Diversity-by-Design for Dependable and Secure Cyber-Physical Systems: A Survey

Qisheng Zhang, Abdullah Zubair Mohammed, Zelin Wan, Jin-Hee Cho, *Senior Member, IEEE*, and Terrence J. Moore, *Member, IEEE*

***Abstract*—Diversity-based security approaches have been studied for several decades since the 1970s. The concept of *diversity-by-design* emerged in the 1980s. Since then, diversity-based system design research has been explored to provide more secure and dependable services in cyber-physical systems (CPSs). In this work, we are particularly interested in providing an in-depth, comprehensive survey of existing diversity-based approaches, their insights, and associated future work directions for building secure and dependable CPSs. This will allow us to provide promising ways of providing quality network and services based on key diversity-by-design principles for those who want to conduct research on developing secure and dependable CPSs using diversity as a system design feature. This survey paper mainly provides: (i) The common concept of diversity based on its multidisciplinary nature along with the historical evolution of the concept of diversity-by-design for providing secure and dependable services; (ii) the key diversity-by-design principles; (iii) the key benefits and caveats of using the diversity-by-design; (iv) the main concerns of CPS environments utilizing the diversity-by-design; (v) an extensive survey and discussions of existing diversity-based approaches based on five different classifications; (vi) the types of attacks considered by diversity-based approaches; (vii) the overall trends of evaluation methodologies used for diversity-based approaches, in terms of metrics, datasets, and testbeds; and (viii) the insights, lessons, and gaps identified from this extensive survey and future work directions.**

***Index Terms*—Diversity-by-design, software diversity, heterogeneity, security, dependability, cyber-physical systems.**

I. Introduction

DIVERSITY has been recognized as a key factor for a system to be managed for providing quality services with the aim of achieving high sustainability even under hostility or dynamics. This line of diversity concept was initially observed in the area of biodiversity [74]. After then, many diversity-based security mechanisms have been proposed in [15], [54], [72], [175]. Now more than ever, security is commonly considered as a key service property called as 'security-as-a-service' (SaaS) [33] and the concept of diversity-by-design has been considered as a way of achieving the SaaS. Common examples of the diversity-by-design as a security service include different implementations of software providing the same functionalities [175], diverse software stacks [82], [83], dynamic configurations of a network topology [181], antenna diversity in hardware for generating a shared secret key [176], and architectural diversity to improve security and dependability of Field Programmable Gate Arrays (FPGA) systems [91]. These various types of diversity-based designs have been applied with many applications in cyber-physical systems (CPSs).

CPSs have received significant interest over the past decade spurred in part by programs run by various organizations like the National Science Foundation and the National Institute of Standards and Technology [126]. As this research continues to develop and CPSs become more prevalent in society, there is an increasing demand for research that may solve current and future issues involving CPSs, which include security, privacy, dependability, functionality, etc. This is more critical when CPSs are being developed as a service [30], [51], where the CPS features such as security and dependability necessarily become part of that service. In this survey, we are particularly interested in investigating how diversity can contribute to enhancing system security and dependability. Dependability and security are defined by their key attributes, which for dependability includes reliability, availability, safety, integrity, and maintainability while for security encompasses confidentiality, integrity, and availability [5]. In this work, we conducted an extensive survey on diversity-based approaches designed to build CPSs more resilient against attacks and faults. To be specific, we focused our survey on the following: the key design principles, the historical evolution of the diversity concept, the key approaches at different system layers, the attacks defended, the evaluation metrics, and the evaluation testbeds used for the proposed diversity-based approaches. In addition, we extensively illuminate the pros and cons of each approach and address insights and lessons learned from this survey that suggest future research directions.

A. Comparison With Other Similar Survey Papers

To clarify the contributions of our survey paper, we identified the key merits of our survey paper, compared to

Manuscript received July 16, 2020; revised January 25, 2021 and June 5, 2021; accepted June 14, 2021. Date of publication June 22, 2021; date of current version March 11, 2022. This research was partly supported by the Army Research Office under Grant Contract Number W91NF-20-2-0140. The associate editor coordinating the review of this article and approving it for publication was C. Fung. *(Qisheng Zhang, Abdullah Zubair Mohammed, and Zelin Wan contributed equally to this work.) (Corresponding author: Qisheng Zhang.)*

This work did not involve human subjects or animals in its research.

Qisheng Zhang, Zelin Wan, and Jin-Hee Cho are with the Department of Computer Science, Virginia Tech, Falls Church, VA 22043 USA (e-mail: qishengz19@vt.edu; zelin@vt.edu; jicho@vt.edu).

Abdullah Zubair Mohammed is with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Arlington, VA 22041 USA (e-mail: abdullahzubair@vt.edu).

Terrence J. Moore is with the Network Science Division, U.S. Army Research Laboratory, Adelphi, MD 20783 USA (e-mail: terrence.j.moore.civ@mail.mil).

Digital Object Identifier 10.1109/TNSM.2021.3091391

1932-4537 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON OF OUR SURVEY AND OTHER EXISTING SURVEYS ON DIVERSITY-BASED SYSTEM DESIGNS

Criteria	Our survey paper	Balakrishnan and Schulze [13] (2005)	Larsen et al. [103] (2014)	Baudry and Monperrus [17] (2015)	Hosseinzadeh et al. [76] (2016)	Hosseinzadeh et al. [77] (2018)
Multidisciplinary concept of diversity	✓	✗	✗	✗	✗	✗
Design principles: benefits & caveats	✓	✓	✓	✓	✗	Limited (summary)
Attributes & properties of dependable and secure CPSs	✓	✗	✗	✗	✗	✗
Diversity-based approaches discussed based on multi-faceted dimensions of CPSs	✓	Limited (obfuscation techniques)	✓	✓	Limited (summary)	✓
Attacks countermeasured by diversity-based approaches	✓	✗	✓	✗	Limited (Statistic table)	Limited (Statistical table)
Validation & verification methodologies (i.e., metrics, datasets, and evaluation testbeds)	✓	✗	✗	Limited (cost)	✗	Limited (cost)
Discussions on insights, limitations, and lessons learned	✓	✓	✓	✓	✓	✓

the existing survey papers discussing diversity-based security approaches [13], [17], [76], [77], [103].

Balakrishnan and Schulze [13] conducted a limited survey based on 14 references and mainly focused on code obfuscation, a typical method to apply software diversity at an instruction level. They covered general code obfuscation techniques used by viruses to hide malicious activities or commercial software to protect valuable assets. This paper analyzed the attack patterns of a virus to show the potential usage of code obfuscation in the security domain. However, this paper didn't provide any security or performance metrics used in the existing obfuscation techniques.

Baudry and Monperrus [17] surveyed recent studies on software diversity. They categorized each work into either managed diversity or automated diversity based on the human involvement. They discussed the core concept of each technique along with its pros and cons. As for performance evaluations, the authors primarily discussed resilience and security to show the effectiveness of diversity techniques. They also provided an overview on the research history of software diversity. However, their work didn't conduct a comprehensive survey on attacks covered and performance metrics used by the existing diversity-based security techniques.

Larsen *et al.* [103] conducted a more extensive survey on software diversity than [13], [17]. Their survey paper provided two different types of classification of software diversity: (i) the scope and level where diversifications occur, including instruction, basic block, loop, function, program, and system levels; and (ii) the development time when diversifications occur, such as implementation, compilation, linking, installation, loading, executing, and updating. They also discussed extensive sets of attacks and diversity metrics employed in the existing diversity-based security approaches. For each work cited in their survey paper, the authors discussed the performance of a corresponding diversity technique evaluated by running time and memory. They discussed the pros and cons of each diversity technique, providing an example.

Hosseinzadeh *et al.* [77] reviewed over three hundred papers related to diversification and obfuscation from 1993 to 2017. This survey provided an overview of the current existing research in this area. Specifically, by expanding their previous work [76], which only discussed the aims and environments of diversity approaches, they categorized papers based on their aims, environments used, attacks mitigated, obfuscated/diversified targets, obfuscation/diversification levels, and development time as in [103]. However, this work mainly introduced different software diversity techniques other than analyzing their pros and cons in detail. The authors also provided the summary of evaluation methods in performance (e.g., memory size, running time) and security against attacks.

B. Key Contributions

As described in Table I summarizing our discussions above, unlike the above existing survey works [13], [17], [76], [77], [103], our survey provided the additional contributions as below.

The **key contributions** of our survey paper are as follows:

1) We conducted an extensive survey on the multidisciplinary concepts of diversity derived from nine different disciplines to provide an in-depth understanding and merits of diversity to maximize their contributions to achieving system security and dependability.
2) We provided design principles to develop diversity-based security techniques in terms of *what-to-diversify*, *how-to-diversify*, and *when-to-diversify* as design strategies to enhance system security and dependability.
3) We provided an extensive survey on diversity-based approaches based on a classification of five different layers from the physical environment to human factors in order to comprehensively discuss the core role of each technique and its pros and cons. In addition, we discussed how the key merit of each technique can contribute to improving security and dependability for CPSs.
4) We conducted a comprehensive survey on the set of attacks that have been considered by the existing diversity-based security techniques. This provides a landscape view of what attacks have been mitigated by diversity, leading to our discussion on what other types of attacks diversity-based approaches may prove fruitful in future research.

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

5) We provided an in-depth survey on evaluation methodologies, in terms of metrics, datasets, and testbeds used for experiments conducted to validate the existing diversity-based security approaches. We also suggested how to improve experimental environments in order to offer more practical help to real world applications in enhancing security and dependability for CPSs.
6) Based on our up-to-date and extensive survey on existing diversity-based approaches and our in-depth discussions of their pros and cons, we offered a list of future research directions that may be highly promising to the diversity-based design of secure and dependable CPSs.

Note that the scope of this survey paper is diversity-based approaches for dependable and secure CPSs. Hence, this survey paper does not consider non-diversity-based approaches for CPSs or diversity-based approaches for application domains other than CPSs for our work to be better focused.

C. Structure of the Paper

The rest of this paper is organized as follows:

- Section II discusses (i) the common concept of diversity based on the concepts of diversity discussed in nine different disciplines; (ii) the evolution of diversity-based security approaches from the 1970s to the 2010s; (iii) the key principles of designing diversity-based approaches to build secure and dependable CPSs, and (iv) the key benefits and caveats of diversity-based CPS designs.
- Section III describes what types of CPSs we address in this work and the key attributes of security and dependability.
- Section IV introduces a variety of existing diversity-based approaches to build secure and dependable CPSs. We discussed the existing diversity-based approaches, in terms of the system layer in which an approach is deployed, covering five layers from physical environments to human-machine interactions, with the discussions of their pros and cons.
- Section V surveys what types of attacks are defended by the existing diversity-based approaches.
- Section VI provides a survey on how existing diversity-based approaches have been verified and validated in terms of metrics, datasets, and evaluation testbeds used.
- Section VII discusses the limitations and lessons learned from this comprehensive survey.
- Section VIII concludes the paper by summarizing our key findings and suggesting future work directions.

II. Concepts and Evolution of Diversity-Based Security, and Their Design Principles

A. Multidisciplinary Concept of Diversity

The concepts of diversity have been discussed in multiple disciplines and have been applied in various forms in the context of each discipline. In this section, we discuss the multidisciplinary concepts of diversity and the key benefits and caveats when applying the concept of diversity as a design feature to achieve system dependability and security. Based on our extensive survey on the multidisciplinary concepts of diversity, including biodiversity [49], geodiversity [22], biology [93], sociology [131], psychology [42], political science [45], organizational management [165], nutrition [152], and computing and engineering [175], we derived one common ideology of diversity as follows:

> *Diversity of components in a system (e.g., a group, community, society, body, ecosystem, and computer system or network) can enhance sustainability originated from the principle of polyculture system components that will be highly resistant against sudden, disastrous changes from external effects. The system sustainability can be achieved by meeting multi-faceted properties of system quality, such as dependability, security, survivability, fault tolerance, resistance, stability, creativity, and resilience.*

B. Evolution of Diversity-Based Security

Diversity-based security has been studied for decades. In the 1970s, Randell [135] proposed recovery blocks in programs to detect potential errors in the execution process and perform spares with diverse implementations. Avizienis [6] introduced the concept of N-version programming (NVP), providing multiple programmings with the same functionalities.

In the 1980s, Avizienis [7] and Knight and Leveson [96] described both experimental results and applications of NVP, which is a fault-tolerance approach that was originally applied to the physical faults and has been reused for software fault-tolerance. Brilliant *et al.* [23] raised a problem that if the NVP comparison is based on the finite-precision number output from multi-version applications, it is impossible to guarantee that two correct applications have a consistent output leading to potential false positives. The terms *design diversity* [8], [9] and *software diversity* are coined from the hardware diversity domain in the 1980s [65].

In the 1990s, Cohen [40] first applied the concept of diversity in software for defending against cyberattacks. Forrest *et al.* [54] first comprehensively described diversity in computer systems and argued its merit in the application of computer security. These authors also highlighted the promise of using diversity for security and forecasting some security issues. Also in the 1990s, other studies tried to combine NVP and *design diversity* [113].

The creation of the World Wide Web in the 1990s generated an escalating demand for access to the Internet for consumer, commercial and governmental interests, but this also created ever-increasing opportunities for groups or individuals with malicious intent. Online threats became increasingly concerning as more diverse and serious attacks are introduced over time [117]. The sophistication of these cyberattacks, such as zero-day attacks, has evolved substantially to the point that systems can be infected without a user's action of mistakenly downloading software, but simply by the user visiting a malicious website [117]. At the same time, also in the 2000s, automate diversity became more widely in use, which spurred the adaptation of diversity-based approaches into novel applications. However, the purposes of diversity were different

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

TABLE II
EVOLUTION OF DIVERSITY-BASED SECURITY AND DEPENDABILITY

1970s	1980s	1990s	2000s	2010s
Emergence of recovery blocks for error detection and redundant diverse implementations; and N-version programming (NVP) for providing multiple programmings with the same functionalities [6, 135]	Enhanced maturity of NVP based on theoretical and empirical analysis; emergence of 'design diversity' and 'software diversity'; and focused on fault tolerance [7, 8, 9, 23, 65, 96]	Emergence of the concept of software diversity; combining NVP and 'design software' for software diversity; and starting to recognizing the use of diversity for security applications [40, 54, 113]	Wide application of automate diversity; different applications of the same concept of diversity; and applied software diversity for preventing malware [15, 34]	Common use of software diversity for security and dependability [18, 44, 57, 175]
Diversity-based software fault tolerance			**Diversity-based security and dependability**	

even if they all use the same concept of diversity [15]. For example, in software engineering, diversity is used to create multiple solutions for solving one problem in order to significantly increase the probability of finding a solution. However, in security, diversity is used to avoid replicated attacks and increase attack complexity so the attacker is forced to redesign its strategy even if it attacks the same target. In the late 2000s, the concept of *software diversity* has been applied to defend against malware propagation [175]. An era of the so-called *diversity for security* has begun [34]. In the 2010s, software diversity-based approaches were commonly used for enhancing system security and dependability [18], [44], [57]. We summarized how diversity-based security research has been evolved from the 1970s to the 2010s in Table II.

C. Key Design Principles of Diversity-Based Approach

In this section, we discuss the following key design principles of what-, how-, and when-to-diversify.

1) What-to-Diversify: This principle refers to what platform a given diversity-based approach is applied to achieve a particular design goal. We discuss the design principle of what-to-diversify for three different systems, including cyber, physical, and cyber-physical systems. What-to-diversify at a different system type is detailed as follows:

- *Diversity-based approaches at cyber systems* have been applied using software stack diversity [10], [82], [87], software version diversity [2], [18], [57], [67], [157], code diversity [19], [72], [77], [78], [88], [98], and programming language diversity [109], [154]. Each approach is detailed in Section IV-D.
- *Diversity-based approaches at hardware systems* have been also used, such as sensors and actuators [94], [116], [164], embedded devices [59], [167], and communication modules [63], [176] to improve the security and dependability of the system. Each approach applied in these categories is detailed in Section IV-C.
- *Diversity-based approaches at CPSs* have been used to improve reliability and safety [94], such as network diversity [177], physical environment diversity [170], and human-machine interaction diversity [81]. The examples using N-variance concepts include multi-version technology, multi-version systems, multi-version projects, and multi-version life-cycles. The applications of these concepts on various industrial test-cases have been discussed to improve safety, security, and survivability [94], [116]. The detail of each approach applied in these categories is given in Section IV-A, Section IV-B, and Section IV-E.

2) How-to-Diversify: This principle refers to a particular technique to realize diversity in given systems (or networks). We categorize the types of techniques based on the existing approaches as: *randomization* (e.g., software stack [10], [87], address space [19], [98], instruction set [78], network shuffling [156]), *dynamic reconfiguration* (e.g., code reconfiguration [88], reconfiguration of antenna systems [63], [176], network topology reconfiguration [181]), *diverse duplication* (e.g., software for malware detection [18], the operating system instances [2], [57], [67], Web-servers [157], code diversity [72], diversified system architecture [59], [167]), and *obfuscation* (e.g., code obfuscation [77], network diversity [120]). Each technique is detailed in Section IV.

3) When-to-Diversify: Diversity-based approaches can be either dynamically applied at the system operation stage (e.g., time-varying dynamic reconfiguration) or statically configured at the system deployment stage (e.g., diversification of software stack). For the dynamic diversification of system configurations, whenever the changes are made, a corresponding cost occurs. Hence, overly frequent changes of system configurations or maintaining too high diversity may introduce some drawbacks. Therefore, there should be adaptive strategies that can maintain diversity for system security and dependability while minimizing performance degradation or overhead.

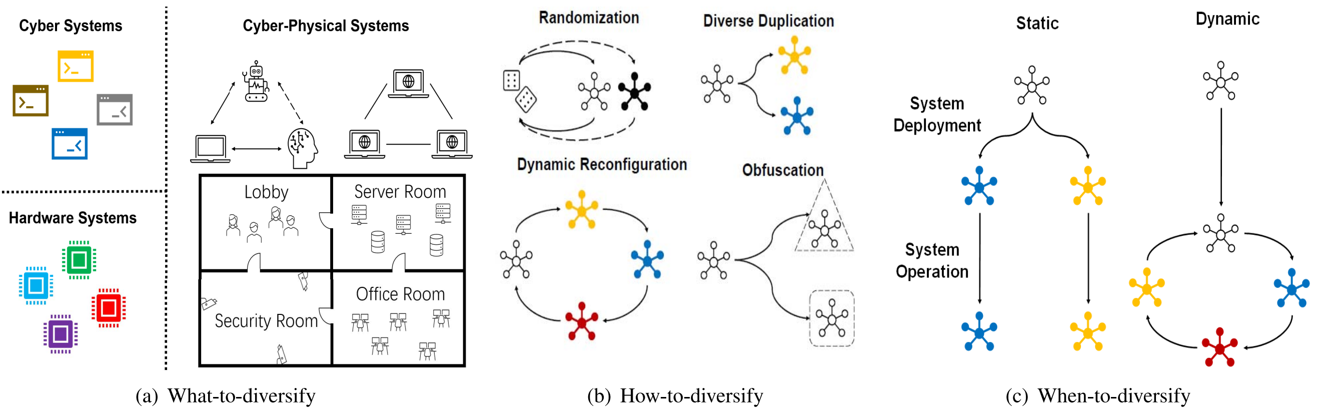
In Fig. 1, we conceptually visualized the key principles of diversity-by-design based on our discussion above.

D. Benefits and Caveats of Diversity-Based System Designs

1) The Benefits of Diversity-Based System Designs Are:

- *Increasing fault tolerance of a system:* Diversity-based system design can introduce high fault tolerance, meaning that the system can be functional even in the presence of attacks. Note that fault tolerance is one of the key attributes of *resilience*, which includes fault tolerance, adaptability, and recoverability [37]. The origin of diversity-based approaches was to enhance fault tolerance [8].
- *Enhancing system availability and reliability:* Software or hardware diversity-based designs allow a system to continuously function even when a system component is being compromised because the system does not consist of homogeneous components exposing the same vulnerabilities. This introduces high fault tolerance of

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.



(a) What-to-diversify (b) How-to-diversify (c) When-to-diversify

Fig. 1. Three key principles of the diversity-by-design.

the system. This also naturally introduces high system availability and reliability since better preparation for attacks increases the capability of providing seamless, uninterrupted services.

- *Leveraging existing technologies:* A basic approach of diversity-based system design is the use of different implementations of software, hardware, or other system components that can provide the same functionalities or services. Instead of developing a new technology, which can be challenging as it requires additional time and effort to ensure performance and security requirements, diversity-based design can easily leverage legacy technologies.

2) The Caveats of Diversity-Based System Designs Are:

- *Potential high cost and performance degradation:* The key caveat of diversity is the potential of greater cost. For many types of systems, maintaining diversification of system components can be costly. Hence, even if diversity-based system designs can provide high security and dependability for CPSs, we should seek critical tradeoff-aware solutions.
- *High challenges in deployment of diversity-based designs:* If diversification is not successfully deployed, system security and dependability may suffer since the high cost, delay, or incompatibility may significantly reduce Quality-of-Service (QoS) possibly resulting in system failure due to highly disruptive services.
- *Lack of positive effect of diversity in poorly designed, unsecure systems or components:* Diversity can be effective only when an individual system component is sufficiently secure. For example, if an individual software package is poorly developed with significant vulnerabilities, using different software packages may not be able to enhance system security or dependability [32].

E. Summary and Discussions

As shown in Table II, originating from notions of diversity design to enhance fault tolerance, diversity-based approaches have been popularly applied to design secure and dependable systems over the past two decades. We categorized these diversity approaches by what, where, and how to diversify. Moreover, we discussed several benefits and caveats of incorporating diversity into the design of systems. While diversity approaches can leverage existing technologies to increase system fault tolerance and enhance system availability and reliability, these approaches can potentially have high cost and implementation difficulties in certain environments. Note that in this paper, we use the term diversity to represent a higher level of system goal while using heterogeneity to represent distinct features of entities or system components. Hence, diversity can embrace both dynamic and static nature of different types of system components or environmental settings while heterogeneity refers to the static state of an entity or environment itself.

III. Types of CPSs and the Key Attributes of Dependable and Secure CPSs

In this section, we discuss three types of CPSs, including Internet-of-Things (IoT), Smart Cities, and Industrial Control Systems (ICSs). In addition, we discuss the dependability and security attributes of CPSs along with how diversity-based designs can enhance the dependability and security of CPSs.

A. Types of Cyber-Physical Systems

1) Internet-of-Things: IoT technologies have become more popular and have been recognized as one type of CPS that can provide effective services to users. IoT encompasses a large number of CPSs including Wireless Sensor Networks (WSNs) integrated with IoT [97]. **The key challenges of designing an IoT [141]** are: (1) Distributed communications, data filtering/processing, and a large amount of data dissemination in highly different forms (e.g., text, voice, haptics, image, video) for distributed communications in large-scale networks by heterogeneous entities (e.g., devices or humans); (2) resource constraints in battery, computation, communication (e.g., bandwidth), and storage; (3) highly adversarial environments, introducing compromised, deceptive entities and data; and (4) high dynamics of interactions between entities, data, network topology and resources available in which each component dynamically changes in time and space.

2) Smart Cities: From a technological perspective, a smart city is considered as a CPS [29]. A smart city encompasses

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

intelligent transportation, smart buildings and infrastructure, smart citizens and governance, and so forth. In addition to the challenges discussed for IoT, **the following additional design challenges should be considered for smart cities** [16], [150]: (1) Effortless connectivity and coordination of all sectors where the infrastructure of a city is significantly distributed and the smart city should work effectively; (2) Effective and efficient data and their integration required where huge amounts of data need to be generated and systems should perform efficient data acquisition, mining, integration, transformation, and additional analysis; and (3) High security and privacy for data [53] where the inherent heterogeneity of smart cities and multiple interfacing systems increases the number of security threats and vulnerabilities [62].

3) Industrial Control Systems (ICSs): An ICS is an umbrella term for all the control systems associated with industrial processes and instrumentation. As the automation and smart control areas have grown, an ICS deploys a CPS for its purposes. Industrial CPSs are deployed in environments that are not easily accessible by humans and are expected to sustain for a long duration [50], [118]. **The key challenges of designing an ICS include**: (1) Control timing requirements and complexity where the complex industrial processes have the stringent requirements on timing for efficient, synchronized and uninterrupted operation [153]; (2) The nature of distributed environments where a system is distributed across a wide geographical area [153]; (3) High availability in which all the domains need to be available at all times [153]; (4) Mitigation of a single failure in one domain, which may result in severe consequences on an entire system [153]; and (5) High vulnerabilities to cyberattacks [52] caused by the large attack surface derived from the distributed nature of the ICS, the system requiring user or device authentication (e.g., a two-way authentication), and communication channels vulnerable to eavesdropping or DoS attacks.

B. Key Attributes of Dependability and Security for CPSs

In this section, we discuss the key attributes of system dependability and security and how diversity-based designs can contribute to build secure and dependable a CPS.

1) Dependability Attributes for CPSs: Avizienis *et al.* [5] defined the primary attributes of dependability as availability, reliability, safety, integrity and maintainability. They discussed robustness as a secondary attribute of dependability to be examined against external faults. However, Cho *et al.* [37] argued that resilience and agility should be also considered as dependability attributes since they can capture the dynamic aspects of a system quality, which have not been fully addressed in the other existing metrics. According to [86], dependability represents a collective term used to describe system availability and its influencing factors, including reliability, maintainability, and maintenance.

2) Security Attributes for CPSs: According to Avizienis *et al.* [5], the primary security attributes consist of confidentiality, integrity, and availability. The secondary security attributes include accountability, authenticity, and non-repudiation. Kharchenko [94] included availability, confidentiality, and integrity as security attributes. Humayed *et al.* [84] studied vulnerabilities and attacks in smart grids, medical CPS, and smart cars, where they interpreted security as availability, one of the security goals. Trivedi *et al.* [159] defined dependability and security as one property where their attributes are defined based on availability, confidentiality, integrity, performance, reliability, survivability, safety, and maintainability. Compared to [5], performance and survivability are additionally considered in [159].

C. Summary and Discussions

In this section, we broadly classified CPSs, based on the areas of applications, into IoTs, Smart Cities, and ICSs. Security challenges for each classification were laid out. We discussed the attributes of dependability and security in the context of CPSs and introduced the solutions provided by diversity-based approaches. Due to multiple versions of systems with the same functionality, diversity-based designs can enhance dependability attributes in all aspects, including availability, reliability, safety, integrity, and maintainability. However, diversity can have multiple, different effects on security attributes, such as confidentiality, integrity, and availability. For example, high diversity design may enhance system availability and integrity while it may not necessarily increase confidentiality.

IV. Diversity-Based Techniques

In this section, we discuss the existing diversity-based techniques for dependable and secure CPSs. To embrace multi-faceted aspects of a CPS, we categorize diversity-based techniques at the following layers: Physical environments, network, hardware, software, and human users.

A. Diversity of Physical Environments

1) Diversified Physical Environments: A CPS incorporates a physical environment to further improve its practicality and effectiveness. These physical environmental factors include guards, cameras, badge readers, physical access policies, biometrics, and so forth [170]. However, this may introduce different types of vulnerabilities in CPSs, implying a widened attack surface.

2) Diversified Physical Topology: Skandhakumar *et al.* [147] proposed a building information model to capture the diversified topology information of the building by integrating different 3D models of objects in the building and creating a 3D map of the building. This information can be used to administrate and enforce the access control systems. Akhuseyinoglu and Joshi [1] built a general access control model to calculate and manage potential risks in access requests. For any incoming accepted access request, their model could check corresponding total risks of multiple diversified action sequences to select the option that can minimize risks to the system. Cao *et al.* [27] developed a risk control mechanism to dynamically assign a clearance level based on a user's history of activities and the related risks of the requests. The authors considered diversified physical topology

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

constraints generated by different combinations of basic role-permissions defined in a unified model to further restrict the access to important assets in CPSs. Many physical environment security issues are related to the human factors because humans can introduce serious security vulnerabilities during their interaction with physical systems [128]. We discuss more on this in Section IV-E.

Pros and Cons: Diversification in physical environments is generally more complicated than that in cyberspaces due to the high uncertainty and inconstancy associated with human mistakes, including human attackers that have high intelligence and ability to launch sophisticated attacks. Access control in physical environments is primarily used to manage human-related risk. However, current approaches still introduce high vulnerabilities to highly intelligent human attackers.

B. Network Diversity

Typically, a CPS includes many networks, such as sensor networks and actuator networks, where network topology is defined as the set of connections between network components. The network diversity research has been explored primarily in terms of three aspects: diversity generation, diversity quantification, and diversity deployment. We discuss each aspect in detail below.

1) Diversity Generation: Network diversity refers to the diversification in network settings [120], [156]. Key factors in network settings include network topology and system components installed in each node of the network. The system component includes hardware or software components. Thus, network diversity can be derived from heterogeneous network topologies or generated from different variants of hardware and software components, which will be detailed in Sections IV-C and IV-D.

2) Diversity Quantification: Diversity metrics have been proposed to measure the degree of network diversity by deploying multiple variants of software. The diversity metrics used in the literature are as follows:

- *Entropy:* Neti *et al.* [120] modeled diversity in a network using a bipartite graph between a set of hosts and a set of vulnerabilities. The diversity, $\mathcal{D}_\alpha$, is quantified using Rényi entropy, which is given by:

$$\mathcal{D}_\alpha = \left(\sum_{i=1}^{n} p_i^\alpha \right)^{\frac{1}{1-\alpha}}, \tag{1}$$

  where p_i is the probability that a randomly chosen host has vulnerability v_i, *n* is the total number of vulnerabilities among all the hosts, and α is the order of entropy. (For $\alpha = 1$, Rényi entropy converges to Shannon Entropy.) The authors use this measure to develop a game theoretic approach to analyze diversity in a network.

  Pros and Cons: Entropy is a common metric to measure the extent of the polyculture of software or hardware to achieve network diversity. However, entropy-based metrics may not be an effective measure of diversity if the variants share vulnerability to the same attack. In this case, a network with high entropy can even introduce higher vulnerabilities to identical attacks. In addition, entropy does not measure topological network diversity.
- *Resilience metrics:* Zhang *et al.* [177] devised three diversity-based metrics (i.e., d_1, d_2, d_3) to evaluate the resilience of a network in the presence of software diversity as follows: (1) d_1 is a biodiversity-based model, using the number of distinct resources, distribution of resources, and a measure of resource variety for evaluation. It is defined as the ratio of the network's effective richness to the total number of variants, i.e.,

$$d_1 = \frac{\frac{1}{\prod_{i=1}^{n} p_i^{p_i}}}{\sum_{i=1}^{n} |\mathrm{res}(h_i)|}, \tag{2}$$

  where p_i is the relative frequency of resource and $\mathrm{res}(h_i)$ is the resource mapping of host h_i to resource types; (2) d_2 evaluates the least attack effort to compromise hosts. It is defined as the ratio of the minimum number of resources in an attack path to the minimum number of steps in the attack path, i.e.,

$$d_2 = \frac{\min_{q \in \mathrm{seq}(g)} R(q)}{\min_{q \in \mathrm{seq}(g)} q'} \tag{3}$$

  where $\mathrm{seq}(g)$ is the number of attack paths to the goal condition g and $R(q)$ is the number of resources in q; and (3) d_3 is like d_2, but it focuses on the average attack effort. It is a probabilistic model defined by the ratio of the probability of given asset being attacked over the probability of a given asset being attacked with the condition of all variants being unique.

  Pros and Cons: Like entropy-based metrics, the d_1 metric works well assuming that variants are alike. The limitation of this metric is that it does not consider the causal relation between variants. The d_2 metric considers the causal relation between variants, but does not focus on enhancing security. It is also computationally costly to evaluate, but the cost can be reduced by estimating its value using heuristics. The d_3 metric provides a global view in terms of average vulnerabilities. Hence, it may not be able to distinguish between two networks with similar average vulnerabilities.

3) Diversity Deployment: Diversity can be further enhanced depending on how the existing diversity can be differently deployed. We call this *diversity deployment* and classify these types into two classes: metric-based and metric-free. We discuss each class as below.

- *Metric-based Diversity Deployment:* This class provides metrics to measure diversity in a proposed algorithm. For example, Temizkan *et al.* [156] proposed a software allocation model using Shannon entropy as a diversity metric (called the Software Diversity Index or SDI) to minimize the total information gain in the software assignment problem. The main drawback of this metric is high complexity as the problem is NP-hard. The authors proposed a heuristic algorithm to reduce the complexity and the SDI is given by:

$$\mathrm{SDI} = \sum_{i=1}^{v} p_i \log p_i \tag{4}$$

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

where p_i is the fraction of the number of node pairs in a network that do not have a common vulnerability *i* and the sum of the number of vulnerability dimensions, *v*, between each node pair. Borbor *et al.* [21] leveraged the software diversity metrics defined in [177] (i.e., d_1, d_2, d_3 metrics) to present their model-based technique. This work solved a software assignment problem aiming to optimize a max-min of d_1 and d_2 and a min-max of d_3 among all nodes in a network. This work leveraged a meta heuristic (i.e., a genetic algorithm) to solve the problem. Zhang *et al.* [178] proposed a metric to quantify the software diversity value of each node in a network and leveraged it to design a scheme used to adapt the corresponding network by edge addition and removal in each node's local network. By extensive comparative analysis, their results showed that the proposed scheme achieved a significant performance boost in both network resilience and security against epidemic attacks.

Pros and Cons: The use of metrics to measure diversity can provide a simple solution via maximizing the metric assuming that high diversity enhances network security. If the validity of a diversity metric does not hold, however, the relationship between network diversity and network security may not hold as well. In addition, high diversity can introduce high overhead as well as potential performance degradation (e.g., incompatibility between nodes). Moreover, solving an optimization problem using a diversity metric may have high complexity. Heuristics introduced to solve high complexity, such as meta heuristics, could be also computationally prohibitive in reaching an optimal solution.

- *Metric-free Diversity Deployment:* This deployment class does not use any metrics to measure diversity; rather it simply uses randomization or dynamic reconfiguration. To discuss metric-free diversity deployment in detail, we further classify this class into the following two sub-classes:
  - *Graph Coloring:* This approach, borrowed from graph theory [89], seeks to color a graph such that all pairwise connected nodes have different colors. This idea is reformulated into its variant in the domain of software diversity. Different software variants, representing different colors, are expected to be installed into each node in a computer communication network. Therefore, coloring techniques aim to assign different software versions to every pair of connected nodes. Leveraging this idea, a software assignment problem is solved by O'Donnell and Sethu [125] by developing different coloring algorithms. In addition, Huang *et al.* [80] studied the order of coloring based on priority determined using different centrality metrics. Taking this approach further, Touhiduzzaman *et al.* [158] introduced a game theoretic approach to solve a graph coloring problem by using different software versions to minimize vulnerabilities to epidemic attacks. In their game, the payoff is the security index (SI) that measures the complexity of the attack to break a security mechanism. The SI of node *i* is given by:
  $$\mathrm{SI}_i = \sum_{j \in N_i} |c_i \Psi_i - c_j \Psi_j| \qquad (5)$$
  where N_i is the set of node *i*'s neighboring nodes, c_i (or c_j) is an integer value representing a color (i.e., a software version) chosen by node *i* (or node *j*) and Ψ_i (or Ψ_j) is the software vulnerability of node *i* (or node *j*). Anwar *et al.* [4] proposed a graph-coloring algorithm for a network administrator to find an optimal diverse software deployment strategy that matches a Nash equilibrium. This Nash equilibrium condition maximizes the reward for a defender and minimizes the number of different types of software (or the cost for software deployment).

    Pros and Cons: Since the coloring problem has been studied for decades in mathematics and other engineering domains, its theoretical validity and maturity for algorithmic effectiveness and efficiency has been already proven and can be reliably leveraged. However, as the coloring problem has predominantly been studied in static networks, its applicability in dynamic network that can ensure efficiency as well as effectiveness is not fully proven. Moreover, simple repetition of the static network-based coloring algorithm may introduce high reconfiguration overhead.
  - *Network Topology Shuffling:* This technique aims to identify an optimal assignment of software variants to maximize the degree of software variants along attack paths. The main idea is to increase attack cost or complexity for an attacker by increasing hurdles in reaching a target node. Hong *et al.* [73] solved a network shuffling problem for software assignment as an online moving target defense (MTD) mechanism. Their proposed algorithm was designed to redirect a certain number of edges such that the reconfigured network topology can be more robust against worm attacks. Zhang *et al.* [179] developed a mechanism for networks with underlying multimedia services in order to redirect routes periodically by Deep Q-learning method, aiming to thwart DoS and targeted attacks to routes. For each step, DRL agents can determine the mutated routes for each node and flow from the source to the destination. Similar to [31], [73], [179] proposed 'a deep reinforcement learning-based moving target defense against DDoS attacks' called *DQ-MOTAG* where the MTD is adopted to shuffle the connections between users and servers in a given CPS. The DQ-MOTAG provides the ability to intelligently shuffle the duration of triggering the MTD operation based on reinforcement learning. The authors demonstrated the outperformance of DQ-MOTAG in terms of the system availability and performance.

    Pros and Cons: Shuffling techniques can cope with dynamic network structures because the cost of redirecting edges is relatively low. Furthermore, network

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

shuffling does not require assigning software variants. However, the complexity of network shuffling algorithms is proportional to the number of edges in a network, which may not scale for large networks. In addition, if a network is required to stay in the same network topology, network shuffling may not be applicable.

To introduce diversity into a system, existing dynamic reconfigurability techniques are often leveraged. Dynamic reconfigurability refers to the ability to dynamically reconfigure system settings, such as network topology and software resources. The reconfiguration process may introduce various types of diversification. For instance, dynamically reconfiguring network topology would bring path diversity while software resources reallocation would result in software diversity. To clarify the scope of our survey paper, we treat dynamic reconfiguration as a subset of diversity-based solutions because diversity-based approaches can be also applied in static network environments.

C. Hardware Diversity

The hardware of a CPS constitutes sensors and actuators, communication modules and antennas, and embedded devices [136]. The sensors and actuators form a bridge between the *cyber* and the *physical* parts. In sensors, their physical information is translated to electrical voltages or currents (the opposite in case of the actuators), usually in the order of milli-volts or milli-amperes. These components are vulnerable to false data injection, generally in the form of intentional electro-magnetic interference (IEMI) [101], [144]. To the best of our knowledge, diversity-based security techniques have not been studied yet for sensors and actuators. Therefore, in this section, we only discuss diversity-based techniques proposed for communication modules and antennas and embedded devices.

1) Communication Modules and Antennas: In a CPS, the communication module is responsible for the transmission and reception of information (i.e., control and data) between nodes. The medium of communication can be wired, for example, in IEEE 802.3 (Ethernet) standard or wireless, as in standards IEEE 802.11 (WiFi), IEEE 802.15.1 (Bluetooth), or IEEE 802.15.4 (Zigbee). In a wireless medium, diversity is usually achieved by employing multiple antennas for communication. From the physical layer security perspective, diversity is used to achieve high *secrecy capacity* or low *intercept probability*. These metrics quantify the ability of a wireless channel to protect its data from a malicious eavesdropper.

Zou *et al.* [182] discussed the effects of diversity on the physical layer security of a communication system, with the following three types of diversity: (1) *Multiple input multiple output* (MIMO) diversity where multiple antennas are used for transmitting and receiving nodes; (2) *Cooperative diversity* in which multiple relays (repeaters) are used between the transmitting and the receiving node; and (3) *Multiuser diversity*, where the transmitter node is communicating with multiple receiver nodes. Via experiments, the authors showed that diversity of the communication channel introduces higher secrecy capacity and lower intercept probability.

Watteyne *et al.* [168] designed routing protocols wherein the transmitting node sends succeeding packets on different communication channels, introducing channel hopping, a type of frequency diversity. The system is protected against communication failures in the path, considerably improving the reliability and connectivity as well as reducing the network churn. Zeng *et al.* [176] proposed a shared secret key generation protocol, called MAKE, for a multiple antenna system by leveraging the increase in randomness. With the contribution of generating high speed key generation and security against passive eavesdropping attacks. Sarkar and Ratnarajah [142] employed the channel diversity of a multiple antenna system to improve the secrecy capacity against an eavesdropper. The authors provided analysis to prove that the maximum-ratio combining diversity, where the signals from multiple antennas are weighted based on their strength and noise levels, improves the secrecy capacity of the communication system. Ghourab *et al.* [63] proposed that using re-configurable operating frequencies on the relay selection schemes improves the channel secrecy capacity and enhances the performance against an eavesdropping attack on the routing path in the network. The authors further enhanced the security by obfuscating the transmitted data via intentional injection of false data, thereby diversifying in both space and time. Tuset-Peiró *et al.* [161] proposed modulation diversity for IEEE 802.15.4-2015 nodes that use Smart Utility Networks (SUN) modulations. By using three different modulation techniques, namely, SUN-FSK, SUN-OQPSK and SUN-OFDM, the authors observe improvements in reliability of the network in terms of packet delivery rate (PDR).

Pros and Cons: Diversity-based security has been mainly developed by diversifying channel frequency, which has been proven highly effective for enhancing system security. However, most of the approaches require multiple antennas at the transmitter and receiver that increases hardware complexity and requires non-trivial signal processing. This leads to high power consumption, thus requiring lightweight solutions to realize diversity of channel frequency.

2) Embedded Devices: An embedded device, including microcontrollers, microprocessors, FPGAs or Application Specific Integrated Circuits (ASICs), is the core of a CPS. The embedded device acts as a central controller of the system as well as provides an interface between the cyber and physical aspects of the system. We discuss diversity-based designs in embedded devices as follows:

- *Architectural Diversity of FPGAs:* Lach *et al.* [102] exploited the intrinsic redundancy, flexibility and reconfigurability in the architecture of FPGAs to improve its fault tolerance and reliability. They suggest partitioning the physical design of the circuit into independent sets of tiles, containing logic blocks and resources. On the occurrence of a fault in a tile, a spare tile can be used to create an alternate but functionally equivalent logic. Karam *et al.* [91] used the architectural diversity in FPGAs to generate a different final executable file (i.e., bitstream) in each of the nodes in the network.

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

This increases difficulty in reverse engineering techniques performed by attacker, leading to enhanced security against tampering and piracy attacks with reasonable overhead.

- *Variants of Physical Layer Identification:* Even if devices are of the same model from the same manufacturer, there exists minor variations in the intrinsic characteristics at their physical layer, such as transients in the radio signals, clock skew, and other features. This diversity can be utilized for physical layer identification (PLI) and device fingerprinting as a security technique defending against impersonation and identity-theft attacks [60]. Gerdes *et al.* [61] used a matched-filter based approach to identify Ethernet devices from the variations in their analog signal. The authors were able to create a signal profile by applying the matched filter on 25 subsequent Ethernet frames and validated their approach by experimenting on 16 Ethernet cards of 3 different models. The cards of different models were near-perfectly detected while few cards of same model were hard to distinguish. Danev and Capkun [47] identified the IEEE 802.15.4 devices, used in the wireless sensor nodes (WSN) domain, by analyzing the unique variations in the turn-on transients of their radio transceivers at the beginning of transmission. They were able to achieve device identification with an equal error rate of 0.24% and perform a security analysis against DoS and hill-climbing impersonation attacks. Cobb *et al.* [39] introduced radio frequency distinct native attribute (RF-DNA) fingerprinting to identify embedded processors from their RF emissions to detect intrusions and prevent against impersonation attacks. Foruhandeh *et al.* [55] proposed a technique to identify Electronic Control Units (ECUs) in a Controller Area Network (CAN) bus architecture by exploiting the variations in their physical layer features. For the purpose of sender (transmitter) identification in the CAN protocol, Kneib *et al.* [95] used the fingerprint derived from signal characteristics based on rising edge of the square pulse of transmission. Their proposed methods achieved identification percentage of 99.98.
  Pros and Cons: This inherent variation between devices increases their resiliency against side-channel attacks. However, side-channel attacks are less effective if an attacker is trained on one device and tested on another device from the same manufacture and with an identical chip [167]. The attacker can improve its efficacy by training on a diverse set of devices and on varied implementations of the cryptographic algorithm. In addition, modulation-based identification as a PLI method may be vulnerable to signal and feature replay attacks while transient-based identification is more robust [47]. Hence, it is important to appropriately use a relevant diversity design to deal with the given attack scenario.

D. Software Diversity

Software diversity-based approaches have been substantially used to enhance system security as the key diversity-based design. Due to the large volume of studies explored in the literature, we classify based on the different types of diversities applied in: (i) operating systems; (ii) firewalls; (iii) intrusion detection systems (IDSs); (iv) malware detection; (v) cryptographic authentication; (vi) instruction diversification; and (vii) code obfuscation.

1) Operating Systems (OS): Forrest *et al.* [54] provided the following guidelines for designing OS-based diversity methods to preserve their convenience, usability, and efficiency: (i) preserve a high-level functionality; (ii) introduce diversity that can disrupt known intrusion most; and (iii) minimize deployment cost and run-time cost while maintaining sufficient diversity.

Pu *et al.* [133] developed a specialization toolkit for helping a programmer to improve the resistance of systematic specialization of OS kernels and against virus and worm attacks. The toolkit protects an OS by dynamically generating various versions of software components at compile-time specialization and run-time specialization.

Nagy *et al.* [119] applied an N-version technique on an OWASP (Open Web Application Security Project) to enhance the robustness against common vulnerabilities. In addition, the authors further discussed the feasibility of detecting zero-day attacks.

Garcia *et al.* [57] analyzed the vulnerabilities in 11 different OSs and showed that many vulnerabilities exist in more than one OS and the number of common vulnerabilities decreases if several OSs are combined. They also proposed a method to identify optimal composition of diverse OSs to improve intrusion tolerance. In addition, the authors found that reducing the number of days of gray-risk and the number of forever-day vulnerabilities are the main challenges for OS security.

Gorbenko *et al.* [67] designed an optimal intrusion-tolerant architecture composed of several different OSs in which a request will pass through these OSs synchronously. If the responses from these OSs are not the same, then an intrusion may happen. The authors showed that a 3-variant system is an optimum configuration providing the least vulnerabilities in availability and integrity.

Bulle *et al.* [24] leveraged OS diversity to build a novel intrusion detection model for SCADA. Their model is capable of proactively changing the underlying OS between Windows and Linux systems once an alarm is raised. Thus, intrusion detection accuracy can be improved by selecting the OS where the IDS behaves more reliably.

Pros and Cons: OS diversity techniques are high-level strategies in software architecture, which allow them to defend an attacker while having a certain number of code errors. However, there always exists a trade-off between the cost and diversity, and this conflict becomes more apparent in this category since most of the techniques require multiple OSs to run in parallel.

2) Firewalls: Liu and Gouda [112] proposed a process of designing diverse firewalls for enterprize security based on three processes: design, comparison, and resolution. The *design phase* lets multiple groups design a firewall policy independently based on the same requirement. The *comparison phase* detects function discrepancies between the multiple policies. The *resolution phase* generates a unified design for all groups. The authors also proposed three algorithms to identify

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

all functional discrepancies and estimate the impact of a policy change at the comparison phase. Based on [110], the authors also conducted a firewall policy impact analysis [111].

Pros and Cons: Diversity-based security mechanisms in firewalls are known to be very effective to deal with zero-day attacks [112]. However, if diverse software is configured or designed by the same group of people, they may share a common problem, which eliminates the advantage of diversity [169]. In addition, the research on the diversity-based firewall to enhance system security is still in its infancy, showing a lack of studies in this research area. This could be because of the overhead and potential errors introduced due to continuous firewall policy changes. Cost-effective firewall policy changes should be considered for security enhancement.

3) Intrusion Detection Systems (IDSs): Reynolds *et al.* [138], [139] proposed an intrusion tolerant control system for protecting identified users on the Internet. Their approach exploits commercial off-the-shelf (COTS)-based diversity, wherein multiple software systems are used in tandem for improving fault tolerance. The outputs (decisions) of each of the software systems are combined to detect and isolate attacks. In their implementation, the status code in the HTTP response from two different servers are compared to take a decision on an attack detection. The framework also provides continuous repair and enables the system to provide uninterrupted service even under the attack. Reference [157] built an IDS for Web servers incorporating COTS diversity to provide a high coverage detection while utilizing a masking mechanism to resolve false positive rate (FPR) generated due to the design differences in the diversified servers. This work considered attacks such as the loss of confidentiality, integrity, and availability of the servers in an offline evaluation test and later an online approach [115]. However, Totel *et al.* [157] also suggested that the FPR performance will be highly dependent on the particular COTS choices and the comparison scheme in the masking mechanism, potentially requiring frequent rule and tuning updates by an administrator to maintain the system.

Cox *et al.* [43] used the automated design of diversity to propose an *N*-variant based framework to provide security against a wide variety of attacks. The variants are generated in such a way that they behave similarly for normal inputs and differently for abnormal inputs (under an attack), hence enabling the monitor to detect anomalies. For example, in memory addressing, two disjoint memory space variants will work as expected for a relative memory address, but atleast one of them will have an error if the absolute memory address is provided. A proof-of-concept implementation is performed against code injection and memory corruption attacks. Majorczyk *et al.* [115] provided a 'masking mechanism' for an IDS to resolve the high FPR when applying COTS-based diversity. Instead of directly assigning a request to diverse components and comparing outputs, the masking function can modify the request before and after the request being processed by diverse components. Gu *et al.* [68] proposed a 'decision-theoretic alert fusion technique' to deal with the alarms from multiple IDSs. This technique is based on the likelihood ratio test (LRT) to combine different alert reports. In addition, since there is little work on analyzing the effectiveness of an IDS ensemble, this technique evaluates the effectiveness of the IDS ensemble by testing the LRT rule on two different datasets in advance.

Qu *et al.* [134] exploited diversity in the implementation of Web applications to develop a technique to defend against code injection attacks. They evaluated 16 Web applications written in four diverse languages, PHP, ASP, ASP.NET, and JSP against SQL injection vulnerabilities from the list of common vulnerabilities and exposures (CVE) [46]. Their results showed that the proposed approach has 0% False Positive Rate (FPR), 25.93% False Negative Rate (FNR), and 98.03% detection accuracy. All these results clearly exceed the single-stage counterpart.

Pros and Cons: IDS research is another field mainly studied within Enterprize Security or Web Server Security in which COTS diversity is used by many projects. The combination of various detectors is a well-known strategy that can achieve better performance. However, one main issue associated with the COTS-based IDS is that the FPR can increase due to the types of detection algorithms used in each detector and the ways of estimating diversity across different detectors. How to decrease the FPR while increasing the use of diversity-based IDSs is a promising direction [115].

4) Malware Detection: Oberheide *et al.* [123] proposed *N*-version protection based on a set of detection engines that run in parallel for detecting malicious files. They evaluated the performance of the proposed scheme with 7,220 unique malware data from NetScout systems [121]. They showed a significant increase of detection coverage as the number of engines increases. In addition, as the file detection threshold (the number of engines that need to be examined before a file is considered unsafe) increases, FPR decreases dramatically while the detection coverage decreases by less than 4%. In an another study, to analyze the potential benefit of the concept of the diverse Antivirus (AV), Gashi *et al.* [58] set up an experiment with 1,599 malware samples collected by deployed honeypots [107]. This experiment demonstrates that increasing the number of different AVs can decrease the detection failure rate.

Leveraging the benefits of using diverse AV software, Silva *et al.* [146] developed an AV system for the e-mail framework using different AVs running in parallel. They evaluated the system on e-mails containing malware. Zhou and Feng [18] reviewed the same dataset as [58] to study the detection gains when utilizing more than two AV products as well as to reduce 'at-risk-time' of the system. Hole *et al.* [71] provided a diversity software for an enterprise networked computer system to slow down or prevent the spreading of infectious malware and prevent zero-day exploits. Smutz and Stavrou [148] proposed a Random Forest (RF) machine learning technique to detect malicious PDF documents. The RF is an ensemble classifier constructed by multiple independent classification trees. Those individual trees are trained by randomly selecting feature data extracted from the structure and metadata of the PDF documents. The RF predicts benign or malicious PDF documents based on the votes from those trees.

Pros and Cons: Diversity-based malware detectors are known to be very effective in defending against zero-day attacks, compared to other security technologies, such as

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

anti-malware or patching. However, their downside is the consumption of more computing resources since different detectors are required to work in parallel, such as Silva's diversity-based antivirus software [146].

5) Cryptographic Authentication: Carvalho [28] proposed a redundancy and diversity-based design for cloud authentication systems to ensure security against unknown zero-day vulnerabilities. The key idea is to use redundant modules in the authentication system and apply design diversification to these modules to realize the diversity of the whole authentication system.

Pros and Cons: In the literature, diversity-based cryptographic authentication has been rarely studied as we only cited one work above [28]. The main reason for the lack of studies in this area would be because using multiple authentication protocols may introduce more complexity in system performance as well as incompatibility with other systems that use different authentication mechanisms.

6) Code Instruction Diversification: This technique is to diversify code instructions to prevent side-channel attacks, code modification attacks, or code replay attacks.

Cohen [40] proposed a method that can make a static program self-evolve over time for increasing attack complexity based on the *Instruction Equivalence* and *Instruction Reordering* technologies. Chew and Song [35] proposed lightweight methods for mitigating buffer overflow by randomizing system call mapping, global library entry point, and stack placement. Xu *et al.* [173] proposed a Transparent Runtime Randomization (TRR) method to defend against a wide range of attacks. The TRR, implemented by a program loader, dynamically relocates stack, heap, shared libraries in the memory space of a program. Kc *et al.* [92] proposed Instruction Set Randomization (ISR), an automated design diversity technique, wherein the machine instructions of the processor is modified by a key-based randomization algorithm and hence preventing the attacker from injecting a code in the *new* language. The instructions are decoded using the key at the time of execution. Reference [15] proposed Randomized Instruction Set Emulator (RISE), an ISR approach for the Valgrind x86-to-x86 binary translator. For every program execution, each byte in the codes is scrambled using a unique key and are de-scrambled during the fetch cycle. They evaluated their approach against multiple stack and heap overflow attacks. While [92] applied ISR in a key-based randomization algorithm, Barrantes *et al.* [15] used ISR in a broader sense by applying it in the whole simulation system. Sovarel *et al.* [151] discusses the effectiveness of the ISR against remote attackers.

Both Hu *et al.* [78] and Williams *et al.* [171] improved the performance of the Instruction Set Randomization (ISR) by leveraging the extended toolchain and combining static and dynamic binary rewriting. The former combined the ISR with the Advanced Encryption Standard (AES) to operate a dynamic translation to software and to improve the efficiency of the ISR. The latter proposed the Strata Virtual-Machine, which is a software dynamic translation (SDT) technique which combines the key ideas of CSD (calling sequence diversity) and ISR for increasing the diversity of binary code. This technique is designed for eliminating the return-to-libc and code injection attack.

Ichikawa *et al.* [85] developed a diversified instruction set architecture (ISA) to increase the redundancy of software. The key idea of the ISA is to change the encoding of opcode while keeping the original functionality of an instruction set. Franz [56] proposed a diversity version of compiler for the *App Store* that can automatically generate a unique but functionally identical software when download is requested. This technique aims to mitigate the risk of monoculture of software, and limit the success of attack on a small fraction of target. This technique also prohibits attackers discovering software vulnerabilities by reverse engineering security patches. Homescu [72] used profile-guided optimization to reduce the overhead in the NOP-based variants while maintaining its security benefits. Their method identifies the hot code in the program, which is the portion of the code which takes up most of the execution time. Diversification by NOP insertion is performed mostly on the hot-code, thereby, reducing the NOP insertion overhead from 25% to 1%. They perform a security analysis of their method against the return-oriented programming (ROP), a class of code reuse attacks. Koo and Polychronakis [98] used an instruction displacement to randomize the starting addresses of gadgets in the binary on installation phase.

Pros and Cons: Along with OS diversification, code instruction diversification research has been substantially explored. The key reason would be its less adverse impact on system performance while maintaining the original functionality of the code. However, it is inevitable that code instruction diversification introduces the complexity of the coding process and incurs high CPU overhead.

7) Code Obfuscation: This technique aims to transform code and make it unintelligible but still functional.

Collberg *et al.* [41] proposed four criteria to evaluate the degree of obfuscation, including *potency* (i.e., how much obscurity is added to a program), *resilience* (i.e., how hard is it for a deobfuscator to crack a proposed obfuscation technique), *stealth* (i.e., how well is the obfuscated code integrated into the program), and *cost* (i.e., how much overhead does the obfuscated code generate).

Crane *et al.* [44] proposed an approach to thwart cache/power side channel attacks by randomizing the control flow of programs. Replicas of certain program fragments are generated to produce an equivalent output with different implementations. Hence, the replicas can create several unique program execution paths over the space of dynamic and systematic choices of the replicas to use for each program fragment execution. This limits the attacker's ability to obtain information by monitoring cache and power consumption. The authors evaluated the resilience of a prototype diversifier against side-channel attacks attempting to discover cryptographic keys and demonstrated the approach can effectively mitigate the attack (fewer key bits found) with moderate overhead (1.5-2x slowdown) depending on parameter settings. Hataba *et al.* [69] proposed a technique to dynamically disrupt the control flow of a program so that the conditional branches will be converted randomly. This method is

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

TABLE III
LAYER-BASED CLASSIFICATION OF DIVERSITY-BASED DESIGNS FOR SECURE AND DEPENDABLE CYBER-PHYSICAL SYSTEMS

Layers	Diversity-based Designs & Techniques	Ref.
Diversity of Physical Environments	Diversified Physical Environments	[170]
	Diversified Physical Topology	[1, 27, 128, 147]
Network Diversity	Diversity generation	[120, 156]
	Diversity quantification	[120, 177]
	Diversity deployment	[21, 31, 73, 80, 89, 125, 156, 158, 178, 179]
Hardware Diversity	Communication modules and antennas	[63, 142, 161, 168, 176, 182]
	Embedded devices	[39, 47, 55, 60, 61, 91, 95, 102]
Software Diversity	Operating system	[24, 54, 57, 67, 119, 133]
	Firewall	[110, 111, 112]
	IDS	[43, 68, 115, 134, 138, 139, 140, 157]
	Malware detection	[18, 58, 71, 123, 146, 148]
	Cryptographic authentication	[28]
	Code instruction diversification	[15, 35, 40, 56, 72, 78, 85, 92, 98, 171, 173]
	Code obfuscation	[41, 44, 69, 99, 100, 129, 174]
Diversity for Human-Machine Interactions	Multiple user role assignment	[38]
	Multiple authentication process	[137]

designed for mitigating the side-channel attack in a cloud platform. Pawlowski *et al.* [129] proposed an obfuscation technique based on probabilistic control flow. The key idea is generating different, multiple execution traces while keeping semantics, given the same input values. Via experiments, the authors proved that their developed obfuscation prototype can effectively ensure divergent traces for the same input while enhancing resilience under dynamic attack analysis.

Kuang *et al.* [99], [100] enhanced the existing VM-based code obfuscation technique. After translating program instructions into bytecode instructions, the VM scheduler can segment those bytecodes into multiple sets and randomly select different but semantically equivalent handlers to insert those bytecode sets into the binary region that linked with a VM library. Xue *et al.* [174] proposed an obfuscation scheme, called *Code Virtualization Protection with Diversity* (DCVP), to increase the complexity even for experienced attackers to uncover the virtual instructions to native code when applying code virtualization for code obfuscation. The underlying idea of DCVP is to obfuscate the mapping between the opcodes and semantics for increasing the diversity of the program behavior.

Pros and Cons: Even though the code obfuscation is designed to transform code and make it unintelligible but still functional, the obfuscation cannot guarantee the irreversibility of the code. However, it can still increase the cost for the attacker to understand the functionality of the code, which can increase the opportunity time to protect the program [77]. A well-known drawback is that many technologies involved in running VMs in parallel require tremendous computational resources. In addition, a dynamic control-flow diversity technique, such as in [44], has an impact on execution time. The issue can be solved by generating replicas in advance at compiling time with the cost of increased memory consumption.

E. Diversity for Human-Machine Interactions

Human-machine interface deficiencies are commonly caused by interface design faults that can introduce data delay display or data misinterpretation. In addition, a single authorization mechanism, such as password only or biometric only, can expose security vulnerabilities. To enhance the security of an authorization system, diversity-based designs can be introduced [38], [137].

1) Multiple Role Assignment: Clark and Wilson [38] proposed a user level diversity strategy called the 'separation of duty' for military security systems. This strategy assigns the complementary roles to different users and makes the sensitive operation executable with different roles.

2) Multiple Authentication Process: Reiter [137] developed a protocol to force a sensitive operation for authorization to be run on different machines or programs controlled by independent operators.

Pros and Cons: Increasing diversity of human-machine interfaces can increase system security. However, due to humans' limited cognition, high diversity of the human-machine interfaces may introduce more mistakes or errors by humans. Huang *et al.* [81] investigated how human error diversity is related to software diversity under various conditions. Depending on the human operators' skill levels, the human error diversity is shown differently under a different level of software diversity. However, regardless of the skill levels of the human operators, the design of software diversity should be considered human-friendly to minimize human-prone mistakes or errors.

F. Summary and Discussions

This section provided an in-depth discussion of existing approaches under five categories: Diversity of physical environments, network diversity, hardware diversity, software diversity, and diversity for human-machine interactions. Under each category, we discussed pros and cons to provide critical insights to the corresponding diversity approaches. Table III provides an overview of the diversity-based approaches surveyed in this section. As shown in the table, most of the existing works focused on software diversity, hardware diversity, and network diversity.

V. ATTACK TYPES CONSIDERED BY DIVERSITY-BASED SECURITY APPROACHES

In this section, we mainly discuss what types of cyberattacks are defended by diversity-based security solutions. In addition, we discussed the limitations and gaps identified from the existing attack model in Section VII along with other limitations and insights learned from this survey paper.

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

A. Attack Types Countermeasured by Diversity-Based Security Approaches

The existing diversity-based security techniques have been designed and used to counter the following attacks:

- *Physical attack [1], [27], [147]:* A physical attack refers to the attack scenario where attackers attempt to break access control systems and physically access CPSs.
- *Zero-day attack [21], [24], [28], [38], [40], [57], [67], [71], [81], [112], [119], [123], [137], [138], [140]:* This attack utilizes unknown vulnerabilities where patches for the vulnerabilities are not available yet. Diverse but redundant authentications are used to thwart such attacks. The authentication is performed by comparing outputs of diverse implementations of system components [28], [119], [138], [140].
- *Worm attack [12], [36], [71], [75], [79], [80], [90], [124], [125], [133], [139], [156], [175], [178]:* A worm is a malicious computer program that can self-replicate and spread to other network computers. After the worm infects a machine, it can edit a file or monitor the machine. Software diversity can enhance the survivability of the Internet against worm attacks [12]. Graph coloring algorithms are also used to increase the diversity of software packages assuming that different software packages have the different degree or types of vulnerabilities [75], [79], [80], [124], [125], [156], [175]. Software diversity-based topology adaptation is another effective way to thwart such epidemic attacks [36], [178].
- *Code injection attack [15], [43], [58], [78], [92], [134], [171], [173]:* This attack injects a payload, which is usually binary, to a running application, and then forces the application to run the payload. Since the injected code can only work when an environment is compatible, random instructions generated for each program have been proposed when they are loaded to memory [92]. Since the attacker won't know the randomization value, the attacker cannot execute the payload properly.
- *Code reuse attack [72], [98], [171]:* This attack changes the function pointer of a program so that the program is going to execute malicious behavior. A profile-guided automated diversity approach to defend against code reuse attacks [98].
- *Return-to-Libc attack [171]:* This attack is often applied when a buffer overflow error occurs. The attacker usually has prior knowledge of the stack address. And then, it replaces the return address with the address of another subroutine, subsequently forcing the application to execute a library function with malicious arguments. To deal with this attack, the calling sequence diversity (CDS) of functions is used [171]. Since different programs have their respective calling sequence with CSD, the attack for one program cannot be easily propagated to other programs. An attacker may guess the key value for each function by constantly observing the function address although the attacker cannot use the same value for other programs.
- *Correlated attack [73]:* A correlated attack refers to an attack scenario where one compromised node's failure can cascade to the class of nodes it belongs to immediately. Under this situation, the attacker is assumed to have direct access to all nodes in the network. Graph coloring algorithms are used to optimally assign software variants to the network to maintain maximum connectivity and security [73].
- *Coordinated attack [158]:* This type of attack allows attackers to target multiple network assets simultaneously. Thus, software monoculture can introduce significant vulnerability in this scenario. Game-theoretic approaches are used to mitigate and thwart this kind of attack by optimally assigning different software packages in the network [158].
- *Buffer overflow attack [19], [35], [54], [132], [151]:* The buffer overflow attack targets services that automatically restart once a machine is crashed. This attack would first scan and read the stack to identify potential vulnerabilities and then remotely perform the write operation to steal a server's binary code. Note that attackers may overwrite the stack with their guesses until services crash and restart, which allows them to try more without being detected.
- *Side-channel attack [44], [69], [91], [160], [167], [180]:* A side-channel attack targets the implementation environment of an algorithm other than the algorithm itself. These environmental factors can be physical, such as power supply and acoustic variables, or non-physical, such as cache and running time during the execution of algorithms.
- *Deobfuscation attack [41], [129]:* This kind of attack tries to perform malicious reverse engineering on obfuscated code. Specifically, the attacker aims to undo the obfuscating transformations on the original program and retrieve valuable information out of it. The attacker may perform analysis through multiple traces in order to efficiently and effectively deobfuscate the target program [129].
- *Impersonation attack [39], [48], [55], [60], [61], [95], [148]:* This type of attack fools the identification system to disguise the malicious behavior if any. This is normally done by mimicking or replaying the features and signals extracted from normal communications with other nodes [48], [60].
- *Tampering attack [91]:* This attack targets a system's physical identity information, such as an IP address. For example, the attacker may keep sending requests to the system and analyze the returning bitstream [91]. After retrieving the identity information, the attacker may tamper with the systems' identity and launch other attacks, such as impersonation attacks.
- *Eavesdropping attack [63], [64], [142], [176], [182]:* This attack allows the attacker to passively gain information through the network communications. This can be done by installing malware or injecting a virus into compromised network clients. The stolen private information would later be stored and analyzed to engage in malicious activities.

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

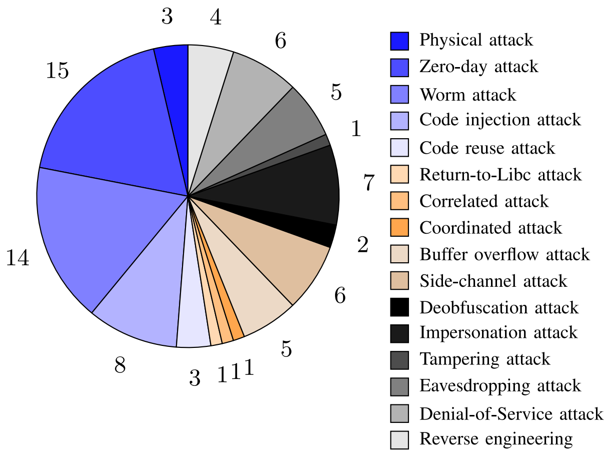


Fig. 2. Frequency of different attack types considered by the surveyed diversity-based security mechanisms for CPSs.

- *Denial of Service attack [31], [47], [66], [146], [157], [179]:* This type of attack typically sends out an extensive number of requests to servers, which causes a system's temporary overload and dysfunction. Normal valid requests would be rejected during the temporary shutdown period.
- *Reverse engineering [41], [56], [99], [174]:* Reverse engineering refers to the process where attackers could analyze and identify system components and their interrelationships so that attackers can further leverage them to reconstruct the system in a similar form.

B. Summary and Discussions

Various types of attacks can be defended by the existing diversity-based approaches. In this section, to provide an overview of those attacks, we summarized what types of attacks are considered in terms of the number of papers considering each attack type. As observed from Fig. 2, the top three common attacks of those considered are zero-day attack, worm attack (e.g., malware or virus propagation), and code injection attack. Since software diversity is a major trend in diversity-based approaches and software assignment research is primarily studied based on the concept of polyculture software following the fundamental principle of diversity in enhancing system survivability, it is natural to observe more efforts made in mitigating worm attacks in the existing literature.

VI. Metrics, Datasets, and Evaluation Testbeds

This section discusses how the existing diversity-based security solutions have been validated by using various types of metrics, datasets, and evaluation testbeds. Due to space constraints, we provided details of metrics, datasets, and testbeds in the Appendices B-D in the supplement document and discuss the key trends observed from our extensive survey in this section.

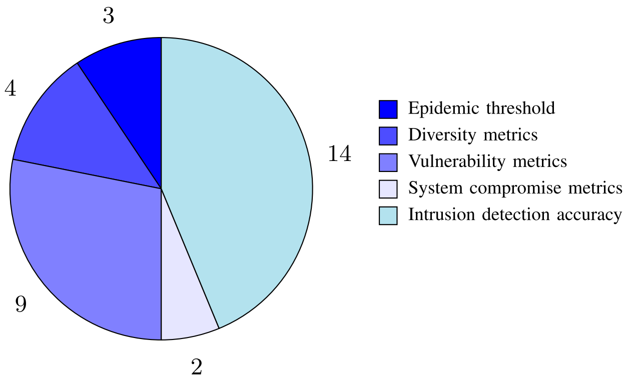


Fig. 3. Types and frequency of security metrics used in the surveyed diversity-based security approaches.

A. System Metrics

We discuss the metrics used to validate the existing diversity-based security solutions in terms of measuring security and dependability, respectively.

1) Security Metrics: Although various types of security metrics have been used to evaluate diversity-based approaches to enhance system security, we identified the following major categories of metrics used in the literature: (i) The epidemic thresholds representing the rate of infecting other nodes in malware or virus propagation; (ii) The extent of diversity measured in code, instructions, or routing paths; (iii) The metrics to capture system vulnerability (or exploitability) to attacks; (iv) The extent of compromised nodes or compromised routes in a given system or network; and (v) Intrusion detection accuracy in diversity-based IDSs. The detailed description of each metric belonging to one of these categories is provided in Appendix B of the supplement document. Based on the summary of these trends in Fig. 3, the majority of the diversity-based approaches have estimated the system security level based on the system's vulnerability to attacks. Although it seems clear that high diversity enhances system security, the adverse effect of using a diversity-based approach on system performance has not been thoroughly investigated.

2) Dependability Metrics: As discussed in Section III-B, dependability embraces availability, reliability, safety, integrity, and maintainability. We extensively surveyed dependability metrics that have been used to validate the quality of diversity-based approaches. However, due to the space constraint, we provided the detail of each dependability metric in Appendix C of the supplement document. Here, we discuss the overall trends found from our survey on the dependability metrics in diversity-based approaches.

In Fig. 4, we summarized the types and frequency of dependability metrics used in the existing diversity-based approaches. The major trends are: (1) Quality-of-Service (QoS) metrics are the dominant metrics used to capture system dependability, such as packet delivery or loss rates or delay; (2) Reliability is also captured based on load reduction caused by attacks; and (3) Maintenance cost is also observed, such as the financial cost to maintain multiple software packages (or versions).

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

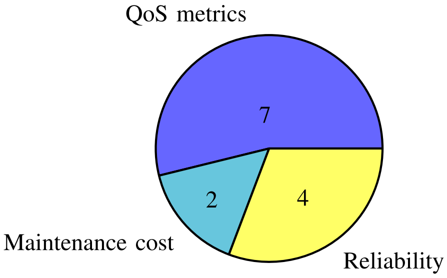


Fig. 4. Types and frequency of dependability metrics used in the surveyed diversity-based security approaches.

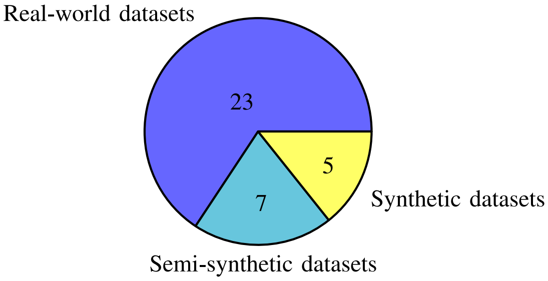


Fig. 5. Types and frequency of datasets used in the surveyed diversity-based security approaches.

B. Datasets

We examined 35 papers that have proposed diversity-based system design for secure and dependable CPSs. Based on this survey, we could categorize the following three types of datasets used for the validation of the proposed mechanisms: real-world datasets, semi-synthetic datasets, and synthetic datasets. As the names explain, the real-world datasets means that the data have been captured from real world environments, such as network traffics or attacks observed in real systems. The synthetic datasets are data generated by simulation that mimic the real world datasets. Sometimes when researchers cannot find the appropriate dataset to evaluate their proposed mechanism, they combined a real world dataset with synthetic dataset in order to make a dataset that can test the system security and dependability of their proposed mechanism. Due to space constraints, we provided Table II in the supplement document that provides the detail of each paper, 35 papers in total. In Fig. 5, we simply summarize the frequency of each dataset type used among the 35 papers surveyed in this work.

Based on Fig. 5, we can clearly observe that the most of the studies leveraged real world datasets to evaluate their proposed diversity-based approaches while only 5 works relied solely on synthetic datasets. We found that the most of the synthetic datasets are mainly for generating synthetic network topologies. Although various types of network datasets are available, there is still a limited amount of real 'communication network' datasets, resulting in generating synthetic datasets for network topologies.

C. Evaluation Testbeds

In the literature, we found mainly the following four types of evaluation testbeds being used: real testbeds, emulation-based testbeds, simulation testbeds, and analytical or probability model-based testbeds. In Section IV, we also discussed the multiple layers a diversity-based solution is deployed at. To grasp the overall picture of what testbeds have been used to validate the diversity-based approaches that are developed at a certain layer, we summarized the testbeds used for the validation of the existing diversity-based approaches in Fig. 6. In Fig. 6, each color represents the layer at which a diversity-based approach is deployed while we showed what evaluation testbed is used to validate each approach. Interestingly, real testbeds are popular for developing both hardware and software diversity-based approaches. The second most popular validation testbed was the analytical model-based one. Particularly for network diversity-based approaches, simulation-based testbeds are popularly used in the existing works.

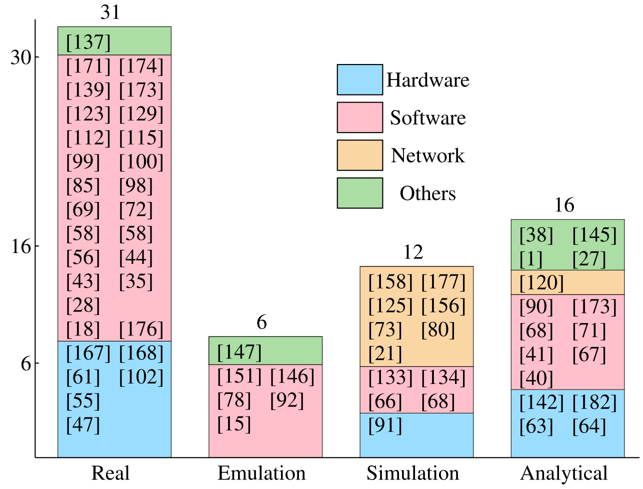


Fig. 6. Evaluation testbeds used in the surveyed diversity-based security approaches.

These observations are well aligned with the datasets used to validate the existing approaches. As discussed in Section VI-B, real-world datasets are the most popular in use while synthetic datasets based on simulation models are mainly used for network topologies, which are used in validating network diversity-based approaches.

D. Summary and Discussions

Security metrics attempt to predict or capture the outcome of adverse events. These metrics can determine phase transitions (e.g., such as in using epidemic thresholds), characterize the difficulty for an adversary in what it expects (e.g., diversity in software use), measure an adversary's success or advantage (e.g., vulnerability), assess the impact of an event (e.g., system compromise), and evaluate security performance (e.g., intrusion detection accuracy). Of these varieties of metrics, the two most prominent metrics are from opposing perspectives. The dependability metrics capture performance aspects, such as measures of services (e.g., Quality-of-Service, or QoS), ability to maintain services (e.g., reliability), and costs of services (e.g., maintainability). The literature reveals a strong

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

preference for real data to validate the security and dependability of CPSs utilizing a diversity paradigm. The different environments are necessary for different types of diversity-based approaches due to the difficulty in implementations. For instance, it is easier to attach a novel hardware to a real network than to recreate it in an emulated environment. Likewise, network diversity-based concepts are too difficult and expensive to evaluate in a real or emulated setting, at least at the basic research level.

VII. Limitations, Insights, and Lessons Learned

We found the following limitations and learned insights and lessons from this extensive survey:

- *Few studies identifying critical tradeoffs between system diversity and other aspects of system quality:* Although it is well-known that diversity-based system designs can significantly enhance system security, it may not be always true [32]. For example, if each software has inherently high vulnerabilities, increasing diversity with poor software components would not contribute to increasing system security. For example, using a set of diverse detectors may not necessarily lead to high system security. When each detector's detection capability is too poor (e.g., < 0.5), the system is still highly vulnerable due to the misdetection by the poor detectors. Although diversity-based design can be easily integrated with legacy security mechanisms and existing technologies, there should be studies investigating the critical tradeoff setting to identify the thresholds for achieving the benefit of diversity-based system designs.
- *Lack of research investigating the drawbacks of diversity-based system designs:* It is well known that introducing more diversity to the system can bring adverse impact on configuration cost, service availability, and economic cost. However, there has been less effort in investigating the key drawbacks of diversity-based system designs and how to mitigate the drawbacks.
- *Need more effort to explore diversity-based system designs in broader areas:* Based on our extensive survey in diversity-based system designs, we found that OSs, IDSs, malware detection, and instruction diversifications have been substantially studied. However, there have been significantly fewer studies developing diversity-based security mechanisms in firewalls and cryptographic authentication. The reason could probably be the benefit of using diversity-based approaches not exceeding that of not using them. However, no clear investigation has been even conducted to confirm this. Based on the critical tradeoff analysis of using diversity and not using it, we can set our research towards a more promising direction.
- *Less adverse impact of diversity-based designs at lower layers on system performance:* Diversity-based designs deployed at a lower layer (i.e., instruction diversification) tend to have a less adverse impact on system performance. On the other hand, when diversity-based designs are considered at higher layers, computational or memory resources tend to be more often required.
- *Integration of hardware diversity and software diversity:* Although there has been a fairly good amount of diversity-based approaches by introducing software diversity or hardware diversity, we have not found any research effort to explore diversity of both hardware and software, investigating the impact of an integrated approach.
- *Lack of research examining the relationships between diversity and other system dependability and security attributes:* As we can observe from Table II, until the 2000s, the primary effort of diversity-based system design was to increase software fault tolerance. Even if there have been more studies explored in the 2000s and the 2010s for investigating system security and dependability attributes, there have been many works that are still focused on enhancing fault tolerance. In addition, the relationships between diversity and other system attributes, such as confidentiality, maintainability, safety, and so forth, are still unclear.
- *Lack of deploying diversity-based approaches under dynamic system environments:* Some recent efforts have explored diversity-based approaches under dynamic system environments [36]. However, most current diversity-based research has been studied under static system environments where system components are fixed once diversity is implemented, such as code diversification, malware detectors, code instruction diversification or obfuscation, and so forth.
- *Limited theoretical understanding of diversity-based approaches:* Most diversity-based approaches have been validated based on simulation or emulation testbeds. Surely, the extensive experiments via simulation and emulation can provide a certain level of confidence on proposed technologies. However, validating their effectiveness and efficiency via mathematical and analytical models can further provide a solid basis of demonstrating their powerful merits on system security and dependability.
- *Lack of valid diversity metrics:* Most literature surveyed in our paper have not devised or used diversity metrics to quantify system diversity. Even though there are some studies that have proposed diversity metrics and their comparative analysis particularly in software assignment research [21], [114], [156], there is still a lack of studies that conduct in-depth analysis of various types of diversity metrics.

VIII. Conclusion and Future Work

In this section, we conclude this work with the summary of the key findings obtained from this extensive survey. And then, we suggest future work directions to develop diversity-based solutions to build secure and dependable CPSs.

From our extensive survey on diversity-based approaches, we obtained the following *key findings:*

- The key principle of diversity-based system designs is to enhance resilience, survivability, or sustainability of the system by increasing attack cost or complexity

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

for attackers to compromise the system by exploiting the same system vulnerabilities. However, deploying diversity-based approaches may introduce additional cost or performance degradation due potential cross incompatibility issues, maintenance cost, or high dynamic system/network reconfigurations.

- While software diversity, hardware diversity, and network diversity are the three most popular approaches used in the literature, diversity-based solutions for physical environments and human-machine interactions to enhance security and dependability are rarely explored.
- Although diversity-based approaches have been explored since the 1970s for enhancing system security and since the 2000s for both system security and dependability, the maturity of diversity metrics has not been reached for them to be used as general metrics like other security or dependability metrics (e.g., mean time to security failure, reliability, or availability). Entropy has been commonly used to capture uncertainty, representing a measure of randomness where higher diversity is assumed to show high uncertainty. However, as high entropy can be shown when there are not many variants of system components, it is highly questionable to simply use entropy as a diversity metric.
- We found that the three most popular attacks considered in the existing diversity-based approaches are worm attacks, zero-day attacks, and code injection attacks based on our survey. Since software diversity-based approaches are popularly used to increase network diversity, it is natural to observe that worm attacks performing epidemic attacks (e.g., malware or virus propagation) are the most popular attack type considered in the existing diversity-based techniques.
- We found that most diversity-based approaches use existing security metrics to capture their effect on security although it is not crystal clear that diversity can enhance security regardless of context or environmental conditions. Most security metrics are mainly based on the extent of system vulnerability to cyberattacks. The existing diversity-based approaches have also used dependability metrics that are most often used to measure Quality-of-Service (QoS) metrics (e.g., message delivery ratio, throughput, delay) while pure dependability metrics, including availability, reliability, or performability, have not been sufficiently considered.
- Unlike other cybersecurity research domains, the majority of diversity-based research used real datasets and real testbeds to validate the proposed diversity-based approaches (see Figs. 5 and 6). Most synthetic datasets and simulation models are used to evaluate network diversity-based approaches where the datasets represent network topologies and simulation models are used to evaluate network resilience under various epidemic attacks.

According to the lessons learned from Section VII, we suggest the following **future research directions**:

1) *Investigate critical tradeoffs between system diversity and other aspects of metric attributes:* Diversity offers potential for increased security and dependability in CPSs. For example, in the software diversity context, a vulnerability or glitch in a particular OS will be limited to the devices utilizing that OS. However, potential drawbacks may exist due to compatibility conflicts, particularly, when systems update. Moreover, as new security issues or vulnerabilities are discovered, these OSs require maintenance in terms of updates and possible upgrades. Therefore, the maintenance cost increases as the diversity increases. There may also be feasibility constraints due to the limited availability of OSs. Market forces can often result in few or even a single option for a required service, and the cost to manufacture or generate diversity may be prohibitive. Generally, for any diversity approach, there will be a need to balance the security and dependability versus potential performance degradation, maintenance or acquisition cost, or compatibility.
2) *Integrate hardware and software diversity-based approaches:* One approach to resolve the tradeoffs discussed above is to leverage whatever achievable benefits that are affordable in each of the different layers. The integration of these different approaches can be interpreted as a layering of diversity-based techniques. There exists substantial research efforts in each of the hardware and software diversity domains, but there has been no real effort to study the integration of these approaches. While this introduces a new set of integration issues to reconcile, this has the potential to multiply the advantages of diversity. If the expense of integration is not prohibitive, then existing approaches can be leveraged in a cost-effective manner. Certainly, this potential warrants investigations into the feasibility and merits of combining these approaches. One way to combine different hardware and software diversities is analyzing different diversity types in a general framework. For example, Laszka *et al.* [104] proposed uniformed metrics to quantify the cost and risk caused by diversity-based designs in their water distribution system and transportation network. Similarly, we can diversify both hardware and software and measure their diversity level to investigate its impact on system security and dependability.
3) *Broaden the areas to deploy/apply diversity-based approaches:* Diversity-based approaches have been applied most often in diversifying network topologies or system components. On the other hand, there has been insufficient research exploring diversity approaches in firewalls [110], [111], [112], cryptographic authentication [28], physical environments [1], [27], [147], [170], and human-machine interactions [38] as discussed in Section IV. This may be the case because introducing diversity-based designs can introduce high complexity, which can offset the positive benefits of enhancing system security and dependability. In order to mitigate such an adverse effect of diversity-based approaches in those domains, we need to develop the techniques to identify optimal settings that increase

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

the benefit of diversity-based approaches based on the three design principles (i.e., what, when, and where to diversify), with the aim of minimizing the performance degradation caused by deployed diversity-based security schemes. To support this, we should be able to quantify the impact introduced by diversity-based approaches, in terms of how much the level of increased diversity improves conventional system security metrics measuring the breach of security goals (e.g., confidentiality, integrity, availability). Further, we need to further delve into solutions for how and where to deploy firewalls, what cryptographic mechanisms to use in which system or network layer, and what security mechanisms machines need to use when interacting with different human users to prevent potential privacy and information leakage.

4) *Develop metrics to capture dynamics of environments:* Environmental conditions and their dynamics require vastly different approaches for diversity-based applications to balance the conflicting goals of system security and performance requirements. Future research directions should focus on developing meaningful metrics that can capture the multiple, critical aspects of system dynamics that affect the outcome of both security and performance goals. Some research has made an effort to address this issue by emphasizing dynamic security metrics with respect to the dynamics of attack arrivals and a system's intrusion detection and response processes [130]. In addition, the importance of dynamic system metrics is discussed in terms of resilience (i.e., a measure of the ability against performance degradation) and agility (i.e., a measure of the ability to deal with unexpected, sudden changes) [37]. However, the efforts are still in its infancy. Dynamic metrics can better characterize gains in security, such as changes in vulnerability, and losses in dependability, such as shifts in overhead, introduced by dynamic-based approaches, which enable optimization strategies to enhance both security and dependability.

5) *Develop meaningful diversity metrics:* As discussed in Section VI-A, the current research in diversity-based system designs mostly uses existing metrics that cannot capture the clear merits of diversity-based approaches in terms of system security and dependability. Although the positive relationships between the extent of diversity in system components or configurations and system security are obvious, the existing diversity metrics have not addressed the disadvantages of too high system diversity which may introduce high complexity or unbearable performance degradation. Developing diversity metrics balancing these two are not trivial but critical for building secure and dependable CPSs. As an example, the entropy metric in Eq. (1) and the resilience metrics in Eq. (2) and Eq. (3) only measure the performance gain by diversity-based design without considering the cost associated with calculating the diversity metrics. The future work also needs to investigate how to efficiently develop those metrics for large-scale networks.

6) *Explore the theoretical validation of diversity-based approaches:* As demonstrated in Figs. 5 and 6, we observed real-world datasets and real testbeds have been popularly used in the existing diversity-based approaches. Simulation or empirical validations are valuable; however, their applications may be limited to the domains considered by those existing approaches. Accordingly, we could not find much work showing theoretical validations of proposed diversity-based approaches, which may introduce hurdles for those proposed approaches to be used as more generic frameworks with high applicability in various domains. For example, the future work needs to investigate how diversity-based designs can introduce enhanced security and dependability in terms of the relationships between the increased level of diversity and system availability, reliability, and dependability in the presence of various attackers.

Acknowledgment

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] N. B. Akhuseyinoglu and J. Joshi, "A risk-aware access control framework for cyber-physical systems," in *Proc. IEEE 3rd Int. Conf. Collaboration Internet Comput. (CIC)*, 2017, pp. 349–358.

[2] H. Alavizadeh, D. S. Kim, and J. Jang-Jaccard, "Model-based evaluation of combinations of shuffle and diversity MTD techniques on the cloud," *Future Gener. Comput. Syst.*, vol. 111, pp. 507–522, Oct. 2020.

[3] J. Antunes and N. Neves, "DiveInto: Supporting diversity in intrusion-tolerant systems," in *Proc. IEEE 30th Int. Symp. Rel. Distrib. Syst.*, 2011, pp. 137–146.

[4] A. H. Anwar, N. O. Leslie, C. Kamhoua, and C. Kiekintveld, "A game theoretic framework for software diversity for network security," in *Proc. Int. Conf. Decis. Game Theory Security*, 2020, pp. 297–311.

[5] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Depend. Secure Comput.*, vol. 1, no. 1, pp. 11–33, Jan.–Mar. 2004.

[6] A. Avizienis, "On the implementation of *n*-version programming for software fault tolerance during execution," in *Proc. COMPSAC*, 1977, pp. 149–155.

[7] A. Avizienis, "The *N*-version approach to fault-tolerant software," *IEEE Trans. Softw. Eng.*, vol. SE-11, no. 12, pp. 1491–1501, Dec. 1985.

[8] A. Avizienis and J. P. Kelly, "Fault tolerance by design diversity: Concepts and experiments," *Computer*, vol. 17, no. 8, pp. 67–80, Aug. 1984.

[9] A. Avizienis and J.-C. Laprie, "Dependable computing: From concepts to design diversity," *Proc. IEEE*, vol. 74, no. 5, pp. 629–638, May 1986.

[10] M. Azab, R. Hassan, and M. Eltoweissy, "ChameleonSoft: A moving target defense system," in *Proc. 7th Int. Conf. Collaborative Comput. Netw. Appl. Worksharing (CollaborateCom)*, 2011, pp. 241–250.

[11] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, "Robust malware detection for Internet of (battlefield) Things devices using deep eigenspace learning," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 1, pp. 88–95, Jan.–Mar. 2019.

[12] M. G. Bailey, "Malware resistant networking using system diversity," in *Proc. ACM 6th Conf. Inf. Technol. Educ.*, 2005, pp. 191–197.

[13] A. Balakrishnan and C. Schulze, "Code obfuscation literature survey," in *Proc. CS 701 Construct. Compilers*, vol. 19, Dec. 2005.

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

[14] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[15] E. G. Barrantes, D. H. Ackley, S. Forrest, T. S. Palmer, D. Stefanovic, and D. D. Zovi, "Randomized instruction set emulation to disrupt binary code injection attacks," in *Proc. 10th ACM Conf. Comput. Commun. Security*, 2003, pp. 281–289.

[16] M. Batty *et al.*, "Smart cities of the future," *Eur. Phys. J. Topics*, vol. 214, pp. 481–518, Nov. 2012.

[17] B. Baudry and M. Monperrus, "The multiple facets of software diversity: Recent developments in year 2000 and beyond," *ACM Comput. Surveys*, vol. 48, no. 1, pp. 1–26, Sep. 2015.

[18] P. Bishop, R. Bloomfield, I. Gashi, and V. Stankovic, "Diversity for security: A study with off-the-shelf antivirus engines," in *Proc. IEEE 22nd Int. Symp. Softw. Rel. Eng.*, 2011, pp. 11–19.

[19] A. Bittau, A. Belay, A. Mashtizadeh, D. Mazières, and D. Boneh, "Hacking blind," in *Proc. IEEE Symp. Security Privacy*, 2014, pp. 227–242.

[20] BlueOnix. *BlueOnix: Network Datasets*. Accessed: Jan. 11, 2020. [Online]. Available: http://www.nvd.org/

[21] D. Borbor, L. Wang, S. Jajodia, and A. Singhal, "Optimizing the network diversity to improve the resilience of networks against unknown attacks," *Comput. Commun.*, vol. 145, pp. 96–112, Sep. 2019.

[22] J. Brilha, M. Gray, D. Pereira, and P. Pereira, "Geodiversity: An integrative review as a contribution to the sustainable management of the whole of nature," *Environ. Sci. Policy*, vol. 86, pp. 19–28, Aug. 2018.

[23] S. S. Brilliant, J. C. Knight, and N. G. Leveson, "The consistent comparison problem in N-version software," *IEEE Trans. Softw. Eng.*, vol. 15, no. 11, pp. 1481–1485, Nov. 1989.

[24] B. B. Bulle, A. O. Santin, E. K. Viegas, and R. R. dos Santos, "A host-based intrusion detection model based on OS diversity for SCADA," in *Proc. 46th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, 2020, pp. 691–696.

[25] T. R. Butler. *The Open Source IA-32 Emulation Project (Home Page)*. Accessed: Jan. 11, 2020. [Online]. Available: http://bochs.sourceforge.net/

[26] J. Canto, M. Dacier, E. Kirda, and C. Leita, "Large scale malware collection: Lessons learned," in *Proc. IEEE SRDS Workshop Sharing Field Data Exp. Meas. Resilience Distrib. Comput. Syst.*, 2008, pp. 1–6.

[27] Y. Cao, Z. Huang, Y. Yu, C. Ke, and Z. Wang, "A topology and risk-aware access control framework for cyber-physical space," *Front. Comput. Sci.*, vol. 14, no. 4, pp. 1–16, Jan. 2020.

[28] R. Carvalho, "Authentication security through diversity and redundancy for cloud computing," M.S. thesis, Inf. Syst. Comput. Eng., TÉCNICO LISBOA, Lisbon, Portugal, 2014.

[29] C. G. Cassandras, "Smart cities as cyber-physical social systems," *Engineering*, vol. 2, no. 2, pp. 156–158, Jun. 2016.

[30] R. Chaâri *et al.*, "Cyber-physical systems clouds: A survey," *Comput. Netw.*, vol. 108, pp. 260–278, Oct. 2016.

[31] X. Chai, Y. Wang, C. Yan, Y. Zhao, W. Chen, and X. Wang, "DQ-MOTAG: Deep reinforcement learning-based moving target defense against DDoS attacks," in *Proc. IEEE 5th Int. Conf. Data Sci. Cyberspace (DSC)*, 2020, pp. 375–379.

[32] H. Chen, J.-H. Cho, and S. Xu, "Quantifying the security effectiveness of network diversity: Poster," in *Proc. 5th Annu. Symp. Bootcamp Hot Topics Sci. Security (HoTSoS)*, 2018, p. 24. [Online]. Available: https://doi.org/10.1145/3190619.3191680

[33] J. Chen and Q. Zhu, "Security as a service for cloud-enabled Internet of Controlled Things under advanced persistent threats: A contract design approach," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2736–2750, Nov. 2017.

[34] P.-Y. S. Chen, G. Kataria, and R. Krishnan, "Software diversity for information security," in *Proc. Workshop Econ. Inf. Security (WEIS)*, Boston, MA, USA, Jun. 2005.

[35] M. Chew and D. Song, "Mitigating buffer overflows by operating system randomization," CMU Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Rep. CMU-CS-02-197, Dec. 2002.

[36] J.-H. Cho and T. J. Moore, *Nature-Inspired Cyber Security and Resiliency: Fundamentals, Techniques and Applications*. Stevenage, U.K.: Inst. Eng. Technol., 2019, pp. 313–342.

[37] J.-H. Cho, S. Xu, P. M. Hurley, M. Mackay, T. Benjamin, and M. Beaumont, "STRAM: Measuring the trustworthiness of computer-based systems," *ACM Comput. Surveys*, vol. 51, no. 6, pp. 1–47, Feb. 2019.

[38] D. D. Clark and D. R. Wilson, "A comparison of commercial and military computer security policies," in *Proc. IEEE Symp. Security Privacy*, 1987, pp. 184–184.

[39] W. E. Cobb, E. W. Garcia, M. A. Temple, R. O. Baldwin, and Y. C. Kim, "Physical layer identification of embedded devices using RF-DNA fingerprinting," in *Proc. Mil. Commun. Conf. (Milcom)*, 2010, pp. 2168–2173.

[40] F. B. Cohen, "Operating system protection through program evolution," *Comput. Security*, vol. 12, no. 6, pp. 565–584, Oct. 1993.

[41] C. Collberg, C. Thomborson, and D. Low, "Manufacturing cheap, resilient, and stealthy opaque constructs," in *Proc. 25th ACM SIGPLAN-SIGACT Symp. Principles Program. Lang.*, 1998, pp. 184–196.

[42] V. Corral-Verdugo, M. Bonnes, C. Tapia-Fonllem, B. Fraijo-Sing, M. Frías-Armenta, and G. Carrus, "Correlates of pro-sustainability orientation: The affinity towards diversity," *J. Environ. Psychol.*, vol. 29, no. 1, pp. 34–43, Mar. 2009.

[43] B. Cox *et al.*, "N-variant systems: A secretless framework for security through diversity," in *Proc. USENIX Security Symp.*, 2006, pp. 105–120.

[44] S. Crane, A. Homescu, S. Brunthaler, P. Larsen, and M. Franz, "Thwarting cache side-channel attacks through dynamic software diversity," in *Proc. NDSS*, 2015, pp. 8–11.

[45] C. Crouch and W. Streeck, *The Diversity of Democracy : Corporatism, Social Order and Political Conflict*. Cheltenham, U.K.: Edward Elgar, Jan. 2006.

[46] *Common Vulnerabilities and Exposures (CVE)*. Accessed: Jan. 11, 2020. [Online]. Available: http://cve.mitre.org/

[47] B. Danev and S. Capkun, "Transient-based identification of wireless sensor nodes," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, 2009, pp. 25–36.

[48] B. Danev, H. Luecken, S. Capkun, and K. El Defrawy, "Attacks on physical-layer identification," in *Proc. 3rd ACM Conf. Wireless Netw. Security*, 2010, pp. 89–98.

[49] D. C. DeLong, "Defining biodiversity," *Wildlife Society Bull.*, vol. 24, no. 4, pp. 738–749, 1996.

[50] D. Ding, Q. Han, Z. Wang, and X. Ge, "A survey on model-based distributed control and filtering for industrial cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2483–2499, May 2019.

[51] J. Dobaj, M. Krisper, and G. Macher, "Towards cyber-physical infrastructure as-a-service (CPIaaS) in the era of industry 4.0," in *Proc. Eur. Conf. Softw. Process Improvement*, 2019, pp. 310–321.

[52] Z. Drias, A. Serhrouchni, and O. Vogel, "Analysis of cyber security for industrial control systems," in *Proc. Int. Conf. Cyber Security Smart Cities Ind. Control Syst. Commun. (SSIC)*, 2015, pp. 1–8.

[53] A. S. Elmaghraby and M. M. Losavio, "Cyber security challenges in smart cities: Safety, security and privacy," *J. Adv. Res.*, vol. 5, no. 4, pp. 491–497, Jul. 2014.

[54] S. Forrest, A. Somayaji, and D. H. Ackley, "Building diverse computer systems," in *Proc. 6th Workshop Hot Topics Oper. Syst.*, 1997, pp. 67–72.

[55] M. Foruhandeh, Y. Man, R. Gerdes, M. Li, and T. Chantem, "SIMPLE: Single-frame based physical layer identification for intrusion detection and prevention on in-vehicle networks," in *Proc. 35th Annu. Comput. Security Appl. Conf.*, 2019, pp. 229–244.

[56] M. Franz, "E unibus pluram: Massive-scale software diversity as a defense mechanism," in *Proc. New Security Paradigms Workshop*, 2010, pp. 7–16.

[57] M. Garcia, A. Bessani, I. Gashi, N. Neves, and R. Obelheiro, "Analysis of operating system diversity for intrusion tolerance," *Softw. Practice Exp.*, vol. 44, no. 6, pp. 735–770, Jun. 2014.

[58] I. Gashi, V. Stankovic, C. Leita, and O. Thonnard, "An experimental study of diversity with off-the-shelf antivirus engines," in *Proc. 8th IEEE Int. Symp. Netw. Comput. Appl.*, 2009, pp. 4–11.

[59] I. Gashi, A. Povyakalo, L. Strigini, M. Matschnig, T. Hinterstoisser, and B. Fischer, "Diversity for safety and security in embedded systems," in *Proc. Int. Conf. Depend. Syst. Netw.*, vol. 26, 2014, pp. 06–2014.

[60] R. Gerdes, "Physical layer identification: Methodology, security, and origin of variation," Ph.D. dissertation, Dept. Elect. Comput. Eng., Iowa State Univ., Ames, IA, USA, 2011. [Online]. Available: https://lib.dr.iastate.edu/etd/10257

[61] R. M. Gerdes, T. E. Daniels, M. Mina, and S. Russell, "Device identification via analog signal fingerprinting: A matched filter approach," in *Proc. NDSS*, 2006, p. 6.

[62] A. Gharaibeh *et al.*, "Smart cities: A survey on data management, security, and enabling technologies," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2456–2501, 4th Quart., 2017.

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

[63] E. M. Ghourab, A. Mansour, M. Azab, M. Rizk, and A. Mokhtar, “Towards physical layer security in Internet of Things based on reconfigurable multiband diversification,” in *Proc. 8th IEEE Annu. Inf. Technol. Electron. Mobile Commun. Conf. (IEMCON)*, 2017, pp. 446–450.

[64] E. M. Ghourab, M. Azab, and A. Mansour, “Spatiotemporal diversification by moving-target defense through benign employment of false-data injection for dynamic, secure cognitive radio network,” *J. Netw. Comput. Appl.*, vol. 138, pp. 1–14, Jul. 2019.

[65] L. Gmeiner and U. Voges, “Software diversity in reactor protection systems: An experment,” in *Safety of Computer Control Systems*. Amsterdam, The Netherlands: Elsevier, 1980, pp. 75–79.

[66] M. S. Gondal, A. J. Malik, and F. A. Khan, “Network intrusion detection using diversity-based centroid mechanism,” in *Proc. 12th Int. Conf. Inf. Technol. New Gener.*, 2015, pp. 224–228.

[67] A. Gorbenko, A. Romanovsky, O. Tarasyuk, and O. Biloborodov, “From analyzing operating system vulnerabilities to designing multiversion intrusion-tolerant architectures,” *IEEE Trans. Rel.*, vol. 69, no. 1, pp. 22–39, Mar. 2020.

[68] G. Gu, A. A. Cárdenas, and W. Lee, “Principled reasoning and practical applications of alert fusion in intrusion detection systems,” in *Proc. ACM Symp. Inf. Comput. Commun. Security*, 2008, pp. 136–147.

[69] M. Hataba, R. Elkhouly, and A. El-Mahdy, “Diversified remote code execution using dynamic obfuscation of conditional branches,” in *Proc. IEEE 35th Int. Conf. Distrib. Comput. Syst. Workshops*, 2015, pp. 120–127.

[70] Hex Rays. *IDA Home Page*. Accessed: Jan. 11, 2020. [Online]. Available: https://www.hex-rays.com/index.shtml

[71] K. J. Hole, “Diversity reduces the impact of malware,” *IEEE Security Privacy*, vol. 13, no. 3, pp. 48–54, May/Jun. 2013.

[72] A. Homescu, S. Neisius, P. Larsen, S. Brunthaler, and M. Franz, “Profile-guided automated software diversity,” in *Proc. IEEE/ACM Int. Symp. Code Gener. Optim. (CGO)*, 2013, pp. 1–11.

[73] J. B. Hong, S. Yoon, H. Lim, and D. S. Kim, “Optimal network reconfiguration for software defined networks using shuffle-based online MTD,” in *Proc. IEEE 36th Symp. Rel. Distrib. Syst. (SRDS)*, 2017, pp. 234–243.

[74] L. Hong and S. E. Page, “Groups of diverse problem solvers can outperform groups of high-ability problem solvers,” *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 46, pp. 16385–16389, Nov. 2004.

[75] S. Hosseini, M. A. Azgomi, and A. T. Rahmani, “Malware propagation modeling considering software diversity and immunization,” *J. Comput. Sci.*, vol. 13, pp. 49–67, Mar. 2016.

[76] S. Hosseinzadeh *et al.*, “A survey on aims and environments of diversification and obfuscation in software security,” in *Proc. 17th Int. Conf. Comput. Syst. Technol.*, 2016, pp. 113–120.

[77] S. Hosseinzadeh *et al.*, “Diversification and obfuscation techniques for software security: A systematic literature review,” *Inf. Softw. Technol.*, vol. 104, pp. 72–93, Dec. 2018.

[78] W. Hu *et al.*, “Secure and practical defense against code-injection attacks using software dynamic translation,” in *Proc. 2nd Int. Conf. Virtual Execution Environ.*, 2006, pp. 2–12.

[79] C. Huang, S. Zhu, and R. Erbacher, “Toward software diversity in heterogeneous networked systems,” in *Proc. IFIP Annu. Conf. Data Appl. Security Privacy*, 2014, pp. 114–129.

[80] C. Huang, S. Zhu, Q. Guan, and Y. He, “A software assignment algorithm for minimizing worm damage in networked systems,” *J. Inf. Security Appl.*, vol. 35, pp. 55–67, Aug. 2017.

[81] F. Huang, B. Liu, Y. Song, and S. Keyal, “The links between human error diversity and software diversity: Implications for fault diversity seeking,” *Sci. Comput. Program.*, vol. 89, pp. 350–373, Sep. 2014.

[82] Y. Huang and A. K. Ghosh, “Introducing diversity and uncertainty to create moving attack surfaces for Web services,” in *Moving Target Defense*. New York, NY, USA: Springer, 2011, pp. 131–151.

[83] Y. Huang, A. K. Ghosh, T. Bracewell, and B. Mastropietro, “A security evaluation of a novel resilient Web serving architecture: Lessons learned through industry/academia collaboration,” in *Proc. IEEE/IFIP Int. Conf. Depend. Syst. Netw. Workshops (DSN-W)*, 2010, pp. 188–193.

[84] A. Humayed, J. Lin, F. Li, and B. Luo, “Cyber-physical systems security—A survey,” *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017.

[85] S. Ichikawa, T. Sawada, and H. Hata, “Diversification of processors based on redundancy in instruction set,” *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 91, no. 1, pp. 211–220, Jan. 2008.

[86] *ISO/TC Document ISO/TC 176/SC 1 N 93*. Accessed: Feb. 1992. [Online]. Available: https://www.iso.org/committee/53888.html

[87] T. Jackson, A. Homescu, S. Crane, P. Larsen, S. Brunthaler, and M. Franz, “Diversifying the software stack using randomized NOP insertion,” in *Moving Target Defense II*. New York, NY, USA: Springer, 2013, pp. 151–173.

[88] A. Jangda, M. Mishra, and B. De Sutter, “Adaptive just-in-time code diversification,” in *Proc. 2nd ACM Workshop Moving Target Defense (MTD)*, 2015, pp. 49–53.

[89] T. R. Jensen and B. Toft, *Graph Coloring Problems*, vol. 39. New York, NY, USA: Wiley, 2011.

[90] J. E. Just and M. Cornwell, “Review and analysis of synthetic diversity for breaking monocultures,” in *Proc. ACM Workshop Rapid Malcode*, 2004, pp. 23–32.

[91] R. Karam, T. Hoque, S. Ray, M. Tehranipoor, and S. Bhunia, “MUTARCH: Architectural diversity for FPGA device and IP security,” in *Proc. 22nd Asia South Pac. Design Autom. Conf. (ASP-DAC)*, 2017, pp. 611–616.

[92] G. S. Kc, A. D. Keromytis, and V. Prevelakis, “Countering code-injection attacks with instruction-set randomization,” in *Proc. 10th ACM Conf. Comput. Commun. Security*, 2003, pp. 272–280.

[93] J. Kephart, G. Sorkin, M. Swimmer, and S. White, *Blueprint for a Computer Immune System*. Berlin, Heidelberg: Springer, 1999, pp. 242–261.

[94] V. Kharchenko, “Diversity for safety and security of embedded and cyber physical systems: Fundamentals review and industrial cases,” in *Proc. 15th Biennial Baltic Electron. Conf. (BEC)*, 2016, pp. 17–26.

[95] M. Kneib, O. Schell, and C. Huth, “EASI: Edge-based sender identification on resource-constrained platforms for automotive networks,” in *Proc. Netw. Distrib. Syst. Security Symp.*, 2020, pp. 1–16.

[96] J. C. Knight and N. G. Leveson, “An experimental evaluation of the assumption of independence in multiversion programming,” *IEEE Trans. Softw. Eng.*, vol. SE-12, no. 1, pp. 96–109, Jan. 1986.

[97] M. Kocakulak and I. Butun, “An overview of wireless sensor networks towards Internet of Things,” in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC)*, 2017, pp. 1–6.

[98] H. Koo and M. Polychronakis, “Juggling the gadgets: Binary-level code randomization using instruction displacement,” in *Proc. 11th ACM Asia Conf. Comput. Commun. Security*, 2016, pp. 23–34.

[99] K. Kuang *et al.*, “Exploiting dynamic scheduling for VM-based code obfuscation,” in *Proc. IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 489–496.

[100] K. Kuang, Z. Tang, X. Gong, D. Fang, X. Chen, and Z. Wang, “Enhance virtual-machine-based code obfuscation security through dynamic bytecode scheduling,” *Comput. Security*, vol. 74, pp. 202–220, May 2018.

[101] D. F. Kune *et al.*, “Ghost talk: Mitigating EMI signal injection attacks against analog sensors,” in *Proc. IEEE Symp. Security Privacy*, 2013, pp. 145–159.

[102] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, “Algorithms for efficient runtime fault recovery on diverse FPGA architectures,” in *Proc. IEEE Int. Symp. Defect Fault Tolerance in VLSI Syst. (EFT)*, 1999, pp. 386–394.

[103] P. Larsen, A. Homescu, S. Brunthaler, and M. Franz, “SoK: Automated software diversity,” in *Proc. IEEE Symp. Security Privacy*, 2014, pp. 276–291.

[104] A. Laszka, W. Abbas, Y. Vorobeychik, and X. Koutsoukos, “Integrating redundancy, diversity, and hardening to improve security of industrial Internet of Things,” *Cyber Phys. Syst.*, vol. 6, no. 1, pp. 1–32, 2020.

[105] C. Leita, “SGNET: Automated protocol learning for the observation of malicious threats,” in *Proc. 7th Eur. Depend. Comput. Conf. (EDDC)*, 2008, pp. 1–11. [Online]. Available: https://www.eurecom.fr/publication/2445

[106] C. Leita and M. Dacier, “SGNET: Implementation insights,” in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, 2008, pp. 1075–1078.

[107] C. Leita and M. Dacier, “SGNET: A worldwide deployable framework to support the analysis of malware threat models,” in *Proc. 7th Eur. Depend. Comput. Conf.*, 2008, pp. 99–109.

[108] J. Leskovec and A. Krevl. (Jun. 2014). *SNAP Datasets: Stanford Large Network Dataset Collection*. [Online]. Available: http://snap.stanford.edu/data

[109] B. Littlewood, P. Popov, and L. Strigini, “Modeling software design diversity: A review,” *ACM Comput. Surveys*, vol. 33, no. 2, pp. 177–208, Jun. 2001.

[110] A. X. Liu and M. G. Gouda, “Diverse firewall design,” in *Proc. Int. Conf. Depend. Syst. Netw.*, 2004, pp. 595–604.

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

[111] A. X. Liu, "Change-impact analysis of firewall policies," in *Proc. Eur. Symp. Res. Comput. Security*, 2007, pp. 155–170.

[112] A. X. Liu and M. G. Gouda, "Diverse firewall design," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 9, pp. 1237–1251, Sep. 2008.

[113] M. R. Lyu and A. Avižienis, "Assuring design diversity in N-version software: A design paradigm for N-version programming," in *Dependable Computing for Critical Applications 2*. Vienna, Austria: Springer, 1992, pp. 197–218.

[114] M. R. Lyu, J.-H. Chen, and A. Avizienis, "Software diversity metrics and measurements," in *Proc. 16th Annu. Int. Comput. Softw. Appl. Conf.*, 1992, pp. 69–78.

[115] F. Majorczyk, É. Totel, and L. Mé, "Experiments on cots diversity as an intrusion detection and tolerance mechanism," in *Proc. 1st Workshop Recent Adv. Intrusion-Tolerant Syst. (WRAITS)*, 2007, pp. 28–32.

[116] I. Malynyak, "Functional diversity design of safety-related systems," *Educ. Rev.*, vol. 2, no. 1, pp. 147–154, 2018.

[117] B. Middleton, *A History of Cyber Security Attacks: 1980 to Present*. Boca Raton, FL, USA: CRC Press, 2017.

[118] Q. Mou, H. Ye, and Y. Liu, "Enabling highly efficient eigen-analysis of large delayed cyber-physical power systems by partial spectral discretization," *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1499–1508, Mar. 2020.

[119] L. Nagy, R. Ford, and W. Allen, "N-version programming for the detection of zero-day exploits," in *Proc. IEEE Topical Conf. Cybersecurity*, Daytona Beach, FL, USA, 2006, pp. 52–59.

[120] S. Neti, A. Somayaji, and M. E. Locasto, "Software diversity: Security, entropy and game theory," in *Proc. HotSec*, 2012, pp. 1–8.

[121] *DDoS Solutions*. Accessed: Jan. 11, 2020. [Online]. Available: https://www.netscout.com/arbor-ddos

[122] D. Nicol, W. Sanders, and K. Trivedi, "Model-based evaluation: From dependability to security," *IEEE Trans. Depend. Secure Comput.*, vol. 1, no. 1, pp. 48–65, Jan.–Mar. 2004.

[123] J. Oberheide, E. Cooke, and F. Jahanian, "CloudAV: N-version antivirus in the network cloud," in *Proc. USENIX Security Symp.*, 2008, pp. 91–106.

[124] A. J. O'Donnell and H. Sethu, "Software diversity as a defense against viral propagation: Models and simulations," in *Proc. 19th Workshop Principles Adv. Distrib. Simulat.*, 2005, pp. 247–253.

[125] A. J. O'Donnell and H. Sethu, "On achieving software diversity for improved network security using distributed coloring algorithms," in *Proc. 11th ACM Conf. Comput. Commun. Security*, 2004, pp. 121–131.

[126] *Winning the Future With Science and Technology for 21st Century Smart Systems*. Accessed: Jan. 11, 2020. [Online]. Available: https://www.nitrd.gov/pubs/CPS-OSTP-Response-Winning-The-Future.pdf

[127] M. Parkour. (2011). *11 355+ Malicious Documents-Archive for Signature Testing and Research*. Accessed: Jan. 11, 2020. [Online]. Available: http://contagiodump.blogspot.com/2010/08/malicious-documents-archive-for.html

[128] A. S. Patrick, A. C. Long, and S. Flinn, "HCI and security systems," in *Proc. Extended Abstracts Human Factors Comput. Syst. (CHI)*, 2003, pp. 1056–1057.

[129] A. Pawlowski, M. Contag, and T. Holz, "ProbFuscation: An obfuscation approach using probabilistic control flows," in *Proc. Int. Conf. Detect. Intrusions Malware Vulnerability Assess.*, 2016, pp. 165–185.

[130] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," *ACM Comput. Surveys*, vol. 49, no. 4, pp. 1–35, Dec. 2016. [Online]. Available: https://doi.org/10.1145/3005714

[131] M. Polése and R. Stren, *The Social Sustainability of Cities: Diversity and the Management of Change*, Toronto, ON, Canada: Univ. Toronto Press, 2000.

[132] M. Prasad and T.-C. Chiueh, "A binary rewriting defense against stack based buffer overflow attacks," in *Proc. USENIX Annu. Techn. Conf. Gen. Track*, 2003, pp. 211–224.

[133] C. Pu, A. P. Black, C. Cowan, J. Walpole, and C. Consel, "A specialization toolkit to increase the diversity of operating systems," in *Proc. ICMAS Workshop Immunity Based Syst.*, Nara, Japan, 1996.

[134] W. Qu, W. Huo, and L. Wang, "Opportunistic diversity-based detection of injection attacks in Web applications," *EAI Endorsed Trans. Security Safety*, vol. 5, no. 16, p. e5, Oct.–Dec. 2018.

[135] B. Randell, "System structure for software fault tolerance," *IEEE Trans. Softw. Eng.*, vol. SE-1, no. 2, pp. 220–232, Jun. 1975.

[136] F. Regazzoni and I. Polian, "Securing the hardware of cyber-physical systems," in *Proc. 20th Unibus Pluram 22nd Asia South Pac. Design Autom. Conf. (ASP-DAC)*, 2017, pp. 194–199.

[137] M. Reiter, "Distributing trust with the rampart toolkit," *Commun. ACM*, vol. 39, no. 4, pp. 71–74, Apr. 1996.

[138] J. Reynolds, J. Just, E. Lawson, L. Clough, and R. Maglich, "On-line intrusion protection by detecting attacks with diversity," in *Research Directions in Data and Applications Security*. Boston, MA, USA: Springer, 2003, pp. 245–256.

[139] J. Reynolds, J. Just, E. Lawson, L. Clough, R. Maglich, and K. Levitt, "The design and implementation of an intrusion tolerant system," in *Proc. Int. Conf. Depend. Syst. Netw.*, 2002, pp. 285–290.

[140] J. C. Reynolds, J. Just, L. Clough, and R. Maglich, "On-line intrusion detection and attack prevention using diversity, generate-and-test, and generalization," in *Proc. 36th Annu. Hawaii Int. Conf. Syst. Sci.*, 2003, p. 8.

[141] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, Jul. 2013.

[142] M. Z. I. Sarkar and T. Ratnarajah, "Enhancing security in correlated channel with maximal ratio combining diversity," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6745–6751, Dec. 2012.

[143] SecurityFocus. *Securityfocus*. Accessed: Jan. 11, 2020. [Online]. Available: https://www.securityfocus.com/

[144] J. Selvaraj, G. Y. Dayanıklı, N. P. Gaunkar, D. Ware, R. M. Gerdes, and M. Mina, "Electromagnetic induction attacks against embedded systems," in *Proc. Asia Conf. Comput. Commun. Security*, 2018, pp. 499–510.

[145] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[146] C. Silva, P. Sousa, and P. Veríssimo, "RAVE: Replicated antivirus engine," in *Proc. Int. Conf. Depend. Syst. Netw. Workshops (DSN-W)*, 2010, pp. 170–175.

[147] N. Skandhakumar, F. Salim, J. Reid, and E. Dawson, "Physical access control administration using building information models," in *Cyberspace Safety and Security*. Heidelberg, Germany: Springer, 2012, pp. 236–250.

[148] C. Smutz and A. Stavrou, "Malicious PDF detection using metadata and structural features," in *Proc. 28th Annu. Comput. Security Appl. Conf.*, 2012, pp. 239–248.

[149] C. Smutz and A. Stavrou, "When a tree falls: Using diversity in ensemble classifiers to identify evasion in malware detectors," in *Proc. NDSS*, 2016, p. 9.

[150] F. Song, Z. Ai, H. Zhang, I. You, and S. Li, "Smart collaborative balancing for dependable network components in cyber-physical systems," *IEEE Trans. Ind. Informat.*, early access, Oct. 9, 2020, doi: 10.1109/TII.2020.3029766.

[151] A. N. Sovarel, D. Evans, and N. Paul, "Where's the FEEB? The effectiveness of instruction set randomization," in *Proc. USENIX Security Symp.*, vol. 10, 2005, pp. 1–8.

[152] N. P. Steyn, J. H. Nel, G. Nantel, G. Kennedy, and D. Labadarios, "Food variety and dietary diversity scores in children: Are they good indicators of dietary adequacy?" *Public Health Nutrit.*, vol. 9, no. 5, pp. 644–650, Aug. 2006.

[153] K. Stouffer, J. Falco, and K. Scarfone, *Guide to Industrial Control Systems (ICS) Security*, document NIST SP 800-82, Jun. 2011.

[154] M. Taguinod, A. Doupé, Z. Zhao, and G.-J. Ahn, "Toward a moving target defense for Web applications," in *Proc. IEEE Int. Conf. Inf. Reuse Integr. (IRI)*, 2015, pp. 510–517.

[155] NWB Team. (2006). *Network Workbench Tool. Indiana University, Northeastern University, and University of Michigan*. [Online]. Available: http://www.med.umich.edu/schnell-lab/software/network-workbench/

[156] O. Temizkan, S. Park, and C. Saydam, "Software diversity for improved network security: Optimal distribution of software-based shared vulnerabilities," *Inf. Syst. Res.*, vol. 28, no. 4, pp. 828–849, Aug. 2017.

[157] E. Totel, F. Majorczyk, and L. Mé, "COTS diversity based intrusion detection and application to Web servers," in *Proc. Int. Workshop Recent Adv. Intrusion Detect.*, 2005, pp. 43–62.

[158] M. Touhiduzzaman, A. Hahn, and A. Srivastava, "A diversity-based substation cyber defense strategy utilizing coloring games," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5405–5415, Sep. 2019.

[159] K. S. Trivedi, D. S. Kim, A. Roy, and D. Medhi, "Dependability and security models," in *Proc. 7th Int. Workshop Design Rel. Commun. Netw.*, pp. 11–20.

[160] E. Tromer, D. A. Osvik, and A. Shamir, "Efficient cache attacks on AES, and countermeasures," *J. Cryptol.*, vol. 23, no. 1, pp. 37–71, Jan. 2010.

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.

[161] P. Tuset-Peiró, F. Adelantado, X. Vilajosana, and R. D. Gomes, "Reliability through modulation diversity: Can combining multiple IEEE 802.15. 4-2015 SUN modulations improve PDR?" in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, pp. 1–6.

[162] P. Tuset-Peiró, R. D. Gomes, P. Thubert, E. Cuerva, E. Egusquiza, and X. Vilajosana, "A dataset to evaluate IEEE 802.15.4g SUN for dependable low-power wireless communications in industrial scenarios," *MDPI Data*, vol. 5, no. 3, p. 64, Jul. 2020.

[163] University of California. (1999). *KDD Cup 1999 Data*. Accessed: Jan. 11, 2020. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[164] M. Van Der Meulen, "On the use of smart sensors, common cause failure and the need for diversity," in *Proc. 6th Int. Symp. Program. Electron. Syst. Safety Relat. Appl.*, 2004, p. 6.

[165] D. Van Knippenberg, C. K. De Dreu, and A. C. Homan, "Work group diversity and group performance: An integrative model and research agenda," *J. Appl. Psychol.*, vol. 89, no. 6, pp. 1008–1022, 2004.

[166] *Virustotal*. Accessed: Jan. 11, 2020. [Online]. Available: https://www.virustotal.com/gui/home/upload

[167] H. Wang, M. Brisfors, S. Forsmark, and E. Dubrova, "How diversity affects deep-learning side-channel attacks," in *Proc. IEEE Nordic Circuits Syst. Conf. (NORCAS) NORCHIP Int. Symp. Syst. Chip (SoC)*, 2019, pp. 1–7.

[168] T. Watteyne, A. Mehta, and K. Pister, "Reliability through frequency diversity: Why channel hopping makes sense," in *Proc. 6th ACM Symp. Perform. Eval. Wireless Ad Hoc Sensor Ubiquitous Netw.*, 2009, pp. 116–123.

[169] W. Weber, "Firewall basics," in *Proc. 4th Int. Conf. Telecommun. Modern Satellite Cable Broadcast. Services (TELSIKS)*, vol. 1, 1999, pp. 300–305.

[170] S. H. Weingart, "Physical security devices for computer subsystems: A survey of attacks and defenses," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2000, pp. 302–317.

[171] D. Williams, W. Hu, J. W. Davidson, J. D. Hiser, J. C. Knight, and A. Nguyen-Tuong, "Security through diversity: Leveraging virtual machine technology," *IEEE Security Privacy*, vol. 7, no. 1, pp. 26–33, Jan./Feb. 2009.

[172] *Xen Project*. Accessed: Jan. 11, 2020. [Online]. Available: https://xenproject.org/

[173] J. Xu, Z. Kalbarczyk, and R. K. Iyer, "Transparent runtime randomization for security," in *Proc. 22nd Int. Symp. Rel. Distrib. Syst.*, 2003, pp. 260–269.

[174] C. Xue *et al.*, "Exploiting code diversity to enhance code virtualization protection," in *Proc. IEEE 24th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, 2018, pp. 620–627.

[175] Y. Yang, S. Zhu, and G. Cao, "Improving sensor network immunity under worm attacks: A software diversity approach," *Ad Hoc Netw.*, vol. 47, pp. 26–40, Sep. 2016.

[176] K. Zeng, D. Wu, A. Chan, and P. Mohapatra, "Exploiting multiple-antenna diversity for shared secret key generation in wireless networks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[177] M. Zhang, L. Wang, S. Jajodia, A. Singhal, and M. Albanese, "Network diversity: A security metric for evaluating the resilience of networks against zero-day attacks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 5, pp. 1071–1086, May 2016.

[178] Q. Zhang, J.-H. Cho, T. J. Moore, and I.-R. Chen, "Vulnerability-aware resilient networks: Software diversity-based network adaptation," *IEEE Trans. Netw. Service Manag.*, early access, Dec. 28, 2020, doi: 10.1109/TNSM.2020.3047649.

[179] T. Zhang *et al.*, "DQ-RM: Deep reinforcement learning-based route mutation scheme for multimedia services," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2020, pp. 291–296.

[180] Y. Zhou and D. Feng, "Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing," in *Proc. IACR Cryptol. ePrint Archive*, vol. 2005, Oct. 2005, p. 388.

[181] R. Zhuang, S. Zhang, S. DeLoach, X. Ou, and A. Singhal, "Simulation-based approaches to studying effectiveness of moving-target network defense," in *Proc. Nat. Symp. Moving Target Res.*, 2012, p. 9.

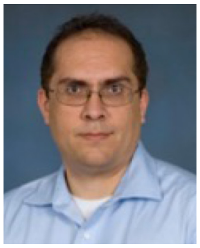
[182] Y. Zou, J. Zhu, X. Wang, and V. C. Leung, "Improving physical-layer security in wireless communications using diversity techniques," *IEEE Netw.*, vol. 29, no. 1, pp. 42–48, Jan./Feb. 2015.

Qisheng Zhang received the B.S. degree in mathematics from Shandong University in 2017, and the M.S. degree in mathematics from the University of Warwick in 2018. He is currently pursuing the Ph.D. degree with the Department Computer Science, Virginia Tech. His research interests include network security and network science.

Abdullah Zubair Mohammed received the B.E. degree in electronics and communication from Osmania University in 2012, and the M.Tech. degree in communication and signal processing from the Indian Institute of Technology Hyderabad in 2017. He is currently pursuing the Ph.D. degree with the Bradley Department of Electrical and Computer Engineering, Virginia Tech. Before starting his Ph.D., he worked as a Design Engineer with Silicon Laboratories, Inc. (formerly, Redpine Signals, Inc.) in 2017. His research interests include hardware security, device fingerprinting, embedded systems security, and cyber-physical systems security.

Zelin Wan received the B.S. degree in computer science from the University of Arizona, Tucson, AZ, USA, in 2019. He is currently pursuing the Ph.D. degree with the Department Computer Science, Virginia Tech, Falls Church, VA, USA. His research interests include game theoretic and machine learning-based cybersecurity and network science.

Jin-Hee Cho (Senior Member, IEEE) received the M.S. and Ph.D. degrees in computer science from the Virginia Tech in 2004 and 2008, respectively, where she has been an Associate Professor with the Department of Computer Science since 2018. Prior to joining the Virginia Tech, she worked as a Computer Scientist with the U.S. Army Research Laboratory, Adelphi, MD, USA, in 2009. She has published over 150 peer-reviewed technical papers in leading journals and conferences in the areas of trust management, cybersecurity, metrics and measurements, network performance analysis, resource allocation, agent-based modeling, uncertainty reasoning and analysis, information fusion/credibility, and social network analysis. She received the best paper awards in IEEE TrustCom'2009, BRIMS'2013, IEEE GLOBECOM'2017, 2017 ARL's Publication Award, and IEEE CogSima 2018. She is a winner of the 2015 IEEE Communications Society William R. Bennett Prize in the Field of Communications Networking. In 2016, he was selected for the 2013 Presidential Early Career Award for Scientists and Engineers. She is a member of the ACM.



Terrence J. Moore (Member, IEEE) received the B.S. and M.A. degrees in mathematics from American University in 1998 and 2000, respectively, and the Ph.D. degree in mathematics from the University of Maryland, College Park, in 2010. He is currently a Researcher with the Network Science Division, U.S. Army Research Laboratory. His research interests include sampling theory, constrained statistical inference, stochastic optimization, network security, geometric and topological applications in networks, and network science.

Authorized licensed use limited to: Bangladesh Univ of Engineering and Technology. Downloaded on May 20,2022 at 08:18:58 UTC from IEEE Xplore. Restrictions apply.