Library Progress International
Vol.44 No. 3, Jul-Dec 2024: P. 18229-18235

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

Secure Chip To Chip Communication Based On Zero Trust Architecture In Embedded Systems

Rahul Shandilya[1]*, R.K. Sharma[2]

[1]*Research Scholar, School of VLSI Design and Embedded Systems, NIT Kurukshetra-136119, Haryana, India, rss.nitk@gmail.com
[2]Professor, Dept. of Electronics and Communication Engineering, NIT Kurukshetra-136119, Haryana, India, mail2drks@gmail.com
.

How to cite this article: Rahul Shandilya, R.K. Sharma (2024) Secure Chip To Chip Communication Based On Zero Trust Architecture In Embedded Systems. *Library Progress International,* 44(3), ***18229-18235***.

ABSTRACT

Nowadays, semiconductor companies frequently outsource the production of chips to meet the rising demand for integrated circuits. As a result, the chip supply chain is now dealing with a number of security problems, like hardware intellectual property theft, trojans, and over-production. In critical systems where adversary assaults have the potential to cause large losses or damage, zero-trust offers a promising method for guaranteeing the validity of Integrated Circuits (ICs). A reliable protocol which makes use of certificates to guarantee the legitimacy of ICs is the Security Protocol and Data Model (SPDM). The work under this study presents a secure chip-to-chip (S2C) zero-trust security architecture based on SPDM protocol, which attempts to authenticate any attached peripheral before using it. The contributions include a comprehensive explanation of the proposed design, the SPDM protocol's implementation, and a discussion of the obstacles that were encountered while executing and implementing.

KEYWORDS: *Chip-to-chip communication, Zero-trust Architecture, SPDM, Embedded Systems*

INTRODUCTION

With the development of embedded systems, the backbone of related automobiles, drones, smart homes, industrial control systems, and the Internet of Things (IoT), the globe is getting progressively more interconnected at an staggering rate. Embedded systems usually rely on Integrated Circuits (ICs) that are made in low-cost production zones by third parties. The reliability of manufactured integrated circuits (ICs) or devices is seriously questioned since modern, innovative foundries are viewed as untrustworthy entities in the IC supply chain [24]. When relying on unverified information, one must carefully consider the risks involved. Any IP component of the system, for example, must necessarily be shared with the untrusted foundry. In addition to the well-known dangers of IP theft, overproduction, and reverse engineering, a backdoor or hardware Trojan can be used to alter the same IP [11]. Simultaneously, malicious entities are uncovering increasingly inventive methods to gain access to embedded devices via the software supply chains that produce them.

LITERATURE REVIEW

The emerging threats exceed existing security frameworks by a slight degree. Two existing IoT-specific standards created in order to protect individual IoT devices are the PSA (Platform Security Architecture) [17] as well as the SESIP (Security Evaluation Standard for IoT Platforms) [18]. ARM's PSA project attempts to offer an isolated execution environment based on hardware. In order to build a secure basis for IoT systems, it provides threat models, security analysis, and hardware/firmware standards.

However, in order to make sure IoT platforms fulfill specific security requirements, the SESIP lays out recommendations for doing just that. In order to increase user and stakeholder trust, this standard assists producers and developers in evaluating the security features and resilience of their products. Various strategies are suggested to counteract the risks caused by unreliable manufacturing, including logic locking [5], obfuscation [9], and Trojan detection [22]. These last circuit-level tactics call for circuit alterations that would make it harder for an enemy to decipher intellectual property. According to some writers, chip-to-chip authentication should come first, then split-chip solutions for trustworthy fabrication [10]. At some point, these remedies are unable to offer fully secure systems. Simultaneously, the cybersecurity, as well as silicon sectors, have recently argued in favor of zero-trust architectures to more thoroughly secure distributed infrastructure, particularly with the available generation of open-source hardware, which would undoubtedly offer a significantly larger attack surface with potentially severe physical consequences. The zero-trust principle enhances security throughout semiconductor supply chains [27]. The semiconductor industry wants to implement this idea to prevent any non-self-authenticating device from connecting with system hardware. This suggests that it is necessary to disregard any manipulations that take place in the foundry or throughout the supply chain. Moreover, the future generation of embedded systems may benefit from a more reliable end-to-end security strategy, which might be facilitated by fusing zero-trust security concepts with current embedded systems security techniques [21]. Intel shares its goals and principles for "A ZeroTrust Approach to Architecting Silicon" [28], which lends credibility to this concept.

A framework called DRLGENCERT is shown in [15], demonstrating the application of deep reinforcement learning (DRL) to automation of certificate verification testing. Using conventional certificates as input, DRLGENCERT generates new certificates that can effectively identify discrepancies. This method improves the procedure overall by using DRL to make intelligent decisions during certificate generation based on previous modifications In [14], the MQTTS protocol utilizing SSL/TLS certificates is employed to protect communication between an IoT ESP32 embedded system & IoT cloud. Without disclosing any information about the methods that were employed, the study concentrates on whether the encryption strategy is accurate. In [13], an architecture is put forth that would allow Internet of Things devices to notarize and authenticate data inside the Ethereum blockchain. By creating a strong hardwaresoftware framework that enables lightweight devices, like Internet of Things sensors, to manage this process, the work expands on this idea. These devices, together with their corresponding public address, include a confidential key within this architecture. Transactions are automatically signed and sent to the blockchain network as they are created. A Secure Chip-to-Chip (S2C) Zero-Trust Architecture is being presented in this study to ensure security for communications between two chips and a mechanism for proving the authenticity of peripherals. S2C is based on a zero-trust processor which uses multiple cryptographic engines for increased security to implement the SPDM protocol. The compilation, optimization, as well as testing of SPDM protocol are the main contributions of this effort. Among these contributions are an overview of the architecture, the SPDM protocol's implementation and a thorough examination of the difficulties in its execution and implementation. Moreover, the research includes the experimental realization of SPDM SPI connection using NXP S32G3 devices as platform.

Assuming that devices authorized within the network can be implicitly trusted, conventional security architectures and models frequently rely on a single network architectural solution [1]. Aside from implementation or architectural arrangement, authorization is a crucial component since the resources and architecture of a network dictate the kind of authorization model that is needed. In ZT contexts, authorization systems like as RBAC, PBAC, and ABAC are frequently employed. Continuous authorization guarantees that access is only authorized when required [2].

The remaining part of the paper is organized as follows: The suggested methods are defined in Section 2, which includes both an integrated design execution and a thorough overview. The obtained experimental outcomes are defined and examined in Section 3. Ultimately, Section 4 concludes the paper and proposes opportunities for further research.

Numerous studies have been carried out recently applying scientometric analysis to determine the growth of research production. Aydin (2017) conducted the research on “Research Performance of Higher Education Institutions”, the article intends to raise awareness of "research performance," which plays a crucial role in university competition. The study makes an effort to summarize the findings of a thorough literature evaluation in the area of higher education research performance in order to achieve this goal. First, basic literature on research performance is discussed together with its concept definition and indicators**.** Then, a thorough presentation of the variables affecting research performance followed. The study concludes with the provision of a conceptual framework that will be useful to all university staff.

METHODS

Architecture Overview

An S2C architecture that can guarantee secure communication between two chips—an initiator and a target—has been suggested in our work is shown in Figure 1. It should be mentioned that the target is a representation of an external peripheral, that could be either passive (no processor) or active (integrated processor). In order to exchange data between the target and initiator, the target needs to be verified.

3.1.1 ZTP ("Zero-Trust Processing"):

This feature serves as a link between the initiator and the target and is responsible for implementing the zero-trust procedure. SPDM [26] and PCIE[16] were examined for this purpose, and the SPDM was chosen because it supports interconnects that are non-PCIe.

3.1.2 ZTM ("Zero-Trust Management"):

It evaluates if ZTP is allowed to interact with the peripheral that is connected. This unit makes use of scheduling based on events. Prior to initiating data exchange during the initialization phase, the Target submits its certificate to the ZTM of the Initiator for validation. For example, in order to declare a verification success or failure, zero trust operations need to be recorded and shared.

3.1.3 CMU ("Certificate Management Unit"):

It allows the maintenance of certificates, including their revocation, modifications, and notifications to other chips of certificate changes. 3.1.4 CSS ("Certificate and Secret Storage"): It allows for the efficient and safe storage of private keys and certificates in on-chip flash memory.

The SPI, I2C, CAN, and other interface protocols are the ways that the S2C can connect with external devices. The article aims to manage the newly inserted devices and implement the chosen authentication scheme. We exclusively work on implementing these above modules and integrated into the whole system.

3.2 Zero-Trust Processing Mechanism

3.2.1 Description of the SPDM protocol:

For secure communication between devices via a variety of transport and physical media, the SPDM protocol specifies formats of messaging, data objects, as well as sequences [26]. Cryptographic engines for digital signatures, hashing, and verification are included in the SPDM. Moreover, the SPDM protocol has recently been updated to include postquantum cryptography techniques in order to make it resistant to quantum computing [7].

3.2.2 Formal verification of the SPDM protocol:

Before a security protocol is implemented, it must be validated. A security protocol ought to be included in hardware systems only when it has successfully completed all formal verification testing. A variety of devices and methodologies are presently accessible for formal verification of security protocols. Through the use of AVISPA ("Automatic Verification of Internet Security Protocols and Applications"), this work validates the SPDM protocol for the suggested S2C [12]. The research community uses AVISPA, a pushbutton interface formal verification tool, extensively. Although it is composed of multiple backends, we use an OFMC (On-theFly Model Checker) to verify it. As previously explained, the protocol has two chips: one is initiator and the other is targeted, which are also known as agents. Consequently, the tests are run over several sessions, that is, when one chip is permitted and when the other is not, with the unauthorized chip functioning as a threat. The tool can be used to help a Dolev-Yao intruder, who has total access to the network and is able to intercept any communication. However, the intruder lacks the necessary cryptographic keys to decrypt data. In our model, a chip that is illegal acts as an intruder. The results of the tests show that SPDM is a secure protocol embedded in hardware. refer [6] for a detailed explanation of formal verification procedure used with the SPDM.

3.2.3 Embedded implementation of SPDM protocol:

To give an example, the suggested architecture will be included in a vehicle so that its ECU can use the CAN or SPI interface to interact with other ECU and automotive electronics modules like the steering wheel unit, breaking control unit etc. with the help of Steer-by-wire (SBW) system, which replaces the mechanical linkage with electric wires. An overview of the implemented prototype environment is shown in Figure 3. It is made up of two NXP S32G3 boards that are linked together by the SPI bus; one of them is set up as a target (slave) and the other as an initiator (master).

3.3 Mechanism of the Certificate Management Unit:

The ZTM's many situations for handling plugged-in peripherals in response to a ZTP request are shown in Figure 1. The ZTM mechanism verifies the certificates of the inserted peripherals during the SPDM protocol's initialization phase. ZTM allows ZTP to initiate a communication session by verifying the certificate of a peripheral. This ensures that unauthorized communication between the initiator and even authentic peripherals is prevented. The program administrator can authenticate unknown peripherals by using a certification process that is suggested within the CMU unit to handle this problem. The ZTM sends a certification request to CMU, requesting clearance from a local agent (Domain Validation Certificate: DVC) or a 3rd party (Extended Validation Certificate: EVC). Information about the peripheral, including its Unified Identifier (UID) and production details, should be present on the plugged-in non-certified devices. The relevant certificates are generated and returned by the EVC after receiving this information via the internet. With libraries such as OpenSSL [20] or EmbedTLS [23], a system administrator can do local certification. The key parameter generation, certificate creation, revocation, and update, message digest computation, and signing and validating processes are all made possible by these libraries.

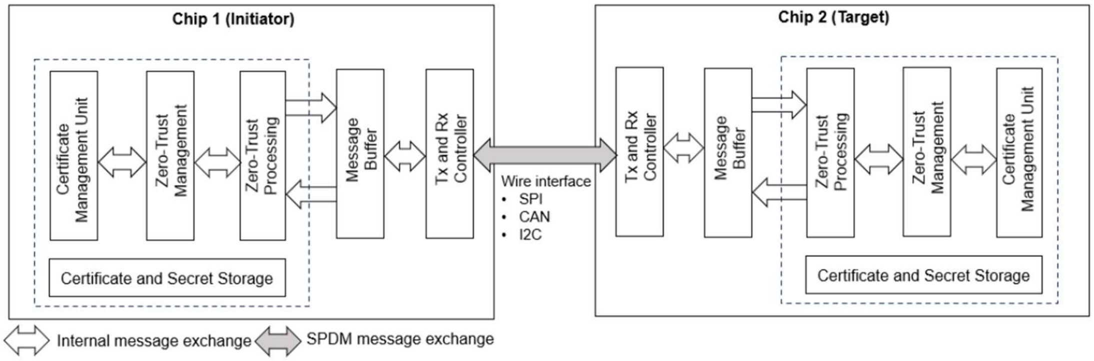

Figure 1. Operational unit in S2C zero-trust architecture.

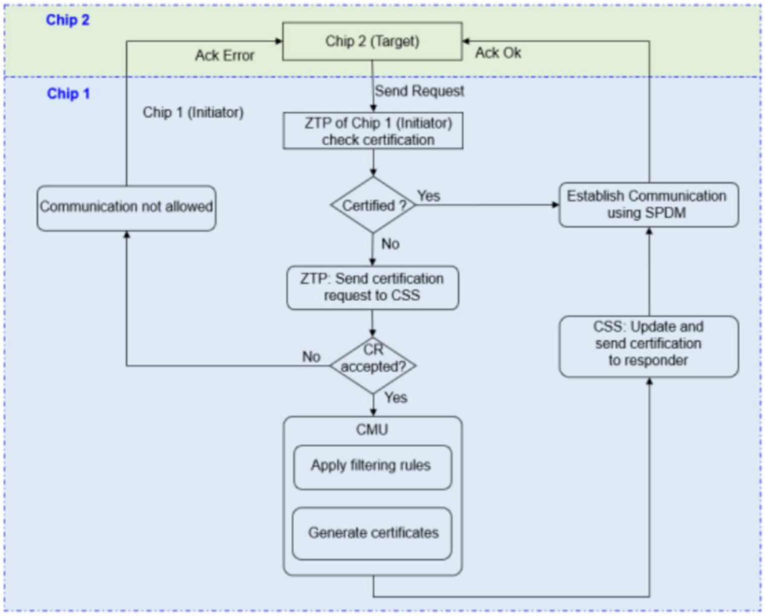
Chip 2
Chip 2 (Target)
Ack Error
Ack Ok
Chip 1
Chip 1 (Initiator)
Send Request
ZTP of Chip 1 (Initiator) check certification
Certified ?
Yes
No
Communication not allowed
Establish Communication using SPDM
ZTP: Send certification request to CSS
CR accepted?
No
Yes
CSS: Update and send certification to responder
CMU
Apply filtering rules
Generate certificates

Figure 2: Certificate management flow diagram

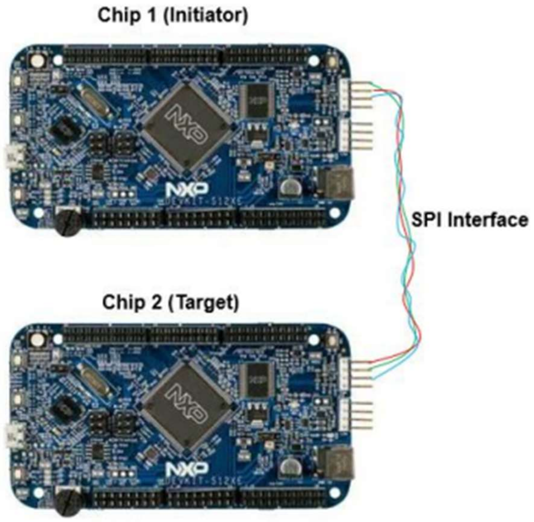

Figure 3: Hardware Prototype

RESULT AND DISCUSSION

Formal Verification of SPDM Protocol

To verify the security of the protocol and its capacity to make sure zero-trust communication between 2 chips, a number of attack scenarios, including replay attacks, were put to the test. The SPDM protocol demonstrates resilience against various threats and facilitates secure communication and authentication, as indicated by AVISPA's findings.

4.2 Performance Evaluation

The first hurdle in implementing SPDM on the NXP S32G3 via SPI connection is the lack of software support for application drivers. Although S32G3 may communicate via several protocols, such as SPI, CAN, and I2C, driver installation is difficult due to the compatibility of variable message buffers between SPI, CAN, and I2C, with differing buffer sizes, Table 1 illustrates the implementation latency at each level of the SPDM protocol. The authentication process takes about 1-2 seconds to complete. The authentication process involves establishing a shared key, exchanging certificates, obtaining user measurements, and starting encrypted connections. The findings demonstrate that a larger buffer has some improvement on latency time because of a hardware does not require to check that buffer is empty or not, by issuing read requests to checks the buffer status during data transfer. The throughput results for various buffer sizes are also summarized in Table 1. It emonstrates that the read and write throughput is considerably reduced with lower buffer sizes. The hardware faces performance drop that occurs when the buffer size is lower, which cause filling of the buffer too frequently which halt the communication during data transfer and wait until it becomes empty by the host. However, this can only be avoided by taking higher size of buffer for read and write accesses. Therefore, every time a master needs to read a single byte, it has to provide 512 read requests in order to clear the buffer. It's important to verify the manufacturer and UID details when it comes to certifying authentic and non-certified peripherals. The system administrator must be aware of peripheral UID cloning, which is a major challenge. Connecting non-certified devices is now the responsibility of the system administrator.

4.3 Benchmarking the Complete Framework The suggested solution is contrasted with existing embedded authentication protocol implementations in Table 2. A number of evaluation criteria are presented, such as the evaluation platform, protocol/model, methodology, security domain, and focus. The proposed program addresses problems including overproduction, hardware trojans, and intellectual property theft while concentrating on protecting the chip supply chain at hardware layer. To reduce the dangers associated with hacked or counterfeit chips, it introduces S2C, which checks the validity of attached peripherals. The SPDM protocol, which was created especially to meet the security needs of the hardware layer, is used by the work for authentication. Using widely available embedded systems, the NXP S32G3 assessment platform, which is on the basis of ARM architecture, shows how feasible it is to implement S2C in practical applications. On the other hand, the studies showcased in

[15]–[13] focus on data certification or certificate generation or verification on the blockchain

Notable security improvements for chip communication are provided by the S2C. However, because S2C may not work with outdated hardware or software, integrating it into current systems or devices may be challenge. Moreover, system performance may be impacted by higher resource usage and authentication latency. These constraints can be controlled by optimizing and designing carefully.

5. Conclusion In this work, we suggested an S2C architecture to implement a secure chip-to-chip communications. The proposed design incorporates SPDM protocol, that allows data communications with integrity as well as confidentiality protection. SPDM has been formally verified using AVISPA tools in order to assess the correctness and attack resilience of the protocol. The protocol is applied on two NXP S32G3 systems as a case study to show that, when the certificates are validated, the two platforms are able to communicate. To create fresh certification credentials for new, authentic, and uncertified peripherals, a certification method is constructed.

In the future, our focus will be on creating an embedded system-on-chip (SoC) that combines the different components of the suggested design, utilizing a RISC-V CPU. In addition, lightweight accelerators cryptographic algorithms will be taken into consideration [25] in order to expedite secure chip-to-chip communication and authentication in the zero trust era.

REFERENCES

1. Liu, Yiliang, et al. "Zero Trust-Based Mobile Network Security Architecture." IEEE Wireless Communications 31.2 (2024): 82-88.
2. Tsai, Mengru, Shanhsin Lee, and Shiuhpyng Winston Shieh. "Strategy for implementing of zero trust architecture." IEEE Transactions on Reliability (2024).
3. Zanasi, Claudio, Silvio Russo, and Michele Colajanni. "Flexible zero trust architecture for the cybersecurity of industrial IoT infrastructures." Ad Hoc Networks 156 (2024): 103414.
4. Stafford, V. "Zero trust architecture." NIST special publication 800 (2020): 207.
5. Zhong, Yadi, and Ujjwal Guin. "Complexity analysis of the SAT attack on logic locking." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 42.10 (2023): 3143-3156.
6. Ahmed, Ashfaq, Abdulhadi Shoufan, and Kais Belwafi. "Light-weight security protocol and data model for chip-to-chip zero-trust." IEEE Access 11 (2023): 60335-60348.
7. Yao, Jiewen, Krystian Matusiewicz, and Vincent Zimmer. "Post quantum design in SPDM for device authentication and key establishment." Cryptography 6.4 (2022): 48.
8. Belwafi, Kais, et al. "Unmanned aerial vehicles' remote identification: A tutorial and survey." IEEE Access 10 (2022): 87577-87601.
9. Rangarajan, Nikhil, et al. "Opening the doors to dynamic camouflaging: Harnessing the power of polymorphic devices." IEEE Transactions on Emerging Topics in Computing 10.1 (2020): 137-156.
10. Karageorgos, Ioannis, et al. "Chip-tochip authentication method based on SRAM PUF and public key cryptography." Journal of Hardware and Systems Security 3 (2019): 382-396.
11. Tehranipoor, Mohammad, and Farinaz Koushanfar. "A survey of hardware trojan taxonomy and detection." IEEE Design & Test of Computers 01 (2016): 1-1.
12. Vigano, Luca. "Automated security protocol analysis with the AVISPA tool." Electronic Notes in Theoretical Computer Science 155 (2006): 61-86.
13. Rafaiani, Giulia, et al. "Implementation of ethereum accounts and transactions on embedded IoT devices." 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS). IEEE, 2022.
14. Nikolov, Neven, and Ognyan Nakov. "Research of secure communication of Esp32 IoT embedded system to. NET core cloud structure using MQTTS SSL/TLS." 2019 IEEE XXVIII International Scientific Conference Electronics (ET). IEEE, 2019.
15. Chen, Chao, et al. "DRLgencert: Deep learning-based automated testing of certificate verification in SSL/TLS implementations." 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2018.
16. Sharma, Debendra Das. "Keynote 1: Compute express link (CXL) changing the game for cloud computing." 2021 IEEE Symposium on HighPerformance Interconnects (HOTI). IEEE, 2021.
17. Jung, Junyoung, Jinsung Cho, and Ben Lee. "A secure platform for iot devices based on arm platform security architecture." 2020 14th International Conference on Ubiquitous Information Management and

Communication (IMCOM). IEEE, 2020.

18. IEEE 2621.1-2022 /UL 2621-1-2022 IEEE/UL Standard for Wireless Diabetes Device Security Assurance Evaluation: Connected Electronic Product Security Evaluation Programs
19. Alshamsi, Sondos, et al. "A low-power remote identification module for drones." 2023 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2023.
20. Walden, James. "OpenSSL 3.0. 0: An exploratory case study." Proceedings of the 19th International Conference on Mining Software Repositories. 2022.
21. Conlon, Chris, and Cesare Garlati. "A new zero-trust model for securing embedded systems." Proceedings of the Embedded World Conference, Nuremberg, Germany. 2019.
22. Jain, Ayush, Ziqi Zhou, and Ujjwal Guin. "Survey of recent developments for hardware trojan detection." 2021 IEEE international symposium on circuits and systems (iscas). IEEE, 2021.
23. Lipp, Moritz, et al. "PLATYPUS: Software-based power side-channel attacks on x86." 2021 IEEE Symposium on Security and Privacy (SP). IEEE, 2021.
24. Bertoni, Guido Marco, and JeanSébastien Coron. Cryptographic Hardware and Embedded SystemsCHES 2013. Springer Berlin Heidelberg, 2013.
25. Laue, Ralf, et al. "Compact AES-based architecture for symmetric encryption, hash function, and random number generation." 2007 International Conference on Field Programmable Logic and Applications. IEEE, 2007.
26. DSP2058: Security Protocol and Data Model (SPDM) Specification, Version 1.2.1, Jun 2022. accessed: 2022-10-09.
27. A. Varas, R. Varadarajan, J. Goodrich, and F. Yinug, "Strengthening the global semiconductor value chain;' Semiconductor Industry Association (SIA)/Boston Consulting Group (BCG), Tech. Rep., April 2021. [Online]. Available:https://www.semiconducto rs.org/wp-content/uploads/2021/05 /BCG-x-SIA-Strengthening-theGlobal-Semiconductor-Value-ChainApril-2021_1.pdf.
28. M. G. Dixon, "A zero trust approach to architecting silicon;' Intel, Tech. Rep., accessed: 2022-10-24. [Online]. Available:https://www.intel.com/co ntent/www/us/en/newsroom/opini on/zero-trust-approach-architectingsilicon.html