International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020

Multi-layer Attestation for Internet of Things using Blockchain

Vytarani Mathane, P V Lakshmi

Abstract: ***Internet of Things (IOT) by its nature comprises of heterogeneous devices with varying degree of resources and capabilities with common attributes that those are connected and uniquely identifiable over the network. Given the always on always connected nature of IoT devices along with virtually limitless applications, the attack surface of constituent IoT device is very large. Hence ability to attest IoT devices for its trustworthiness is very important factor in determining trustworthiness of IoT network. In past significant amount of research has focused on possible attestation mechanisms for IoT but all those proposals invariably depend on specific hardware implementation like TrustZone, SGX, TPM, RTC, memory with OTP etc. Sine all such security primitives are either architecture or manufacturer specific it is not possible to build common unified attestation scheme for all constituent IoT devices in a typical IoT network using any of those primitives. This research work proposes different pragmatic approach to define such common and scalable attestation scheme that all IoT devices within IoT network could deploy. The proposed scheme makes use of memory management which is one of most basic features of any processor or controller to build common and scalable attestation mechanism for all types of IoT devices. The approach is to understand threat model and then develop mitigations in pragmatic manner.***

Keywords: ***IOT, attestation, security, distributed ledger, heterogeneous devices, network integrity, device integrity, data integrity, factory automation, low power devices, low resource devices.***

I. INTRODUCTION

Over last few years IOT has emerged as an overarching all-inclusive term used to address every device which is either connected to internet directly or indirectly and is uniquely identifiable remotely. Implications of such an evolution is that the terminology IoT corresponds to very wide range of heterogenous devices with varying hardware, software, functionalities and computing resources. IoT devices most commonly work as end points which collect vital information from field and feed it to data aggregators. In turn such data aggregators make decisions based upon aggregated data using additional software intelligence. These decisions drive responses to various events occurring on the field where data was collected. Even though these decisions and responses will vary in criticality, those will alter course of events, controls or execution in real life.

Hence reliability of all such data generating out of wide array of IoT devices is very important. Significant amount of research efforts has continued over last few years to improve reliability of IoT devices as well as data by strengthening its security in variety of ways.

This research work briefly analyzes some of significant approaches towards attestation techniques for IoT proposed so far over last few years on the parameters of feasibility, return on investment (RoI), resilience and gaps. These authors then proceed to define holistic but differing view of security needs of IoT segment followed by comprehensive and pragmatic solutions to meet those needs.

II. RELATED WORK

Ayoade et al. (2018) primarily deals with management & auditing of data generated in an IoT network using Intel's SGX trusted execution environment and Distributed ledger Tangle technology. This approach does not address trustworthiness of the device itself and relies on very specific hardware technology available only to very small but costly subset of IoT devices. Thus, it lacks scalability and ability to secure origin of data which is device itself. At the same time, it requires significant amount of computing resources often not available to IoT segment to manage security of data as every transaction is secured using Distributed ledger Tangle. Tan, Tsudik and Jha (2017) acknowledge heterogeneity of IoT devices very distinctly and proposes scheme to utilize hierarchical design where nodes with TPM could take over more active responsibilities for attestation of nodes without such primitives. It does require usage of shared secret key which is not reliable when attacker attacks the prover IOT device. Scalability of such scheme would indeed depend upon largely on number of such superior nodes in IoT network.

Park and Kim (2017) address trustworthiness of data transaction with remote attestation but does not comprehend trustworthiness of device itself by assuming initial status of device is trusted during registration & subsequent updates assumes.

Revised Manuscript Received on February 05, 2020.
* Correspondence Author
Vytarani Mathane*, research scholar, GITAM (Deemed to be University).
P V Lakshmi, professor, GITAM (Deemed to be University).

This solution also mandates use of specific hardware technologies like ARM TrustZone, e-fuses, Real Time Clock, expects each device to have secured storage, layered privileged yet unstandardized software architecture calling for separation of Rich Execution Environment (REE) & Trusted Execution Environment (TEE), all of which are typically present in higher end devices meant for mobiles & handhelds. This approach also mandates use of multiple level of PKI keys and operations apart of maintaining mapping of device ID against those multiple levels of PKI keys. Second level and onwards of these PKI keys are generated in pseudorandom fashion weakening security based upon those keys. Every transaction to be attested carries device measurements to be verified in runtime adding significant overhead in verification, mining etc. Again, this scheme asks for significant amount of computing and storage resources as each transition generating out of IoT is meant to be attested. Compromised REE's ability to extract TCB of device is not addressed leaving significant attack door open. Devices used once in an IoT network are not capable of redeployment owing to use of fuses to manage registration of device.

Branden burger et al. (2018) analyses combining Blockchain with TEE and SGX, argues that in such combination of TEE and Blockchain, TEE resident smart contracts could not be held accountable for rollback prevention but Blockchain must handle it by design. Next it goes on to propose alternate scheme by combining Intel SGX with Blockchain where SGX enclaves execute smart contracts to provide mitigation for possible rollback attacks. This work though does not address an important consideration of malware executing itself as smart contract with SGX and be part of Blockchain.

Hristozov et al. (2018) describes runtime attestation for IoT devices limited to certain layers of stack in multi-layer and multi-privileged software environment. This scheme offers resiliency against non-physical remote attacks executed via memory corruption. This scheme assumes devices to have adequate firewalling or isolation mechanism built-in silicon to quarantine some portion of volatile and non-volatile memory, device unique identifier accessible to only ROM and software stack divided into multiple privilege level. During device boot these software components execute in descending order of their privilege level whereby each layer measures next layer. Return Oriented Programming (ROP) or Denial of Service (DoS) attacks are not under scope of these mitigations.

Abera et al. (2016) surveys different kinds of attestation techniques based upon some hardware and software techniques, hybrid approaches, run time attestations along with subsequent limitations and challenges. This survey literature brings out scalability and robust runtime attention as key challenge for IOT attestation. According to Asokan et al. (2015) SEDA is the attestation scheme for swarm of devices with static and dynamic topologies. But it uses symmetric key cryptography which is not suitable when the prover is compromised.

All of these proposals rely on very specific hardware technologies like SGX, TrustZone, MPUs equipped with isolation techniques, at times need hardware components like RTC, TPM, memories with secured storage, components and required privilege based layered software architecture. These proposals attempt to address different aspects of IoT security depending upon computing resources assumed as pre-requisites.

III. PROBLEM STATEMENT

In the light of above-mentioned multiple research trends for IoT attestation authors attempt to revisit definition of what needs to secured or attested as first step and subsequently extrapolate that definition to the context of IoT. Such an exercise leads to more apt contextual attestation model for IoT.

It is not practical as of today and in distant future to lay out specifics of hardware, software resources as prerequisites for an IoT device and standardize those as adoption of such standard definition by hardware & software vendors is not realistic. At the cost of sounding repetitious it is worth to impress that IoT encompasses very vast spectrum of heterogenous devices operating in even more diversified application domains like computing for personal, industrial, defense, environmental and many more technologies. Needless to say, devices selected for any of such varied functional domain will depend upon application & cost sensitivities making it impossible to force any one particular hardware or software technologies.

For the sake of further discussion lets us use word device as representing processor, controller, micro-processor, micro-controller, SoC or any such computing unit. Let us also define attestation as merely ability to prove trustworthiness of certain entity. The trustworthiness itself will have varying meaning depending upon what needs to be attested. Above mentioned past research implies two separate trends i.e. attestation of device versus attestation of data generated by device.

Much has been discussed in past over merits of boot time static attestation versus dynamic or on-demand runtime attestation and it is said static attestation is not enough to prove trustworthiness of device. But any such discussion to define attestation needs to be framed more specifically in the context of how particular device behaves upon any change to its software.

This present research work claims that within defined threat model, data generated or provided by a device is trustworthy as long as integrity and authenticity of the software executing on the device has not been compromised. Receiver should trust data sent by uncompromised device. The threat model here does not dwell in details on the man in the middle attacks because such attack has limited benefit potential as it is typically deployed on the current transaction over specific communication link or channel and is carried out one attack at a time limiting its compromise potential assuming end devices use basics security measures like session based key for securing exchanges.

A. Approach

With these arguments as foundation for further discussions these authors propose very pragmatic methods to establish and verify trustworthiness of the devices participating in typical IoT network.

These methods are scalable to wide variety of IoT devices despite their heterogeneity.

Proposed methods in this research rely on the most fundamental attributes of a device called memory management. More specifically it leverages availability or lack of availability of hardware unit that handles memory management on device and is called as memory management unit, hereafter referred as MMU.

Availability of MMU allows device to enable virtual memory with various types of techniques to map virtual memory to physical memory. Specifics of those techniques are beyond the scope and necessity of the current discussion. This fundament property of enabling virtual memory space has implications how device behaves after software update and subsequently to the boot flow and execution of the software on the device. This behavior upon software update is closely analyzed in following text -

- MMU less device has flat address space; all processes can access all physical address space. When device powers on it would copy its ROM and loadable software to its system RAM and execute out of it. When software of such device is updated the device needs to reboot in order to use updated software.

- MMU enabled device has hardware backed layered memory model enabling two or more rings of execution. Privileged software like kernel operates in higher privileged ring typically called as ring 0 and has access to all physical memory, rest of unprivileged software operates at ring 1-3, typically in ring 3 and understand only virtual memory. When privileged software is updated device needs to reboot to use new software whereas unprivileged software update does not warrant rebooting device.

Now integrity of an IoT network relies on trustworthiness of each of its device and data exchanged between these devices as shown in Fig. 1.

With this in mind this research work proposes multi layered security schemes to protect overall trustworthiness of the IoT network. Essential element of the proposed scheme is using off-the-host mechanism to protect and verify trustworthiness of constituent IoT devices.

Trustworthiness of device is quantified as measurement of its critical software. Further simple methods like hashing software of device using sufficient key strength like 256 bits and using off-the-host container mechanism to protect & verify such measurements will achieve desired goal of protecting trustworthiness of constituent of IoT devices and hence of IoT network itself. Approach to use off-the-host mechanism to protect and verify trust measurements of all device is required because IoT devices vary in their on-the-host capabilities widely to be able to define uniform scheme for all of them.

Proposed off-the-host container to protect trust measurements is based on distributed ledger technology commonly known as DLT or Tangle (Popov 2018). This is truly distributed and efficient for following reasons -

- There are no hierarchical nodes e.g. mining nodes that could become bottleneck of the ledger as transactions keep growing over the time.
- There is financial reward for mining in Tangle. Mining is in a way off-loaded to each node and for each incoming transaction a node is required to verify two other transactions.
- Advantage of above requirement of verifying two other transactions to insert new transaction is that regardless of how many transactions are added to Tangle over period of time, Tangle itself offers relatively fixed transaction time for each incoming transaction request which equals to time required to verify two previous transactions.

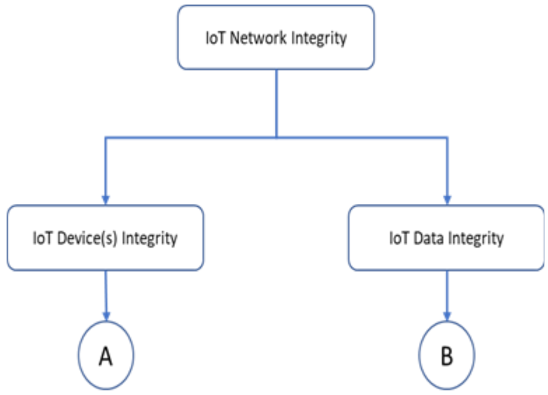

Fig. 1. IoT networks integrity is maintained as long as its devices and data is not compromised.

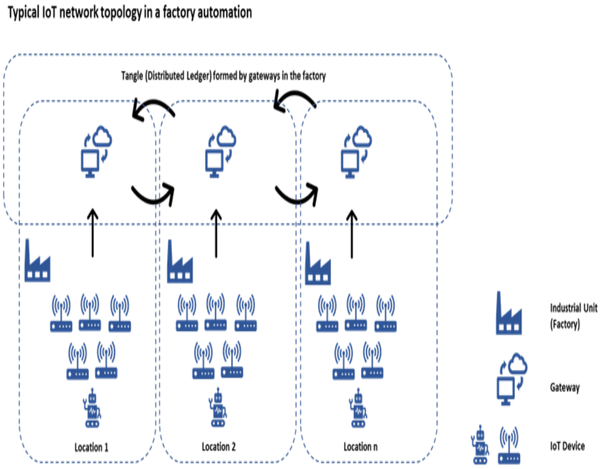
Typical IoT network topology in a factory automation
Tangle (Distributed Ledger) formed by gateways in the factory
Location 1
Location 2
Location n
Industrial Unit (Factory)
Gateway
IoT Device

Fig. 2. IoT devices network topology in typical large factory environment

IV. DESIGN OF MULTI-LAYER ATTESTATION FOR IOT NETWORK

The solution presented here comprises of two different methods to address two distinct security challenges of IoT devices. The key to solution is scalability so that despite heterogeneity of IoT devices, one common scheme could be deployed on all participant devices of the IoT network. Fig. 2 represents generalized representation of typical IoT network common in industrial, factory, weather monitoring or any such distributed application.

Solution to provide common attestation scheme is based upon following points-

- Devices without MMU should use distributed ledger technology (DLT) based Tangle like architecture to maintain their trustworthiness by protecting hash of their software as transactions of Distributed ledger Tangle, upon every update old transaction should be blacklisted and new to replace it. This corresponds to path A in Fig. 1 and further C & D in Fig. 3.
- Devices with MMU capable computing units should use distributed ledger technology (DLT) based Tangle like architecture to maintain the trustworthiness of their privileged software kernel by protecting hash of their software kernels as transactions of Tangle, upon every update old transaction should be blacklisted and new to replace it. This corresponds to path A in Fig. 1 and further C & D in Fig. 3.
- Devices with MMU capable computing units should use adequate control flow protection methods such as one proposed in this research work to ensure malware does not succeed in capturing execution of software execution and thus compromise data originating out of it. This corresponds to path A in Fig. 1 and further E in Fig. 3.
- All the devices irrespective of MMU status should use efficient key negotiation protocol ideally session based but the one with capabilities to generate keys in advance of its usage to protect data exchange between two nodes. This corresponds to B in Fig. 1 and further F & G in Fig. 4. This research paper does not intend to cover path G as there are numerous existing techniques already established to handle these needs like TLS, DH or ECDHA etc.

In the following section, this research work intends to provide simple and scalable solutions to scenarios tagged as C, D, E & F, while as mentioned before there are many mechanisms available to solve G.

A. Device Software Attestation (C, D, F)

As stated, MMU less devices would reboot when their software is updated and MMU enabled devices exhibit similar behavior of rebooting when privileged software like kernel is updated. This behavior implies that even in case of malicious attempt to alter such privileged software or firmware will result in rebooting device.

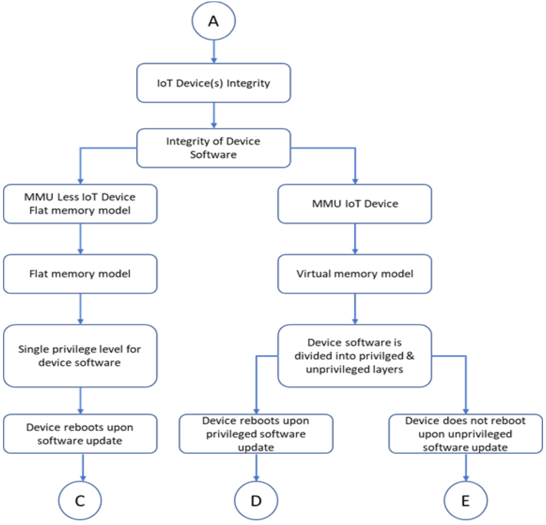
A
IoT Device(s) Integrity
Integrity of Device Software
MMU Less IoT Device Flat memory model
MMU IoT Device
Flat memory model
Virtual memory model
Single privilege level for device software
Device software is divided into privilged & unprivileged layers
Device reboots upon software update
Device reboots upon privileged software update
Device does not reboot upon unprivileged software update
C
D
E

Fig. 3. Classifying IoT devices based upon their memory model & behavior on software update

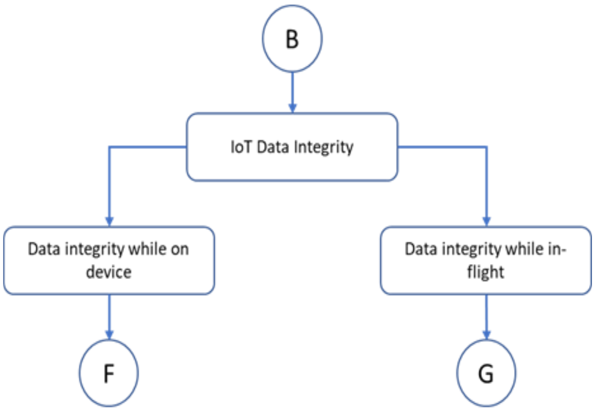
B
IoT Data Integrity
Data integrity while on device
Data integrity while in-flight
F
G

Fig. 4. IoT data integrity needs to be maintained while on device & in-flight

The solution proposed is to enforce off-the-host attestation of the boot image before allowing it to execute on software. Since majority of the IoT device are low power and low resource devices by & large these devices do not have security primitives like Trusted Platform Module (TPM) or Trusted Execution Environment (TEE) which are isolated and have ability to enforce trusted boot on device. In addition to that TEEs are not standardized and with such heterogeneity in IoT devices it is not possible to rely on availability of TEE, TPM and hence on-the-host attestation of boot image.

Off-the-host attestation offers scalability to deploy it across variety to IoT devices at the expense some overhead in of boot time. This solution assumes three different roles viz. prover (IoT device) seeking attestation, verifier (gateway in an IoT network) attesting device and distributed ledger or Tangle which provides reference attestation parameters.

The scheme is built to prevent replay attack on boot attestation parameters, spoofing of prover or verifier identity. Usage of newer distributed ledger technology called Tangle is significant in this scheme. It could be replied to maintain trustworthiness of attestation parameters in immutable fashion, allows to update said attestation parameters of the device as and when required. It also offers significant advantage over traditional Distributed ledger Tangle technology as it does not suffer from financial and computational overhead of mining operations and promises constant “time to complete” transaction.

Following flow describes how device boot attestation parameters are added to the Tangle. This flow required verifier to generate following three device unique identifiers

dev_id – Device unique identity assigned by verifier to each device on its IoT network and used further to identify device until it is part of network

p_id & v_id – There are generated on device in manner that those could be used by prover to verify verifier’s identity. It could be as simple as binary compliments or NAND logic which is used by device to confirm correctness of verifier. These being device unique and never leaving device outside trusted pairing environment, could be trusted.

Following Fig. 5 explains detailed sequence of action for registration of device to the verifier and Tangle. This flow holds good even for software update scenario except that dev_id is not regenerated, p_id and v_id could be regenerated or not depending upon what level of security is required.

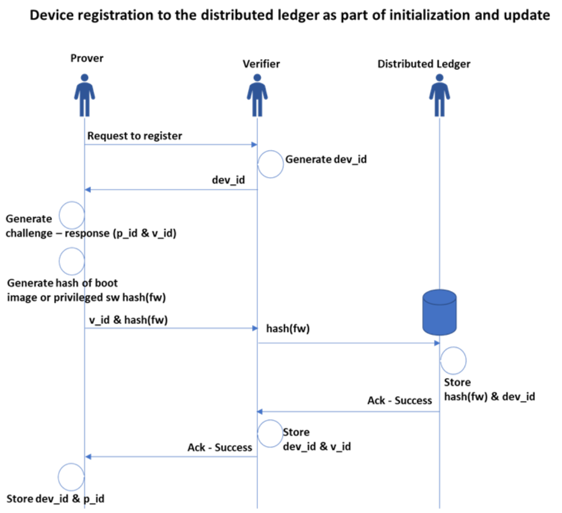

Fig. 5. Registering device boot attestation parameters during initialization or software update

Following Fig. 6 explains sequence of events or actions each time device reboots and seeks attestation of its trustworthiness based upon agreed attestation parameters. dev_id is used by verifier to identify device and used to retrieve required measurements from the Tangle. p_id and v_id are used to ensure verifier is not spoofed as they should have defined results on certain operation carried out within attesting device.

Addition of nonce to the firmware image while attesting boot image or privileged software ensures resistance against replay attacks executed out of device or within the device. The overhead comes out of hashing data twice on device hash(fw) and then hash (hash(fw) + nonce). This is required as without this verifier would need to store actual firmware or software image corresponding to each device and that implies single point of failure. Adding such payloads to DLT Tangle will also increase transaction time and memory requirements exponentially.

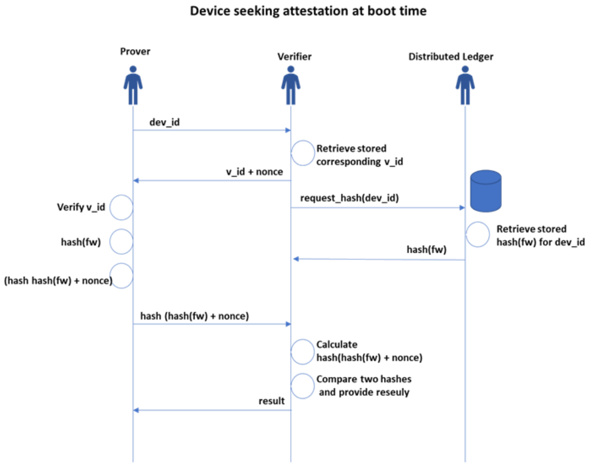

Fig. 6. Device seeking attestation at boot time

This scheme also offers solution to scenario F since as long as software running on device is trusted, data on the device is also equally trusted.

B. Device Software Attestation(E)

The remaining part of the problem is how to ensure trustworthiness of the software on MMU enabled devices. This layer upon change does not require reboot so there would be no way to identify tampering. Also, this layer seems to be very fragmented with its file structure, various standalone ap0070lication and is usually very large in size.

While providing usual boot image or privilege software type protection where every byte is counted is not practical, there are some prominent ways to attack this layer and among them return oriented programming ROP is most recurring. Its one way to hijack execution control from legitimate software and preempt it with malicious software loaded into RAM. This is executed typically with the help of buffer overflow are typically used to overwrite return address of the caller function on the stack memory.

Various ways are being explored to mitigate such attack. Some of those involve hardware change to maintain shadow stack or at least some of the key element of stack like return address of caller in hardware backed program in accessible registers. None of those have materialized yet. Another research attempted to solve this with cryptographic ways to maintain integrity of pointers in the program memory. These mitigations either require costly hardware modifications or computing intensive cryptographic operations.

This research work contends that as long as privileged software like kernel which does scheduling & context switch of processes is trusted for its integrity; it could be used to mitigate ROP attacks by allowing it to take copy of return address per stack push and verify it with return address on stack. This is comparatively not very resource intensive and can be achieved by simple modifications to kernel scheduling code.

V. IMPLEMENTATION

To verify this proposal authors used two sets of hardware viz. one mainstream Intel Core i7 with 16 GB RAM and Linux OS. Second set of experiments was carried out with ARM Cortex M 200 MHZ memory.

A. Results

For Intel i7 running multi-threaded Linux time taken to calculate hash on 32KB of data from Linux file storage vary from 2 mS to 9 ms. For ARM Cortex M operating at 96MHz with single threaded RTOS environment time taken to calculate hash of 256KB of data is around 6 ms.

The average time taken to send a hash data transaction from client to the node in tangle is 27.3 ms and average time taken to retrieve a hash from tangle is 22.8 ms. Fig.7. represents the average time taken to send hash to a node in tangle from a client. Fig. 8. represents the average time taken to retrieve hash from tangle.

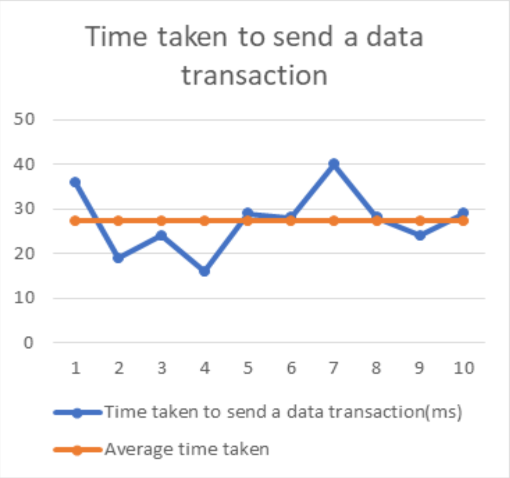

Fig. 7. Average time taken to send a data transaction from client to node in a tangle.

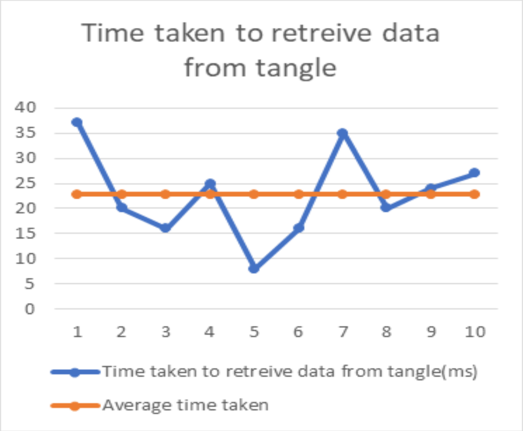

Fig. 8. Average time taken to retrieve hash from tangle

Various proposals analyzed earlier in this research work similar or greater overhead in terms of computational resources like code and execution time. Such overhead though depends greatly upon the underlying techniques used and entity that is being protected. Multi-tier attestation using IoT nodes equipped with TPM for attestation of lower end IoT devices has performance overhead proportional to the topology of the devices, the time taken for such an attestation is in the range of several 10s of seconds. Research work showed close to 30 seconds overhead for few thousand of devices in network. The time it takes would thus also depend on number of IoT devices in network.

On the contrary schemes using combination of TEE along with SGX to protect transactions against any attacks required on an average 9k lines of code in both trusted and untrusted environments, 3.5Kb of data overhead during individual transactions and latency of close to 100mS to achieve such operations. Using traditional techniques like HMAC & ECDSA to achieve attestation of boot time measurements shows wide variance in overhead depending upon nature of operation i.e. symmetric cryptography vs asymmetric cryptography. Figures indicate up to 2.2 seconds of overhead. Usage of proprietary Blockchains like TM-Coin mechanism on the other hand requires 27mS for both key pair generation as well as signing.

In contrast while on one hand methodologies presented in this research work are scalable to all types of IoT devices and on the other hand overhead incurred is minimal and constant irrespective of the devices participating in IoT network.

VI. CONCLUSION

Off-the-host boot image verification adds overhead of few milliseconds to boot time in return for not having any attestation capabilities of device. Since IoT devices are of always-on-always-connected nature, this delay is once is while and provides sufficient and scalable scheme for deployment across categories of devices.

REFERENCES

1. Gbadebo Ayoade, Vishal Karande, Latifur Khan and Kevin Hamlen, “Decentralized IoT Data Management Using Blockchain and Trusted Execution Environment,” IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 6-9 July 2018, DOI: 10.1109/IRI.2018.00011.
2. Hailun Tan, Gene Tsudik, Sanjay Jha, “MTRA: Multiple-tier remote attestation in IoT networks,” IEEE Conference on Communications and Network Security (CNS), Las Vegas, NV, USA, 9-11 Oct. 2017, DOI: 10.1109/CNS.2017.8228638.
3. Jaemin Park, Kwangjo Kim, “TM-Coin: Trustworthy management of TCB measurements in IoT,” IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 13-17 March 2017, DOI: 10.1109/PERCOMW.2017.7917640.
4. Marcus Brandenburger, Christian Cachin, Rüdiger Kapitza, Alessandro Sorniotti, “Blockchain and Trusted Computing: Problems, Pitfalls, and a Solution for Hyperledger Fabric,” Distributed, Parallel, and Cluster Computing (cs.DC); Cryptography and Security (cs.CR), arXiv:1805.08541 [cs.DC].
5. Stefan Hristozov, Johann Heyszl, Steffen Wagner, Georg Sigl, “Practical Runtime Attestation for Tiny IoT Devices,” Conference: Workshop on Decentralized IoT Security and Standards, January 2018, DOI: 10.14722/diss.2018.23011.
6. Tigist Abera, N. Asokan etc.,“Invited: Things, Trouble, Trust: On Building Trust in IoT Systems,” 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, USA, 5-9 June 2016, DOI: 10.1145/2897937.2905020.
7. N. Asokan, Ferdinand Brasser etc.,“ SEDA: Scalable Embedded Device Attestation,” 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, USA, 12- 16 October, 2015, pp. 964-975, Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security.
8. M. The Tangle April 30, 2018. Version 1.4.3 Serguei Popov - Available at https://uk.sagepub.com/sites/default/files/sage_harvard_reference_style_0.pdf.

AUTHORS PROFILE

Vytarani Mathane, research scholar, GITAM (Deemed to be University).

P V Lakshmi, professor, GITAM (Deemed to be University).