Establishing Cyber Resilience in Embedded Systems for Securing Next-Generation Critical Infrastructure

Establishing Cyber Resilience in Embedded Systems for Securing Next-Generation Critical Infrastructure

Fahad Siddiqui, Matthew Hagan, Sakir Sezer
The Centre for Secure Information Systems (CSIT), Queen's University Belfast
Belfast, United Kingdom
f.siddiqui, m.hagan, s.sezer@qub.ac.uk

***Abstract*—The mass integration and deployment of intelligent technologies within critical commercial, industrial and public environments have a significant impact on business operations and society as a whole. Though integration of these critical intelligent technologies pose serious embedded security challenges for technology manufacturers which are required to be systematically approached, in-line with international security regulations.**

This paper establish security foundation for such intelligent technologies by deriving embedded security requirements to realise the core security functions laid out by international security authorities, and proposing microarchitectural characteristics to establish cyber resilience in embedded systems. To bridge the research gap between embedded and operational security domains, a detailed review of existing embedded security methods, microarchitectures and design practises is presented. The existing embedded security methods have been found ad-hoc, passive and strongly rely on building and maintaining trust. To the best of our knowledge to date, no existing embedded security microarchitecture or defence mechanism provides continuity of data stream or security once trust has broken. This functionality is critical for embedded technologies deployed in critical infrastructure to enhance and maintain security, and to gain evidence of the security breach to effectively evaluate, improve and deploy active response and mitigation strategies. To this end, the paper proposes three microarchitectural characteristics that shall be designed and integrated into embedded architectures to establish, maintain and improve cyber resilience in embedded systems for next-generation critical infrastructure.

***Index Terms*—Cyber Resilient Embedded System, Cyber Resilience, Cyber-Physical Embedded System, Critical Infrastructure, Active Defence, Response, Recover, Security Regulation.**

I. Introduction

Proliferation of intelligent connected technologies are opening venues to new service and computing models, providing diverse socio-economic benefits. These technologies are giving rise to wide range of intelligent applications including smart home, smart city, smart grid and intelligent transportation systems [1], [2]. Estimates from market leading industry predict that intelligent connected technologies will proliferate to a trillion devices by 2035 [3]. This rapid growth of intelligent consumer and industrial solutions is leading to significant growth in smart embedded devices, such as wearables and critical infrastructure components, that provide information and communication functions to the users and businesses.

These smart embedded devices will be integrated and deployed in public and private environments for commercial and non-commercial purposes, to enhance business and consumer experiences by sharing and analysing generated data [4], [5]. This data can be used in a variety of ways, enhancing the customer's experience, bringing new business models and market opportunities using artificial intelligence, machine learning and data analytics, to make better informed decisions. However, where this sharing of data brings benefits and opportunities, it simultaneously presents risks [6]. The large-scale integration and deployment of smart embedded devices and related services within critical infrastructure environments to control critical tasks, poses serious design, supply chain, security and safety challenges [2], [7], [8], [9].

As reliance on these technologies has grown, opportunities have arisen for adversaries to attack and compromise public and commercial critical infrastructure systems [4], [5], [6]. Therefore, international government agencies have released cyber security regulations [10], [11], [12] to curtail this problem by advocating businesses and technology manufacturers to comply and adhere to these regulations. These cyber security regulations pose a need for smart embedded devices and intelligent technologies to be *Cyber Resilient*. Device manufacturers therefore should design, develop and deploy security within their products to maintain compliance, consumer confidence and market share. However this need for harnessing security to comply with cyber security regulations, has compelled embedded designers and security architects to deploy defences that are often ad-hoc and passive in nature. As they have been designed to mitigate a certain class of known attacks [13]. Nevertheless, this strategy has been found vulnerable and compromised due to software vulnerabilities, microarchitectural weaknesses and poor use of secure design practices [8], [14], [15], [16], [17], [18].

Open literature and reported events show that attack methods are evolving and becoming sophisticated, software vulnerabilities are inevitable, embedded architectures are insecure and are therefore susceptible to diverse attacks [13], [16]. A successful launch of an attack on a device can expose private and confidential data of the user and enterprise to adversaries. To best of our knowledge, no existing embedded security microarchitecture or defence mechanism provides continuity of data stream and the information that can be extracted to gain and establish an evidence caused by the security breach for *Cyber Forensics*.

Considering these diverse cyber security challenges, there is a need for adopting a holistic rather than continue pursuing passive approach to achieve cyber resilience in embedded

systems. The architecture shall harness, maintain and ensure design and operational security. Moreover, it shall be capable of both detection and recovery from a launched attack, and preserve crucial security requirements of embedded device deployed within public and private critical environments.

This paper will present the core security functions set out by the international security authorities in Section II. A comprehensive review of existing embedded security practices and mapping of core security functions to existing embedded security landscape will be presented in Section III which will be used to derive the security requirements of a cyber resilient embedded system. The shortcomings of well established embedded security microarchitectures will be discussed in Section IV and microarchitectural characteristics of a cyber resilient embedded system will be proposed in Section V.

II. Cyber Resilience & Cyber Security Regulations

Currently, major differences exist in the way companies are using technologies and adopting security practices into their design, development and operational processes making it more difficult to mitigate and fight against cyber attacks [14]. This problem has been elevated by the lack of adoption of security and cyber resilient posture by the stakeholders. IT Governance is a global provider of cyber risk and privacy management solutions that defines *Cyber resilience* [19] as:

"The ability of a system to identify, prevent, and respond to cyber attacks, intended to disrupt the system's operational capabilities while maintaining confidentiality and integrity of the data"

To streamline these security issues, the *National Institute of Standards and Technology* (NIST) and the *National Cyber Security Centre* (NCSC) have released the following frameworks and regulations to improve security:

- NIST **R**isk **M**anagement **F**ramework (RMF)
- NIST **C**yber **S**ecurity **F**ramework (CSF)
- NSCS Security of **N**etwork and **I**nformation **S**ystems Regulations (NIS)

The NIST *Risk Management Framework* [11] is a guidance document designed to help organisations and enterprises assess and manage risks to their information and infrastructure. It enables a process that integrates security and risk management activities within the system development life cycle as shown in Figure 1. It provides means to *select*, *implement*, *assess*, *authorise* and *monitor* controls. This involves identification of critical components based on their security requirements followed by selection and implementation of effective monitoring controls, that are aligned with the system's operational behaviour. This process enables security architects to identify risks, select suitable mitigation strategies and deploy countermeasures. This also avoid vulnerabilities which might be overlooked in product functional specification.

The NIST *Cyber Security Framework* [10] aims to improve the security of critical infrastructure from cyber attacks. It provides a set of guidelines for technology manufacturers to follow and better prepare to handle cyber attacks, particularly where a lack of security standardisation exists. The framework defines five core security functions (*identify, detect, protect, respond* and *recover*) to establish, maintain and improve cyber resilience as illustrated in Figure 1.

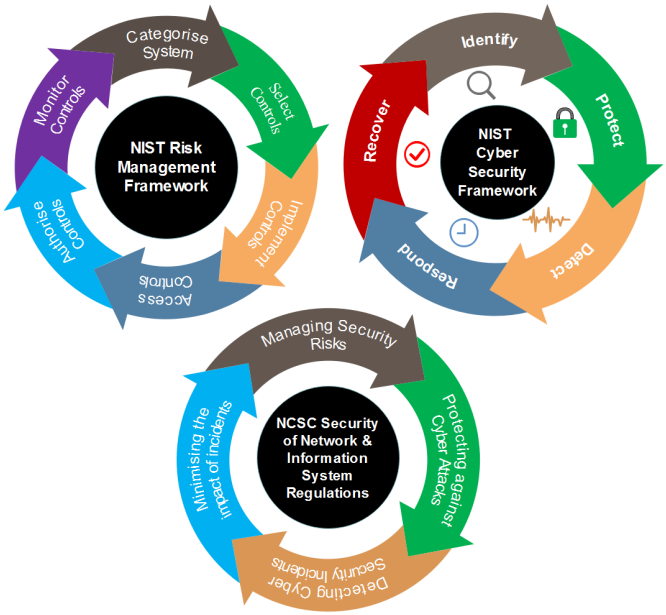

Fig. 1. Core security functions, principles and activities of NIST Risk Management Framework [11], NIST Cyber Security Framework [10] and NCSC Security of Network and Information System Regulations [12].

The primary focus of NCSC Security of *Network and Information Systems* (NIS) [12] regulation is to respond to rising cyber security challenges faced by public/private organisations and enterprises by minimising the risks of disruption to services caused by the failure of digital technologies. One of the key objectives is to establish and improve cyber resilience of intelligent technologies by identifying and managing risks of potential causes of failure by gaining and establishing an evidence of the caused security breach. For this purpose, the regulation introduces four security principles *(managing security risks, protecting against cyber attacks, detecting cyber security incidents and minimising the impact of incidents)* as shown in Figure 1.

The discussed frameworks and regulations advocates that it is essential for technology manufacturers and their involved partners, including semiconductor and original equipment manufacturers (OEMs), to manage their risks by implementing appropriate and proportionate embedded security measures for next-generation critical infrastructure.

III. Security Requirements of Cyber Resilient Embedded System

Cyber Resilience in embedded systems can be achieved by identifying the security requirements and incorporating them into the product life cycle. Intrinsically, the discussed Cyber Security regulations in Section II do not render security requirements for cyber resilient embedded system. Instead they

TABLE I

ASSOCIATION OF NIS SECURITY PRINCIPLES AND CSF CORE SECURITY FUNCTIONS, THEIR RESPECTIVE OPERATIONAL SECURITY AND DERIVED EMBEDDED SECURITY REQUIREMENTS FOR A CYBER RESILIENT EMBEDDED SYSTEM. MAPPING OF EXISTING EMBEDDED SECURITY LANDSCAPE ON TO THE DRIVEN SECURITY REQUIREMENTS IS ALSO PRESENTED.

NSCS NIS [12]	**NIST CSF** [10]	**Operational Security Requirements**	**Cyber Resilient Embedded Security Requirements**	**Existing Embedded Security Practices, Methods and Microarchitectures**
Managing Security Risks	**Identify**	**Asset Management** • Understand and Assess • Identify Risks • Prioritise and Evaluate • Comply and Review	**Embedded Security Modelling** • Risk Assessment • Threat and Security Modelling • Attack surface identification • Secure-by-design practises	✪ STRIDE, PASTA, CVSS, DREAD, HARA ❖ IEC 61508, ISO2626 (ASIL A-D), ISO/IEC 15408 ❖ Common Criteria, FIPS 140-2, ETSI TVRA ❖ ISO/IEC 27005, SAE J3061, ISO/IEC 27001
Protecting against Cyber attack	**Protect**	**Awareness Control** • Protect Data • Protection Technology • Manage & Adopt	**Protection Method** • Chain of Trust • Data Confidentiality and Integrity • Secure Provisioning & Attestation • Isolation and Segregation	Root of Trust, Trusted Technologies, ✪ Secure boot ✪ AES, ECC, RSA, EDSA, ECCDSA, SHA, SSL ✪ Digital Certificate, Public-Private Key Infrastructure ✪ ARM TrustZone, Intel SGX
Detecting Cyber Security Incidents	**Detect**	**Event Discovery** • Discover & Determine • Continuous Monitoring • Detect Anomalies • Alert Events	**Detection Method** • Platform Security Architecture • Trusted Execution Environment • Static & Dynamic Flow Integrity • Access Control and Policing	✪ ARM Platform Security Architecture ✪ Global Platform, ARM TEE, QSEE, Kinibi ✪ Dover [20], ✷ ARMHEx [21] ✷ SECA [22]
Minimising the impact of cyber security incidents	**Respond**	**Response Planning** • Analyse detected events • Response Strategy • Mitigation Strategy • Report & Improve	**Response Method** • Platform Security Manager • Physical Security • Passive countermeasure • Active countermeasure	✪ Trusted Platform Module ✪ Side-channel countermeasure ✪ Reboot, Reset, Key zeroisation
	Recover	**Recovery Planning** • Repair and Update • Improve and Train • Communicate • Evidence Collection	**Recovery Method** • Roll-back and Roll-forward • Fault avoidance and tolerance • Static and Dynamic Redundancy • System Monitoring	✪ Secure Firmware Update, On-the-air update ✪ Single event upset, Parity, Error Correction Codes ✪ Hardware/Software redundancy, Process pairs ✪ Voltage, clock and temperature monitors

❖ International Standard ; ✪ Commercially Available ; ✷ Academic Research Frameworks/Solutions

yield a blueprint which can be used to articulate and derive security requirements for cyber resilient embedded system. Table I shows the association between NIS security principles and CSF core security functions and their operational security requirements. This includes *asset management, awareness control, event discovery, response planning* and *recovery planning* which are used to derive the security requirements of a cyber resilient embedded system. To bridge the research gap between information security and embedded security, the mapping of each driven embedded security requirement onto existing embedded security landscape is presented in Table I.

① **IDENTIFY** and manage cyber security risks by conducting *asset management* which involves detailed understanding of an application use case and respective deployment scenario. This requires decomposition of system components and evaluation of their interactions with internal and external entities to identify their associated risks and threats [23]. This is followed by evaluating and prioritising tasks, where potential damage to the system and its infrastructure for each identified threat.

In embedded domain this process is well established which involves creating an abstraction of the embedded system [24] known as *Threat and Security Modelling* [25], [26]. This builds profiles of a potential attacker, their goals and methods, which then used to define and deploy countermeasures either to mitigate, minimise the impact of the attack or making less attractive for an attacker. Table I list some of the risk and threat assessment modelling methods and international standards. They provide detailed guidelines and specifications to model, implement and comply diverse security in embedded systems.

② **PROTECT** against cyber attacks by introducing system *awareness control*. This required deployment of appropriate data security and protection methods to build a security foundation based on the principles of information assurance [24].

- **Confidentiality**: Ensuring that information is disclosed only to intended individuals, entities and processes.
- **Integrity**: Maintaining and assuring the accuracy and completeness of information over its life cycle.
- **Availability**: Ensuring that information must be available when needed by individuals, entities and processes.
- **Authentication**: Ensuring that information is accessible by only authorised individuals, entities and processes.

In the embedded security domain, well-established cryptography-based protection methods have been published as shown in Table I. These protection methods require strong trust anchor to establish and maintain confidentiality, integrity and authentication [24], [27], [28]. In addition, embedded access control protection methods such as ARM TrustZone and Intel SGX have been widely used to achieve resource isolation and segregation by dividing system into subsystems and isolating their memory spaces.

③ **DETECT** cyber security incidents using *event discovery* methods. This requires detection of malicious activity by continuous monitoring of system critical resources and comparing it against the healthy behaviour. Once malicious activity is detected, generate an alert to initiate a mitigation strategy.

In the embedded security domain, there is a significant published literature on signature, anomaly and information flow-based detection methods [21], [22] as shown in Table I. Within embedded architectures, these security mechanisms

have been deployed at hardware and software layers managed by a *Trusted Execution Environment* (TEE) [29].

④ **RESPOND** to detected threats and malicious activities by planning and deploying an effective response and mitigation strategy to limit and reduce the impact of the cyber attack. *Machine-to-Machine* (M2M) communication is an enabling technology for critical infrastructure [1], which brought serious security challenges to secure, verify and avoid man-in-middle attacks in embedded systems. The existing embedded systems lack the capability to respond against attacks, making a need for active response against attacks a fundamental security requirement for cyber resilient embedded systems. Nevertheless, constantly evolving cyber attacks demand continuous re-evaluation for effective response and mitigation strategies.

Existing embedded security microarchitectures are largely focused on trust-based security and protection. They are limted and provide passive countermeasures such as watchdog timer, brownout reset, voltage and temperature monitoring and anti-tamper. Where, the vast majority system do not have any response mechanism and curtail such attacks using system reboot and reset. Nevertheless, trust management between device manufacturers and service providers is still a formidable challenge [2]. Clearly, there is a strong need for embedded response methods and microarchitectures that fulfil the security requirements of cyber resilient embedded system.

⑤ **RECOVER** system data and resources back to the device healthy provisioned state, by repairing, updating and patching the system. However, effective cyber strategy requires identification of the causes and method of system failure by collecting evidence from the compromised system. It allow to establish, conduct and communicate critical administrative tasks with the actors involved, during the system life cycle, to effectively ensure and maintain safety and security of the critical systems.

The existing embedded security architectures are limited to the principles of reliability to achieve recovery, and thus are insufficient to provide a system-level information or evidence that can be used to improve the cyber strategy. They often make use of fault avoidance and fault tolerant design practises by incorporating redundant system resources and roll-back patches to return the system to a healthy state.

The mapping of existing embedded security approaches in Table I clearly indicates a research gap and need for active response and recovery methods. Section IV extends this by discussing challenges and shortcomings of existing embedded security microarchitectures.

IV. Challenges and Shortcomings in existing Embedded Security Micorarchitectures

Embedded Security has been the subject of extensive research in the context of general-purpose computing, signal processing, programmable architectures and communications systems [7], with significant published work on various fine and coarse grained embedded security challenges [7], [8], [9], [30]. Security is often misapprehended by security architects and system designers as the addition of security features to make the system secure. Instead, security is a process that should be considered and managed throughout the life cycle of the embedded system specially for devices deployed in critical infrastructure. Therefore, this section first presents challenges of existing security microarchitectures:

- The majority of embedded security microarchitectures follow *Device Trust Architecture* [31]. It is a specification that provides a method to design and develop secure component technologies by building trust and secure services from the boot mechanism to the device operating system and application layer. Hence, the security of the system is strongly reliant on building and maintaining a strong *chain of trust* [6] which comprises of a series of nested assumptions and as vulnerable as its weakest link. If broken, compromises the security of the whole system. In the commercial domain, *Secure Boot* is a well established and widely used secure component, which has been found vulnerable [15], [16].
- A lack of clear ownership of device security, insufficient adoption of security-aware practises and an absence of baseline security requirements. Practically design engineers do not perceive themselves accountable for security requirements and effectively embedding them into the device life cycle. This includes a lack of formal security risk assessment, with management of security technology outsourced to third parties for design, development and formalised security patch management process. As a result, this integration of third party services leads to security inconsistencies and vulnerabilities.

These challenges have posed immense need for harnessing security, in compliance with cyber security regulations. This in turn, has compelled embedded security architects to design and deploy defence mechanisms that are often ad-hoc and passive in nature, targeting and mitigating certain attacks or classes of attacks after they have been discovered [13]. This approach may be effective to rectify software vulnerabilities or bugs through a software update, but insufficient to realise effective microarchitecture security which cannot be updated after release. The following are widely adopted embedded security methods has been found vulnerable due too poor usage of secure design practises, software vulnerabilities and microarchitectural weaknesses:

- **Trusted Computing**: Trusted software services uses cryptographic digital signatures to verify the integrity of the firmware and applications which has been exploited to gain access to the device [16]. This has occurred due to lack of roll-back prevention, as the system was using the same digital signature to verify the application. A similar attack has been performed against commercial TEE [32].
- **Processor virtualisation and logical isolation of resources**: In existing embedded security architectures, processor virtualisation has been used to achieve logical isolation between secure and non-secure system resources. This has been attacked through a covert cache-based attack, resulting in leaking of information using mi-

croarchitecture side-channels. The recently demonstrated Spectre [18] attack leverage speculative loads which circumvent access checks to read memory-resident secrets, transmitting them to an attacker using cache timing or other covert communication channels. Meltdown [17] is another microarchitectural attack that exploits out-of-order execution to leak the targets physical memory. These attacks exploit the fact that both secure and non-secure processes shares the same physical memory resource and pointer. Maene *et al.* have proposed a data encryption mechanism [27] that allows automatic encryption and decryption of data between the main memory and cache though found infeasible due to large area overhead.

- **Pointer Authentication**: To circumvent the microarchitecture side-channel leakage attacks, a pointer authentication mechanism has been introduced [33]. This guarantees the integrity of pointers by extending each pointer with authentication code, allowing verification using special instructions that are part of the code executing on the same physical computing resource and managed by the software. Similarly, to mitigate branch prediction attacks, deployment of separate stacks and their pointer registers has been introduced in ARM Cortex-M33 processors.
- **Vulnerable system communication**: A security evaluation of the ARM TrustZone technology has demonstrated that it is possible to modify hardware security attributes and communication bus handshaking signals [34]. This has demonstrated by integrating ARM TrustZone technology with reconfigurable hardware logic.

These microarchitectural weaknesses clearly indicates the need for cyber resilient embedded security microarchitecture that support active detection, response and recovery mechanisms to effectively realise diverse cyber security strategies through the life cycle of an embedded device. To this end, Section V proposes micro-architectural characteristics of a cyber resilient embedded system.

V. Microarchitectural Characteristics of a Cyber Resilient Embedded System

As discussed, there is no active method in existing embedded microarchitectures to establish and maintain the security of a device once its trust is compromised. This leads to exposure of confidential data to the adversary, often without leaving any evidence trail. Considering the derived security requirements of cyber resilient embedded system (Table I), security functionality is not limited to *protection*. The device must *detect* malicious cyber activities and attacks, *respond* against them by deploying active countermeasures and *recover* system back to its healthy state. These are crucial security requirements for embedded devices deployed in critical infrastructure as well as to facilitate forensic analysis, to study behaviour and method of cyber attacks. Using existing embedded security microarchitectures, this is difficult and implausible to recover data due to lack of *continuity of data stream*, *runtime monitoring* and *system-level visibility*. The following are proposed three core microarchitectural characteristics that shall allow to establish historical system data stream by continuous monitoring of system resources and activities, keeping track of events to achieve system-level visibility:

1) An **Independent Active Runtime System Security Manager** shall be responsible for *protection*, *detection*, *response* and *recovery* security functions while complimenting existing security mechanisms. It shall continuously monitor system resources, use gathered information to detect benign or malicious system behaviour, respond to detected malicious (system or resource-specific) activities by deploying active countermeasures and recover system back to its healthy state. It is crucial that *system security manager* must be physically independent and isolated so its memory resources from the general purpose processor. This physical limiting of attack surface, will make the system robust and significantly less susceptible to software vulnerabilities and attacks as was in the case of the TEE. As TEE shares the same physical processor and memory resources with the general purpose processor. Effective realisation of this *system security manager* requires resource-level visibility and monitoring of system's critical components which leads to the second characteristic.
2) An **Active Runtime Resource Monitors** shall actively monitor resource specific behaviour to *detect* malicious activity and report it to the *System Security Manager*. These active monitors are essential as embedded architectures are becoming complex, designers are consolidating diverse functionalities into a single application often involves mixing of sensitive data with non-sensitive data and physical actuation. These active monitors shall generate fine-grained resource specific information which would enable the *system security manager* to articulate, analyse and evaluate system-level behaviours and initiate appropriate *response* and *recovery* strategies. In addition, this gathered information would facilitate continuity of data stream and to extract crucial information necessary to establish evidence of the caused security breach.
3) An **Active Response Manager** shall be responsible for implementing *response* and *recovery* embedded security requirements of a cyber resilient embedded system. It shall actively enforce and execute the response and recovery strategies initiated by the *System Security Manager*. This involves initiating active countermeasures to mitigate and curtail the detected threat to maintain and ensure security of the system. Moreover, depending on the microarchitecture of the *active runtime resource monitors*, the active response manager can enforce various system-level security strategies, where a compromised resource can be physically isolated from the system. This would allow opportunities to gracefully degrade the system functionality while maintaining critical services in next-generation critical infrastructure.

A detailed System-on-Chip (SoC) platform architec-

ture [28], [35] and security modelling approach [25] that realises the proposed embedded microarchitectural characteristics of a cyber resilient embedded system have been published.

VI. Conclusion

This paper has presented the increasing security challenges and requirements, in light of international cyber security regulations for intelligent connected technologies deployed in critical infrastructure. Embedded security requirements has been derived from these regulations to improve cyber resilience and achieve conformance. The paper establishes a strong need for embedded cyber resilience for smart technologies, due to lack of active detection, response and recovery security functionalities within existing embedded security systems.

This is due to the majority of embedded security technologies being guided by trust, which has been compromised due to a lack of runtime monitoring and system-level visibility of resources and system activities. Therefore, this paper proposed three embedded microarchitectural characteristics, allowing independent active runtime system monitoring and active response functions to enhance, maintain and ensure secure operation during the life cycle of the device.

References

[1] J. Wan *et al.*, "M2M Communications for Smart City: An Event-Based Architecture," in *Proc. IEEE International Conference on Computer and Information Technology*, Chengdu, China, Oct. 2012, pp. 895–900.

[2] V. Sharma *et al.*, "Security, Privacy and Trust for Smart Mobile-Internet of Things (M-IoT): A Survey," *CoRR*, vol. abs/1903.05362, 2019. [Online]. Available: http://arxiv.org/abs/1903.05362

[3] P. Spark, "White Paper: The route to a trillion devices: The outlook for IoT investment to 2035," ARM, Tech. Rep., 2017. [Online]. Available: https://community.arm.com/iot/b/blog/posts/white-paper-the-route-to-a-trillion-devices

[4] Cabinet Office, National security and intelligence, HM Treasury, and The Rt Hon Philip Hammond MP, "National Cyber Security Strategy 2016 to 2021," HM Government, UK, Tech. Rep., 2016. [Online]. Available: https://www.gov.uk/government/publications/national-cyber-security-strategy-2016-to-2021

[5] U.S. Department of Homeland Security, "Strategic Principles for Securing the Internet of Things," US Government, Tech. Rep., 2016. [Online]. Available: https://www.gov.uk/government/publications/national-cyber-security-strategy-2016-to-2021

[6] A. Ukil, J. Sen, and S. Koilakonda, "Embedded security for Internet of Things," in *Proc. IEEE National Conference on Emerging Trends and Applications in Computer Science*, Shillong, India, Mar. 2011, pp. 1–6.

[7] S. Ravi *et al.*, "Security in Embedded Systems: Design Challenges," *ACM Trans Embed. Comput. Syst. (TECS)*, vol. 3, no. 3, pp. 461–491, Aug. 2004.

[8] N. Apthorpe *et al.*, "Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic," *CoRR*, vol. abs/1708.05044, 2017.

[9] D. N. Serpanos and A. G. Voyiatzis, "Security Challenges in Embedded Systems," *ACM Trans. Embed. Comput. Syst. (TECS)*, vol. 12, no. 1s, pp. 66:1–66:10, Mar. 2013.

[10] "Framework for Improving Critical Infrastructure Cybersecurity," National Institute of Standards and Technology (NIST), Tech. Rep., Apr. 2018. [Online]. Available: https://www.nist.gov/cyberframework/framework

[11] "Risk Management Framework for Information Systems and Organizations," National Institute of Standards and Technology (NIST), Tech. Rep., Dec. 2018. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-37/rev-2/final

[12] "Guidance on implementing the EU Directive on the security of Network and Information Systems," National Cyber Security Centre (NCSC), Tech. Rep., Oct. 2018. [Online]. Available: https://www.ncsc.gov.uk/collection/nis-directive

[13] D. Meng *et al.*, "Security-first architecture: deploying physically isolated active security processors for safeguarding the future of computing," *Cybersecurity*, vol. 1, no. 1, p. 2, Jun. 2018.

[14] M. Choraś *et al.*, *Cyber Threats Impacting Critical Infrastructures*, 2016, pp. 139–161.

[15] M. McClintic *et al.*, "Keyshuffling Attack for Persistent Early Code Execution in the Nintendo 3DS Secure Bootchain," *CoRR*, vol. abs/1802.00092, 2018. [Online]. Available: http://arxiv.org/abs/1802.00092

[16] C. Yue *et al.*, "Downgrade Attack on TrustZone," *Computing Reseach Repository (CoRR)*, Jul. 2017.

[17] M. Lipp *et al.*, "Meltdown," *CoRR*, vol. abs/1801.01207, 2018.

[18] V. Kiriansky and C. Waldspurger, "Speculative Buffer Overflows: Attacks and Defenses," *CoRR*, vol. abs/1807.03757, 2018.

[19] "What is Cyber Resilience?" IT Governance UK, Tech. Rep., 2018. [Online]. Available: https://www.itgovernance.co.uk/cyber-resilience

[20] "Dover Architecture: Hardware Enforcement of Software-Defined Security Policies," Dover Microsystems, Tech. Rep., 2016. [Online]. Available: https://web.cs.wpi.edu/ rjwalls/nesd/slides/DraperFall2016Dover.pdf

[21] M. A. Wahab *et al.*, "ARMHEx: A hardware extension for DIFT on ARM-based SoCs," in *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, Sep. 2017, pp. 1–7.

[22] J. Coburn *et al.*, "SECA: Security-enhanced Communication Architecture," in *Proceedings of the 2005 International Conference on Compilers, Architectures and Synthesis for Embedded Systems*, ser. CASES '05. New York, NY, USA: ACM, 2005, pp. 78–89.

[23] R. Khan *et al.*, "STRIDE-based threat modeling for cyber-physical systems," in *Proc. IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Sep. 2017, pp. 1–6.

[24] S. Ray *et al.*, "System-on-Chip Platform Security Assurance: Architecture and Validation," *Proc. IEEE*, vol. 106, no. 1, pp. 21–37, Jan. 2018.

[25] M. Hagan, F. Siddiqui, and S. Sezer, "Policy-Based Security Modelling and Enforcement Approach for Emerging Embedded Architectures," in *Proc. 31st IEEE International System-on-Chip Conference (SOCC)*, Arlington, USA, Sep. 2018, pp. 84–89.

[26] M. Hagan *et al.*, "Enforcing Policy-Based Security Models for Embedded SoCs within the Internet of Things," in *Proc. IEEE Conference on Dependable and Secure Computing (DSC)*, Kaohsiung, Taiwan, Dec. 2018, pp. 1–8.

[27] P. Maene *et al.*, "Atlas: Application Confidentiality in Compromised Embedded Systems," *IEEE Trans. Depend. Sec. Comput.*, pp. 1–1, 2018.

[28] F. Siddiqui, M. Hagan, and S. Sezer, "Embedded policing and policy enforcement approach for future secure IoT technologies," in *Living in the Internet of Things: Cybersecurity of the IoT*, Mar. 2018, pp. 1–10.

[29] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted Execution Environment: What It is, and What It is Not," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Helsinki, Finland, Aug. 2015, pp. 57–64.

[30] D. Papp, Z. Ma, and L. Buttyan, "Embedded systems security: Threats, vulnerabilities, and attack taxonomy," in *Proc. 13th IEEE Internation Conference on Privacy, Security and Trust (PST)*, Izmir, Turkey, Jul. 2015, pp. 145–152.

[31] GlobalPlatform, "Introduction to Device Trust Archietcture," GlobalPlatform, Tech. Rep., Jul. 2018. [Online]. Available: https://globalplatform.org/resource-publication/introduction-to-device-trust-architecture/

[32] "Trust Issues: Exploiting TrustZone TEEs," Google: Project Zero, Tech. Rep., Jul. 2017. [Online]. Available: https://googleprojectzero.blogspot.com/2017/07/trust-issues-exploiting-trustzone-tees.html

[33] R. Avanzi, "The QARMA Block Cipher Family. Almost MDS Matrices Over Rings With Zero Divisors, Nearly Symmetric Even-Mansour Constructions With Non-Involutory Central Rounds, and Search Heuristics for Low-Latency S-Boxes," *IACR Transactions on Symmetric Cryptology*, vol. 2017, no. 1, pp. 4–44, Mar. 2017.

[34] E. M. Benhani, L. Bossuet, and A. Aubert, "The Security of ARM TrustZone in a FPGA-based SoC," *IEEE Transactions on Computers*, pp. 1–1, 2019.

[35] F. Siddiqui, M. Hagan, and S. Sezer, "Pro-Active Policing and Policy Enforcement Architecture for Securing MPSoCs," in *Proc. 31st IEEE International System-on-Chip Conference (SOCC)*, Arlington, USA, Sep. 2018, pp. 140–145.