to whether to proceed with a cross-chain transfer.
- *Legal trust framework for operating rules*: The technical means to provide node attestations can play an important role in certain types of blockchain networks. For example, in a consortium-based permissioned (private) blockchain network, attestations can be a key factor in the consortium's *Legal Trust Framework* (LTF) that governs the operating rules of the membership.

  Operational standards (profiles) that are co-developed and defined by the members of the consortium – based on stable technical standards published by standards-organization such as the TCG, IEEE and IETF – becomes a quantifiable input into the contracts that make-up the legal framework of the consortium. The legal contracts can be specific regarding member's obligations in the realm of deployment of nodes and related services. (e.g. members' nodes must use valid X.509 device certificates at all times). This co-development of operational standards means easier acceptance by the membership. A joint development of operational standards allows costs-sharing among members. Over time it lowers the shared operational costs of running the network as a whole.

  However, if a member is able to misrepresent their contribution to the shared operational costs, then the consortium looses its unbiased governance abilities. Attestations among consortium member nodes ensures operating rules are applied fairly across all members. Node attestations essentially provide the basis for *technical trust*, which in turn allows *business trust* to be attained by virtue the operational specifications being agreed-upon and observed in the consortium. Other related contracts and Service Level Agreements (SLAs) for the network as a whole can also defined in terms of these operational capabilities. For communities (e.g. industry sectors) seeking to employ the consortium model for permissioned blockchain, the operating rules of the community must include defining the mechanisms and procedures to be employed when changes are to be made to governing rules itself. These procedures can be human-based (e.g. paper based voting), and need not necessarily employ stakes-based electronic voting on the blockchain (e.g. DAO [63]).

3 Overview of the concept of attestations

In this section we briefly review at a high-level a number concepts which underlie the notion of device attestations that is core to the area of trusted computing.

3.1 The notion of the TCB

As computer systems evolved in the 1980s and 1990s, and as Local Area Networks (LANs) and peripheral devices proliferated the question of the security of computers systems became crucial in the networked world. This is true not only in the context of government and defense sectors, but also in the broader networked computing world. In the mid-1980s efforts such as Project Athena at MIT [61, 65] represented the leading edge of networked computing technology, and numerous security challenges – such as scalable authentication and authorization – were identified in these early years.

In the context of trustworthy computing, a landmark event in December 1985 was the publication of the Trusted Computer System Evaluation Criteria (TCSEC) by the U.S. Department of Defense. The TCSEC was a significant step forward because it defined the notion of the *Trusted Computing Base* (TCB). The core notion of the TCB is that if protecting the entire computer system was too costly or technically infeasible, then a portion of the system needs to be isolated that provides trustworthy behavior. That is, a domain "boundary" needs to be identified or defined in the system within which security can be guaranteed. This TCB domain boundary must demarcate "the security-relevant portions" of the system. This concept of the TCB became fundamental to ensuing efforts in the area of trustworthy computing, and the TCB portion became the focus of attention of new technical innovations in the following two decades. All subsequent expressions of trustworthy computing and security policy would be described in terms of impact and relevance to the TCB.

Although the TCSEC criteria focused mostly on defining the operating system security domain, it is important to remember that the operating system is not the sole TCB component in a computing platform. The hardware also plays a significant role, notably in the context of memory page isolation where a central tenet is the idea of kernel-mode isolation (namely *ring-0*) and application-mode (namely *ring-3*) process separation contexts [62]. From the operating system perspective the hardware is thought to be trusted because the operating system has no alternative way to test and verify that the hardware is behaving correctly.

The threat of hardware vulnerability motivated the computing industry to form the Trusted Computing Group (TCG) [68] in the late 1990s where the notion of a hardware root-of-trust was used to distinguish the security relevant portions of a hardware platform. The TCG defined trusted computing more organically by building upon granular components that were described as *shielded locations* and *protected capabilities*. Shielded locations are "...A place (memory, register, etc.) where it is safe to operate on sensitive data; data locations that can be accessed only by protected capabilities". Protected capabilities are "...the set of commands with exclusive permission to access shielded locations." By extension, all components that could be classified as shielded locations or protected capabilities is what defines the hardware security domain wherein the TCB software executes.

3.2 The attester and attesting environment

Implied in the TCSEC definition of the TCB is the required ability for a TCB to be reviewed. Originally, security review was a manual process involving a certification process. However, subsequent evolution and automation of security review makes feasible the reporting of much of the TCBs internal state – or what we refer to as *attestations*. The fundamental idea of attestations of a "thing" (e.g. a computing device) is that of the conveyance of truthful information regarding the (internal) state of the thing being attested to. In the related literature on trustworthy computing the term "measurement" is used to mean the act of collecting (introspecting) claims or assertions about the internal state, and delivering these claims as evidence to an external party or entity for automated review and security assessment.

However, as we know today computing environments can be structurally complex and may consist of multiple elements (e.g. memory, CPU, storage, networking, firmware, software), and computational elements can be linked and composed to form computational pipelines, arrays and networks. Thus, the dilemma is that not every computational element can be expected to be capable of attestation. Furthermore, attestation-capable elements may not be capable of attesting every computing element with which it interacts. The attestation capability could in fact be a computing environment itself. The act of monitoring trustworthiness attributes, collecting them into an interoperable format, integrity protecting, authenticating and conveying them requires a computing environment – one that should be separate from the one being attested. Figure 1 illustrates the recognition of this distinction, namely of the *target environment* being attested to, and the *attesting environment* that performs the work stated above.

The complexity of the problem has led to a number of efforts in industry to define an *attestation architecture* that incorporates some of these key concepts – such as the concept of the root-of-trust – and to develop standards that implement attestation concepts. The roles and functions of the attestation architecture is shown in Figure 1 and will be discussed at length in the ensuing sections. In a nutshell, in Figure 1 an attester conveys evidence of trustworthiness (of the attested target environment) to a verifier entity. The verifier operates based on policies that are supplied by the owner of the verifier.

We believe that an attestation architecture should define the attestation roles in the ecosystem (i.e. Attester, Verifier, Endorser, Relying Party, and Owner), the messages they exchange, their structure and the various ways in which roles may be hosted, combined and divided amongst the various entities involved in real-world deployments. These roles should remain true independent of the specific use-cases or deployments of systems having attestation functions. Furthermore, the attestation messages should be built on an information model that defines its trustworthiness semantics as well as on a data model that

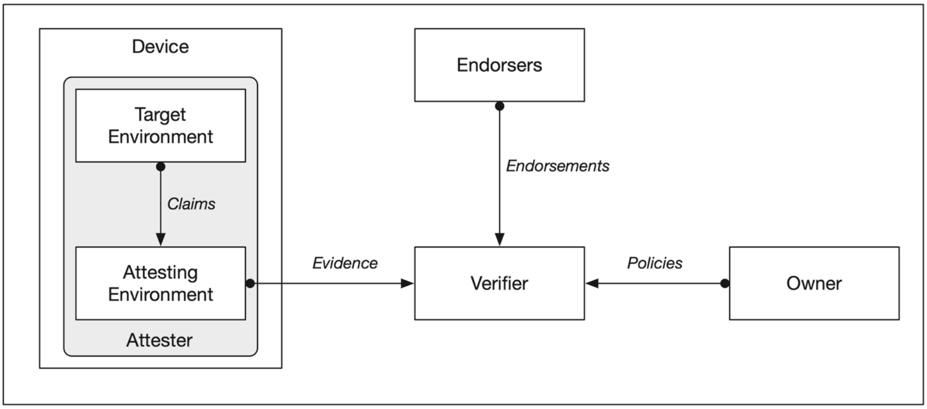

Figure 1 Overview of the concept of the attester and attesting environment

supports broad interoperability options. The information model and various data model representations would then be realized as data structures, data structure encodings and protocol bindings for conveying attestation messages, that are aimed at specific deployment cases or "profiles" (e.g. PC client devices, constrained IoT devices, various network equipment such as routers, mobile devices, server chassis, etc.).

Finally, from architecture perspective we believe that the verifier should be able to understand the trustworthiness properties of both the target environments as well as the attestation capability itself (at the attester). This must be true for any set of assertions within an attestation flow between the attester and the verifier. Thus, the trustworthiness of the attestation capability itself should be a core consideration of a well-designed attestation architecture. This may mean that the attestation architecture should anticipate the possibility of recursive or layered TCBs, each having believable and verifiable trust properties, something that may complicate implementations considerably.

3.3 Reference values and endorsements

Another key concept in trustworthy computing is that of *endorsements* from supply-chain entities regarding one or more components that are incorporated into the target environment. In practice, there is a point at which ultimately a portion of the computing environment trustworthiness must be established via non-automated means. A *root-of-trust* refers to a TCB element that ascribes trust through non-automated means. These non-automated means include things such design reviews, manufacturing process audits and physical security. A trustworthy attestation mechanism depends on trustworthy manufacturing and supply-chain practices. This manufacturer's claims of trustworthiness (of its product) is expressed through endorsements, which most commonly take the form of digital certificates signed by the manufacturer [29, 30].

Thus, a manufacturer of a component (e.g. firmware for hardware component) can publish a "known good" or "expected" value for a given firmware file. In the simplest form, this endorsement could be a cryptographic digest (i.e. hash) of the firmware file which is authenticated to its vendor or place of origin using a digitally signed structure (e.g. X.509 attribute certificates [21]). The significance of signing by the manufacturer using its public-private key pair is that it asserts these values to be authentic, as an *endorsement* for that product. Entities seeking to use the product (i.e. firmware file) can validate that attested values corresponding to the digests found in the endorsed values are known or expected. Thus, we refer to the digest as a *known good value* in this context.

As we will discuss below, an attestation architecture should distinguish between these more static endorsements issued by a supply-chain entity from the *evidence* issued by an attester during its runtime. As shown in Figure 1 the Attester creates attestation evidence (signed assertions) that are conveyed to a Verifier for appraisal. The appraisal process compares the received evidence against the known good values (i.e. endorsements) obtained from supply-chain entities – referred to as *endorsers*. A good architecture should support multiple forms of appraisals (e.g. software integrity verification, device composition and configuration verification, device identity and provenance verification, etc.). Out of this appraisal process the attestation *results* are generated, signed and then conveyed to relying parties. The attestation results provide the operational integrity basis by which relying parties may determine the level of confidence to place in the application data or other application-specific operations that follow.

4 A canonical attestations architecture

Recently, the notion of attestations has garnered interest within different technical standards organizations and industry consortiums, beyond the TCG alliance (e.g. FIDO Alliance [47], Global-Platform [26], IETF [42]).

A broader set of use-cases are also emerging, ranging from attestations by routing fabrics to attestations by low-power Internet-of-Things (IoT) networks. Regardless of the use-case, the notion of attestations holds true due to the universal need in the digital world to obtain assurance regarding the expected working environments and their security and resiliency properties. As an increasing portion of the economy moves onto digital platforms operating using complex and often invisible infrastructures (e.g. cloud platforms, virtualization, edge computing, mobile wallets, etc.), the more crucial the need for attestation underpinnings as a supporting infrastructure.

In this section, we discuss further the notion of attestations and present a canonical architecture that is currently being developed by several industry organizations (e.g. TCG [69], IETF [42]). The hope is that a canonical attestation architecture will allow standards to be developed that implement the various protocols and flows for relevant sectors and products (routers and network equipment [23, 70], mobile devices [26], cloud stacks [18, 56], etc.) By having a common reference architecture, different efforts can share common terminologies, concepts and implementations and therefore affect a reduction in costs of developing and deploying the infrastructures supporting cyber-resilience and trustworthy computing generally.

4.1 Entities, roles and actors

The attestation architecture of [64] defines of a set of *roles* that implement attestation flows. Roles are hosted by *actors*, where actors are deployment entities. Different deployment models may coalesce or separate various actor components and may call for differing attestation conveyance mechanisms. However, different deployment models do not fundamentally modify attestation roles, the responsibilities of each role, nor the information that flows between them. In the following sections, we may use the actor and role terminology interchangeably when appropriate in order to simplify discussion (see Figure 2).

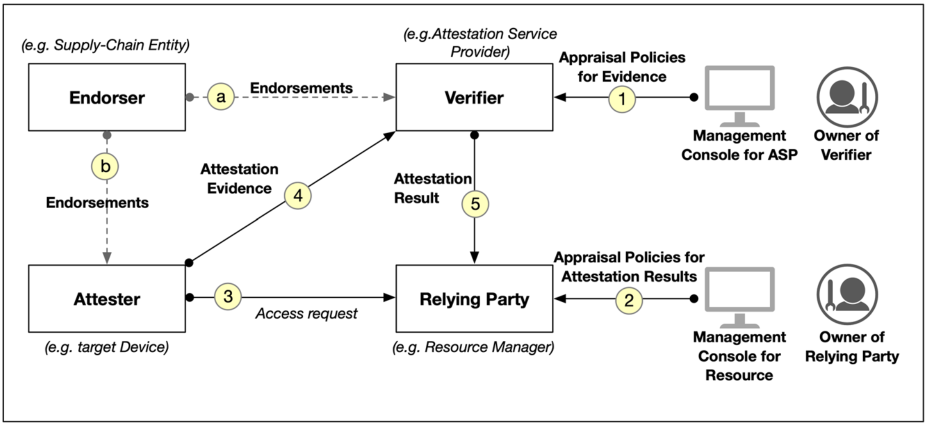

Figure 2 Canonical architecture for attestations (after [4, 64])

- *Attester*: The Attester (e.g. target device) provides attestation Evidence to a Verifier. The Attester must have an attestation identity that is used to authenticate the conveyed Evidence and establishes an attestation endpoint context. The attestation identity is often established as part of a manufacturing process that embeds identity credentials in the entity that implements an Attester.
- *Verifier*: The Verifier accepts Endorsements (from Endorsers) and Evidence (from the Attester) then conveys Attestation Results to one or more Relying Parties. The Verifier must evaluate the received Endorsements and Evidences against the internal *appraisal policies* chosen or configured by the owner of the Verifier [14]. The Attestation Service Provider (ASP) is typically the actor which implements the Verifier role. An example of the ASP is described in [78].
- *Relying Party*: The Relying Party (RP) role is implemented by a resource manager that accepts Attestation Results from a Verifier. The Relying Party trusts the Verifier to correctly evaluate attestation Evidence and Policies, and to produce a correct *Attestation Result*. Thus, we assume that the RP and the Verifier has a business relationship (e.g. see the SAML2.0 [55] model for a similar business relationship assumption) or some other basis for trusting one another. The Relying Party may further evaluate Attestation Results according to Policies it may receive from an Owner. The Relying Party may take actions based on the evaluation of the Attestation Results.
- *Endorser*: An Endorser role is typically implemented by a supply chain entity that creates reference Endorsements (i.e., claims, values or measurements that are known to be authentic). Endorsements contain assertions about the device's intrinsic trustworthiness and correctness properties. Endorsers implement manufacturing, productization, or other techniques that establish the trustworthiness properties of the Attesting Environment. This is shown as flows (a) and (b) in Figure 2.
- *Owner of Verifier*: The Verifier Owner role has policy oversight for the Verifier. It generates Appraisal Policy for Evidence and conveys the policy to the Verifier. The Verifier Owner sets policy for acceptable (or unacceptable) Evidence and Endorsements that may be supplied by Attesters and Endorsers respectively.

  The policies determine the trustworthiness state of the Attester and how best to represent the state to Relying Parties in the form of Attestation Results. The Verifier Owner manages Endorsements supplied by Endorsers and may maintain a database of acceptable and/or unacceptable Endorsements. The Verifier Owner authenticates Verifiers and maintains lists of trustworthy Endorsers, peer Verifiers and Relying Parties with which the Verifier might interact.

- *Owner of Relying Party*: The Relying Party (RP) Owner role has policy oversight for the Relying Party (RP). The RP-Owner sets appraisal policy regarding acceptable (or unacceptable) Attestation Results about an Attester that was produced by a Verifier. The RP-Owner sets appraisal policies on the Relying Party that authorizes use of Attestation Results in the context of the relevant services, management consoles, network equipment, an enforcement policies used by the Relying Party. The Relying Party Owner authenticates the Relying Party and maintains lists of trustworthy Verifiers and peer Relying Parties with which the Relying Party might interact.
- *Evidence*: The Attestation Evidence is a role message containing assertions from the Attester role. Evidence should have freshness and recentness claims that help establish Evidence relevance. For example, a Verifier supplies a nonce that can be included with the Evidence supplied by the Attester. Evidence typically describes the state of the device or entity. Normally, Evidence is collected in response to a request (e.g. challenge from Verifier).

  Evidence may also describe historical device states (e.g. the state of the Attester during initial boot). It may also describe operational states that are dynamic and likely to change from one request to the next. Attestation protocols may be helpful in providing timing context for correct evaluation of Evidence that is highly dynamic.
- *Endorsements*: Endorsement structures contain reference *Claims* that are signed by an entity performing the Endorser role (e.g. supply-chain entity or manufacturer of the target device). Endorsements are reference values that may be used by Owners to form attestation Policies.

  Some endorsements may be considered "intrinsic" in that they convey static trustworthiness properties relating to a given actor (e.g., device, environment, component, TCB, layer, RoT, or entity). These may exist as part of the design, implementation, validation and manufacture of that actor implementation.

  An Endorser (e.g. manufacturer) may assert immutable and intrinsic claims in its Endorsements, which then allows the Verifier to carry-out appraisal of the Attester (e.g. device) without requiring Attester reporting beyond simple authentication. It is worth noting that an Endorser can be viewed as an *Oracle* for truthful assertions regarding endorsements in the practical sense of Bellare-Rogaway.

4.2 Summary of an attestation event

Figure 2 illustrates the canonical attestation model. When an Attester (e.g. target device) seeks to perform an action at the Relying Party (e.g. access resources or services controlled by the Relying Party) the Attester must first be evaluated by the Verifier. Among its inputs, the Verifier obtains endorsements from the Endorser (e.g. device manufacturer) in flow (a) of Figure 2. Prior to allowing any entity to be evaluated by the Verifier, the Owner of the Verifier must first configure a number of appraisal policies into the Verifier for evaluating Evidences. The policies are use-case specific but may require other information about the Attester (or User) to be furnished to the Verifier. This is shown in Step 1 of Figure 2. Similarly, in Step 2 the owner of the Relying Party (e.g. resource or service) must configure a number of Appraisal Policies for Attestation Results into the Relying Party.

When the Attester requests access to the resources at the Relying Party (Step 3), it will be redirected to the Verifier (Step 4) – the understanding being that the Attester must deliver attestation Evidence to the Verifier. Included here are the endorsement(s) that the Attester obtained previously from the Endorser (flow(b) of Figure 2). The flow represented by Step 3 may be multi-round and may include a nonce challenge that the Attester must include in its computation of the Evidence as a means to establish freshness.

After verification and appraisal of the Attester completes, the Verifier delivers the Attestation Result to the Relying Party in Step 5. The Relying Party in its turn must evaluate the Result against its own policies (set previously in Step 2). If the Relying Party is satisfied with its evaluation of the Attestation Result regarding the Attester, it will provide the Attester with permission to complete the action it seeks to perform (e.g. access resources at the Relying Party).

4.3 Variations in the attestation flows

There are various possible variations to the message flows shown in Figure 2. These variations may be useful and applicable to use-cases where certain constraints are present (e.g. IoT device with minimal computing power, devices with limited connectivity, etc.).

Two (2) variations are shown in Figure 3. In the first case in Figure 3(a), the Attester delivers its Attestation Evidence to the Verifier as before. However, the Verifier returns the signed Attestation Result to the Attester, which then wields it to the Relying Party. This variation is akin to the "front-channel" flow in the Web-Browser Single Sign-On (Web-SSO) [55] model based on a mediated authentication service by a trusted third-party [31]. Here the Attester's task is to convey unchanged the (signed) Attestation Results produced by the Verifier. This flow is referred to as the "passport flow" in [42] because the Attester is wielding the Attestation Results in the manner of a signed passport or permit.

In the second variation shown in Figure 3b, the Attester delivers its attestation Evidence direct to the Relying Party (i.e. resource manager). Being a reliant party – reliant on the Verifier to evaluate attestation Evidences – the Relying Party simply forwards the Evidence to the Verifier. After the Verifier completes appraisal of the attestation Evidence, it returns the Attestation Results to the Relying Party directly. This flow is referred to as the "background check flow" in [42] – or "back-channel" in SAML2.0 literature [55] – because the delivery of the Attestation Result occurs between the Verifier and the Relying Party over a one-to-one channel without the assistance of the Attester.

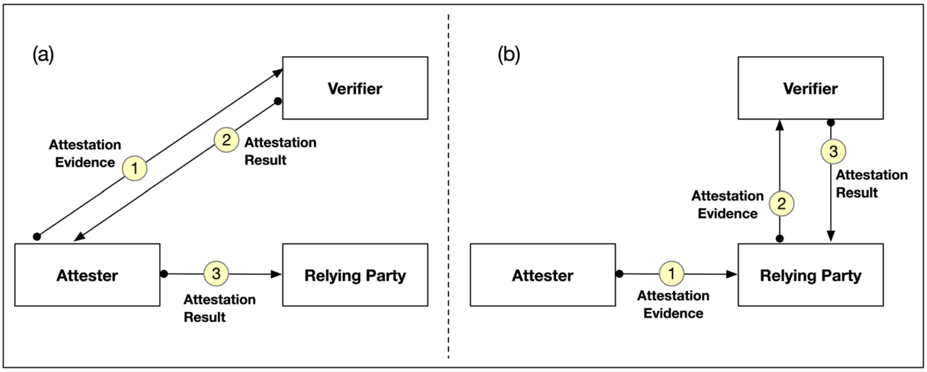

Figure 3 Two variations in attestation flows: **a** The Passport flow, and **b** Background-check flow (after [4])

4.4 Composite attestations

In some cases, an attestation Evidence yielded by an Attester may in fact consists of other Evidence (e.g. from other local components) collated by that Attester (Figure 4).

We refer to this kind of attester as the *Composite Device Attester* and the evidence as *Composite Device Evidence*. In a composite attester scenario, we assume local components have attestation capabilities that generate evidence. This evidence is conveyed locally to a *lead attester* that assembles the various sets of evidences, possibly including evidence that it directly collects as well. Subsequently, the composite device lead attester conveys composite evidence to the relevant Verifier. The composite attester may assert a claim that it was the entity that assembled a piece of component evidence and include this assertion in the composite device evidence it supplies.

4.5 Layered attestations

Another mode of deployment for the attestation architecture is in the appraisal of software (firmware) modules relating to the boot-up sequence within a given device. As mentioned previously, the Trusted Computing Group (TCG) defined a number of hardware-based “roots-of-trust” (RoT) related to the TPM chip [73]. The idea is that because the TPM hardware is tamper-resistant and provided shielded memory and storage, these features could be used as a root-of-trust for ensuring that a TPM-enabled device could boot-up safely and correctly.

However, the TPM-based approach may not be suitable for various other use-cases, and not all devices can suitably support a TPM. Constrained devices such as IoT devices (Internet of Things) – such a low-power sensors and tiny low-cost devices – may not be able to support a TPM or any dedicate security processor. More importantly, not every device architecture maps onto the assumptions underlying the TPM design. In some cases, once the device completes loading the low-level software – whose integrity can be protected by keeping a hash of the code in the registers of the TPM – there comes a point in the boot-sequence where the TPM ceases to be (i.e. unable to be) the root-of-trust for the next piece of software being loaded. That is, there needs to be a variant of the attestation model which

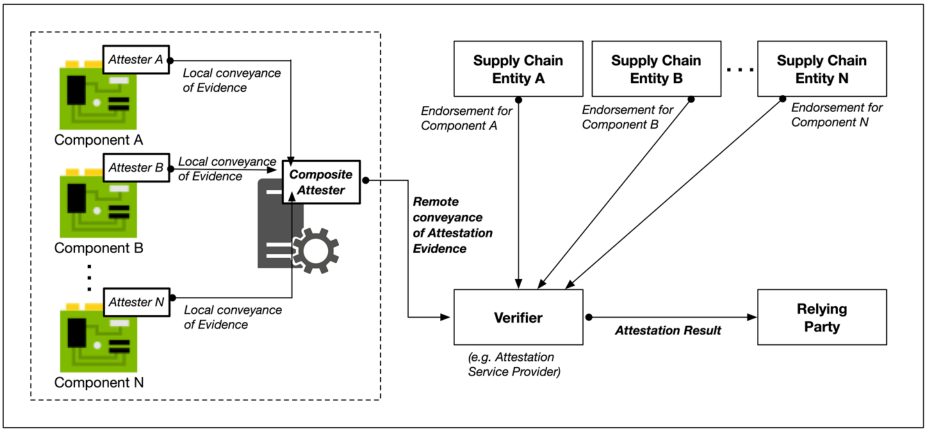
Attester A
Local conveyance of Evidence
Component A
Attester B
Local conveyance of Evidence
Composite Attester
Component B
Attester N
Local conveyance of Evidence
Component N
Remote conveyance of Attestation Evidence
Supply Chain Entity A
Supply Chain Entity B
Supply Chain Entity N
Endorsement for Component A
Endorsement for Component B
Endorsement for Component N
Verifier
(e.g. Attestation Service Provider)
Attestation Result
Relying Party

Figure 4 Example of composite device attestations

can be aided in its initial phases by the use of some hardware-based functions and relevant manufacturer endorsements, but which would require it to be reliant on attestations by other pieces of software in the boot-sequence. We refer to this generically as *layered attestations* (Figure 5).

One specific example of layered attestations can be found in the *Robust Internet-of-Things* (RIoT) architecture [19]. The approach employs a combination of a device-secret that is set by the device manufacturer during production (e.g. fusing during manufacturing). The core idea is to use the device secret and a keyed hash function to derive other secrets (e.g. keys) to be used by the next layer in the boot-sequence.

Although a detailed discussion of layered attestations is beyond the scope of the current work, there are a number of desirable properties for layered attestations (see Section 5 of [32] for an extensive discusion). First, each layer in the sequence must be unambiguously distinguishable (e.g. using a key derivation scheme that provides a unique key for each layer). Secondly, the next layer must be "inspectable" be the current layer. Inspection may simply be to compute a hash value for computation of a layer identity or may involve more rigorous proofs of integrity. Thirdly, there must be a way to achieve layer sequencing – which may different for each type of device. Finally, there must be a way for a layer to provide Attestation Evidence of itself that includes evidences for all previous layers in the sequence. Trust in a current layer depends on the trustworthiness of all previous layers. Consequently, a Verifier of layered attestation must evaluate Attestation Evidence of all the dependent layers before it can reason about trust in the current layer. The set of layered evidence must therefore be communicated within the evidence flow emanating from the Attester.

The TCG has formalized and standardized the notion of layered attestations in the *Device Identity Composition Engine* (DICE) specifications [71, 72]. This standardization process is important not only from a device-manufacturers perspective (and other supply-chain entities), but also from the perspective of the various service providers (e.g. ASPs and Relying Parties) that together with the supply-chain entities form the ecosystem that supports interoperable implementations of the attestation architecture. Currently, the DICE approach

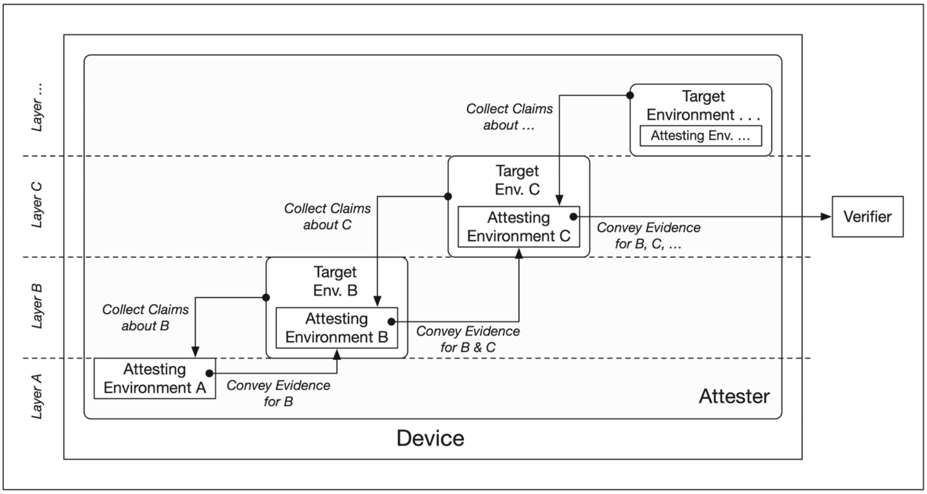

Figure 5 Overview of the concept of the layered device attestations

has been implemented for hardware intended for cloud platforms (e.g. Project Cerberus [43]).

5 Attestation of nodes in blockchain networks

In this section we explore the application of the canonical attestation architecture in Section 4 to the broad case of blockchain networks. We pay close attention to *permissioned* (private) blockchain networks as a means to constrain the scope the problem. We envisage that attestation architectures and technologies discussed in the current work may be of interest in the first instance to communities arranged as consortiums seeking to employ blockchain technology and DLTs generally to solve a specific problem in the community. Examples of communities seeking to use blockchain and DLT technologies to solve industry-specific problems include supply chain management of goods (e.g. TradeLens [51]) and provenance tracking and reducing counterfeit pharmaceutical products (e.g. PharmaLedger [52]). In the financial industry, several organizations are exploring the notion of private blockchains for the purpose of increasing the efficiency of business settlements within their network [16].

There are several high-level desirable features related to the attestation of nodes in a blockchain network:

- *Independence of nodes in verifying other nodes*: A node must be able to take-on the role of a Verifier of the attestation Evidence generated by peer nodes. This means that nodes must have awareness of the identity of its peer nodes. Similarly, a node must be able to generate a signed Attestation Evidence as required (e.g. when demanded by its owner, by the consortium, by peer nodes, or as required by the consensus-protocol in the network).
- *Selectable components or layers to be attested to*: Following the canonical architecture of [64], a Verifier must have the option to request from an Attester node the Attestation Evidence for specific components or layers in the stack (hardware, firmware, software, applications) that make-up that node. This request may be implemented explicitly as a stand alone request-response protocol, or the evidence may be conveyed as part of another protocol (e.g. as part of an access request by the attester node [23]).

  This ability is notably important in the context of smart contracts that are to be executed within Trusted Execution Environments (TEE), such as SGX [15, 50] and ARM Trustzone [53]. Although the technical details are beyond the scope of the current discussion, here the Attestation Evidence must include proof that a specific smart contract (i.e. the correct one) has been loaded into the TEE, that it was authorized by the contract-owner, and that state-changes in the TEE is reportable as proof that the smart contract was truly executed in the TEE hardware. Efforts such as Teechain [46] and Hyperledger Fabric [6] have begun to explore the potential use of TEEs for blockchains, while other efforts (e.g. [12, 66]) are exploring the use of TEEs as the foundation of confidential computing more broadly.
- *Persistence of transaction signing key over reboots*: Since a node's transaction signing key may be used in some consensus-protocols to receive remunerations (e.g. BTCs, "gas"), this private-public key pair must be persistent (i.e. not lost) across reboots of the node-device. This must be true independent of the hardware/software implementation of the node.

- *Inaccessible transaction signing keys when in unapproved configuration*: Since a node must only operate in a configuration and state that is known to and approved by the node's owner, the transaction signing key(s) must not be usable or accessible by the node if it is in an "unhealthy" (i.e. unapproved) state. Among others, this is to prevent malware (i.e. viruses) from using transaction signing keys to perform unauthorized transactions, thereby harming its owner.
- *Observance and enforcement of governance polices*: In a consortium arrangement, there must be a way for the consortium organization to mandate (force) observance by a node of the consortium-wide policies and operating rules. There are various mechanisms that can achieve this effect, one of which is to use Attestation Results as one of the inputs into the consensus-protocol (e.g. if a node is not in one of several configurations approved by the consortium, then the node will never be selected to forge new blocks in a Proof of Stake protocol).

Figure 6 illustrates a number of attestation flows that may occur in a consortium-based permissioned blockchain network. The flow (a) in Figure 6 illustrates situations where the consortium governance administration seeks to verify or appraise the attestation-evidence produced by nodes belonging to members. This "right to verify" may be enshrined within the governance operating contracts of the consortium. In flow (b) nodes are performing mutual attestations of peer nodes in an independent manner (i.e. independent of any centralized entity such as the consortium administration entity). Here, one node conveys its attestation-evidence and another node appraises it. In flow (c), a member may employ its own verifier (e.g. off-chain service) to appraise the evidence conveyed from nodes belonging to other members in the consortium.

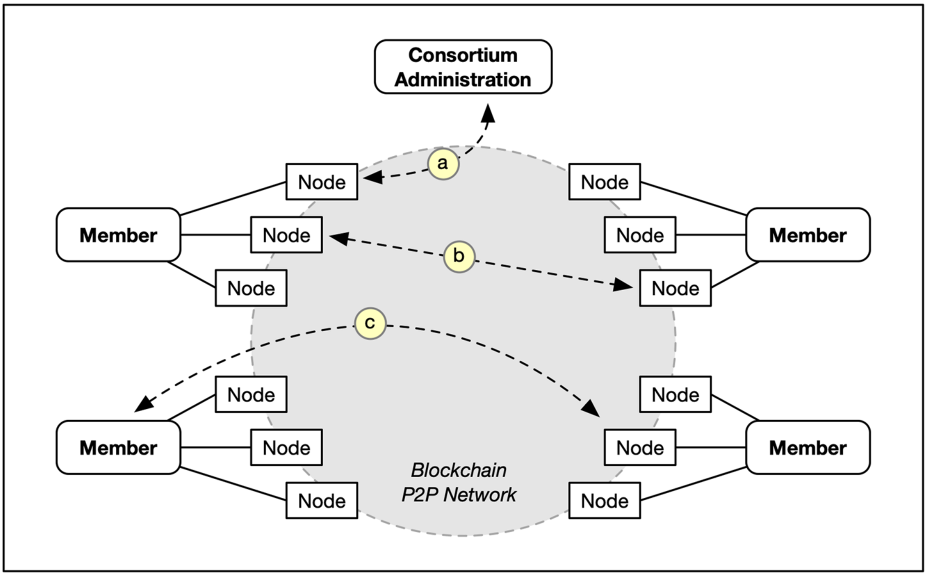

Figure 6 Overview of a consortium arrangement of a permissioned blockchain network

5.1 Domains, nodes and functions

Following from the general entities and roles defined in Section 4.1, in the following we apply these to the entities that participate in a blockchain network (Figure 7):

- *Consortium Verifier*: The consortium as a community owns and operates one or more of its own Verifiers (e.g. Attestation Service Provider) for the purpose of appraising Evidence conveyed by nodes in the network against the consortium's Appraisal Policies.
- *Consortium-wide Appraisal Policies*: Part of the governance of the community is the establishment of a shared set of appraisal policies for Attestation Evidences and Attestation Results. Step (a) of Figure 7 illustrates the conveyance of the consortium-wide appraisal policies to the domains (e.g. the policy store in the management console of domain owner). The intent of Step (a) is to denote that the consortium-wide appraisal policies must be communicated down to the Local Verifier in each node in that domain. Copies of the Appraisal Policies are also recorded on the ledger (i.e. digest or hash of the policies) to allow for future independent audit, such as comparing member policies against the member-agreed baseline policies of the consortium.
- *Member Verifier*: Each member owns and operates one or more or its own Domain Verifier. Thus, for example, in Figure 7 a Member X owns the Domain Verifier DV1 in Domain D1. Each member also operates a local protected *Log*, which is only accessible by nodes and other entities in the domain.
- *Member Appraisal Policies*: Each member may set its own appraisal policies for Attestation Evidences and Attestation Results for its domain. Ideally, member-level policies should not conflict with consortium-level policies. The consortium has the opportunity to use independent audit and compliance entity to ensure that correctness evaluation can be performed. Step (b) and Step (c) in Figure 7 illustrates the conveyance of the consortium-wide appraisal policies and the domain specific appraisal policies to the nodes of the corresponding member (with digests recorded on the ledger).

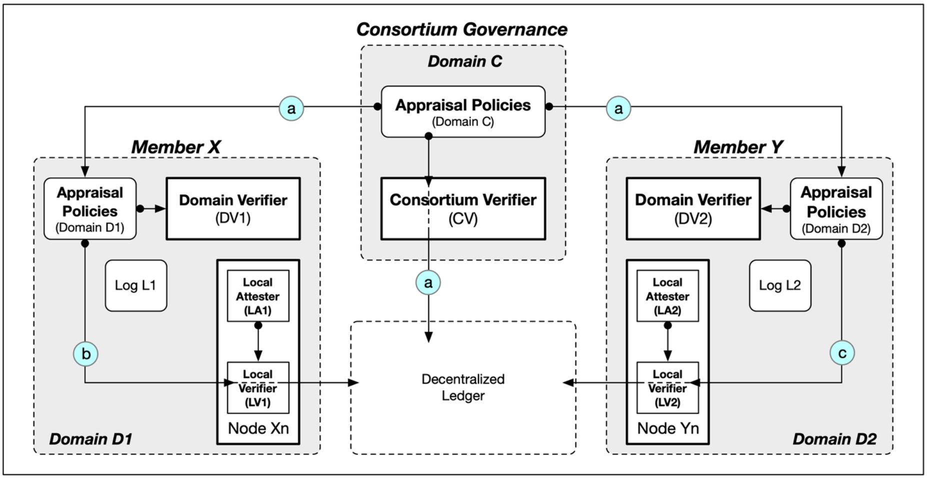

Figure 7 Overview of an attestations architecture for a permissioned blockchain network

- *Node Local Verifier and Local Attester*: Each node in the blockchain network implements a Local Verifier (LV) and Local Attester (LA). The Local Attester creates attestation Evidence about the node and conveys the evidence to a Verifier (i.e. its own Local Verifier, its Domain Verifier, the Consortium Verifier, or another node's Local Verifier).
- *Audit Log*: A domain maintains an audit log system as means to store and retain attestation Evidences (from devices within its domain) and Attestation Results coming from the various Node Local Verifiers as well as the Consortium Verifier (CV). The entries of the local log (i.e. hash of entries) can also be recorded on the ledger. This permits independent audit and compliance entities to check the results are reasonable appraisal results.

There are two broad categories of scenarios that can benefit from attestations. In the first scenario, a Local Attester in a given node conveys attestation Evidence to a Verifier that is either located in its home domain or in the consortium domain. In the second scenario, a Local Attester in a given node conveys attestation Evidence to a Verifier located in another node in a peer-to-peer fashion. We discuss these scenarios below.

5.2 Appraisal by domain verifiers

There are a number of use-cases related to manageability of nodes/devices that may benefit from the ability for a node to convey Evidence regarding its current configuration and other computations state. We use Figure 8 to illustrate.

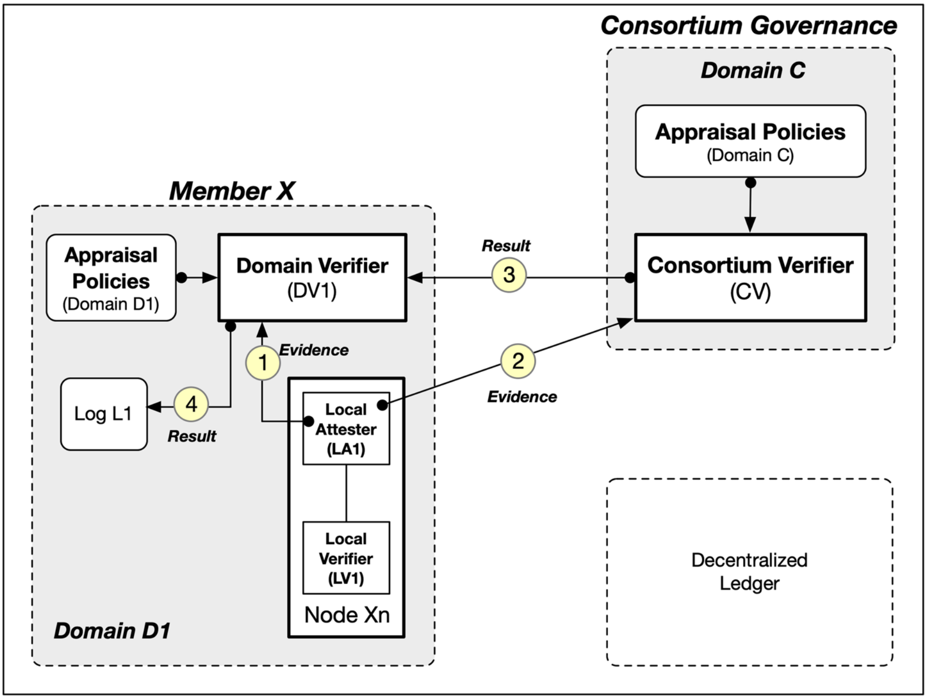

Figure 8 Attestations of a node by its domain verifier

In Step (1) and Step (2) of Figure 8 the Local Attester (LA1) in the node conveys attestation Evidence to (i) its own Domain Verifier (DV1) and (ii) to the Consortium Verifier (CV). Depending on the specific-use case, these two Evidences may differ. Thus, for example, the Domain Verifier may be concerned about both the health of its node and other system attributes of its node (e.g. did a GPU card just malfunctioned). The Evidence conveyed by the Local Attester (LA1) to the Consortium Verifier (CV) may include information that is of interest to the consortium administration. For example, the Consortium Verifier may be seeking status information regarding the versions of the firmware and software on the node, when the last patch was installed, and so on. The goal of the consortium's appraisal policies is to ensure that members and their nodes comply to the operating rules of the consortium organization as defined in its legal trust framework and other related membership contracts.

Note that these two areas of interest – of the Domain Verifier on one hand and the Consortium Verifier on the other hand – are complementary to each other. The understanding here is that the survival of the permissioned blockchain network as a whole is a function of the survival of individual nodes in that network. When the nodes are "healthy" then the network is also healthy.

It is worthwhile to note that the Relying Parties (RP) in these two scenarios are the Domain Owner (i.e. node owner) and the Consortium Administration respectively. Thus, in Step (3) and Step (4) the attestation results are conveyed by the Consortium Verifier and by the Domain Verifier (DV1) respectively. The Domain Owner as a relying party to both of these attestation results may take action based on the received results. For example, it could update the firmware of the node, to bring the node offline, and so on. Thus, the owner's foremost concern could be the visibility into the state of its nodes and maintaining the security and integrity of these nodes.

5.3 Appraisal by peer nodes

One of the key tenets of decentralized computing as exemplified by blockchain systems such as Bitcoin is the autonomy of nodes in performing computations (e.g. Proof-of-Work [54], Proof-of-Stake [9], etc.). To this end, ideally nodes must be able to appraise other nodes as part of a consensus-making protocol. The ability to provide attestation Evidence (e.g. regarding a node's current configuration) enhances the acceptability of the outcomes of peered computations such as PoW and PoS. Other participants in the network obtain some degree of assurance that fairness has been maintained (e.g. that nodes have equal hash-powers or hash-rates). This is especially important for consortium organizations whose members may be competitors.

Another important use-case pertains to nodes that act as gateways between two distinct blockchain networks [32]. Here, the goal of gateways is the establishment of trust for scenarios involving the cross-blockchain transfer of high-value virtual assets. For certain types of virtual assets (e.g. proof of legal ownerships of real-world assets) a change of legal ownership effected on a blockchain may necessitate that the evidence or proof of ownership be moved from one blockchain to another (e.g. from the seller's blockchain to the buyer's preferred blockchain). This movability of virtual assets across blockchain systems is crucial for the scalability of blockchains as an economic medium for business transactions at a global scale.

The appraisal by peer nodes is represented by Figure 9. Here a node X_n is required to provide attestation Evidence to other nodes in the network (e.g. node Y_n) as part of consensus-making (e.g. node X_n to be selected to forge new blocks in PoS [9]). In Step (1) of Figure 9 the attestation Evidence from node X_n is conveyed to the Local Verifier (LV2)

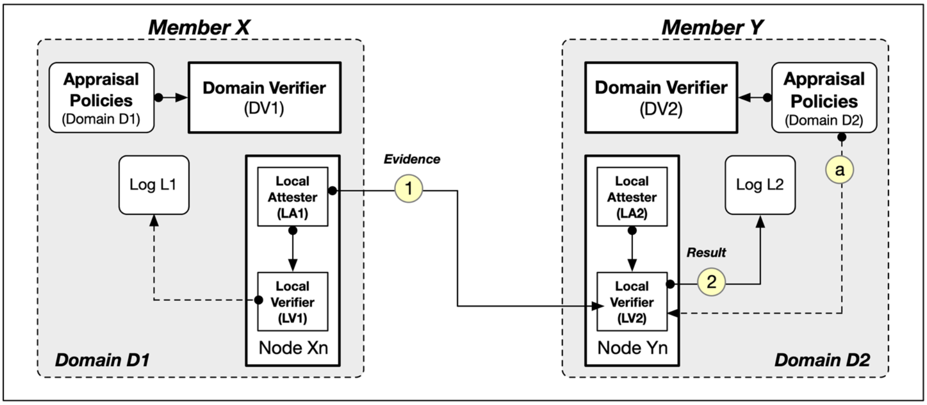

Figure 9 Appraisal of attestations by peer nodes

within node Y_n. Now, LV2 will evaluate the received Evidence based on the appraisal policies (for Evidences and Attestation Results) which it possesses. These policies were previously configured by the Domain Verifier owner (node owner) in Step (a) of Figure 9. If the results of the attestations conforms to the appraisal policies, the node Y_n (as its own Relying Party) may then take action (e.g. confirm proposed new blocks in PoW).

6 Attestation of virtualized nodes: future challenges

Since the emergence of the Bitcoin system [54] in 2008, there has been significant development and departure from the original Bitcoin conception of the topology of nodes (i.e. mining nodes). One key idea in Bitcoin is that the use of physically-separate nodes (i.e. mining rigs) – each with its own copy of the full ledger – would provide a degree of resilience of the network to attackers who wish to skew or compromise the network. An attacker would need to successfully attack a majority (e.g. 51 percent) of the physically-separate nodes in order to compromise the network as a whole [20, 25]. The Ethereum system [10] represented a break from classic Bitcoin topology by expanding the programmability of the blockchain through the use of "smart contracts" that operate within the Ethereum Virtual Machine (EVM). The EVM is essentially stack machine [76] that operates in a virtual space made available on the nodes of Ethereum.

6.1 Cloud computing, CaaS and BaaS

Given the complexity of operating nodes and the resources needed to maintain a network of nodes, it is reasonable to expect that the nascent *virtual assets* [22, 38] industry will look to cloud computing as a means to increase scale while reducing operational costs. Indeed, currently new forms of cloud-based offerings have begun to emerge marketed to various blockchain-based use cases. These offerings range from *Container as a Service* (CaaS) to *Blockchain as a Service* (BaaS). In the CaaS case, the node-owner can create an image (e.g. Docker image) of the node and have it execute in the CaaS infrastructure (e.g. IBM cloud [41]). In the BaaS case, the entire blockchain network can be hosted on a third-party infrastructure (e.g. Microsoft Azure BaaS [44]).

However, several challenges remain to be addressed with regards to the CaaS and BaaS models for blockchain networks. Some of the challenges related to nodes implemented in a BaaS platform include: (i) the security of cryptographic keys of nodes in the cloud; (ii) the integrity – and in some cases the confidentiality – of the ledger data held by nodes; (iii) the secure migration of nodes – or processes implementing the node – from one virtualized stack to another; (iv) malicious interference by adjacent processes in multi-tenant deployments; (v) geographic diversity of the nodes; and so on.

6.2 Geographic diversity of nodes

Historically, geographic diversity has not been reliably enforced as part of a blockchain network. For example, Bitcoin's Proof-of-Work (PoW) focuses mainly on how how quickly the miner can solve the PoW cryptographic puzzle. For the resilience of the network, ideally nodes (miners) should be geographically spread, and geo-politically diverse. Geopolitical diversity has ramifications on the stability of the value of the virtual assets transacted on the blockchain (e.g. see [48]). The attestation architecture we have described above permits nodes/devices to provide evidence of its geolocation. For example, the work of [49] includes the ability to report location coordinates (latitude, longitude and altitude) of the attester device. In turn, this can be reinforced with geo-fence policies relevant to the specific deployment scenario.

We believe that both geo-location evidence and geo-fence policy compliance should in fact be integrated into the consensus protocol of the blockchain network (e.g. PoW, PoS or future variants) where the consensus protocol enforces geo-diversity. This should be true including for nodes implemented using CaaS or BaaS technology. Using the Proof-of-Stake (PoS) example, a node's eligibility factors – to be selected as the validator node to forge the next block – should include the geo-location of the node and the geo-locations of the population of the other eligible nodes. Using the peer-appraisal approach outlined in Section 5.3, a community of nodes could collectively self-enforce this geo-diversity requirement as part of their consensus-making algorithm.

A similar notion in the context of IP routing – referred to as *trusted path routing* – has been proposed in [74], where routers (i.e. routing nodes) in a traditional IP network employ attestation of peers to exclude routers whose attestation evidence does not meet a policy agreed upon by the community of nodes. Although trusted path routing envisages a distributed attestation appraisal solution that approaches a distributed consensus algorithm, more work could be directed at developing an "equivalent" of consensus algorithm but for routers eligible to be in the trusted-path (i.e. route selection versus mining/forging blocks).

6.3 Integrity of cloud platforms

We believe that a secure attestations capability for nodes implemented on a BaaS (or CaaS) platform represents an important factor for the business value-proposition of the BaaS model. Attestations capability are needed at different layers of the virtualization stack according to the corresponding "consumer" (Verifier and Relying Party) of the attestation information. Thus, for example, the BaaS provider needs visibility into the state of the platform as a whole, while the node-owner needs visibility into the integrity of its node. Counter-parties in transactions may wish to see attestation results for nodes running on the BaaS platform, and so on.

For cloud providers generally, there are a number of challenges in providing a reliable and safe virtual computing infrastructures [60]. There needs to be a way to validate the integrity of all firmware updates as a follow-up after the *first-instruction integrity* has been completed [13]. Thus, after the integrity of the first code or data loaded (e.g. from mutable non-volatile media) has been verified (e.g. by either the cloud provider or the manufacturer), the integrity of all firmware updates must also be achieved in a verifiable manner [19]. Achieving this higher level of integrity implies that there could be several roots of trust (RoT) or chains of trust (CoT) that are integral to the platform.

Furthermore, there needs to be a way to *detect unauthorized access or corruption* to the software or firmwares during operations, and then take recourse to remediate this problem – possibly in an automated or semi-automated manner. There needs to be a way to *restore firmware to state of integrity* in cases where corruption has been detected [60]. Given the nature of cloud data centers, recovery must be achievable in an automated fashion without immediate attendance of the IT administrator. Manual recovery must of course be supported. Recovery in this case generally means automatic self-recovery of the critical-to-boot portion(s) of the firmware.

6.4 Decentralized roots of trust

Another challenge pertains to the notion of *decentralized roots of trust* (d-RoT) put forward in [32], where the logic of trust for a given multi-node (multi-component) environment should be based on a decentralized cooperation of those nodes (components). The patterns of roles, interaction and relationships we saw previously in Section 4 – namely endorsements, attestation-evidence, appraisal and attestation-results – must hold true within individual computer systems (which may consists of a complex interaction of parts and components) as well as within a group of distributed computer systems (e.g. nodes on a blockchain network).

For example, within a single computer system a root-of-trust in a hardware CPU or a core package containing multiple cores may have a design that justifies it as a root-of-trust. However, when put together with other components on a system bus there maybe other roots-of-trust present (e.g. system-on-chip (SoC), IP blocks and peripherals, etc.) that may have equal access to the system bus and therefore can assert themselves as a root-of-trust. In this case there must be a distributed mechanism for bootstrapping trust in the system as a whole based on a collective action or consensus of these roots-of-trust. Thus, there should be (must be) be a distributed trust logic being applied at this *microcosm* level of components ("nodes") within one environment (the computer system as a whole).

Equally, the distributed trust logic must also apply at the *macrocosm* level consisting of multiple computer systems, each of which are acting as a distinct node participating with other nodes within a blockchain network. The blockchain network becomes a natural extension of the principles and properties described in [32] (e.g. group reporting, group computation participation, etc.), with the appropriate attention given to constructs at the macro level, such as correct integration with the consensus algorithms and architecture for establishing a d-RoT at the macro (peer-nodes) level.

6.5 Efficiency of node attestations

The application of attestations to nodes within a blockchain system must be carefully integrated into the decentralization infrastructure of the blockchain, informed by the architecture of the blockchain itself (e.g. type of consensus protocol, type of nodes, etc.).

One key challenge is for nodes to have access to the correct and authentic endorsements pertaining to the node hardware and software components (i.e. of all nodes in the network) in a timely manner. Thus, for communities (e.g. consortiums) seeking to employ attestation technologies, sharing a common repository of signed endorsements may alleviate this challenge.

Another challenge pertains to the latency and increase in computation workloads on the part of the nodes – in gathering and reporting attestation evidences and for other nodes to verify these evidences. Aside from regular keep-alive reporting of attestation evidences, other strategies may be employed to increase the efficiency of reporting. A strategy that could used in blockchain designs that require each node to independently race to solve the next block (e.g. mining in Bitcoin) is for the node that wishes to claim mining success to include attestation evidence when broadcasting the claim. This strategy may be used to reduce the frequency of fresh keep-alive reporting of evidences. A similar strategy may be applicable to other types of blockchains that employ an election approach of subsets of nodes to perform critical tasks (e.g. Orderers in Hyperledger Fabric).

Further research is needed to understand the latency impact of attestations when nodes employ Trusted Execution Environment (TEE) technologies and where smart contracts are executed inside the TEEs (e.g. to provide contract confidentiality). The work of [6] has identified several challenges with the regards to the application of smart contracts to TEEs in the context of Hyperledger Fabric. A simplified smart contracts approach has been proposed in [37] where a community of contract service providers (CSP) establish a blockchain system with a fixed set of pre-approved smart contracts (i.e. no end-user programmable contracts). This means that nodes and their TEEs have a pre-defined small number of smart contracts that they are permitted to execute, thereby reducing the range of components (i.e. contract variability) that must be attested to by a given node in the CSP community. This approach is reminiscent of attestations in the context of routers in a domain [23, 74], with each router device having a fixed set of hot-swappable software modules and where the routers are accessible only to the IT administrator in the domain.

7 Areas for innovation

Despite the notion of device attestations nearing two-decades in age [73], the concepts around attestations – such as endorsements, validations and freshness – are just recently coming into wider attention in the broader industry. We believe more research needs to be applied, and several areas of innovation still await the industry as a whole. In the context of the application of the attestation architecture to blockchain networks the following represents a brief list of possible areas for future innovation:

7.1 Dynamic governance-policy setting based on attestations

For permissioned blockchains in a consortium arrangement, the governance-level appraisal policies should be dynamic in that their governance parameters should be subject to orchestrated change that adapts to the shifting environment of software evolution and hardware replacement cycles. This should be done in accordance with a combination of the parameter for category type (e.g. node diversity) and the overall state of the population of nodes in the network.

Thus, given a node that addresses diversity (software and hardware diversity), if for some reason the population of nodes become increasingly non-diverse (homogeneous) up to defined threshold, then the governance policies should change to account for the loss of expected diversity – and subsequently influence (i.e. modify) the consensus algorithm toward stasis of an intended diversity metric. This could be a direct change in the parameter of the consensus algorithm (e.g. majority parameter raised to 70 percent of population from 51 percent), or it could be an indirect change through new policies being pushed into the domain-level (member level) appraisal policies (e.g. Local Verifier in a node belonging to a subset of diverse nodes should prioritize members of that same subset when accepting proposed consensus outcomes). The goal should be, among others, to incentivize members of the consortium to deploy nodes that satisfy the diversity category (software and hardware diversity) for the sake of the resilience of the community as a whole.

7.2 Attestation for migration of containerized-nodes

In the context of trusted hardware used in virtualized platforms, one difficult challenge facing cloud data centers is *trusted migration* of containers and functions [1, 24]. This is especially important for containerized nodes that carry sensitive information such as application-layer cryptographic keys and related keying material, which may need higher availability.

By design, a container is unaware of "where" it executes (i.e. the specific hardware environment – model/version). However, as we have seen above, execution environment endorsements necessarily describe a piece hardware (with its firmware and software). This is crucial for reasoning about trust among multiple interrelated nodes. Given that attestation-derived trust reaches past the containers' presumed isolation domain, the problem becomes complicated by the need for prescribing destination hardware (to which a container is to be migrated) that provides "equivalent trust", but where the migration target might not exactly mimic the currently executing container. Properly secured migration policy expects equivalent or better trust in the target migration environment.

A related challenge facing trustworthy container migrations is how to automate the verification (comparison) between the current execution environment versus the target migration environment.

7.3 Attestation for nodes of permissionless blockchain networks

To avoid confusion when discussing permissionless blockchains, we employ the NIST definition of permissionless blockchains [77], namely a system where all users' permissions are equal and not set by any administrator or consortium. Permissionless blockchain networks are decentralized ledger platforms open to anyone publishing blocks, without needing permission from any authority (see Section 2.1 of [77]).

One possible application of attestations in permissionless blockchains with anonymous nodes (e.g. miners) is for the gradual establishment of a subset of attestation-capable nodes that periodically report attestation-evidence at a regular basis to the public ledger. For simplicity, we refer to these as "honest-nodes". Honest users wishing to choose to have their transactions be processed by one or more of this subset of honest-nodes can pre-register (i.e. self-declare) their public-keys to these honest-nodes. In turn, when an honest-node searches through the list of unprocessed transactions (e.g. in the UTXO model [5]), the node can choose only those new transactions which originated from pre-registered user public-keys, according to the attestation policy common among the honest-nodes. Users can directly remunerate the honest-nodes (who successfully created a block containing transaction from honest-users) by sending coins to the address of these known honest-nodes.

In effect, this creates a *segregation of honest-nodes* from the broader population of anonymous nodes in the permissionless blockchain, something that might be considered a form of *semi-permissioned* blockchain. This may provide a path forward for blockchains that today suffer from the imbalance of hash-power, where some nodes (i.e. mining pools) control too much hash-power and therefore introducing potential instability into the blockchain. Such an approach is outlined in [34] based on the use of certain types of anonymity-preserving keys in the TPM hardware (e.g. DAA [7], EPID [8]).

7.4 Blockchain-based retention of supply-chain endorsements

Beyond the challenges to ensure that endorsements are correct and source-authentic, there is the industry challenge of ensuring the continual availability and update of endorsements, and the availability infrastructures and services (e.g. certificate services) which support attestation evidence verifications.

In the context of the continuous availability of manufacturer-signed endorsements, two challenges are that: (i) a manufacturer ceases to exist, and/or that (ii) an issuing CA ceases to exist. A manufacturer may go out of business entirely, or undergo mergers with other business entities. This could mean that the manufacturer is no longer able to manage the repositories holding its signed endorsements. Secondly, the CA that issued the corporate signing-certificate – for the now-defunct manufacturer – could itself go out of business. This means that the CA is not longer able to manage its CA infrastructure (i.e. to renew the manufacturer signing certificate).

Blockchain technology may be able to help address these two challenges, in the following manner. The basic notion is that when a manufacturer today creates a product and the endorsements for that product, these endorsements could be "registered" on a special blockchain (referred to as the *Endorsements Ledger*) that acts as a decentralized notarization service for the endorsements. This approach complements the manufacturer signature on an endorsement, and it allows a future verifier in possession of the product to use the entries in the blockchain – referred to as *endorsement-records* – to validate that: (1) the endorsement matches the product, and that (2) the endorsement is source-authentic from the manufacturer at the time that the endorsement-record was created (i.e. today) – even though at the future time of the verification the manufacturer may no longer exist.

The blockchain endorsement-record must also include pointers to locations (e.g. archival repositories) on the Internet where copies of the software/firmware for the product (and the corresponding endorsement objects) may be found. Since a manufacturer's endorsement for its software or firmware product typically include a hash of the software/firmware, a future verifier who fetches the endorsement-record from the blockchain obtains assurance that the copy of the product in its possession is a genuine product (i.e. unmodified). Figure 10 provides an overview of this idea.

In the example of Figure 10, the endorsement-record (Step (b)) collates the hashes (digests) of the various objects that support the verification of a manufacturer's endorsement (e.g. the endorsement or RIM [30, 45], manufacturer's signing certificate, CA's root certificate, etc.) from Step (a). In essence, the endorsement-record becomes akin to the root (top) of a Merkle hash-tree. The manufacturer also stores copies of the relevant verification objects in a decentralized file repository system (e.g. IPFS [58]) to ensure the high-availability of these objects (Step (c)). The decentralized file repository must ensure availability of these object long into the future. Other traditional methods, such as distributing via CD/DVD discs (e.g. for consumers of the product) could also be performed. Finally, the manufacturer transmits the endorsement-record from Step (b) onto the Endorsements Ledger in Step (d). The endorsement-record (as a transaction on the blockchain) needs to

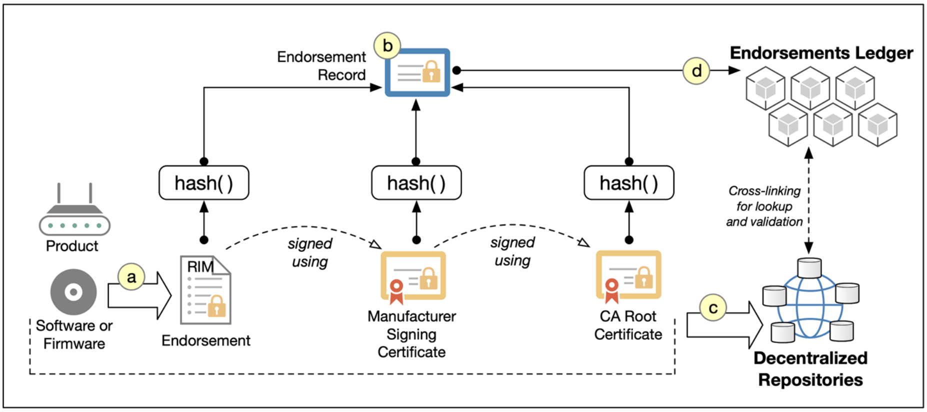

Figure 10 Overview of blockchain-based retention of endorsements

include pointers to (e.g. URL/URI) to the locations of these verification objects, such that any verifier can later easily fetch these objects and perform endorsement verification.

8 Conclusions

As mentioned in the opening, we believe that there is a strong role for trusted computing technologies, and more specially attestations technologies, in the growing area of blockchain networks. The security, resiliency and interoperability of blockchain systems are important factors in the adoption of blockchain networks as the foundation of the future digital economy.

In the current work we have provided a high-level review of the notion of attestations, and described the evolving new standard architecture for device attestations. The application of this attestations architecture to the blockchain environment have been discussed. The blockchain industry's use of a common standard attestations architecture can ensure the best chance for the interoperability of systems and networks, and offers the best path forward towards achieving the survivability of various blockchain networks.

Several challenges remain to be addressed as new modes of implementations of blockchain networks, such as virtualization and containerization, become attractive for deployers of blockchains.

Acknowledgements We thank Sandy Pentland, Alexander Lipton and Stephen Buckley at MIT for support for this work.

We thank the following from Intel Corporation for their support in this work and other efforts in trusted computing: David Grawrock, Abhilasha Bhargav-Spantzel, Rajesh Poornachandran, Michael J. Reed, Andrew Draper, Jose Benchimol, Xiaoyu Ruan, Alberto J. Munoz, Dan Middleton, and Vincent R. Scarlata.

We thank the various members of the TCG who have devoted a considerable part of their lives over the past two decades towards making trusted computing a reality. In particular, in connection with the attestation related efforts in the TCG we thank (alphabetical): Ronald Aigner, Henk Birkholz, David Challener, Paul England, Guy Fedorkow, Jessica Fitzgerald-McKay, Nick Grobelny, Steve Hanna, Dennis Mattoon, Amy Nelson, Graeme Proudler, Monty Wiseman, and David Wooten. From the IETF RATS working group we thank the following: Nancy Cam-Winget, Roman Danyliw, Thomas Fossati, Ben Kaduk, Kathleen Moriarty, Michael Richardson, Dave Thaler, Hannes Tschofenig, and Eric Voit.

References

1. Alder, F., Asokan, N., Kurnikov, A., Paverd, A., Steiner, M.: S-FaaS: Trustworthy and Accountable Function-as-a-Service using Intel SGX. [Online]. Available: arXiv:1810.06080.pdf (2018)
2. Atzei, N., Bartoletti, M., Cimoli, T.: A survey of attacks on Ethereum smart contracts. IACR Cryptology ePrint Archive **2016**, 1007 (2016). [Online]. Available: http://eprint.iacr.org/2016/1007
3. Balacheff, B., Chen, L., Pearson, S., Plaquin, D., Proudler, G.: Trusted Computing Platforms: TCPA Technology in Context. New York, Prentice Hall (2002)
4. Birkholz, H., Thaler, D., Richardson, M., Smith, N., Pan, W.: Remote Attestation Procedures Architecture, IETF, Internet-Draft draft-ietf-rats-architecture-07. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-rats-architecture/ (2020)
5. Bitcoin.org, Unspent Transaction Output (UTXO) (2020). Available at https://bitcoin.org/en/glossary/unspent-transaction-output. Accessed 7 May 2020
6. Brandenburger, M., Cachin, C., Kapitza, R., Sorniotti, A.: Blockchain and Trusted Computing: Problems, Pitfalls, and a Solution for Hyperledger Fabric. [Online]. Available: https://arxiv.org/pdf/1805.08541.pdf (2018)
7. Brickell, E., Camenisch, J., Chen, L.: Direct anonymous attestation. In: Proceedings of the 11th ACM Conference on Computer and Communications Security CCS2004, pp. 132–145. ACM (2004). https://doi.org/10.1145/1030083.1030103
8. Brickell, E., Li, J.: Enhanced privacy ID: a direct anonymous attestation scheme with enhanced revocation capabilities. IEEE Transactions on Dependable and Secure Computing **9**(3), 345–360 (2012)
9. Buterin, V.: Proof of Stake FAQ. [Online]. Available: https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ (2019)
10. Buterin, V.: Ethereum: a Next-Generation Cryptocurrency and Decentralized Application Platform, Bitcoin Magazine, Report. https://bitcoinmagazine.com/articles/ethereum-next-generation-cryptocurrency-decentralized-application-platform-1390528211/ (2014)
11. Challener, D., Yoder, K., Catherman, R., Safford, D., Van Doorn, L.: Practical Guide to Trusted Computing. New York, IBM Press (2008)
12. CCC: Confidential Computing Deep Dive v1.0 - A Publication of The Confidential Computing Consortium, October 2020. [Online]. Available: https://confidentialcomputing.io
13. CSA: Firmware Integrity in the Cloud Data Center, Cloud Security Alliance (CSA), Whitepaper, 2018. [Online]. Available: https://downloads.cloudsecurityalliance.org/assets/research/firmware/firmware-integrity-in-the-cloud-data-center.pdf
14. Coker, G., Guttman, J., Loscocco, P., Herzog, A., Millen, J., Ohanlon, B., Ramsdell, J., Ariel, J., Segall, S., Sniffen, B.: Principles of remote attestation. International Journal of Information Security **10**, 63–81 (2011). [Online]. Available: https://doi.org/10.1007/s10207-011-0124-7
15. Costan, V., Lebedev, I., Devadas, S.: Secure Processors Part I: Background, Taxonomy for Secure Enclaves and Intel SGX Architecture. Boston: Now Publishers Inc. vol. 11, no. 1-2. [Online]. Available: http://dx.doi.org/10.1561/1000000051 (2017)
16. del Castillo, M.: Citi, Goldman Sachs Conduct First Blockchain Equity Swap On Ethereum-Inspired Platform, Forbes. [Online]. Available: https://www.forbes.com/sites/michaeldelcastillo/2020/02/06/citi-goldman-sachs-conduct-first-blockchain-equity-swap-on-ethereum-inspired-platform (2020)
17. Dickerson, T., Gazzillo, P., Herlihy, M., Koskinen, E.: Adding concurrency to smart contracts. In: Proceedings of the ACM Symposium on Principles of Distributed Computing PODC'17, pp. 303–312. New York, Association for Computing Machinery (2017). [Online]. Available: https://doi.org/10.1145/3087801.3087835

18. E. Palmer (Ed.): Attestation of System Components v1.0 - Requirements and Recommendations - Open Compute Project (OCP), November 2020. [Online]. Available: https://www.opencompute.org/projects/security
19. England, P., Marochko, A., Mattoon, D., Spiger, R., Thom, S., Wooten, D.: RIoT - A Foundation for Trust in the Internet of Things, Microsoft Research, Tech. Rep. MSR-TR-2016-18. [Online]. Available: https://www.microsoft.com/en-us/research/publication/riot-a-foundation-for-trust-in-the-internet-of-things/ (2016)
20. Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. In: Financial Cryptography and Data Security - 18th International Conference, FC 2014, pp. 436–454 (2014)
21. Farrell, S., Housley, R.: An Internet Attribute Certificate Profile for Authorization. IETF Standard RFC3281. [Online]. Available: http://tools.ietf.org/rfc/rfc3281.txt (2002)
22. FATF: International Standards on Combating Money Laundering and the Financing of Terrorism and Proliferation, Financial Action Task Force (FATF), FATF Revision of Recommendation 15, October 2018, available at: http://www.fatf-gafi.org/publications/fatfrecommendations/documents/fatf-recommendations.html
23. Fedorkow, G., Voit, E., Fitzgerald-McKay, J. In: TPM-based Network Device Remote Integrity Verification, IETF, Internet-Draft draft-fedorkow-rats-network-device-attestation-05 (2020). [Online]. Available: https://datatracker.ietf.org/doc/draft-fedorkow-rats-network-device-attestation/
24. Fowler, M.: Available at https://martinfowler.com/articles/serverless.html. Accessed 7 May 2020 (2018)
25. Gervais, A., Karame, G.O., Capkun, V., Capkun, S.: Is bitcoin a decentralized currency? IEEE Security & Privacy **12**(3), 54–60 (2014)
26. GlobalPlatform: GlobalPlatform and the Trusted Computing Group Form Work Group to Drive Mobile Security Standards and Solutions, June 2012. [Online]. Available: https://globalplatform.org
27. Gray, J.: The transaction concept: virtues and limitations. In: Very Large Data Bases – Proceedings of the 7th International Conference, Cannes, France, pp. 144–154 (1981)
28. Hardjono, T.: Building trust through strong digital identity. Embedded Computing Design: 13–18 (2008)
29. Hardjono, T.., Smith, N. (Eds.): TCG Infrastructure Reference Architecture for Interoperability (Part 1) – Specification Version 1.0 Rev 1.0, Trusted Computing Group, TCG Published Specification, June 2005. [Online]. Available: https://trustedcomputinggroup.org/wp-content/uploads/IWG_Architecture_v1_0_r1.pdf
30. Hardjono, T.., Smith, N. (Eds.): TCG Infrastructure Working Group architecture (Part 2) – Integrity Management – Specification Version 1.0 Rev 1.0, Trusted Computing Group, TCG Published Specification, November 2006, Available at http://www.trustedcomputinggroup.org/resources
31. Hardjono, T.: Federated Authorization over Access to Personal Data for Decentralized Identity Management. In: IEEE Communications Standards Magazine – The Dawn of the Internet Identity Layer and the Role of Decentralized Identity, vol. 3, no. 4 (2019). [Online]. Available: https://doi.org/10.1109/MCOMSTD.001.1900019
32. Hardjono, T., Smith, N.: Decentralized Trusted Computing Base for Blockchain Infrastructure Security, Frontiers Journal - Special Issue on Finance, Money & Blockchains, vol. 2. [Online]. Available: https://doi.org/10.3389/fbloc.2019.00024 (2019)
33. Hardjono, T.: Blockchain Interoperability and Survivability. Presentation 2018 IEEE Global Blockchain Summit, NIST, Gaithersburg, MD (17-19 September 2018) (2018)
34. Hardjono, T., Pentland, A.: Verifiable Anonymous Identities and Access Control in Permissioned Blockchains, MIT Connection Science & Engineering, Technical Report. Available at arXiv:1903.04584 (2016)
35. Hardjono, T., Lipton, A., Pentland, A.: Towards an interoperability architecture blockchain autonomous systems. IEEE Transactions on Engineering Management **67**(4), 1298–1309 (2019). [Online]. Available: https://doi.org/10.1109/TEM.2019.2920154
36. Hardjono, T., Hargreaves, M., Smith, N.: An Interoperability Architecture for Blockchain Gateways, IETF, Internet-Draft draft-hardjono-blockchain-interop-arch-01. [Online]. Available: https://datatracker.ietf.org/doc/draft-hardjono-blockchain-interop-arch/ (2020)
37. Hardjono, T., Lipton, A., Pentland, A.: A Contract Service Provider Model for Virtual Assets, in 6th International Workshop on P2P Financial Systems, London. [Online]. Available: arXiv:2009.07413 (2020)
38. Hardjono, T., Lipton, A., Pentland, A.: Towards a public key management framework for virtual assets and virtual asset service providers. Journal of FinTech **1**(1). Available at arXiv:1909.08607. [Online]. Available: https://doi.org/10.1142/S2705109920500017 (2020)
39. Hardjono, T., Lipton, A., Pentland, A. In: Pentland, A., Lipton, A., Hardjono, T. (eds.): Interoperability of Distributed Systems, in Building the New Digital Economy. MIT Press, Cambridge (2021)

40. Herlihy, M.: Blockchains from a distributed computing perspective. Communications of the ACM **62**(2), 78–85 (2019). [Online]. Available: https://doi.org/10.1145/3209623
41. IBM: IBM Blockchain Platform , IBM Corporation, Technical Overview, September 2019. [Online]. Available: https://www.ibm.com/cloud/blockchain-platform
42. IETF: Remote ATtestation ProcedureS (RATS) Working Group – Approved Charter, Internet Engineering task Force, March 2019. [Online]. Available: https://datatracker.ietf.org/wg/rats/about/
43. Kelly, B.: Project Cerberus Security Architecture Overview Specification, Open Compute Project, Published Specifications. [Online]. Available: https://github.com/opencomputeproject/Project_Olympus/blob/master/Project_Cerberus/Project (2017)
44. Lardinois, F.: Microsoft launches a fully managed blockchain service, Techcrunch. [Online]. Available: https://techcrunch.com/2019/05/02/microsoft-launches-a-fully-managed-blockchain-service/ (2019)
45. Lear, E., Droms, R., Romascanu, D.: Manufacturer Usage Description (MUD) Specification (RFC8520). [Online]. Available: https://tools.ietf.org/html/rfc8520 (2019)
46. Lind, J., Naor, O., Eyal, I., Kelbert, F., Pietzuch, P., Sirer, E.G.: Teechain: A Secure Payment Network with Asynchronous Blockchain Access. [Online]. Available: arXiv:1707.05454.pdf (2019)
47. Lindemann, R., Jones, M.B.: FIDO 2.0: Key Attestation Format, FIDO Alliance, FIDO Alliance Proposed Standard. [Online]. Available: https://fidoalliance.org/specs/fido-v2.0-ps-20150904/fido-key-attestation-v2.0-ps-20150904.html (September 2015)
48. Lipton, A., Pentland, A.: Breaking the bank. Sci. Am. **318**(1), 26–31 (2018)
49. Mandyam, G., Lundblade, L., Ballesteros, M., O'Donoghue, J.: The Entity Attestation Token (EAT), IETF, Internet-Draft draft-ietf-rats-eat-03. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-rats-eat/ (2020)
50. McKeen, F., Alexandrovich, I., Anati, I., Caspi, D., Johnson, S., Leslie-Hurd, R., Rozas, C.: Intel software guard extensions (Intel SGX) support for dynamic memory management inside an enclave. In: Proc. Workshop on Hardware and Architectural Support for Security and Privacy (HASP) 2016, Seoul (2016). http://caslab.csl.yale.edu/workshops/hasp2016/program.html
51. Miller, R.: IBM teams with Maersk on new blockchain shipping solution, Tech Crunch. [Online]. Available: https://techcrunch.com/2018/08/09/ibm-teams-with-maersk-on-new-blockchain-shipping-solution/ (2018)
52. Morris, N.: 12 global pharmaceutical firms join EU blockchain consortium PharmaLedger, Ledger Insights. [Online]. Available: https://www.ledgerinsights.com/pharmaledger-pharmaceutical-blockchain-eu/ (2020)
53. Müller, C., Brandenburger, M., Cachin, C., Felber, P., Göttel, C., Schiavoni, V.: TZ4Fabric: Executing Smart Contracts with ARM TrustZone. [Online]. Available: aarXiv:2008.11601.pdf (2020)
54. Nakamoto, S.: Bitcoin: a Peer-to-Peer Electronic Cash System. [Online]. Available: https://bitcoin.org/bitcoin.pdf (2008)
55. OASIS: Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005, available on http://docs.oasisopen.org/security/saml/v2.0/saml-core-2.0-os.pdf
56. OCP: Open Compute Project, 2020. [Online]. Available: https://www.opencompute.org
57. Pentland, A.: Building the New Economy: What We Need and How to Get There. In: Pentland, A., Lipton, A., Hardjono, T. (eds.) Building the New Digital Economy. MIT Press (2021)
58. Protocol Labs: Inter Planetary File System (IPFS) (2019). Available at https://docs.ipfs.io. Accessed 23 September 2019
59. Proudler, G., Chen, L., Dalton, C.: Trusted Computing Platforms: TPM2.0 in Context. New York, Springer (2014)
60. Regenscheid, A.: Platform Firmware Resiliency Guidelines, National Institute of Standards and Technology, NIST Publication SP 800-193. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-193/final (2018)
61. Rosenstein, M.A., Geer, D.E., Levine, P.J.: The athena service management system. In: Proceedings of the USENIX Winter Conference. Dallas, Texas, USA, January 1988, pp. 203–211. USENIX Association (1988)
62. Saltzer, J.H.: Protection and the control of information sharing in MULTICS. Commun. ACM **17**(7), 388–402 (1974)
63. Siegel, D.: Understanding the DAO Attack, Coindesk. [Online]. Available: https://www.coindesk.com/understanding-dao-hack-journalists (2016)
64. Smith, N. (Ed.): TCG Attestation framework, Trusted Computing Group. TCG Draft Specification – Version 1.0, November 2020
65. Steiner, J.G., Neuman, B.C., Schiller, J.I.: Kerberos: an authentication service for open network systems. In: Proceedings of the USENIX Winter Conference. dallas, Texas, USA, January 1988, pp. 191–202 (1988)

66. Sturzenegger, D., Sardon, A., Deml, S., Hardjono, T.: Confidential Computing for Privacy-Preserving Contact Tracing. [Online]. Available: arXiv:2006.14235.pdf (2020)
67. Traiger, I.L., Gray, J., Galtieri, C.A., Lindsay, B.G.: Transactions and Consistency in Distributed Database Systems. IBM Research Report. vol. RJ2555 (1979)
68. TCG: Trusted Computing Group. http://www.trustedcomputinggroup.org
69. TCG: Attestations Working Group, Trusted Computing Group, March 2020. [Online]. Available: https://members.trustedcomputinggroup.org
70. TCG: TCG Remote Integrity Verification (RIV): Network Equipment Remote Attestation System Version 1.0, Rev. 0.9b, Trusted Computing Group, TCG Draft Specifications, June 2019. [Online]. Available: https://trustedcomputinggroup.org/wp-content/uploads/TCG-NetEq-Attestation-Workflow-Outline_v1r9b_pubrev.pdf
71. TCG: TCG Implicit Identity Based Device Attestation Version 1.0, Rev. 0.93, Trusted Computing Group, TCG Published Specifications, March 2018. [Online]. Available: https://trustedcomputinggroup.org/wp-content/uploads/TCG-DICE-Arch-Implicit-Identity-Based-Device-Attestation-v1-rev93.pdf
72. TCG: TCG Symmetric Identity Based Device Attestation Version 1.0, Rev. 0.95, Trusted Computing Group, TCG Published Specifications, January 2020. [Online]. Available: https://trustedcomputinggroup.org/wp-content/uploads/TCG_DICE_SymIDAttest_v1_r0p95_pub-1.pdf
73. Trusted Computing Group: TPM Main – Part 1 Design Principles – Specification Version 1.2, Trusted Computing Group, TCG Published Specification, October 2003, available at http://www.trustedcomputinggroup.org/resources/tpm_main_specification
74. Voit, E.: Trusted Path Routing using Remote Attestation, IETF, Internet-Draft draft-voit-rats-trusted-path-routing-01. [Online]. Available: https://datatracker.ietf.org/doc/draft-voit-rats-trusted-path-routing/ (2020)
75. Weber, A.: Lagarde Says Her 'Hunch' is That ECB Will Adopt Digital Currency, Bloomberg. [Online]. Available: https://www.bloomberg.com/news/articles/2020-11-12/lagarde-says-her-hunch-is-that-ecb-will-adopt-digital-currency (2020)
76. Wikipedia, Stack Machine (2020). Available at https://en.wikipedia.org/wiki/Stack_machine, Accessed 7 May 2020
77. Yaga, D., Mell, P., Roby, N., Scarfone, K.: Blockchain technology overview. National Institute of Standards and Technology Internal Report 8202. https://doi.org/10.6028/NIST.IR.8202 (2018)
78. Zic, J., Hardjono, T.: Towards a cloud-based integrity measurement service. Journal of Cloud Computing: Advances, Systems and Applications (2013)