# IoTCP: A Novel Trusted Computing Protocol for IoT

Shuangbao (Paul) Wang, Amjad Ali, Ujjwal Guin, Anthony (Tony) Skjellum

*Abstract - The ability to understand, predict, secure and exploit the vast array of heterogeneous network of things is phenomenal. With the ever-increasing threats to cyber physical systems and Internet of Things, security on those networks of data-gathering sensors and systems has become a unique challenge to industries as well as to military in the battlefield. To address those problems, we propose a trusted computing protocol that employs discrete Trusted Platform Modules and Hardware Security Modules for key management, a blockchain-based package verification algorithm for over-the-air security, and a secure authentication mechanism for data communication. The IoT-based Trusted Computing Protocol implements integrated hardware security, strong cryptographic hash functions, and peer-based blockchain trust management. We have tested the protocol under various circumstances where devices have built-in securities while others do not. We apply the new protocol to a SCADA system that contains more than 3,000 edge devices. The preliminary results show that proposed protocol establishes trust, improves security, integrity, and privacy.*

**Keywords**

*Trusted Computing, IoT, IoBT, Blockchain, TPM*

## 1 INTRODUCTION

The Internet of Things (IoT) is fed with billions of devices and trillions of sensors. IoT networks undergird critical infrastructure, such as electric grids, transportation hubs, and nuclear power plants. They also link to systems containing valuable and sensitive personal information, such as hospitals, schools, and government institutions. A failure in of one of these systems or a cascade of such failures across systems, either in their operations or security, could lead to potentially catastrophic consequences for the population of that region, city and beyond [15, 29]. Yet many of the hardware and software elements used to control, monitor, and connect these systems are not designed with built-in security, while others are outmoded and may not interface with newer technologies. For this reason, every IoT project must address the security and trust, and that can be hardened against tempering and compromise [30]. The ideal IoT, Internet of Battlefield Things (IoBT), and Cyber Physical Systems (CPS) should be able to continue operating under attacks, and provide guaranteed performance [25–28]. Therefore, it is critical to secure edge devices and actuators, and make IoT networks resilient in the face of cybersecurity threats [1, 2, 10, 30].

To ensuring the security and authenticity of IoT infrastructure is challenging since the edge devices have been manufactured in an environment with limited trust and government oversight [30]. These devices often have resource constraints such as low power requirements, low area budget, limited memory, and/or extremely low-cost. The attacks can originate either from the hardware or the software. Hardware attacks against a system can occur with physical tampering of an edge device and/or by the introduction of a cloned/counterfeit device into the IoT system [31-35]. On the other hand, software attacks against the system can be performed through network attacks, such as Phishing, Denial of Service (DoS), and data spoofing [36-37].

Trusted Computing (TC) is to guarantee the trustworthy of IT systems [8, 9]. It has shown promises in research, industrial projects [22], and military implementations. Practically, deploying a large system with Trusted Computing

Base (TCB) is still a challenge especially in IoT systems, where various protocols and legacy systems exist.

This paper first discusses security issues in IoT/CPS systems and then proposes a novel IoT-based Trusted Computing Protocol (IoTCP) that integrates hardware security, strong cryptographic hash functions, and blockchain technology to establish trust among devices and to secure data communications. We apply machine learning and intelligent services to deliver adaptive cyber-physical capability and services necessary to enable effective command and control across IoT/CPS systems. Finally, we illustrate an implementation of a testbed SCADA system with a dashboard and data analytics features.

## 2 BACKGROUND

Trusted computing is to build digital identities and increase assurance of trust among devices that are connected to the network. It adds a level of security on top of what is provided by the operating systems and hardware. Trusted computing base adds a hardware device, which has non-volatile storage and cryptographic execution engines on each device.

Trusted Platform Modules (TPM) and Hardware Security Modules (HSM) are cryptographic hardware that improve the overall security of a system. TPM is usually embedded into a device. For a device that do not come with a TPM, an HSM can be added.

Message Queue Telemetry Transport (MQTT) [12] is a device-to-device IoT connectivity protocol. It has the advantages of small code footprint, low network bandwidth requirement, and lightweight therefore a good candidate especially for IoT/CPS systems.

A blockchain is a distributed ledger [13]. For IoT/IoBT/CPS systems, it records all actions and operations [6]. Each block contains the data and a hash of a previous block. If any of the previous block is tempered with, it will affect the hash of the current block. The temper-proof property makes it a perfect candidate to guarantee packages authenticity for over the air updates. Guin et al. recently proposed to integrate blockchain technology to authenticate resource-constrained, low-cost edge devices [38]. SRAM-based physically unclonable functions (PUFs) were used to generate a unique "digital fingerprints" to identify edge devices.

### 2.1 Trusted Platform Modules

A TPM is a robust chip that is integrated into the systems providing hardware security and establishing trust within devices such as computers, weapons, vehicles, robots, wearables, actuators, and storage [5]. TPM enables server, gateways, and sensors to extend secure authentication [8, 11] and integrity with a TPM chip on each device. Mutual authentication of devices is required at session start and signing; and decipher are performed on the device [4].

TPMs are basic building blocks used in many specifications, for providing an anchor of trust. They can be used for validating basic boot properties before allowing network access, or for storing platform measurements, or for providing self-measurement to provide anchors of trust to hypervisors (in virtualization). The TPM 2.0 Profile Specification allows subsets of proven security to be implemented in a variety of devices, from traditional clients to embedded and IoT systems, with smaller footprints, lower power consumption, and lower cost [16].

### 2.2 TPM-based Security Systems

Bosch uses TPM in its cameras and other edge devices that act like a cryptographic coprocessor to the device. The TPM runs its own firmware, which is continuously maintained to provide optimal protection against possible threats known to the threat intelligent sensors [18]. Communication between the device firmware and the TPM chip happens via a secure agent inside the TPM. TPMs provide application program interfaces and commands for applications. It is impossible for the firmware or operating systems (OS) to modify anything inside the TPM directly.

Chrome OS uses TPMs for a variety of tasks, including software and firmware rollback prevention, protection of user data encryption keys, and attestation of the mode of a device [7].

Microsoft Windows OS uses TPMs to offer features such as disk encryption, virtual smart cards, and device health attestations [14]. During the start-up process, the TPM releases the decryption keys only after comparing a hash of the OS configuration values with a snapshot taken earlier. This verifies the integrity of the Windows OS start-up process.

Virtual smart cards use TPM to emulate the functionalities of physical smart cards, rather than requiring the use of a separate physical smart card and reader. Virtual smart cards are created in the TPM and offer similar properties to physical smart cards. Their keys are not exportable from the TPM, and the cryptographic component is isolated from the rest of the system [7, 14].

### 2.3 Hardware Security Modules

A hardware security module (HSM) is a hardware appliance that provides secure key storage and cryptographic operations within a tamper-resistant hardware module [1]. HSMs provide both logical and physical protection to those devices, including cryptographic keys, from non-authorized users and potential adversaries. The cryptographic function handled by most HSMs are asymmetric key pairs (and certificates) used in public-key cryptography [3, 19]. Some HSMs can also handle symmetric keys and other arbitrary data.

### 2.4 Challenges with TPMs and HSMs

Though TPM provides a good level of security, abusing remote validation by manufacturers may decide what software would be allowed to run. In addition, the user actions may be recorded in a proprietary database without the user actually knowing. This has happened in smart TVs, smart toys, and other voice-activated devices. As a result, privacy becomes an issue. Modern HSMs allow disabling certain functions in a lock-down mode to improve security. However, an attack may extract a number of bits from a secret key and use the short key to launch a brute force attack (CVE-2015-5464). Hash Message Authentication Code (HMAC) obfuscates private keys makes it hard to steal. However incidents such as replay attacks may still occur, due to the fact that the hash value is a constant over time. More cryptographic features must be added to avoid such attacks.

## 3 DESIGN OF A SECURE TRUSTED COMPUTING PROTOCOL

IoT devices such as IP cameras are the most exposed devices facing the most threats. Besides cyber threats, data can be hacked and stolen. Such might happen as the ultimate attempt by an attacker to retrieve certificates and keys to later-on simulate an edge device by his own equipment, trying to hack deeper into the IoT systems [3].

To build trust and secure communications, we propose a novel protocol that uses discrete TPM (dTPM) and HSM for establishing trust and key management. We use HMAC to generate strong cryptographic key and use MQTT protocol for authentication and data communication.

### 3.1 Key Management with Discrete TPM

A discrete TPM is an isolated, separate feature chip that all necessary computing resources are contained within the discrete chip package. A discrete TPM has full control of dedicated internal resource including RAM, non-volatile memory, and cryptographic logic. Due to the architecture, vulnerabilities exist.

A device without a TPM must store private keys in its file system, where it might reside in an especially encrypted file but the key to this must also be stored somewhere in the file system. If hacking into a device's certificate store does not reveal what is being looked for, a side-channel attack may do, such an attack uses analytic hardware equipment to listen to the data of the system while performing its tasks. When triggering the authentication process, at some point, the key will appear unencrypted. This leaves vulnerabilities to attackers.

IoT systems consist of a network of devices, gateways, and sensors. Many are not security-enabled, which run on a variety of old protocols. Since dTPMs are only integrated into some devices, those do not have built-in dTPMs are not able to use the security protocol. To solve this problem, we use HSM to provide hardware security.

### 3.2 Package Verification with Blockchain

Private keys, if loaded with a certificate, are stored inside the TPM and then are no longer retrievable. They can then only be used through cryptographic operations provided by the TPM, specifically its secure agents. The private key is password protected and is secret until safe storage within the TPM. The encryption engine provides key handling support for symmetrical encryption such as AES with up to 256-bit key length. Once the key is delivered, the AES encryption or decryption for communication or over payload is then done by the hardware accelerated encryption engine in the main CPU.

We implemented a Multi-Factor Package Verification algorithm (MFPV) for private management and over-the-air update security. MFPV has a secure cryptographic hash (H) that hashes a message along with the key. It is computationally difficult to create the same hash without the key. The original message with its hash value can only be verified at an edge device with the same key. Note that the hash is computed twice in order to resist some forms of cryptologic analysis such as the birthday attack or Nostradamus attack [17]. Software packages are hashed first on the gateways before sending to the edge devices. When a package is pushed to an edge device, the device broadcasts a ledger to form a blockchain to guarantee authenticity and to prevent modification. The detailed package verification steps are as follows:

Step 1. A hash is computed using MFPV algorithm using the equation:

*MFPV (H, key, message) = H(APPEND(message, H(APPEND(key, message))* where, key is a unique ID on an edge device. The transmitting message to be delivered as follows:

*APPEND(message, MFPV (SHA3, key, message))*

For SHA3-512, we can obtain equivalent security of 256 bits. Here, the block size is 576 bits with unlimited message size. HMAC with SHA3 [19] has been mathematically proven strong in counter attacks. However, data leaks could still happen if a gateway is compromised.

Step 2. Each edge device serves as a sensor by broadcasting to the network notifying a new package has been published and verified by the device. The verification can be performed as follows:

*MFPV (APPEND(message, MFPV (APPEND(key, message))*

where, key is obtained directly from the Non-Volatile Memory (NVRAM). The information (ledger) each device broadcasts forms a blockchain, which is resistant to package modification.

Step 3. After a blockchain is verified, a confirmation is sent out to indicate the package is trusted.

Step 4. With both a correct hash value and the confirmation from peers in the blockchain, the edge device starts the OTA update.

### 3.3 Light-weight Secure Authentication

For secure communication between edge devices and actuators, we use MQTT, a dTPM and a device SDK - client libraries to connect, authenticate, and exchange message between devices and an actuator. MQTT enables low power usage, minimized data packets, small code footprint, and, most importantly, low network bandwidth. Its advantages make it a perfect candidate for sensor communication and mobile applications. As MQTT supports TLS, mutual authentication can be implemented with strong encryption. MQTT consists of three main components: MQTT broker, MQTT subscriber, and MQTT publisher [12].

We also use device shadows to replicate all connected devices such that each edge device has an identical shadow stored in the cloud. The device and its corresponding shadow are constantly compared making sure the integrity of data and interactions. In the event of a security breach, the monitoring agent will issue an alarm indicating there is a conflict of states between the edge device and the corresponding shadow. An action must be taken to resume communication. Adding a device shadow can assure that a device can still interact with applications even when they are offline. The combination of MQTT and device shadow not only secures authentication and communication but also improves fault tolerance.

Figure 1 depicts the handshake process before a device is authenticated. The MQTT Broker can accept or reject the connection based on device authentication results. For secure communication, device authorization is necessary. The MQTT Broker checks the authorization policy to determine whether the device is authorized to publish/subscribe. When both authentication and authorization conditions are met, the device establishes its session to start communication.

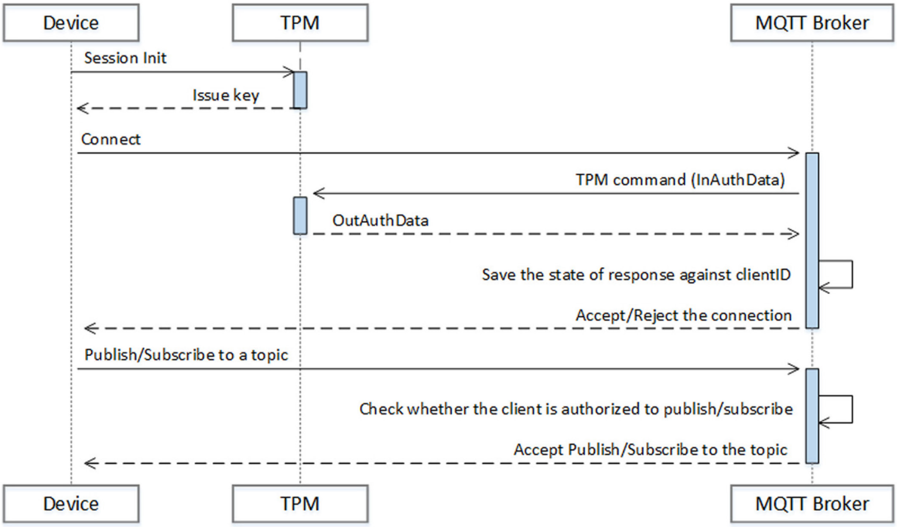
Device
TPM
MQTT Broker
Session Init
Issue key
Connect
TPM command (InAuthData)
OutAuthData
Save the state of response against clientID
Accept/Reject the connection
Publish/Subscribe to a topic
Check whether the client is authorized to publish/subscribe
Accept Publish/Subscribe to the topic
Device
TPM
MQTT Broker

*Figure 1: Secure Edge Device Authentication Protocol*

## 4 CASE STUDY AND PRELIMINARY RESULTS

Using the newly proposed IoTCP protocol, we implemented on a water treatment SCADA system that has more than 3,000 PLCs [21, 24]. The data acquisition layer uses US Department of Defense developed API to collect data [20, 21]. The defense-in-depth architecture reduces the risk to CPS networks from being hacked or data from being stolen. We connect devices that are equipped with TPMs directly to the actuators using the newly developed security communication protocol. We add HSM modules to legacy devices so that they can connect with gateways with security and trust. The blockchain based package verification algorithm ensures OTA security. Secure authentication with MQTT provides low bandwidth requirement and enhanced data security. Initial tests show that the proposed protocol has the advantages of easy connecting with various devices and reducing the risks of cyber intrusions, as a result of the lightweight and low bandwidth protocol. dTPM not only establishes trust between devices and actuators but also provides security for OTA updates. Figure 2 is a diagram of the water treatment SCADA system.

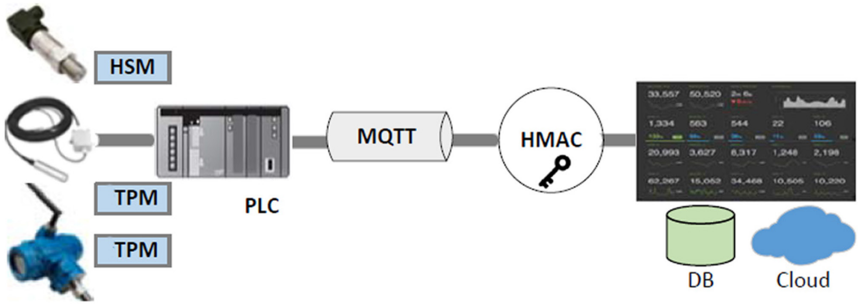

*Figure 2: A Water Treatment SCADA System Diagram*

IoTCP integrates dTPM and HSM to provide hardware level security, uses blockchain to improve OTA security, and uses MQTT with HMAT key scheme to secure authentication and communications.

We utilized the research project as a case study in our master-level cybersecurity courses [23, 29]. Students were divided into four teams. One team focused on overall security assessment and tested the IoTCP against a number of security assessment tools. The team especially focused on replay attack and remote rollback attack. The second team focused on blockchain and MFPV package verification algorithm. The third team worked on secure communication, it verified the secure authentication protocol. The forth team built a dashboard to monitor and display the sensor data, service status and other data analytics results.

## 5 CONCLUSION

IoT, IoBT, and CPS networks connect home appliances, sensors, traffic, vehicles, medical aids, weapons, smart grids, and industrial automation. The heterogeneous collection of microcontrollers, sensors, data interfaces and networks makes it difficult to establish trust and secure the data communication. IoTCP combines dTPM and HSM to secure key management, uses blockchain for package verification and OTA update security, and uses a low bandwidth and lightweight protocol - MQTT to ensure authentication and communication security. Preliminary tests show that IoTCP is especially useful for connecting heterogeneous edge devices including legacy protocols. The device shadows add another level of security, enhance integrity and improve privacy. We are in the process of applying machine learning in modeling device behaviors based on the vast amount of xAPI data we gathered. Once the model is trained with acceptable accuracy, we will use it to predict whether a device can be trusted or actions have to be taken.

## 6 ACKNOWLEDGEMENT

This research is funded in part by grants from US National Science Foundation (NSF) [EAGER-1419055, DGE-1439570, 1821926, and 1755733] and National Security Agency (NSA) [S-004-2017].

## REFERENCES

[1] HP Study Reveals 70 Percent of Internet of Things Devices Vulnerable to Attack. (2014). Retrieved 3/1/2017 from http: //www8.hp.com/us/en/hp-news/press-release.html?id=1744676#.WLd1-ThF04c

[2] The Internet Things Reference Model. (2014). Retrieved 3/4/2017 from http://cdn.iotwf.com/resources/71/IoT_Reference_Model_ White_Paper_June_4_2014.pdf

[3] Internet Security Threat Report. (2016). Retrieved 3/4/2017 from https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

[4] Security and Identity for AWS IoT. (2017). Retrieved 3/4/2017 from http://docs.aws.amazon.com/iot/latest/developerguide/iot-security-identity.html

[5] Will Arthur, David Challener, and Kenneth Goldman. 2015. History of the TPM. Apress, Berkeley, CA. 1–5 pages. https://doi.org/10. 1007/978-1-4302-6584-9_1

[6] Francesco Buccafurri, Gianluca Lax, Serena Nicolazzo, and Antonino Nocera. 2017. Overcoming Limits of Blockchain for IoT Applications. In Proceedings of the 12th International Conference on Availability, Reliability and Security (ARES '17). ACM, New York, NY, USA, Article 26, 6 pages. https://doi.org/10.1145/3098954.3098983

[7] Lisa J. K. Durbeck, Peter M. Athanas, and Nicholas J. Macias. 2014. Secure-by-construction Composable Componentry for Network Processing. In Proceedings of the 2014 Symposium and Bootcamp on the Science of Security (HotSoS '14). ACM, New York, NY, USA, Article 27, 2 pages. https://doi.org/10.1145/2600176.2600203

[8] Trusted Computing Group. 2016. Secure the Internet of Things. (2016). Retrieved 3/11/2017 from https://trustedcomputinggroup.org/trusted-computing/

[9] Kylene Hall, Tom Lendacky, and Emily Ratliff. 2005. Trusted Computing and Linux. TCG, 91–110.

[10] Privacy Rights Clearing House. 2017. Data Breaches. (2017). Retrieved 3/7/2017 from https://www.privacyrights.org/data-breaches breaches.

[11] W. Hufstetler, M. Ramos, and P. Wang. 2017. NFC Unlock: Secure Two-Factor Computer Authentication Using NFC. IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2017) (2017), 507 – 510.

[12] MQTT.org. 2017. (2017). http://mqtt.org/ Downloaded March 3, 2017.

[13] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. (2008), 1–9.

[14] Himanshu Raj and et. al. 2016. fTPM: A Firmware-based TPM 2.0 Implementation. (2016), 1–22.

[15] Devarshi Shah, Anthony Skjellum, Jin Wang, and Peter He. 2017. Challenges and Opportunities for IoT-Enabled Cybermanufacturing: Some Initial Findings from an Iot-enabled Manufacturing Technology Testbed. (2017). https://pdfs.semanticscholar.org/9319/fb10f1f3154f982d5baed6ab5919ed63ef18.pdf?_ga=1.6028727.52173338.1488642784 Paper 66.

[16] Jianxiong Shao, Yu Qin, Dengguo Feng, and Weijin Wang. 2015. Formal Analysis of Enhanced Authorization in the TPM 2.0. In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (ASIA CCS '15). ACM, New York, NY, USA, 273–284. https://doi.org/10.1145/2714576.2714610

[17] Mark Stamp. 2005. Information security - principles and practice. Wiley.

[18] Bosch Security Systems. 2016. Trusted Platform Module explained. (2016).

[19] Apostol Vassilev. 2016. Approved Security Functions for FIPS PUB 140-2, Security Requirements for Cryptographic Modules. (2016). Retrieved 3/7/2017 from http://csrc.nist.gov/publications/fips/fips140-2/fips1402annexa.pdf NIST.

[20] Paul Wang and William Kelly. 2015. A Novel Threat Analysis and Risk Mitigation Approach to Prevent Cyber Intrusions. Colloquium for Information System Security Education (CISSE) 3 (2015), 157–174. Issue 1.

[21] Shuangbao Wang. 2016. Dual-Data Defense in Depth Improves SCADA Security. Signal (2016), 42–44. Issue 10.

[22] Shuangbao Wang, Amjad Ali, and William Kelly. 2015. Data Security and Threat Modeling for Smart City Infrastructure. IEEE Cyber Security of Smart Cities, Industrial Control System and Communications (2015), 1–6.

[23] Shuangbao Wang and William Kelly. 2014. inVideo - A Novel Big Data Analytics Tool for Video Data Analytics. IEEE/NIST IT Pro Conference, 1–19. https://doi.org/0.1109/ITPRO.2014.7029303

[24] Shuangbao Wang and William Kelly. 2017. Smart Cities Architecture and Security in Cybersecurity Education. The Colloquium of Information Systems Security Education (CISSE) (2017). Issue 2.

[25] Shuangbao Wang and Robert Ledley. 2007. Modified Neumann Architecture with Micro-OS for Security. CIICT, 303–310.

[26] Shuangbao Wang and Robert S. Ledley. 2006. Connputer - A Framework of Intrusion-Free Secure Computer Architecture. In Proceedings of the 2006 International Conference on Security & Management, SAM 2006, Las Vegas, Nevada, USA, June 26-29, 2006. 220–225.

[27] Shuangbao Wang and Jiayin Zhang. 2014. A Video Data Search Engine for Cyber-Physical Traffic and Security Monitoring Systems. IEEE/ACM Fourth International Conference on Cyber-Physical Systems, 225–226.

[28] Shuangbao Paul Wang and Robert S. Ledley. 2013. Computer Architecture and Security. Wiley.

[29] Zhi-Kai Zhang, Michael Cheng Yi Cho, and Shiuhpyng Shieh. 2015. Emerging Security Threats and Countermeasures in IoT. In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (ASIA CCS '15). ACM, New York, NY, USA, 1–6. https://doi.org/10.1145/2714576.2737091

[30] U. Guin, A. Singh, M. Alam, J. Canedo, and A. Skjellum, "A Secure Low-Cost Edge Device Authentication Scheme for the Internet of Things", in International Conference on VLSI Design, 2018.

[31] M. M. Tehranipoor, U. Guin, and S. Bhunia, "Invasion of the hardware snatchers," IEEE Spectrum, vol. 54, no. 5, pp. 36–41, 2017.

[32] U. Guin, S. Bhunia, D. Forte, and M. Tehranipoor, "SMA: A System-Level Mutual Authentication for Protecting Electronic Hardware and Firmware", IEEE Transactions on Dependable and Secure Computing (TDSC), 2016.

[33] M. M. Tehranipoor, U. Guin, and D. Forte, Counterfeit Integrated Circuits: Detection and Avoidance. Springer, 2015.

[34] U. Guin, K. Huang, D. DiMase, J. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," Proceedings of the IEEE, vol. 102, no. 8, pp. 1207–1228, Aug 2014.

[35] U. Guin, D. DiMase, and M. Tehranipoor, “Counterfeit integrated circuits: Detection, avoidance, and the challenges ahead,” Journal of Electronic Testing, vol. 30, no. 1, pp. 9–23, 2014.

[36] T. Borgohain, U. Kumar, and S. Sanyal, “Survey of security and privacy issues of Internet of things,” arXiv preprint arXiv:1501.02211, 2015.

[37] H. He, C. Maple, T. Watson, A. Tiwari, J. Mehnen, Y. Jin, and B. Gabrys, “The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence,” in Evolutionary Computation (CEC), 2016 IEEE Congress on. IEEE, 2016, pp. 1015–1021.

[38] U. Guin, P. Cui, and A. Skjellum, “Ensuring Proof-of-Authenticity of IoT Edge Devices using Blockchain Technology”, IEEE International Conference on Blockchain, 2018.