Distributed IoT Attestation via Blockchain (Extended Version[1])

Dartmouth Computer Science Technical Report TR2020-877

Ira Ray Jenkins, Sean W. Smith
Dartmouth College
Hanover, New Hampshire, USA
{jenkins, sws}@cs.dartmouth.edu

***Abstract*—The growing number and nature of Internet of Things (IoT) devices makes these resource-constrained appliances particularly vulnerable and increasingly impactful in their exploitation. Current estimates for the number of connected "things" commonly reach the tens of billions. The low-cost and limited computational strength of these devices can preclude security features. Additionally, economic forces and a lack of industry expertise in security often contribute to a rush to market with minimal consideration for security implications.**

It is essential that users of these emerging technologies, from consumers to IT professionals, be able to establish and retain trust in the multitude of diverse and pervasive compute devices that are ever more responsible for our critical infrastructure and personal information. Remote attestation is a well-known technique for building such trust between devices. In standard implementations, a potentially untrustworthy *prover* attests, using public key infrastructure, to a *verifier* about its configuration or properties of its current state.

Attestation is often performed on an ad hoc basis with little concern for historicity. However, controls and sensors manufactured for the Industrial IoT (IIoT) may be expected to operate for decades. Even in the consumer market, so-called "smart" things can be expected to outlive their manufacturers. This longevity combined with limited software or firmware patching creates an ideal environment for long-lived zero-day vulnerabilities. Knowing both if a device is vulnerable and if so when it became vulnerable is a management nightmare as IoT deployments scale. For network connected machines, with access to sensitive information and real-world physical controls, maintaining some sense of a device's lifecycle would be insightful.

In this paper, we propose a novel attestation architecture, DAN: a distributed attestation network, utilizing blockchain to store and share device information. We present the design of this new attestation architecture, and describe a virtualized simulation, as well as a prototype system chosen to emulate an IoT deployment with a network of Raspberry Pi, Infineon TPMs, and a Hyperledger Fabric blockchain. We discuss the implications and potential challenges of such a network for various applications such as identity management, intrusion detection, forensic audits, and regulatory certification.

***Index Terms*—IoT, Remote Attestation, Blockchain, Security, Distributed Systems**

This material is based upon work supported by the U.S. Department of Energy under Award Number DE-OE0000780. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of United States Government or any agency thereof.

[1]This is an extended version of the paper accepted to SIoTEC 2020. [37]

I. Introduction

The growth of computing technology has borne out Moore's Law for half a century. Computers and their peripherals have become smaller, faster, and cheaper to manufacture. Embedded systems are now ubiquitous and often invisible to us. Their use cases span across our daily lives including our critical infrastructure, industrial controls, automobiles, medical devices, and consumer applications. By their very nature, these resource-constrained devices present unique security challenges. For nearly two decades, security experts have been warning of the insecurity of many of these devices, e.g., [40], [69].

The Internet of Things (IoT) is the latest incarnation of the Internet's evolution. Connecting machines together via the Internet is not new. However, the relatively recent convergence of small and powerful embedded systems, cheap networking hardware, and pervasive network accessibility has led to a new world of connected "things." With machine learning and cloud computing, many of these things are becoming "smart." By various estimates there will 20-50 billion devices connected to the Internet this year [22], [23], [27], and a trillion devices by 2035 [59].

IoT devices manifest many of the same resource constraints as prior generations of disconnected embedded systems, such as low power consumption and stringent availability requirements (e.g., sensors, controls, and monitoring devices), limited hardware support and capabilities (e.g., reduced compute power and lack of hardware security), and prolonged life cycles with minimal oversight, management, or patching. The economics of design and manufacturing often dictate these limitations. Unfortunately, the IoT has also inherited a plethora of security vulnerabilities from hardware, firmware, and software, as well as those generally associated with networked and Internet-enabled devices. A rush to market and a lack of security expertise and awareness by device vendors means that many devices are built and deployed with known vulnerabilities and minimal, if any, infrastructure for ongoing maintenance and support [31]. The always-on, inconspicuous, and noninteractive nature of the IoT, combined with general systems security failings, has meant that IoT-targetted attacks are now a legitimate concern [9], [25], [42], [53].

For end-users and IT professionals, building and maintaining trust in these small, cheap, and naturally vulnerable IoT systems is essential as they become more pervasive and imperceptible in our lives, with access to and control over our critical infrastructure and increasingly personal and private data. Standards and best practices for IoT security are being proposed [13], [24], [35], [45], secure hardware does exist [3], [49], along with secure operating systems [38], [39] and network protocols [29]. However, vendor buy-in and consumer awareness are still lagging behind.

Trusted Computing has been a standard in more traditional computing environments for years. A key tenant in trusted computing is the expectation of consistent behavior from compute devices. *Attestation* is one mechanism for satisfying such expectations remotely by providing verifiable evidence tied to a device's hardware, firmware, or software. The IoT's often purpose-built, simple sensors and controls would seem a perfect application for such concepts.

One of the key economic factors in IoT adoption is the promise of a multitude of diverse, connected things interacting autonomously to provide services, convenience, and efficiency at scale. Here we take particular interest in the potential for heterogeneous networks of devices that may interact beyond the traditional borders of organizations or entities. For example, an *energy delivery system (EDS)* of grid operators, consumers, and Federal regulators creates, in effect, a network of *coopetition*, or cooperative competition. Each party is required to trust the devices (sensors, controls, and smart meters) operating within the network; however, security interests and incentives may be lacking or wholly unaligned.

In this paper, we consider the problems of trust and device management for such resource-constrained IoT systems. We propose a *Distributed Attestation Network (DAN)* that relies on blockchain technologies to store and distribute device information. We present a virtualized simulation of a DAN, and a prototype system of IoT-analogues using a network of Raspberry Pi, Infineon TPMs, and a Hyperledger Fabric blockchain. We discuss the implications and potential challenges of such a network, as well as various applications such as identity management, intrusion detection, forensic audits, and regulatory certification.

The remainder of this paper is organized as follows: Section II reviews attestation and blockchain technologies. Section III introduces our distributed attestation network. Section IV discusses our implementation. Section V presents quantitative and qualitative analysis. Section VI discusses general network feasibility. Section VII considers works related. We present our concluding thoughts in Section VIII.

II. Background

In this section, we provide relevant background on trusted computing, specifically attestation, and blockchain technologies.

A. Trusted Computing

The idea of trust in computer systems is not new [68]. However, the term *Trusted Computing* has been promoted by the Trusted Computing Group (TCG), formerly the Trusted Computing Platform Alliance (TCPA), a consortium of hardware and software vendors and computer manufacturers interested in *secure* and *trustworthy* computing since the late 1990s. Unlike its colloquial usage, trust according to the TCG relies on the expectation of behavior. That is, a computer system should operate reliably and consistently as intended (i.e., according to some specification). Various debates have long raged over "whose" intentions are to be expected and enforced. Regardless, trusted computing has found some success in a wide range of applications.

A key to the TCG's invocation of trust is the ability to prove trustworthiness, or otherwise provide evidence of a system's state or properties. As such, the TCG created specifications for a *Trusted Platform Module (TPM)*, a secure cryptoprocessor with protected storage and cryptographic computational abilities. Using this special hardware, the TCG also promoted the concept of *remote attestation*. Attestation is the process by which the aforementioned evidence is requested and supplied. While certain definitions are often tied to specific use-cases, we prefer this more general and flexible definition of attestation.

In general, a potentially untrusted *prover* or *target* device attests, or provides evidence through certain protocols, to a remote *verifier* or *appraiser*. The evidence may consist of hashes, checksums, or computational results linked to hardware, firmware, or software configurations or properties of the prover's current state. For example, a common application has been software integrity checks initiated by a challenge-response protocol and secured using public key infrastructure (PKI). Immediately, the question arises as to why one would trust the provided evidence? A standard solution, as proposed by the TCG, is for the verifier to assume the existence of a trusted component on the prover. This *root of trust* or *trust anchor* is typically a TPM with various trusted cryptographic keys and functionality.

The TPM provides a *hardware* root of trust for attestation. While typical TPM implementations are small, and relatively cheap they still represent undesirable overhead for many IoT vendors, with added cost, space, and power considerations. Recently, the TCG announced an IoT-centric TPM specification that aims to reduce these factors, and ultimately provide a "tiny TPM... (that) can be integrated directly within the host chips [66]". Other proposals seek a more light-weight solution, with no TPM and minimal hardware capabilities, such as SMART [21], SANCUS [49], Intel's TrustLite [41], and self-protecting modules (SPMs) [62]. All of this attention on secure, isolated computation has led to the inclusion of Trusted Execution Environments (TEEs) in modern processor designs, like Intel's SGX [7], [19], [32], [34], [46] and ARM's TrustZone [4], [10], [11], [48]. For devices that lack the necessary hardware support, especially legacy devices, there have been several proposals for *software-based attestation* [43], [52], [55]–[57], [60].

Traditionally attestation has been *static*, with a focus on measured binaries and disk images [54]. One disadvantage to

these techniques is the assumption that a program on disk or in memory is executing as intended. However, thanks to ever-present *buffer overflows* and more recent *return-oriented programming* attacks, these assumptions cannot hold. More recent work has focused on *dynamic* attestation which seeks to verify runtime integrity [2], [20], [71].

B. Blockchain

Commonly, attestation is performed in an ad hoc manner. In response to a challenge, the prover generates and sends the requested information. Even when attestations are made periodically, trust decisions, on the part of the verifier, are usually made based on the most contemporaneous evidence. However, knowing a device is currently trusted, by all best evidence, tells one little about its prior states. Given the desired longevity of IoT devices, historicity is important.

When considering a data storage solution for a heterogeneous, multi-organization IoT attestation network, it should be distributed and decentralized. Given the potential for mutual-distrust, each member should be party to storing the attested data. A further requirement is that transactions amongst the network must be immutable, transparent, and secure. Haber and Stornetta apprear to be the first to describe a tamper-resistant process for cryptographically linking distinct data, in their case timestamps [30], in such a way as to make future forgeries and retroactive changes infeasible and evident. In the seminal Bitcoin work, the pseudonymous Nakamoto introduced a timestamp server that linked items (transactions) into a cryptographically secure *chain of blocks* [47]. Driven by the explosion of popularity of Bitcoin, research into the underlying technology, this *blockchain*, soon recognized applications beyond cryptocurrencies [67].

Blockchain is a cryptographically secure, decentralized *distributed ledger*. Distributed amongst a network of computers, this shared ledger contains synchronized data that is cryptographically signed and hash-chained to create a secure, verifiable audit of all transactions. Blockchains are decentralized in peer-to-peer (P2P) networks, with each peer maintaining the replicated ledger. Blockchains may be public or private (*permissioned*), with varying degrees of access controls within the ledger. Blocks contain the synchronized data and may be created concurrently by peers, creating the potential for fragmentation and divergent histories within the chain. Therefore, block inclusion must be achieved through a consensus protocol amongst the participating peers, e.g., Bitcoin's reliance on proof-of-work.

We chose a Hyperledger Fabric, version 1.4, blockchain implementation for our proof-of-concept DAN. *Hyperledger* is an open source, collaborative project to facilitate and encourage cross-industry blockchain technologies [65]. *Fabric* is a modular and extensible open source framework for building and deploying *permissioned* blockchains [8], [14]. Fabric allows flexible, plug-and-play services for key blockchain functionality, such as membership, cryptography, and consensus. Additionally, Fabric allows distributed applications written in general purpose programming languages, such as Go, Java, and Node.js.

Fabric relies on an *execute-order-validate* architecture, the first of its kind. *Smart contracts* supply application logic during the execution phase and are executed on peer nodes within a container environment for isolation. An *ordering service* receives endorsed (consensus) transaction outputs and totally orders them before being broadcasting to peers for validation. In Fabric, an ordering service may be associated with multiple blockchains in *channels*. Peers maintain a local copy of the ledger as an append-only blockchain.

One of the primary features of DAN is the integration between attestation and the blockchain. This integration is achieved through smart contracts, or *chaincode* in Hyperledger. According to Szabo [63], smart contracts "... combine protocols with user interfaces to formalize and secure relationships." In the context of Hyperledger, chaincode defines the interface to and access controls for the underlying ledger. In this way, read (query) and write (update) access to channels is programmatically controlled, with execution isolated in a secured and separate container from a peer's endorsement (consensus) logic. For example, an attestation application would need to define a priori the structure and access controls associated with submitting and viewing attestations.

C. IoT Challenges

Attesting all IoT devices is infeasible. The scale alone, of billions of individual devices, makes the idea impractical. However, one might consider instead just attesting the devices within a single deployment. Typical IoT deployments might be as small as dozens of devices in a "smart home" to tens of thousands of sensors in industrial settings. Managing these networks will still be hard given the heterogenous nature of IoT devices and the fragmentation within the market.

Many IoT devices contain batteries for their operation or as a backup to external power. Given the relative efficiency of some of these low-power devices, it is feasible that restarts and power interruptions may be rare. Consequently, static methods such as boot or load-time attestation are clearly not enough.

In addition, many IoT devices are envisioned to be long-running, in a "set it and forget it" manner. This means firmware and software may operate well beyond their intended lifespan. Further, the evidence clearly shows that long-lived, undisclosed, zero-day vulnerabilities are not a myth [16].

Understanding *which* devices are operating within a network, *what* software is running on those devices, and *how* that software is behaving is critical to IoT security and trust. In fact, there is growing regulatory constraints on these networks and devices given their often privileged access to data and control of our physical world [1], [61].

III. Distributed Attestation via Blockchain

In this section, we define our system and threat model assumptions, detail the architecture of our Distributed Attestation Network (DAN), and discuss potential use cases.

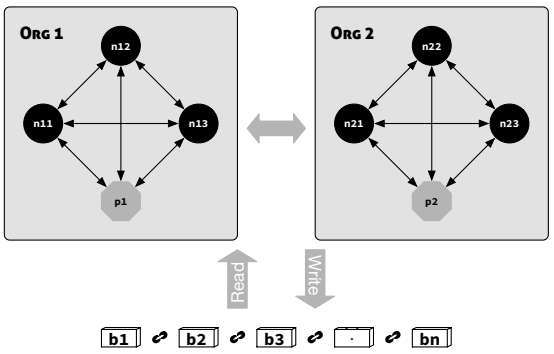

Fig. 1. DAN organizational chart featuring two organizations, each with multiple networked nodes, and a shared blockchain.

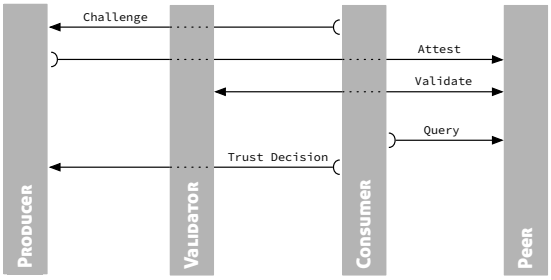

Fig. 2. Example challenge-response protocol within a DAN. The consumer initiates attestation. The producer submits attested information to the blockchain. A validator submits validation to the blockchain. The consumer awaits validation, and finally makes a trust decision.

A. System Model

We consider a heterogenous network of IoT-like devices that may be resource-constrained in power consumption, computational ability, communication frequency and reliability, and storage capacity. We assume a dense network such that each network node has multiple immediate neighbors, and all nodes are free to logically communicate (although this may require intermediate physical hops, such as in a mesh network). Additionally, we allow that nodes within this network may be under the authority and control of diverse, cooperative, but mutually competitive or distrusting organizations or entities. Finally, we assume that the devices on the network that will participate in attestation must have some root of trust or trust anchor.

Requirements. As described by Coker et al. [17], an ideal attestation architecture should satisfy five (5) principles: information should be (1) *fresh* and (2) *comprehensive* with (3) *constrained disclosure* and (4) *semantic explicitness*, all backed by (5) *trustworthy mechanisms*.

B. Threat Model

The adversary or attacker may be passive or active. We consider primarily an adversary that may introduce malware into a (small) fraction of devices over a given time period. Additionally, the adversary may introduce new devices into the network environment, and supply malicious inputs to public interfaces. As is standard in single-prover attestation schemes, we ignore both denial of service (DoS) and physical attacks in this work.

C. System Design

Our distributed attestation network is a generic and flexible attestation architecture. DAN does not specify the attestation protocols, or define the data and properties of interest, or limit the actors and relationships involved. ***We believe DAN is the first such architecture to utilize blockchain as more than simple distributed storage, but as a prime actor in an attestation protocol***.

An example organization chart is shown in Figure 1. As shown, there are two organizations involved in this sample network. Each organization maintains a number of nodes, $\{n_{11}, n_{12}, n_{13}\}$ for Org1 and $\{n_{21}, n_{22}, n_{23}\}$ for Org2. Additionally, each organization maintains a number of *peer* nodes, $\{p_1\}$ in Org1 and $\{p_2\}$ in Org2. Peer nodes will be discussed in more detail below. As mentioned prior, we assume that nodes within the network are free to communicate amongst themselves, even across organizations. For example, node n_{11} might communicate with nodes n_{13} and n_{22}.

Peer nodes here represent those nodes with access to the blockchain ledger. A network may have one or many peer nodes with which a single node can communicate. In the case of traditional computers, every node in a network might be a peer; however, given the resource constraints of individual IoT devices, it may be necessary to offload the blockchain management to an intermediate and more powerful node, for example a gateway or base station. This of course introduces additional security concerns which we discuss later in Section VI.

A goal of DAN is for the individual nodes within the network to be attested. This is accomplished by nodes called *producers*, submitting attestations into the blockchain. This may be the result of a challenge-response attestation protocol involving another node in the network. This approach may leave the door open for an adversary to submit many challenges and monopolize the resources of a device via computing attestations. However, we do not consider DoS attacks in this work. Alternatively, attestations may simply be submitted to the blockchain on a periodic and autonomous schedule.

Once attestations are present within the blockchain, other nodes within the network, so called *validators*, are able to evaluate the attested information. The results of such validation are then included in the blockchain, linking a producer and validator with a timestamped and immutable record of attestation. The final node type within a DAN are the *consumers*. Consumer nodes want to interact with producer nodes; however, there may be (rightfully so) distrust in this transaction. Consumers rely on the validated attestations within the blockchain to formulate trust decisions about producers. Thus decoupling the trust relationship between producer and consumer. Figure 2 depicts this simple scenario.

Individual nodes within the network may fulfill a variety of roles. For example, producer nodes may in fact be validating nodes for specific neighbors. And indeed, a producer node

may also be a consumer of other producing nodes. This sort of network aligns nicely with the envisioned IoT, where devices may perform multiple tasks which in turn depend upon the services of other network devices. Consider, for example, the previously mentioned EDS network comprising grid operators, consumers, and Federal regulators. Grid operators may manage a variety of sensors, controls, and smart meters within their network. An individual consumer may have a variety of "smart" things attached to the network, such as meters, backup generators, and alternative energy production mechanisms. Regulatory officials, while not operating network nodes themselves, have direct interests in validating and monitoring compliance of such critical infrastructure. In this scenario, all parties rely on the proper functioning of the network nodes; however, consumers are incentivized to use more energy than they pay for, grid operators are incentivized to "check-off" compliance boxes with minimal expenditure on overhead and liability, and regulators are responsible to society-at-large for enforcing the national security and reliability of the grid.

With a DAN, the devices within the network can be attested in a semi-public fashion, i.e., to those directly involved in the network. Grid operators may be context-dependent producers, consumers, and validators. Service consumers may act as both producers and consumers. Instead of relying on grid operators to self-report, or the overhead of manpower to manually verifying compliant operation of these devices, regulators can now query the devices themselves. In this case, regulators may act as validators, having direct knowledge of the hardware, firmware, or software deployed, or at least the relevant security properties of interest. Some consumers, additionally, may derive trust from such government oversight; however, building a more diverse trust portfolio for this example might involve the inclusion of original equipment manufacturers (OEMs), distributors, or contractors. In this way, trust relationships can be chosen and distributed, with ground truth device-dependent and decentralized.

D. Use Cases

The DAN architecture facilitates a variety of current use cases, and opens the door to many other potential scenarios.

Authentication and Identity Management. In heterogeneous IoT networks, device identity is crucial for security. Common solutions might involve MAC or IP address whitelisting, behavioural and environmental analysis, or challenge-response protocols using PKI. More robust methods would rely on a root of trust, whether hardware-based like TPMs and *physically unclonable functions (PUFs)* or hybrid architecture-based methods like TEEs. Many proposals already exist for linking identity and blockchain, e.g., [36], [50]. Most of these efforts focus on *permissionless* blockchains; however, our concern has been with those inter-organizational applications that already rely on some business relationship. Part of Hyperledger's Fabric is its customizable membership services that allows the permissioned blockchain and various access controls for both data and chaincode. By default, Fabric itself already contains the identity information in the form of traditional CAs and PKI, but could easily be extended with roots of trust.

Intrusion Detection. Various methods of intrusion detection exist, from simple identity authentication to behavioural analysis and anomaly detection mechanisms. With this data stored within the blockchain, and accessible to network participants, dynamic trust decisions can be made autonomously based on heuristic algorithms and the preponderance of evidence. Consider a network in which devices attest to their static and dynamic execution, and anomalous behaviour is recorded by neighboring devices. Allow an attacker to subvert some small subset of devices that begin participating in a distributed denial of service attack (DDOS) against other nodes in the network or even remotely. Observation and attestation of this behaviour by the rest of the network might quickly result in the isolation of the compromised devices.

Post-event Forensic Audits. It is often challenging for an average computer user, when presented with a vulnerability disclosure, to effectively understand their risk on personal devices. For example, even with the authors' experience with computer technology and security, understanding the personal risk associated with the Heartbleed vulnerability was non-trivial. At the scale of IoT deployments, even the best IT professionals can be expected to struggle given the diversity of devices, firmware, and software. While typical attestation systems are ad hoc and transactional, including attestation within a blockchain can potentially create a historical record, within the ledger, of a device's entire lifecycle. This allows for not only managing software deployments (e.g., knowing which version of OpenSSH is on what devices), but would also allow forensic audits of a device's history. Consider an attestation system that utilizes DAN, that records identity, deployed software and firmware, dynamic attestation, and external behavioural analysis and anomaly detection. After a vulnerability disclosure, it becomes not only easy to identify potentially vulnerable devices, but also *when* they became vulnerable, how long they have been vulnerable, and what—if any—questionable actions they may have taken during that time. In addition, a DAN's flexibility with smart contracts means that much of this might be automated.

Regulatory Certification. The cargo cult mindset of computer security is not good. However, regulatory compliance is a fact of life. The National Institute of Standards and Technology are formulating what could be the beginning of just such regulations [24]. In "Considerations for Managing IoT Cybersecurity and Privacy Risks", NIST considers three goals: (1) device security, (2) data security, and (3) individual privacy [13]. Asset, vulnerability, and access management are listed as areas of risk mitigation. As seen above, DAN facilitates each of these areas, in addition to the areas of incident detection and information flow management. Further, consider the scenario in which regulators are participants within a DAN, and able to verify desired principles and functionality based on the verified self-reporting of the devices themselves. For example, consider a piece of critical infrastructure equipment that contains the necessary roots of trust and can attest to a DAN.

It is plausible that regulators, security auditors, or liability insurers might have certain criteria for security, e.g., hardware with specific features enabled or "up-to-date" firmware and software versions. These entities may be tasked with certifying compliance; however, they may not have the expertise or best knowledge to adequately make these determinations. The hardware manufacturer knows best what capabilites a device has. Firmware and software vendors (should) know best the specifications of their products. Third-party security-related groups might be more current on state-of-the-art vulnerabilties. By allowing these external parties to validate the attestations of a device, those in charge of compliance can certify that those with best knowledge approve or disprove of a device.

IV. Implementation

In this section, we discuss implementation details and describe our work virtualizing, simulating, and creating a proof-of-concept DAN.

We began with a *virtualized simulation* of a DAN using only Docker containers. This was primarily to familiarize ourselves with the Fabric framework, and to quickly experiment with storing and retrieving attestation data. In this simulation, we utilized one node for each producer, consumer, and validator. Additionally, we relied on a single peer node for blockchain interactions including membership services, transaction ordering and endorsement. In our initial implementation, we followed the example challenge-response protocol shown in Figure 2. Attestation and validation were triggered manually. In this case, we did not rely on a trusted computing base (TCB), but assumed the attestations reported were valid. For attestation data, we simulated TPM quotes by generating random measurement lists, and calculating hashes for "known good" states. For these trials, we ignored PKI and signatures on the attested data.

To extend this we built a practical testbed, implementing a DAN using eight Raspberry Pi, models 2B and 3B+, as our IoT nodes. The Pi has become synonymous with IoT research and development. They are low-cost ($35), credit-card sized, relatively powerful, and highly expandable. The model 2B has a Broadcom BCM2837 system on a chip (SoC) with a 900 MHz Quad Core ARMv8-A Cortex-A53 cpu. The 3B+ sports a slightly beefier Broadcom BCM2837B0 SoC with a 1.4 GHz Quad Core ARMv8-A Cortex-A53 cpu, with the addition of on-board WiFi and PoE. Both systems have 5V and 3.3V supplies, and support 40 general-purpose input/output (GPIO) pins with UART, I2C, SPI, and I2S functionality. This flexibility has led to a large number of add-on boards and peripherals. To provide our TCB, we used Infineon Optiga SLB 9670 TPM2.0 iridium boards that are TCG compliant and fully CC(EAL4+) and FIPS certified. These boards communicate over SPI using Infineon's Embedded Linux TPM Toolbox 2 (ELTT2).

We setup two "organizations" each with 3 compute nodes and a single *gateway* node. Recall Figure 1. Each organization was assigned a blockchain peer that ran inside a Docker container on a consumer laptop or desktop. The same protocol transactions were tested for consistency. Our simulated attestations were produced from random measurement lists that were hashed into the TPM PCRs and read out again. Again, for these trials we ignored PKI and signatures on the attested data.

Producer nodes were scripted to "ping" consumers at regular intervals. Attestations of the producer nodes were submitted to the blockchain on a periodic basis. Additionally, random groups of validators were chosen to verify attestations. We tested several scenarios in which consumers required single and multiple validators to sign off on attestation. Trust decisions by consumers were made on varying aggregate validations, e.g., certain consumers only trust specific validators, or require multiple validations, or require more *fresh* validation. On a failed validation, consumers would instantiate an iptables firewall rule to `DROP` traffic from the abusing producer. Manual triggers were used introduce potential errors such as to interrupt or delay attestations from a producer, force submit invalid attestations into the blockchain, or interrupt validators from reading attestations or writing their results.

V. Evaluation

Primary concerns for IoT attestation are the cryptographic and blockchain operations on resource-constrained devices. We consider the feasibility of the proof-of-concept attestation scheme by analyzing the potential performance impacts of computation, communication, storage, and energy overheads. In our example DAN, the TPM must generate hashes over static measurement lists. To isolate this computation, we performed a comparison between SHA1 and SHA256 hashing on various randomized payloads ranging from 1 to 100 bytes. Each test was repeated 1000 times.

Computation Overhead: Naturally, the Infineon TPM requires less CPU time, and is roughly twice as slow (in system time) as the Raspberry Pi on any payload size. These results are shown for both SHA1 and SHA256 in Figures 3 and 4. On average, the wall-clock time to complete these tests required 54 seconds on the TPM, and only 17 seconds on the Pi. These results were expected given the TPM's 43Mhz transfer rate on the SPI interface, and the relative power of the Raspberry Pi.

Communication Overhead: Communication within the DAN is ad hoc. A proving device need only attest periodically or when requested. Additionally, verifiers and consumers primarily communicate with blockchain peers. In our conception, these peers are somewhat more robust than the IoT devices on the edge of the network. Thus, communication is primarily limited by the throughput of the blockchain peers, as well as the bandwidth and latency of the network. For our proof-of-concept, these factors were neglibile; however, more testing should be done to evaluate the scaling limits of a representative network.

Storage Overhead: Because our proof-of-concept is simple and relies on Raspberry Pi, storage is not a real concern. Each entity within the network needs only the credentials to interact with the blockchain peers. More advanced attestation schemes may require more storage; however, proving devices need only

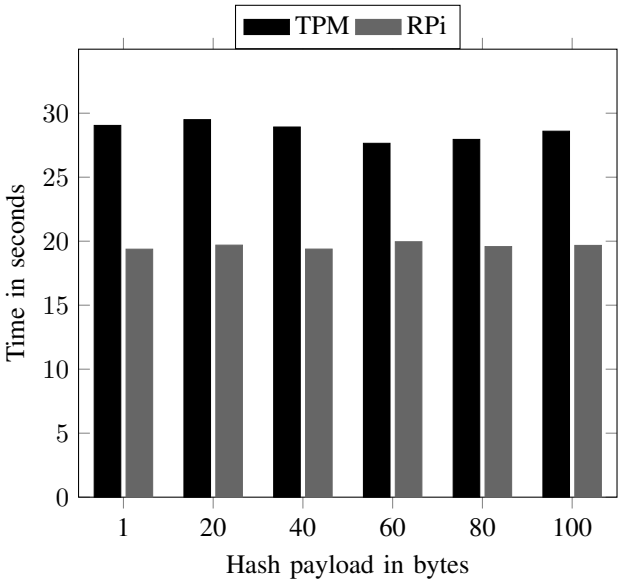

Fig. 3. Timing of SHA1 hashing on Infineon TPM and Raspberry Pi 3B+.

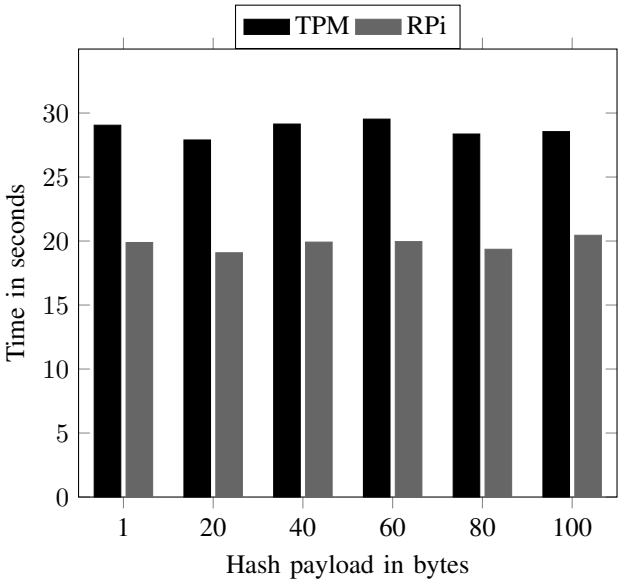

Fig. 4. Timing of SHA256 hashing on Infineon TPM and Raspberry Pi 3B+.

enough storage to generate attestation since all the protocol data elements are stored within the blockchain on peer nodes. Additionally, it is worth mentioning that by adding a physical TPM, a device gains access to some additional non-volatile RAM (NVRAM) (in the case of our Infineon TPM, 6962 bytes).

Energy Overhead: During the hashing experiments mentioned above, we also monitored the power usage with the combination of a consumer-grade wall-outlet electricity usage monitor and multimeter. The average power consumption for SHA1 and SHA256 experiments are shown in Figures 5 and 6. While these measurements may lack resolution, their relative comparison is instructive. On average, the Raspberry Pi requires 10% more power than the TPM during hashing. Additionally, at peak load, the Pi draws 5% more power based on watt calculation. These results suggest that, even though the TPM is powered by the Pi, when performing these cryptographic operations the TPM is more efficient. Further experiments are required to eliminate potential effects of power supply inefficiency.

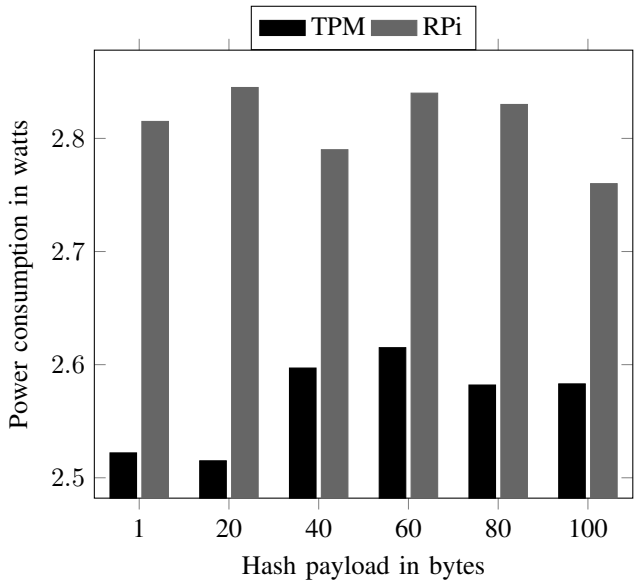

Fig. 5. Average power consumption during SHA1 hashing on Infineon TPM and Raspberry Pi 3B+.

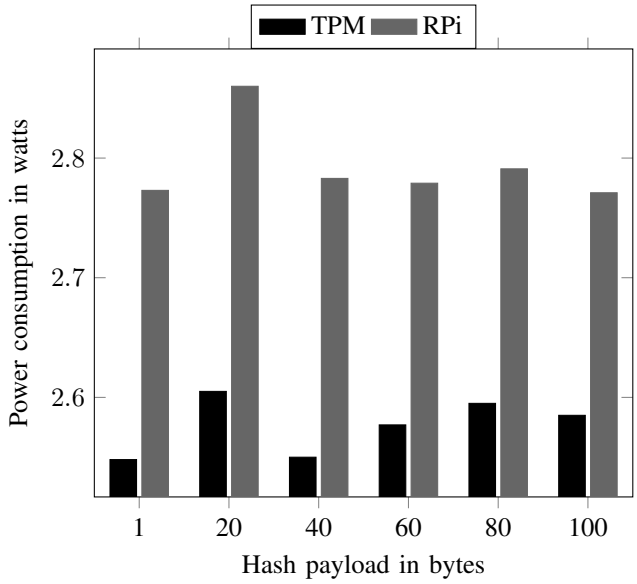

Fig. 6. Average power consumption during SHA256 hashing on Infineon TPM and Raspberry Pi 3B+.

A. *Qualitative Analysis*

In building our proofs of concept, we learned several things. We address some of these questions below.

What are the necessary hardware requirements and are they feasible given IoT resource constraints? In general, attestation requires some root of trust. Hardware roots of trust typically provide more robust security guarantees; although, they do come with a price. However, given the TCG's latest proposals, and the growing research interests in Trusted Execution Environments, these hardware roots of trust will only become more practical.

A major IoT constraint is often power, with many devices relying on batteries. Adding a hardware root of trust, such as the Infineon TPMs we used in our proof of concept represents a design decision. While TPMs are relatively efficient, with hardware implementations of things like SHA-1 and SHA-256, they do represent an additional draw on a small device. Using TEEs as a root of trust may be more efficient. In the future, we'd like to compare the power draw during common attestation and cryptographic operations between our hardware TPMs and SGX or TrustZone. However, any attestation represents additional overhead. For some applications, runtime attestation may be unnecessary. Attestation may only be needed occasionally, with long (e.g., days) periods between successive attempts. Hardware roots of trust will only become more efficient and smaller over time. Alternatively, some applications may only need the security guarantees of a software-based attestation method.

Is this sort of network efficient given the general concerns about proof-of-work blockchains? The Hyperledger Fabric that we have utilized is modular and allows pluggable consensus mechanisms. By default it runs Byzantine Fault Tolerance (BFT) which is well-known, highly scalable, and efficient for the permissioned environment that we consider.

Can the network nodes, as we've described them here, support blockchain operations efficiently? In the proof of concept that we built, blockchain operations are performed by peer nodes within docker containers on consumer laptops and desktops. In a real-world deployment, these may remain containerized and be virtualized within the cloud. Alternatively, there are current efforts to build so-called "light clients" that may allow resource-constrained IoT devices to interact individually with the blockchain as peers [51]. However, storage will remain a problem. It may be feasible to move blockchain storage to an intermediate device like a gateway or base station, further into a data center, or even the cloud.

Does the blockchain scale with the theorized IoT deployment numbers? It clearly depends on what information is being included into the blockchain, by how many devices, and how often. Naturally, we have concerns given the long running estimation of IoT devices, and the desire to maintain a complete lifecycle history. One potential mitigation for these concerns is checkpointing [33]. Additionally, at least in the case of the current generation of IoT devices, firmware and software updates are few and far-between. When considering dynamic attestation, there are efforts to increase efficiency by only considering security critical sections of code. All of these factors may help to reduce the overhead of blockchain storage and computation.

What are the security concerns of offloading to base station/gateway? Naturally, the communication between the more anemic network nodes and the intermediate device needs to be secure. Adding such a node within the network makes for a great man-in-the-middle (MITM) target, and potentially introduces a single point of failure that must be considered. Additionally, attestation now relies on a hierarchy as the base station must be attested trustworthy along with any downstream nodes.

VI. Discussion

DAN is the first distributed attestation architecture that utilizes blockchain as a principal component to facilitate remote attestation. It provides flexibility in establishing a given attestation protocol. It mirrors the envisioned topology of IoT systems. And it naturally satisfies the five constraints of an attestation system provided by Coker et al. [17]:

Measurement of diverse aspects of the target attestation. DAN is not limited to a particular attestation method. Both static, load-time measurements and dynamic runtime measurements may be recorded within the blockchain.

Separation of domains for measurement. In the prototype we built, we rely on hardware TPMs to facilitate our root of trust. As discussed earlier, hardware mechanisms for trust are evolving to satisfy the constraints of the IoT. Trust anchors are transparent to DAN implementations.

Self-protecting trust base. The trusted base for an individual device's attestation may vary depending on services being attested, or the requirements of the validator that is used.

Attestation delegation. The principles of DAN naturally allow multiple verifiers or validators of attestation. In fact, it is easily conceivable that different attestations may be verified by multiple different validators. This creates the opportunity for trust decisions based on a more comprehensive assessment of a target device.

Attestation Manager. The blockchain can naturally realize this concept by providing a distributed ledger containing all of the measurement and attestation tools currently supported by various devices. The DAN, as we have implemented it with Hyperledger Fabric, can also enable the constrained disclosure requirements by utilizing customized membership services and standard PKI.

VII. Related Work

In this section, we provide an overview of related prior work. Attestation is a mature, and robust field of study. Our design of a distributed attestation network utilizing blockchain does not specify an attestation method or protocol. As such, we focus on distributed attestation and IoT systems.

BIND [58] introduced granularity to attestation. The authors describe a process of annotating programs to guide the attestation of critical code sections, and then link that code to the resulting data. DAN supports the idea of granularity by being agnostic to what data is being attested.

Yang, Wang, Zhu, and Cao [70] consider the problem of identifying compromised nodes within unattended sensor

networks. They propose two distributed software-based algorithms for detecting such compromise even when multiple compromised nodes collude during attestation. A key assumption is made that every node within the network must participate in attestation. The design and implementation of DAN does not require this constraint. And indeed, it is a constraint for IoT networks that may be heterogeneous, and interdependent with respect to services. DAN allows attestation between a single prover or many.

Chen and Wang [15] consider homogeneous wireless sensor networks, and show optimal metrics for how often to attest and how many neighbors to require. DAN allows both ad hoc and periodic attestations and validations to be recorded into the blockchain.

SEDA [12] and SANA [5] provide collective attestation techniques for swarms. The network topology considered in swarms requires that nodes only be able to communicate with their direct neighbors. Our implementation of DAN does not assume this constraint. In fact, DAN can be used to facilitate all communication via the blockchain with requests for attestation, and the subsequent results and verification all published. Additionally, most swarm attestation techniques result in each device within the swarm being attested, while DAN could facilitate either single or multi-party attestation and verification.

Fremantle et al. [26] proposed the use of blockchain for IoT devices with a reliance on intermediate, more powerful devices, termed *pythia*. The current implementation of DAN similarly relies on intermediate devices to perform blockchain operations; however, promising work is being done to create light-clients that may make it feasible for resource-constrained devices to communicate directly with the blockchain. For DAN, the primary concern would be storage of the blockchain, especially for long lived devices.

Liang et al. [44] use SGX-based remote attestation and Hyperledger to build a secure membership services platform. In contrast, DAN relies on a hardware root of trust; however, software-based attestations are equally supported. DAN is flexible enough to allow various roots of trust on edge devices.

Tan et al. [64] consider a multi-tier attestation protocol that relies on more powerful devices with hardware TPMs, and more anemic devices utilizing software roots of trust. Unlike their work, we do not assume a single trusted device to issue attestation challenges; instead, DAN is flexible, and allows a variety of attesting relationships across multiple organizations.

WISE [6] is a flexible swarm attestation scheme that allows subsets of devices to be attested based on the history or characteristics of each device. An assumption for swarm attestation is still that all devices be attested by each other. While DAN could support this sort of attestation because of the history already contained within the blockchain, it is not necessary for any given transaction amongst various IoT devices that *every* device in the network be attested.

CIoTA [28] is a framework for anomaly detection that relies on blockchain. An anomaly detection model is attested to the blockchain, and used to iteratively build a combined, dynamic model that is distributed to each device. Our implementation of DAN is more generalizable in that we are not focused on anomaly detection, but encourage multiple and varied attestation schemes in addition to our interests in asset and identity management and forensic audits.

RADIS [18] is a protocol for distributed service attestation. The authors rightly consider the cascading effect of compromised dependent services and describe a method of attestation in which attesting service 1 on device 2, may require attesting service 2 on device 3. Our implementation of DAN could facilitate this sort of cascading given the requisite attestation of each device within the blockchain. Care must needs be given to the implemented protocol to prevent time-of-check to time-of-use (TOCTOU) vulnerabilities, race-conditions, and circular dependencies.

VIII. Conclusions

In this paper, we introduced DAN: a distributed attestation network that utilizes blockchain technologies to decentralize, and distribute attestation. DAN is the first attestation architecture in which blockchain integrates directly with the attestation protocol. DAN allows a variety of complex relationships to exist between producers, consumers, and validators. Additionally, DAN is flexible enough to support a multitude of validators using different attestation mechanisms at the same time. By relying on the blockchain, new and interesting applications are possible, for example device lifecycle histories and forensic audits. In the future, we plan to extend current state of the art attestation protocols into DAN, and continue to use our testbed for quantitative evaluation and feasibility testing, such as comparing the power efficiency between hardware and software roots of trust. The security of DAN is currently based upon the assumption of guarantees provided by the underlying hardware, protocols, and cryptographic infrastructure of the blockchain. A more formal model of these primitives and interactions is needed to prove the security and scalability of IoT attestation.

References

[1] 116th Congress, "IoT cybersecurity improvement act of 2019," H.R. 1668, March 2019.

[2] T. Abera, N. Asokan, L. Davi, J. Ekberg, T. Nyman, A. Paverd, A. Sadeghi, and G. Tsudik, "C-FLAT: Control-flow attestation for embedded systems software," in *Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS '16)*. Vienna, Austria: ACM, October 24–28 2016, pp. 743–754.

[3] M. Ahmad, "A tale of two hardware security solutions for IoT designs," TheCircuit [Blog post], Digi-key Electronics, March 2019.

[4] T. Alves and D. Felton, "TrustZone: Integrated hardware and software security," *Technology In-Depth*, vol. 3, no. 4, pp. 18–24, 2004.

[5] M. Ambrosin, M. Conti, A. Ibrahim, G. Neven, A. Sadeghi, and M. Schunter, "SANA: Secure and scalable aggregate network attestation," in *Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS '16)*. Vienna, Austria: ACM, October 24–28 2016, pp. 731–742.

[6] M. Ammar, M. Washha, and B. Crispo, "WISE: Lightweight intelligent swarm attestation scheme for IoT (the verifier's perspective)," in *Proceedings of the 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '18)*. Limassol, Cyprus: IEEE, October 15-17, 2018, pp. 1–8.

[7] I. Anati, S. Gueron, S. Johnson, and V. Scarlata, "Innovative technology for CPU based attestation and sealing," in *Proceedings of the Second International Workshop on Hardware and Architectural Support for Security and Privacy (HASP '13)*. Tel-Aviv, Israel: ACM, June 2013.

[8] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger Fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the 13th EuroSys Conference (EuroSys '18)*. Porto, Portugal: ACM, April 23-26, 2018, pp. 1–15.

[9] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai botnet," in *Proceedings of the 26th USENIX Security Symposium (SEC '17)*. Vancouver, BC, Canada: USENIX Association, August 16–18, 2017, pp. 1093–1110.

[10] ARM Limited, "Building a secure system using TrustZone technology," White Paper, April 2009.

[11] ——, "Arm® TrustZone technology for the Armv8-M architecture," White Paper, October 2018.

[12] N. Asokan, F. Brasser, A. Ibrahim, A.-R. Sadeghi, M. Schunter, G. Tsudik, and C. Wachsmann, "SEDA: Scalable embedded device attestation," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*. Denver, CO, USA: ACM, October 12–16 2015, pp. 964–975.

[13] K. R. Boeckl, M. J. Fagan, W. J. Fisher, N. B. Lefkovitz, K. N. Megas, E. M. Nadeau, B. M. Piccarreta, D. G. O'Rourke, and K. A. Scarfone, "Considerations for managing Internet of Things (IoT) cybersecurity and privacy risks," NIST Interagency/Internal Report (NISTIR) - 8228, National Institute of Standards and Technology, Gaithersburg, MD, USA, June 2019.

[14] C. Cachin, "Architecture of the Hyperledger blockchain Fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers (DCCL '16)*, Chicago, IL, USA, July 2016.

[15] I. Chen and Y. Wang, "Reliability analysis of wireless sensor networks with distributed code attestation," *IEEE Communications Letters*, vol. 16, no. 10, October 2012.

[16] S. Clark, S. Frei, M. Blaze, and J. Smith, "Familiarity breeds contempt: The honeymoon effect and the role of legacy code in zero-day vulnerabilities," in *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC '10)*. Austin, TX, USA: ACM, December 6-10, 2010, pp. 251–260.

[17] G. Coker, J. Guttman, P. Loscocco, A. Herzog, J. Millen, B. O'Hanlon, J. Ramsdell, A. Segall, J. Sheehy, and B. Sniffen, "Principles of remote attestation," *International Journal of Information Security*, vol. 10, no. 2, pp. 63–81, June 2011.

[18] M. Conti, E. Dushku, and L. V. Mancini, "RADIS: Remote attestation of distributed IoT services," in *Proceedings of the Sixth International Conference on Software Defined Systems (SDS '19)*. Rome, Italy: IEEE, June 10-13, 2019, pp. 25–32.

[19] V. Costan and S. Devadas, "Intel SGX explained," Cryptology ePrint Archive, pp. 1–118, January 2016. [Online]. Available: ia.cr/2016/086

[20] L. Davi, A. Sadeghi, and M. Winandy, "Dynamic integrity measurement and attestations: Towards defense against return-oriented programming attacks," in *Proceedings of the Fourth Annual Workshop on Scalable Trusted Computing (STC '09)*. Chicago, IL, USA: ACM, November 2009, pp. 49–54.

[21] K. El Defrawy, A. Francillon, D. Perito, and G. Tsudik, "SMART: Secure and minimal architecture for (establishing a dynamic) root of trust," in *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS '12)*. San Diego, CA, USA: The Internet Society, February 2012, pp. 1–15.

[22] Ericsson, "Ceo to shareholders: 50 billion connections 2020," Press Release, April 2010. [Online]. Available: http://mb.cision.com/Main/15448/2246220/662223.pdf

[23] D. Evans, "The Internet of Things: How the next evolution of the internet is changing everything," White Paper, Cisco Systems, Inc., April 2011. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[24] M. J. Fagan, K. N. Megas, K. A. Scarfone, and M. Smith, "Recommendations for IoT device manufacturers: Foundational and core device cybersecurity capability baseline," NIST Interagency/Internal Report (NISTIR) - 8259 (2nd Draft), National Institute of Standards and Technology, Gaithersburg, MD, USA, January 2020.

[25] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *Proceedings of the 37th IEEE Symposium on Security and Privacy (S & P '16)*. San Jose, CA, USA: IEEE, May 23–25, 2016, pp. 636–654.

[26] P. Fremantle, B. Aziz, and T. Kirkham, "Enhancing IoT security and privacy with distributed ledgers - a position paper," in *Proceedings of the Second International Conference on Internet of Things, Big Data and Security (IoTBDS '17)*. Porto, Portugal: SciTePress, April 24-26, 2017, pp. 334–349.

[27] Gartner, "Gartner says 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016," Press Release, February 2017. [Online]. Available: https://www.gartner.com/newsroom/id/3598917

[28] T. Golomb, Y. Mirsky, and Y. Elovici, "CIoTA: Collaborative IoT anomaly detection via blockchain," in *Proceedings of the First Workshop on Decentralized IoT Security and Standards (DISS '18)*. San Diego, CA, USA: Internet Society, February 18 2018.

[29] J. Granjal, E. Monteiro, and J. Sá Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, August 2015.

[30] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," in *Proceedings of the 10th Annual International Cryptology Conference (CRYPTO '90)*, ser. Lecture Notes in Computer Science, vol. Advances in Cryptology, no. 537. Santa Barbara, CA, USA: Springer-Verlag, August 11-15, 1990, pp. 437–455.

[31] Hiscox, "2018 Hiscox cyber readiness report," White Paper, February 2018. [Online]. Available: https://www.hiscox.com/sites/default/files/content/2018-Hiscox-Cyber-Readiness-Report.pdf

[32] M. Hoekstra, R. Lal, P. Pappachan, V. Phegade, and J. Del Cuvillo, "Using innovative instructions to create trustworthy software solutions," in *Proceedings of the Second International Workshop on Hardware and Architectural Support for Security and Privacy (HASP '13)*. Tel-Aviv, Israel: ACM, June 2013.

[33] G. D. Hunt and L. Koved, "Checkpoints for permissionless blockchains," United State Patent 10 523 421, December 31, 2019.

[34] Intel Corporation, *Intel® 64 and IA-32 Architectures Software Developer's Manual*, 325384-071US, October 2019.

[35] IoT Security Foundation, "Establishing principles for Internet of Things security," White Paper, September 2015. [Online]. Available: https://www.iotsecurityfoundation.org/wp-content/uploads/2015/09/IoTSF-Establishing-Principles-for-IoT-Security-Download.pdf

[36] O. Jacobovitz, "Blockchain for identity management," The Lynne and William Frankel Center for Computer Science, Department of Computer Science, Ben-Gurion University, Beer Sheva, Israel, Tech. Rep. 16-02, December 2016.

[37] I. R. Jenkins and S. W. Smith, "Distributed IoT attestation via blockchain," To appear in Proceedings of the 1st Workshop on Secure IoT, Edge and Cloud systems (SIoTEC '20). Melbourne, Australia: IEEE Computer Society, May 11–14 2020.

[38] Kaspersky, "Secure OS for the Internet of Things," White Paper, 2018. [Online]. Available: https://os.kaspersky.com/media/KasperskyOS-for-IoT-En.pdf

[39] G. Klein, K. Elphinstone, G. Heiser, J. Andronick, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish, T. Sewell, H. Tuch, and S. Winwood, "seL4: Formal verification of an OS kernel," in *Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles (SOSP '09)*. Big Sky, MT, USA: ACM, October 2009, pp. 207–220.

[40] P. Kocher, R. Lee, G. McGraw, and A. Raghunathan, "Security as a new dimension in embedded system design," in *Proceedings of the 41st Annual Design Automation Conference (DAC '04)*. San Diego, CA, USA: ACM, June 7–11, 2004, pp. 753–760.

[41] P. Koeberl, S. Schulz, A. Sadeghi, and V. Varadharajan, "TrustLite: A security architecture for tiny embedded devices," in *Proceedings of the Ninth European Conference on Computer Systems (EuroSys '14)*. Amsterdam, Netherlands: ACM, April 13–16, 2014, pp. 1–14.

[42] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDOS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, July 2017.

[43] Y. Li, J. M. McCune, and A. Perrig, "VIPER: Verifying the integrity of peripherals' firmware," in *Proceedings of the 18th ACM Conference*

on Computer and Communications Security (CCS '11). Chicago, IL, USA: ACM, October 2011, pp. 3–16.

[44] X. Liang, S. Shetty, D. Tosh, P. Foytik, and L. Zhang, "Towards a trusted and privacy preserving membership service in distributed ledger using intel software guard extensions," in *Proceedings of the 19th International Conference on Information and Communications Security (ICICS '17)*, Beijing, China, December 6-8, 2017, pp. 304–310.

[45] G. McGraw, "Software security," *IEEE Security & Privacy*, vol. 2, no. 2, pp. 80–83, March 2004.

[46] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar, "Innovative instructions and software model for isolated execution," in *Proceedings of the Second International Workshop on Hardware and Architectural Support for Security and Privacy (HASP '13)*. Tel-Aviv, Israel: ACM, June 2013.

[47] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," White Paper, November 2008.

[48] B. Ngabonziza, D. Martin, A. Bailey, H. Cho, and S. Martin, "TrustZone explained: Architectural features and use cases," in *Proceedings of the Second International Conference on Collaboration and Internet Computing (CIC '16)*. Pittsburgh, PA, USA: IEEE, November 2016, pp. 445–451.

[49] J. Noorman, P. Agten, W. Daniels, R. Strackx, A. Van Herrewege, C. Huygens, B. Preneel, I. Verbauwhede, and F. Piessens, "Sancus: Low-cost trustworthy extensible networked devices with a zero-software trusted computing base," in *Proceedings of the 22nd USENIX Security Symposium (SEC '13)*. Washington, D.C., USA: USENIX Association, August 14–16 2013, pp. 479–494.

[50] A. S. Omar and O. Basir, "Identity management in IoT networks using blockchain and smart contracts," in *Proceedings of the 2018 IEEE Conferences on Internet of Things (iThings), Green Computing and Communications (GreenCom), Cyber, Physical and Social Computing (CPSCom), Smart Data (SmartData), Blockchain, Computer and Information Technology (CIT), Congress on Cybermatics*. Halifax, Canada: IEEE, August 2018, pp. 994–1000.

[51] K. R. Özyılmaz and A. Yurdakul, "Work-in-progress: Integrating low-power IoT devices to a blockchain-based infrastructure," in *Proceedings of the 13th International Conference on Embedded Software Companion (EMSOFT '17)*. Seoul, Republic of Korea: ACM, October 15-20 2017, p. 2.

[52] T. Park and K. G. Shin, "Soft tamper-proofing via program integrity verification in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 297–309, March 2005.

[53] E. Ronen, A. Shamir, A. Weingarten, and C. O'Flynn, "IoT goes nuclear: Creating a ZigBee chain reaction," in *Proceedings of the 38th IEEE Symposium on Security and Privacy (S & P '17)*. San Jose, CA, USA: IEEE, May 22–24, 2017, pp. 195–212.

[54] R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn, "Design and implementation of a TCG-based integrity measurement architecture," in *Proceedings of the 13th USENIX Security Symposium (SEC '04)*. San Diego, CA, USA: USENIX Association, August 9–13 2004, pp. 223–238.

[55] A. Seshadri, M. Luk, E. Shi, A. Perrig, L. Van Doorn, and P. Khosla, "Pioneer: Verifying code integrity and enforcing untampered code execution on legacy systems," *ACM SIGOPS Operating Systems Review*, vol. 39, no. 5, pp. 1–16, October 2005.

[56] A. Seshadri, A. Perrig, L. Van Doorn, and P. Khosla, "SWATT: Software-based attestation for embedded devices," in *Proceedings of the 25th IEEE Symposium on Security and Privacy (S & P '04)*. Oakland, CA, USA: IEEE, May 9–12, 2004, pp. 272–282.

[57] M. Shaneck, K. Mahadevan, V. Kher, and Y. Kim, "Remote software-based attestation for wireless sensors," in *Proceedings of the Second European Conference on Security and Privacy in Ad-Hoc and Sensor Networks (ESAS '05)*. Visegrad, Hungary: Springer-Verlag, July 2005, pp. 27–41.

[58] E. Shi, A. Perrig, and L. van Doorn, "BIND: A fine-grained attestation service for secure distributed systems," in *Proceedings of the 26th IEEE Symposium on Security and Privacy (S & P '05)*. Oakland, CA, USA: IEEE, May 8-11, 2005, pp. 154–168.

[59] P. Sparks, "The route to a trillion devices," White Paper, ARM Limited, June 2017. [Online]. Available: https://pages.arm.com/rs/312-SAX-488/images/Arm-The-route-to-trillion-devices_2018.pdf

[61] State of California, "Security of connected devices," CA Civ Code § 1798.91.04 (2018).

[62] R. Strackx, F. Piessens, and B. Preneel, "Efficient isolation of trusted subsystems in embedded systems," in *Proceedings of the 6th International ICST Conference on Security and Privacy in Communication Networks (SecureComm '10)*. Singapore: Springer, September 2010, pp. 344–361.

[63] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, September 1997.

[64] H. Tan, G. Tsudik, and S. Jha, "MTRA: Multiple-tier remote attestation in IoT networks," in *Proceedings of the 5th Annual IEEE Conference on Communications and Network Security (CNS '17)*. Las Vegas, NV, USA: IEEE, October 9-11, 2017, pp. 1–9.

[65] The Linux Foundation, "Linux Foundation's Hyperledger project announces 30 founding members and code proposals to advance blockchain technology," Press Release, February 2016.

[66] Trusted Computing Group, "TCG launches initiative to develop "world's tiniest TPM"," Press Release, June 2019. [Online]. Available: https://trustedcomputinggroup.org/tcg-launches-initiative-to-develop-worlds-tiniest-tpm/

[67] S. Underwood, "Blockchain beyond Bitcoin," *Communications of the ACM*, vol. 59, no. 11, pp. 15–17, November 2016.

[68] US Department of Defense, "Trusted computer system evaluation criteria," in *DoD 5200.28-STD*, December 26, 1985.

[69] J. Viega and H. Thompson, "The state of embedded-device security (spoiler alert: It's bad)," *IEEE Security & Privacy*, vol. 10, no. 5, pp. 68–70, October 2012.

[70] Y. Yang, X. Wang, S. Zhu, and G. Cao, "Distributed software-based attestation for node compromise detection in sensor networks," in *Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems (SRDS '07)*. Beijing, China: IEEE, October 10-12, 2007, pp. 219–230.

[71] S. Zeitouni, G. Dessouky, O. Arias, D. Sullivan, A. Ibrahim, Y. Jin, and A. Sadeghi, "ATRIUM: Runtime attestation resilient under memory attacks," in *Proceedings of the 36th IEEE/ACM International Conference on Computer-Aided Design (ICCAD '17)*. Irvine, CA, USA: IEEE, November 13–16, 2017, pp. 384–391.