Build with intrinsic security: Trusted autonomy security system

Shen He, Jing Huang and Penglin Yang

Abstract
Security of Internet of things network is difficult to manage because Internet of things devices are heterogeneous and always deployed in remote places. Based on these features, attacks against Internet of things network like distributed denial-of-service and system intrusion are easier than traditional network. In order to enhance Internet of things network’s security, in this article we propose the concept of intrinsic security to establish a trusted and autonomy Internet of things network. Intrinsic security emphasizes that network itself must be trusted and security autonomous to address the security problems which are caused by remote and heterogeneous. Based on this thought, we designed the architecture of Trusted Autonomy Security System. Trusted Autonomy Security System maintains a LAN characterized by trusted and security autonomy, in which all Internet of things network gateways formed a decentralized autonomy system to evaluate if other gateways or Internet of things devices are secure. Compared to traditional Internet of things networks, this architecture uses trusted computing technique to build the basic of security, and makes Internet of things network available to manage security automatically. In addition, the decentralized architecture could reduce security risks caused by compromise of single node in Internet of things network. With these advantages, it can be expected that this novel idea will have more development in Internet of things scenarios with high security requirements.

Keywords
Autonomy security, block chain, intrinsic security, trusted computing

Date received: 25 August 2020; accepted: 28 October 2020

Handling Editor: Yanjiao Chen

Introduction
Along with the rapid development of Internet of things (IoT) technique, this method has been involved in various fields such as medical, industrial, transportation, and finance. The society relies more and more on IoT, extra attentions also need to be paid on its security risk. Usually, IoT networks are deployed in remote and distributed places, and network gateways communicate with these devices and report relevant information to administrator. In normal conditions, all the data reported to administrator are assumed to be trusted. However, with potential security risks like system vulnerabilities and intrusion, the data collected and sent by IoT gateways could be forged and tampered. It may be too late for the administrator to find out the abnormality and make reaction. In extreme cases, the merge of attack flow of IoT network could cause large-scale Internet security incidents, like Dyn DDoS Attack and Mirai DDoS Attack.

The deep reason of these attacks is because IoT network is vulnerable and all the IoT network flow is converged without any supervision and autonomic reaction. Intrinsic security thought brings new ideas to IoT network security. This concept is different to the hardware-
based intrinsic security like. Nili et al. focused on hardware technique like nonlinear conductance variation to prevent electron component from cloning, which is not related to our topic. In this article, we would like to create a new network system which is built with trust and security autonomy. This security architecture prevents IoT attacks from or near the source. As a consequence, scale of DDoS attack will be reduced dramatically.

Intrinsic security insists that network needs to be born or designed with security features. More specifically, trusted is the basic, and security autonomy is network’s organization form. By this way, security incidents in IoT network can be handled inside the IoT network domain and prevent the spread of attacks. Figure 1 shows the difference between normal IoT security architecture and intrinsic security architecture. Intrinsic security changes the security topology of the existing system which has a vulnerable center node, embeds security capability to every domain in IoT network. In these domains, network gateways have security information process ability and trusted root. In intrinsic security network there is no central node with absolute authority, every gateway needs to communicate with each other to make comprehensive security decisions. Also, changes of security status of gateways during lifecycle will also be dynamically detected and evaluated.

Compared with normal IoT security model, advantages of intrinsic security are listed in the following:

1. Trusted: every domain in IoT network has trusted root to execute security actions. Security risks can be limited inside domain without spreading out.
2. Decentralization: Intrinsic security does not require specific centralized node to manage system’s security. The decentralized structure reduced security risk from single node’s compromise.
3. Easy to deploy: It is easy to deploy without any central security management node.

The specific sections of this article are structured as follows. Section “Related work” describes the related work of decentralized autonomy and trusted computing. Section “Architecture of trusted autonomy security system” describes the architecture of trusted autonomy security system. Section “Trust anchor” describes the design of trust anchor and its implementation in TASS. Section “Autonomy” focuses on how to implement decentralized autonomy architecture in TASS. In section “Evaluation of TASS,” the security and performance of the prototype are discussed. In the end, we make a conclusion and discuss the future work.

**Related work**

Trust and decentralization are TASS’s basic features. Relevant research and technologies have been studied for a while and approached certain achievements.

**Decentralized autonomy**

In decentralized autonomy field, blockchain is an emerging technology, through which decentralized consensus and storage can be approached. Blockchain is composed by a series of functions like decentralized networking, data tamper-proofing, consensus-based trust construction, and the mechanism of automatic execution of contracts. These provide a distributed autonomy platform and establish connections between multiple partners. Blockchain technology changed the traditional network structure from “series” to “parallel.” For example, in the architecture of Dapp, applications run in a decentralized way in different terminals. Blockchain is also widely used in IoT area, such as blockchain-based identity authentication, tracing of hardware, and threat intelligence sharing. In our thought of intrinsic security, Blockchain could be used as the decentralized platform and make security autonomy practicable.

Typical distributed consensus algorithms include KAFKA, PBFT, POW, and RAFT. Among which, POW relies on workload proof to realize decentralized computing and consensus. For reaching consensus, the mining function’s computing scale need to be large enough which is unacceptable by IoT devices with limited computing resources. Like Bitcoin who...
uses POW to mine coin prize can barely cover electric charge right now. In IoT, the feasible choice should be the limited-scale decentralized algorithm such as RAFT and PBFT. Normally, RAFT is used in private chains which only have malfunctioning nodes such as power off or offline. But in security situations, we have to assume that there may have evil nodes in distributed system. That means we need to use consortium chains to construct consensus in IoT network, in which PBFT is the appropriate choice. From the principle of PBFT, when less than 1/3 nodes failed, it still can be considered that the distributed consensus mechanism is effective, and the data in the block chain is complete and not tampered.

**Trusted computing**

Trust is the basis of TASS, trust computing techniques like TPM,15 SGX,16 DICE,17 and Trust Zone18 provide trusted roots in hardware. In IoT gateways, lack of trust in IoT will lead to the freedom of selecting untrusted or even malicious data to upload to chains. And without trust, the perception layer, transportation layer and application layer of IoT network will also be affected.19 Distributed consensus algorithm guarantees that data after uploaded to chains can reach consensus and keep integrity, but it cannot guarantee whether the device have chosen the real data to upload to chains. If one device is malicious, the data it chose to upload cannot be trusted even the consensus mechanism is valid. When PBFT, RAFT, and other private chain algorithms are used for decentralized IoT system, malicious IoT nodes can upload fake information to puzzle decentralized system without any security restrictions.

As far as we know, several studies have tried to solve this problem in different ways. Urien20 tries to provide security of distributed system by TLS/DTLS stack, Link Key, and WPA. Khalifa et al.21 presented trust requirement model of block chain system. Samaniego and Deters22 attempts to settle trust problem in block chain IoT system by means of Zero-Trust.23 This theory was first proposed by Google Beyond Corp, which is featured with context aware and dynamic access control. It evaluates device by its dynamic behaviors instead of static certificates. Chen et al.24 attempted to secure block chain information between communication terminals in form of encryption. Although these schemes have adopted different methods to address a security decentralized system, yet they have not addressed the question that the contents before uploaded to decentralized platform cannot be trusted.

**Architecture of trusted autonomy security system**

The diversity of IoT devices’ architectures and computing resources make them almost impossible to apply a unified security framework. Fortunately, IoT gateways construct network domains for IoT devices to access, and they are also the direct manager to manage IoT devices. IoT gateways could have functions such as network mandatory access control and device status detection. In TASS architecture, all IoT devices gains access to network by gateways. And gateways could connect to each other by means of flexible network protocol like Mesh25 or wired network. In traditional network topology, IoT gateways need to use centralized management center to manage their security. But in TASS, IoT gateways connected with each other in a decentralized way. In this network, each IoT gateway manages and evaluates IoT devices that have accessed to it. And security information will be shared among gateways in the whole decentralized system. In Figure 2, we presented the specific architecture of TASS in gateways.

In our design, TASS is composed of two components. The first component is trust anchor, which provides trusted endorsement for TASS both in booting time and runtime. The concept of trust anchor is derived from trusted computing technique, but with different details. In TASS, trust anchor not only provide root of trust for measurement and storage, but also provide root of trust for transfer. In decentralized systems, trust information need to be shared in all trust nodes to compose a “trust alliance.” This function makes trust anchor a unified logic domain that can be relied by decentralized system like TASS.

The second component is decentralized security autonomy platform, which is used to provide decentralized architecture and security management. Decentralized security autonomy system is in application level, it maintains blockchain subsystem and security management subsystem. Blockchain maintains a consortium chain to store and distribute security information.
information during TASS’s lifecycle. Security management focuses on access devices that attached to each gateway. IoT devices’ network status and behaviors also will be detected and reported by security management subsystem in real time.

**Trust anchor**

We chose the concept of TPM to construct trust anchor. TPM was first provided by TCG group. It uses a hardware chip as trusted root, and insures system’s booting integrity by trusted measurement, trusted encryption, trusted report and trusted connection. However, TPM has its defect that it cannot meet the security needs of TASS. For example, TASS need runtime trust and transfer trust in gateways to build an entire lifecycle trust system, which cannot be provided by TPM.

In order to fit IoT network’s security requirements, we use trust anchor to replace traditional TPM concept. Trust anchor is not only part of trust chain, but also an intrinsic structure designed in TASS. In trust anchor, boot anchor replaces the original TPM + BIOS trusted root, and runtime anchor make runtime integrity detection available. In addition, we designed trust transfer in trust anchor to make trust information flow in different IoT gateways. Figure 3 shows the logic location of trust anchor in TASS. Trust anchor makes TASS a trusted system with distributed trusted nodes.

**Boot anchor**

In traditional concept, a static trust chain generated by TPM can represent the trust of system. But in IoT network, there has been a flaw for static trust chain. In TPM, system’s trusted root is derived from CRTM (Core Root of Trust for Measurement), which is composed of TPM and first part of BIOS. IoT system usually uses Uboot as the initial boot program, which can be easily programmed or tampered through serial port in motherboard. Hence, for a higher security purpose, we use TPCM (Trusted Platform Control Module) as trusted root to instead TPM. TPCM changes IoT device motherboard’s start-up sequence to make sure the hardware trusted computing chip is the first equipment to be powered up. It avoids software like BIOS as part of CRTM, make booting period totally depend on trusted computing chip.

We use TPCM to store HASH of Uboot, and based on electronic encryption, this digest cannot be modified by serial port or other programming methods. Also, TPCM have a control module to prevent CPU from powering up at the beginning. When IoT device is powered up, TPCM will execute self-test first.

Specific steps of boot anchor’s establishment are shown in Figure 4. In step 1, reference value of Uboot will be loaded to TPCM’s nonvolatile store. Then, IoT gateway can powered up. Step 2, controller of TPCM, makes CPU keep power-off status. TPCM verifies the image of Uboot in NOR Flash. Step 3, if the measurement result of Uboot equals to reference value, TPCM

![Figure 3. Trust anchor in trusted autonomy security system.](image-url)

![Figure 4. Start-up sequence of boot anchor.](image-url)
will handle its control right to CPU of this device, which is step 4. In step 5, CPU acquires control of motherboard and launches Uboot image. Next, steps of measurement will be executed in sequence.

In boot anchor, we constructed a strict trust chain of IoT gateway. In IoT environment, there is a possibility that IoT gateway can be accessed by hacker in physical. If there is no security mechanism to protect gateway’s storage like Uboot and kernel image, trusted root in it can be tampered easily. But boot anchor guaranteed that even in physical contact or under other advanced attacks, trusted root of gateway can still be trusted. As a consequence, all the data generated by gateways can be uploaded to blockchain as real information.

Runtime anchor

Another reason we use trust anchor is that TPM only guarantees system’s integrity in start-up stage, the runtime stage of system cannot be guaranteed or detected properly. Boot anchor only settled the problem of trusted booting, runtime security of gateway also needs to be protected. Runtime anchor focuses on files change when system is running. It makes sure that during the runtime files like elf, so, and ko will not be tampered.

Runtime anchor is based on IMA (Integrity Measurement Architecture). In the design of IMA (Integrity Measurement Architecture), it records every change of files and upload it to a list called ML (Measurement List). At the same time, IMA provides appraise mode as its control module to prevent files from unauthorized change. However, IMA has no limitations to users with root authority. If gateway leaks its root authority or user privilege escalation attack, the protection of IMA will be inefficient. In order to solve this potential issue, we changed the logic of IMA appraise mode. We modified the judgment logic of IMA to deny file write request from root user. After our modification, any overwrite of ko, elf, config, so, and sh are not available. Files under this decision can also be updated, but the only way is rewriting OS image in motherboard.

In runtime anchor, in order to evaluate every gateway’s runtime security status, we also monitored the runtime information of processes in system. We set hooks in runtime anchor to collect process information in OS kernel. And this information is listed in the following:

- process creation log
- process running list
- file access log
- file update log

AS an IoT device with settled application scenario, gateway in TASS should not have any unexpected behaviors. The process list and file operation log should follow certain rules that can be predicted to all gateways in the same network.

Trust transfer

Figure 5 shows the logic path of trust transfer. Logically, trust is originated from trusted computing chip. Then trust is transferred to boot anchor, then to the runtime anchor, then to the decentralized platform, and finally to the IoT network.

While the physical path of trust transfer in TASS is shown in Figure 6. In this figure, trust anchor is first created by gateways respectively. Then, gateways pass the trust information to decentralized platform. In this platform, trust information like ML and process log need to reach consensus between different gateways. When a gateway tries to participate in TASS, it first needs to use password configured by administrator to join the LAN. Second, it needs to upload its AIK (Attestation Identity Key) pub key to let other anchors know its identity. Then the new gateway will upload its trust information like ML and process information.
signed by AIK private key to let other anchors evaluate its trust status. If the number of agreement nodes is equal or greater than \( f + 1 \) and \( 3f + 1 \leq N \), then a trust consensus is reached. Here \( N \) is the number of total nodes in LAN, \( f \) is the maximum evil node number in LAN. The consensus process is listed below.

1. \( \text{password} \rightarrow \text{LAN} \)
2. \( \text{AIK}_{\text{pub.key}} \rightarrow \text{consensus} \)
3. \( \text{AIK}_{\text{priv.key}} \cdot \text{Sign(PCR + process log)} + \text{ML} \rightarrow \text{consensus} \)

In single trusted computing model, we assume the probability of trust failure is \( p \). In TASS, we use PBFT as decentralized consensus algorithm. This algorithm will be invalid when 1/3 or more nodes are compromised. This means only one-third or more gateways in TASS invalid could cause compromise of TASS. As a consequence, the probability of trust failure in TASS is 

\[
p^3
\]

It is obvious that TASS is more secure and stable than single trust model because the risk is separated by decentralized system. The combination of trusted computing and decentralization could form a complete trusted chain from device to decentralized platform and IoT network. This benefit ensures that contents in IoT network are trusted and could map IoT status directly. However, decentralized architecture can also reduce the influence of single trusted node’s invalidation. Based on this analysis, TASS can not only have the flexibility of networking and deployment but also has higher security and credibility.

### Autonomy

Autonomy system in TASS has two modules: blockchain platform and security management. In which, security management is in charge of access side device management, and blockchain is used to provide decentralized platform and consensus algorithm. These two parts make IoT gateways achieve the function to manage IoT devices in a distributed way.

#### Blockchain to decentralize

Block chain is the mechanism to realize decentralized management and consensus. In section “Trust transfer,” we already introduced blockchain to run trust transfer. From the perspective of TASS system, blockchain is application layer software. All the gateways in TASS maintain this consortium blockchain system. Gateways use blockchain system to reach consensus and transfer trust information and security management information.

In Figure 7, there are two kinds of transactions: trust anchor transaction and security management transaction. Local data of trust information and security management information in IoT gateways will be uploaded to blockchain. And chain code will also monitor blockchain to accept information from other gateways. In blockchain system, every transaction of trust and security will be recorded in blocks. Gateways in TASS could use these blocks to evaluate IoT network and devices’ security status. With the advantage of decentralized consensus of blockchain, these blocks cannot be tampered by any single node.

---

**Figure 7.** Architecture of blockchain system.
Security management

Since the purpose of TASS is to form an IoT network and manage IoT devices, it is necessary to address how to monitor IoT devices that have accessed to TASS. Usually, hardware information like MAC and certificates are the features to evaluate access side device’s identity and security status. But without trust anchor in access side devices, this information may be tampered or at least untrusted. In addition, the heterogeneous of access side devices makes it difficult to deploy unified security architecture inside.

An appropriate way to manage access side device is dynamic awareness, which could evaluates device by its dynamic network behaviors. In TASS, dynamic awareness evaluation can be implemented by analyzing access device’s network flow. And according to security strategy requirements, there are two way to implement this dynamic analysis: enforce evaluation and flex evaluation. Enforce evaluation is similar to whitelist mode. In this mode, only flows in which target addresses and URLs have been permitted by TASS are allowed to access and transmit in gateways. This mode is simple to deploy that administrator only need to list out whitelist behavior. Also this mode is too strict to forbid any unexpected behavior.

Flex evaluation is similar to blacklist mode, it only estimate if an action or access is forbidden by blacklist in TASS. Obviously, it is impossible to enumerate all malicious addresses in one local blacklist file. A feasible method is to create a network behavior analysis model to estimate if the behavior of access side device is legal, and machine learning is an appropriate method. For example, KNN, SVM, and LOF can be used to train a behavior model to judge whether the access side device is abnormal and infer the security state of the device later. This approach reduced the scale of blacklist in TASS and is feasible in lightweight IoT gateways. The disadvantage is that the application scenarios and network behavior of access side device need to be pre-collected and trained. When application scenario changes, network behavior analysis model needs to be updated to fit new network behavior. Also, precision of flex model is not as accurate as traditional blacklist model.

Evaluation of TASS

Based on the description above in this article, we made a prototype of TASS as shown in Figure 8. In our system, TASS gateway is based on the hardware of MTK7621, the detail parameter of which is shown in Table 1. Meanwhile, the start-up sequence of this motherboard is modified to deploy TPCM module to support trust anchor. As for blockchain system, we use hyperledger fabric and PBFT algorithm to realize decentralization and consensus. And security management is set up as enforce mode. Also, we simulated several virtual gateways to communicate with the physical one to evaluate the performance of TASS.

Figure 8. Prototype of TASS gateway.
TASS: Trusted Autonomy Security System.

Table 1. Information of MTK7621.

| Information of MTK7621. |
|-------------------------|
| MTK7621 SoC            |
| Operating system       | OpenWrt |
| CPU                    | 880 MHz dual-core MIPS 1004KEc |
| RAM                    | 500 MB |
| EMMC                   | 4 GB |
| TPCM                   | Nation Technology |

TPCM: Trusted Platform Control Module.

decentralization and consensus. And security management is set up as enforce mode. Also, we simulated several virtual gateways to communicate with the physical one to evaluate the performance of TASS.

Trust anchor performance

Trust anchor is the fundamental of TASS’s security. When gateway is installed with trust anchor, both its booting time and response time will be affected. Figure 9 shows the comparison of booting time between boot anchor and regular motherboard. The average booting time of motherboard with trust anchor is 153 s, which is 60 s longer than regular architecture. The main reason of this delay is that the trusted computing module needs to measure the image of Uboot and establish an integrated trusted chain. Fortunately, gateway is not a device that need to boot frequently, which eased the impact of this delay.
The influence of runtime anchor is another factor that may reduce gateway’s performance. In this customized gateway, we use applications in BusyBox to test the latency of runtime anchor. The result shows that the average delay of runtime anchor is about 3%, which is a slight amount in overall performance. We also compared CPU and memory usage between original gateway and customized gateway. The original CPU usage rate is 10%, and TASS increased 1% of CPU usage rate. At the same time, memory usage increased 12 MB on the basis of 80MB. To sum up, the influence of trust anchor is limited for gateway.

Decentralized security autonomy platform performance

In order to reach consensus by PBFT, there are five steps need to be executed: request, pre-prepare, prepare, commit, and reply. Once transaction is triggered, it will be uploaded to blockchain after consensus is reached. We use one TASS prototype and certain QEMU virtual machines to measure the autonomy performance. The QEMU virtual machines load TASS image and run as virtual gateways in host OS. In order to improve the accuracy of performance evaluation, the configurations of QEMU virtual machines are the same as our prototype. And the monitoring point of transaction time is set in TASS prototype to record the time span from request to replay. The specific results of consensus time are shown in Figure 10.

Figure 9. Booting time comparison.

![Booting time comparison](image)

Figure 10. Consensus time in different node. (a) Node Num: 5. (b) Node Num: 7. (c) Node Num: 9.
When there are five gateways in TASS, the average time of consensus is 322 ms, and the standard deviation is 174 ms. When there are seven gateways in TASS, the average time of consensus is 342 ms, and the standard deviation is 155 ms. When there are nine gateways in TASS, the average time of consensus is 413 ms, and the standard deviation is 185 ms. The comparison result shows that the consensus time grows along with the node number. In fact, the time complexity of PBFT is $O(N^2)$.

Based on this time complexity and data fitting process, we can make an estimate of performance trend. The detail is shown in Figure 11. And data fitting function is listed in the following

$$y = 6.375x^2 - 66.5x + 495$$

It indicates that the consensus time of TASS will increase to almost 60 s when there are 100 nodes inside. This delay cannot be accepted in most IoT security systems. This also suggests that in PBFT algorithm there should better have less than 100 nodes. Fortunately, this defect could be solved by the architecture of hierarchy. Hierarchy could separate nodes in different levels like multi-way tree to downsize the consensus scale, and we will research this issue in our future work. To sum up, when nodes number is in a relative small number like 20, the hardware of MTK7621 could form an IoT LAN featured by TASS appropriately.

**Conclusion**

In this work, we proposed the concept of intrinsic security of IoT network. In order to resist network attacks like DDoS and system intrusion, this concept insists that security system must be trusted and autonomous. And based on this idea, we designed a decentralized security architecture TASS, which uses trusted computing technique to create security anchor and establish trust basis, uses decentralized algorithm to transmit security information and operate security management.

It is obvious that TASS has certain advantages compared to traditional system: trusted, decentralization, easy to deploy.

Based on this prototype of TASS, we will conduct more research about the theory like hierarchical consensus and application of intrinsic security, like in the NFV network, deterministic network, and industrial Internet. We believe that this security concept and architecture will have more progress and application in the near future.

**Acknowledgements**

The authors would like to thank the colleagues in security technology department of china mobile research institute for their technical support and consultation.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) received no financial support for the research, authorship, and/or publication of this article.

**ORCID iD**

Penglin Yang https://orcid.org/0000-0003-2547-7297

**References**

1. Gupta Brij B (ed.). *Computer and cyber security: principles, algorithms, applications, and perspectives*. Boca Raton, FL: CRC Press, 2018.

2. Abhishta A, Van Rijswijk- Deij R and Nieuwenhuis LJM. Measuring the impact of a successful DDoS attack on the customer behaviour of managed DNS service providers. *ACM SIGCOMM Comput Commun Rev* 2019; 48(5): 70–76.

3. Kolias C, Kambourakis G, Stavrou A, et al. DDoS in the IoT: Mirai and other botnets. *Computer* 2017; 50(7): 80–84.

4. Nili H, Adam GC, Hoskins B, et al. Hardware-intrinsic security primitives enabled by analogue state and non-linear conductance variations in integrated memristors. *Nat Electron* 2018; 1(3): 197–202.

5. Crosby M, Pattanayak P, Verma S, et al. Blockchain technology: beyond bitcoin. *Appl Innov* 2016; 2(6–10): 71.

6. Papadodimas G, Palaiokrasas G, Litke A, et al. Implementation of smart contracts for blockchain based IoT applications. In: 2018 9th International conference on the network of the future (NOF), Poznan, 19–21 November 2018, pp.60-67. New York: IEEE.

7. Malik N, Nanda P, Arora A, et al. Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks. In: 2018 17th
IEEE international conference on trust, security and privacy in computing and communications: 12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE), New York, 1–3 August 2018, pp.674–679. New York: IEEE.

8. Islam MN, Patii VC and Kundu S. On IC traceability via blockchain. In: 2018 International symposium on VLSI design, automation and test (VLSI-DAT), Hsinchu, Taiwan, 16–19 April 2018, pp.1–4. New York: IEEE.

9. Riesco R, Larriva-Nov o X and Villagra VA. Cybersecurity threat intelligence knowledge exchange based on blockchain. *Telecommun Syst* 2020; 73(2): 259–288.

10. Shi W. On design of a trusted software base with support of TPCM. In: International conference on trusted systems, Beijing, China, 17–19 December 2009, pp.1–15. Berlin: Springer.

11. Sailer R, Zhang X, Jaeger T, et al. Design and Implementation of a TCG-based Integrity Measurement Architecture. In: *SSYM’04: Proceedings of the 13th conference on USENIX security symposium*, vol. 13, San Diego, CA, 9–13 August 2004, pp.223–238. Berkeley, CA: USENIX Association.

12. Xu G, Cao Y, Ren Y, et al. Network security situation awareness based on semantic ontology and user-defined rules for Internet of Things. *IEEE Access* 2017; 5: 21046–21056.

13. Ongaro D and Ousterhout J. In search of an understandable consensus algorithm. In: *USENIX ATC’14: Proceedings of the 2014 USENIX conference on USENIX annual technical conference*, Philadelphia, PA, 19–20 June 2014, pp.305–319. Berkeley, CA: USENIX Association.

14. Mingxiao D, Xiaofeng M, Zhe Z, et al. A review on consensus algorithm of blockchain. In: 2017 IEEE international conference on systems, man, and cybernetics *(SMC)*, Banff, AB, Canada, 5–8 October 2017, pp.2567–2572. New York: IEEE.

15. Fuchs A, Krauß C and Repp J. Advanced remote firmware upgrades using TPM 2.0. In: *IFIP international conference on ICT systems security and privacy protection*, Ghent, 30 May–1 June 2016, pp.276–289. Cham: Springer.

16. Costan V and Devadas S. Intel SGX explained. *IACR Cryptol* 2016; 2016(86): 1–118.

17. Jäger L, Petri R and Fuchs A. Rolling dice: lightweight remote attestation for cots IoT hardware. In: *Proceedings of the 12th International Conference on Availability, Reliability and Security*, Reggio Calabria, 29 August–1 September 2017, pp.1–8. New York: ACM.

18. Pinto S and Santos N. Demystifying arm trustzone: a comprehensive survey. *ACM Comput Surv* 2019; 51(6): 1–36.

19. Tewari A and Gupta BB. Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework. *Future Gener Comput Syst* 2018; 108: 909–920.

20. Urien P. Introducing TLS/DTLS secure access modules for IoT frameworks: concepts and experiments. In: 2017 IEEE symposium on computers and communications *(ISCC)*, Heraklion, 3–6 July 2017. New York: IEEE.

21. Khalifa D, Madjid NA and Svetinovic D. Trust requirements in blockchain systems: a preliminary study. In: 2019 Sixth International Conference on Software Defined Systems *(SDS)*, Rome, 10–13 June 2019, pp.310–313. New York: IEEE.

22. Samaniego M and Deters R. Zero-trust hierarchical management in IoT. In: 2018 IEEE international congress on Internet of Things *(ICIOT)*, San Francisco, CA, 2–7 July 2018, pp.88–95. New York: IEEE.

23. Kindervag J. Build security into your network’s DNA: the zero trust network architecture. Cambridge, MA: Forrester Research Inc, 2010, pp.1–26.

24. Chen LF, Li YS, Wen H, et al. Block Chain Based Secure Scheme For Mobile Communication. In: 2018 IEEE conference on communications and network security *(CNS)*, Beijing, China, 30 May–1 June 2018, pp.1–2. New York: IEEE.

25. Liu Y, Tong KF, Qiu X, et al. Wireless mesh networks in IoT networks. In: 2017 International workshop on electromagnetics: applications and student innovation competition, London, 30 May–1 June 2017, pp.183–185. New York: IEEE.

26. Gupta Brij B, Quan Z and Sheng (eds). *Machine learning for computer and cyber security: principle, algorithms, and practices*. Boca Raton, FL: CRC Press, 2019.

27. Li Y, Fang B, Guo L, et al. Network anomaly detection based on TCM-KNN algorithm. In: *Proceedings of the 2nd ACM symposium on information, computer and communications security*, Singapore, 20–22 March 2007, pp.13–19. New York: ACM.

28. Schuster F, Paul A, Rietz R, et al. Potentials of using one-class SVM for detecting protocol-specific anomalies in industrial networks. In: 2015 IEEE symposium series on computational intelligence, Cape Town, South Africa, 7–10 December 2015, pp.83–90. New York: IEEE.

29. Paulauskas N and Bagdonas AF. Local outlier factor use for the network flow anomaly detection. *Secur Commun Netw* 2015; 8(18): 4203–4212.

30. Androulaki E, Barger A, Bortnikov V, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains, 2018, https://arxiv.org/abs/1801.10228

31. Sukhwani H, Martinez JM, Chang X, et al. Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric). In: 2017 IEEE 36th Symposium on Reliable Distributed Systems *(SRDS)*, Hong Kong, China, 26–29 September 2017, pp.253–255. New York: IEEE.

32. Mijumbi R, Serrat J, Gorricho JL, et al. Network function virtualization: state-of-the-art and research challenges. *IEEE Commun Surv Tutor* 2016; 18(1): 236–262.

33. Li S, Xu LD and Zhao S. 5G Internet of Things: a survey. *J Ind Inf Integr* 2018; 10: 1–9.

34. Li JQ, Yu FR, Deng G, et al. Industrial Internet: a survey on the enabling technologies, applications, and challenges. *IEEE Commun Surv Tutor* 2017; 19(3): 1504–1526.