Software Security Requirement Engineering for Risk and Compliance Management

D. Kavitha, S. Ravikumar

Abstract: ***The objective of the research work is to propose a software based security requirement engineering model using categorical and morphisms theory. The earlier security requirement engineering models focus different viewpoints on parallel processing and develop rewrite based knowledge centred models but does not include different functional mappings between the security objects to select the best strategy. The security models have not considered the needed security functions that are to be implemented in different environments with different levels of executions. The proposed requirement engineering model is based on the formal theory of category of objects and the morphisms between them in addition to n categories and multiple morphisms that were used to organize the security requirement functional objects of different categories. The on demand security requirement objects, morphisms and the uncertain events in any one of the subsystems are considered to manage this security requirement category as an algebraic data types. The collection of security requirement objects using classification and clustering techniques are implicitly applied by the formation of category and morphism. The risk and compliances both in the form of direct and indirect categories are mapped so as to provide a security assurance functors with minimum risk on the requirements to the next design state. An 'n' category and 'n' morphic model for software security requirement model is proposed towards for minimum security risks through efficient compliance management techniques.***

Keywords: Categorical Theory, Security Objects, Requirement Morphisms, Functions and Functors, Compliance Management.

I. INTRODUCTION

Security requirements are most important so that it can be done at the early stage of software lifecycle. It is a non-functional requirement and it has typical security properties like confidentiality, integrity, availability, accountability and access control. Software engineering design methodologies, formal verification, simulation, and various other techniques have been devised to aid in the production of error free software [1]. Dynamic network security architecture was built on the technologies of software defined networking, Virtual Machine (VM) traffic redirection, network policy unified management, software defined isolation networks, vulnerability scanning, and software updates. The security supporting mechanisms, tenants should have the ability to know the vulnerabilities of their VMs and the vulnerability scanning framework acts as a service to tenants, based on a lightweight monitoring agent for IaaS platforms [2]. In multi-tenant cases, the network boundaries are blurring. With the increase of tenants, the data centre network topology along becomes complicated. Multiple tenants put their data in cloud and the same tenant may utilize different servers with multiple backups, making the network boundaries between each tenant become blurred and virtualized rather than traditional physical isolation [3]. The security requirements in VANET include message authentication, identity privacy preservation, traceability, unlinkability, non-repudiation and various attack resistance [4]. The distinct IoT security and privacy features, including security requirements, threat models, and attack taxonomies from the health care perspective. IoT-based healthcare services are expected to reduce costs, increase the quality of life, and enrich the user's experience. From the perspective of healthcare providers, the IoT has the potential to reduce device downtime through remote provision. In addition, the IoT can correctly identify optimum times for replenishing supplies for various devices for their smooth and continuous operation [5]. Requirements like authorization, authentication, or access control were mentioned in six of the articles while requirements like anonymity, unlinkability, resilience to failure, or emergency access where only mentioned in one article [6]. Secure software systems in the software development process are known as Security Requirements Engineering [7]. software security cannot be just added after a system has been built and delivered to customers as seen in today's software applications whereas security requirements elicitation and analysis based upon the construction of a context for the system, representation of security requirements as constraints, and satisfaction arguments for the requirements in the system context[8,9]. Security is about the prevention of several difficulties due to the presence of attackers behaving malicious activities so that security should be tackled at the beginning of the software lifecycle Requirement elicitation is the process of determining, understanding, reporting, and realizing the user requirements and constraints for the system. Requirements analysis is the process of focusing the user's requirements and constraints [10]. The specification and the testing of such policies are the fundamental steps in the development of a secure system since any error in a set of rules is likely to harm the global security. To ensure that a certain level of security is always maintained, the system behaviour must be restrained by a security policy. A security policy is a set of rules that regulates the nature and the context of actions that can be performed within a system, according to specific roles [11].

Manuscript received on March 03, 2021.
Revised Manuscript received on March 11, 2021.
Manuscript published on March 30, 2021.
* Correspondence Author
D.Kavitha*, Department of Computer Science and Engineering, SRM Valliammai Engineering College, Chennai, India. Email: dkavitha2005@gmail.com
S.Ravikumar, Department of Information Technology, SRM Valliammai Engineering College, Chennai, India. Email: dr.sravikumarit@gmail.com

The security issue in distributed system includes security of information, physical security in distributed system and security of network and authentication policy [12]. Security goals and security requirements aim to protect assets from harm. . Primary security goals are operationalized into primary security requirements, which take the form of constraints on the functional requirements sufficient to protect the assets from identified harms. Primary security requirements are, consequently, preventative. Feasibility, trade-off, and conflict analyses may lead to the addition of secondary security goals, which result in additional functional and/or secondary security requirements [13, 14]. Safety-critical and security-critical software systems are dynamic and interactive resulting in having unintentional hazards. The upgrading process is continuous as the main objective of monitoring the residual risk and its compliance to the standards and certificate [15][17][18][19][20][21][22][23][24][25].

II. CATEGORICAL THEORY IN SOFTWARE SECURITY REQUIREMENTS AND COMPLIANCE

. Category theory is a mapping between object and morphism whereas object can be thought of as sets and arrows and they are not limited to interpret by set theory. The object is related to other objects in the category through morphism. The security category and compliance category have much number of objects so that in order to map the objects in the category set theory is not sufficient for mapping between objects and morphism so category theory is used for object mapping and morphism. An 'n' category is an algebraic structure consists of objects and morphism between two objects and more which leads up to 'n' morphism. The figure 5 describes the composition of zero dimension and zero morphism which corresponds to gluing together an arrow Leads to: Direct attack☐ Direct threat and an arrow Extends: Direct threat☐ Direct method, whereas the arrow includes: Direct method☐Direct damage so as to obtain leads to: includes: Direct attack ☐ Direct damage is shown in figure 1.

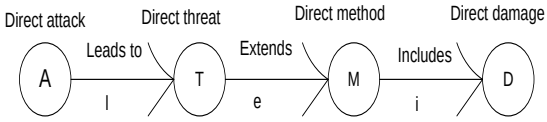

Figure 1: 0 Dimension and 0 morphism

The Zero morphism for the zero dimension object can be written as:

$$l :: A \rightarrow T$$

$$e :: T \rightarrow M \quad and \quad e \ after \ l \rightarrow M$$

$$i :: M \rightarrow D \ are \ used \ so \ as \ to \ obtain$$

$$i \ after \ e \ after \ l = A \rightarrow D$$

$$i \circ e \circ l = A \rightarrow D \qquad \rightarrow \qquad (1)$$

When visualizing the 2-morphisms as 2-dimensional, and compose 2-morphisms in a way that corresponds to gluing together 2-dimensional shapes. Of course, we should choose some particular shapes for our 2-morphisms. For example,

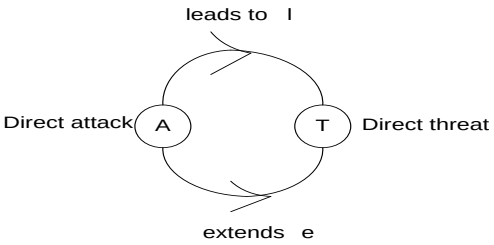

Figure 2: Two Dimension and Two morphism

The two morphism for the two dimension object can be written as:

l :: A→T

e :: A→T to obtain

e after l = A→T

e ∘ l = A →T → (2)

Figure 2 describes two dimension with 2 morphism whereas leads to: Direct attack →Direct threat and the arrow Extends: Direct attack→ Direct threat so as to obtain leads to : extends Direct attack→ Direct threat.

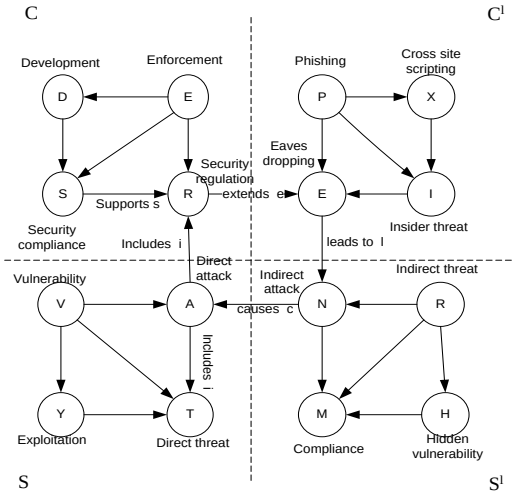

Fig 3: Mapping Security and Compliance

Figure 3 shows the mapping of security category objects with compliance category objects can be done and the morphism between the objects can be written as follows:

s :: S →R

e :: R →E and

i :: E →N

c :: N →A and

i :: A →T to obtain

i after c after i after e after s =S →T

i ∘ c ∘ i ∘ e ∘ s = S→T → (3)

III. CATEGORICAL MAPPING BETWEEN SECURITY AND COMPLIANCE OBJECTS

The software security requirement engineering includes objects or services belonging to various categories like the infrastructure category, platform category, vulnerability category, direct attack category, services category, exploitation category, direct threat category, direct data category, direct method category and direct damage category are grouped under security which is denoted as S.

The indirect attack category, insider threat category, indirect management category, indirect damage category, compliance category, hidden vulnerability category, indirect risk category, platform category, hidden services category and virtual infrastructure category are grouped under security inverse category which is denoted as Sl. The direct risk category, legality and law category, maintenance category, enforcement category, development category, design and architecture category, security governance category, security management category, security compliance category and security regulation category are grouped under Compliance category which is denoted by C. The side effect category, site channel category crypto attack category, phishing category, eaves dropping category, covert channel category, cross site scripting category, insider threat category, denial of service category, fraud category and hacker category are grouped under Cl which is described in figure 4

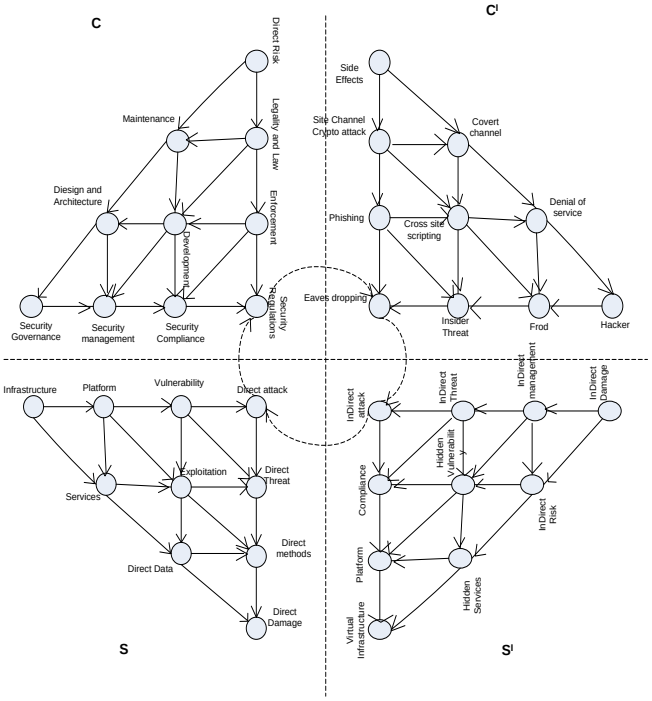

Fig 4: Security Compliance Requirements (Direct Vs Indirect) as 'n' category and 'n' morphism

The Security infrastructure supports the security platform whereas it is compatible with the security services, security exploitation may occur in the platform which extends vulnerability. The security platform may leads to security vulnerability which also includes a flaw in the security services. The vulnerability tends to direct attack, it also extends to exploitation and also leads to direct threat. The security services may leads to exploitation which causes direct threat which includes direct methods and direct data. The security vulnerability leads to exploitation which also tends to direct attack on the system. The security exploitation causes direct threat, extends direct method and includes direct data whereas direct threat includes direct method and direct data leads to direct damage to the system. The security direct attack leads to direct threat and it extends to direct methods which m ay includes direct damage to the system.

The compliance category represents that direct risk supports legality and law which is compatible with the maintenance of the system. The legality and law category includes the maintenance and the security enforcement category of the system. The maintenance category tends to development and the enforcement category includes security regulation category. The development category in the compliance category includes design and architecture and leads to security management category, whereas design and architecture category extends security governance category and security management category supports security compliance category. The enforcement category includes security regulation category which includes security compliance category. The design and architecture category leads to security governance category which supports security management category. The security management category in the compliance category extends security compliance category which includes security regulation category in the compliance category. The maintenance category in the compliance category supports the design and architecture category which tends to the development category in the compliance category which is described in figure 5

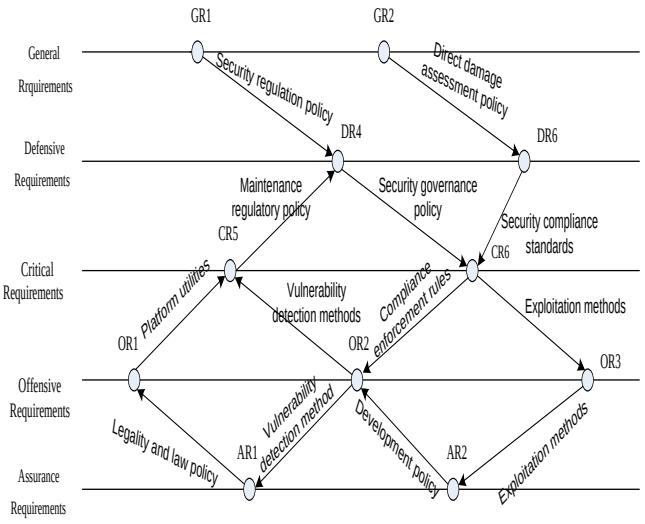

Fig 5: Security Requirements Dependency

The requirements are categorized in to general requirements, defensive requirements, critical requirements, offensive requirements and assurance requirement. The GR1 in general requirement communicate to DR4 in defensive requirement through security regulation policy. The DR4 in defensive requirement communicate to CR5 in critical requirement through maintenance regulatory policy. The CR5 in critical requirement communicate to OR1in offensive requirements through platform utilities. The OR1 communicate to AR1 in assurance requirement through legality and law policy. The AR1 communicate OR2 through compliance enforcement rules. The OR2 communicate CR6 through security compliance standards. The CR6 in communication requirement to DR6 through security compliance standard. The DR6 communicate to GR2 through direct damage assessment policy which is shown in figure 5.

IV. SECURITY REQUIREMENT MANAGEMENT (CSRM)

A practical framework is developed based on model based security risk assessment by exploiting the synthesis of risk analysis. The CORAS risk assessment modelling technology is used to describe the target of assessment at the right level of abstraction , it also acts as a medium for communication and interaction between different groups of users involved in risk assessment, and it also documents the risk assessment results on the assumption on which the results depends. Figure 6 describes the attacker in the auto driver mode and the attacker injects vulnerability to make loss of speed control then the car suddenly accelerated with a maximum speed due to the injection of malware by the attacker. This situation makes catastrophic effect in the control of the system in the webserver. The user in the manual driver mode makes a change in the driver mode based on the road condition attacker can also inject vulnerability which results in delay in action of sensor. The decision has to be made based on the road condition so that it causes a moderate effect of cause in the system which can be done through collaboration server. The security regulator regulates the compliance and standards of the sensors associated with the system whereas the misconfiguration of the sensor standards is made by the malware injection of the attacker whereas it makes slow response of the system sensor so that it makes a catastrophic effect in the real time communication server platform.

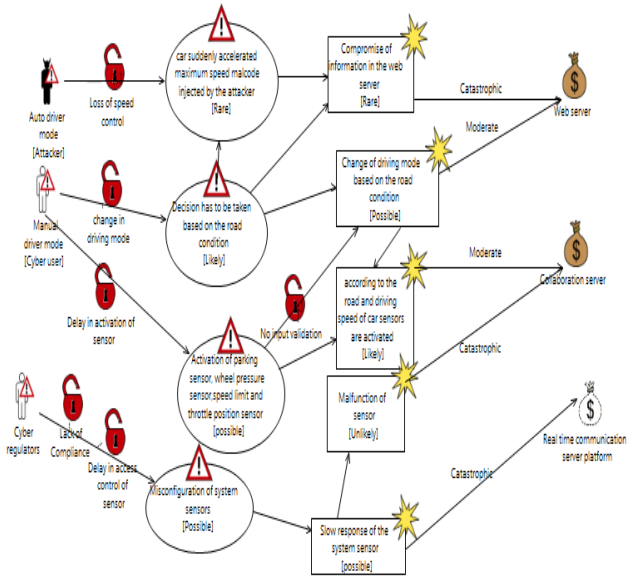

Fig 6: Security threat and risk management

The security requirement objects with different classes and domains have different morphism on their different codomains. Security requirement objects forms different categories since the attacks and vulnerabilities are different. Isomorphism and homomorphism are the techniques existing to collect the security requirement objects in this case we need a mathematical approach to collect classify and cluster all the security requirement objects which should be in the manageable form to port and to certify the available module as needed. Based on the proposed 'n' category – morphism model we categorize the objects in the security and compliance category as general requirement object, critical requirement objects, offensive requirement objects, defensive requirement objects and assurance requirement objects. These objects are grouped and filtered by criticality filter, offensive filter, defensive filter, general requirement filter and critical filter based on the security requirement weight (SRW) assigned by the 0 morphism 1 morphism, 2 morphism, 3 morphism and n morphism. Figure 6 shows the software requirement document and it also describes the general requirement, defensive requirement, critical requirement, offensive requirement and assurance requirement.

V. RESULT AND DISCUSSION

Table 2 describes the category whereas S represents Security category and C represents compliance category. The objects in the security category are infrastructure which supports morphism and compatible to one morphism and it does not transform up to n morphism. The platform object in security category leads to morphism and extends to one morphism and includes up to n morphism. The vulnerability object tends to morphism which it causes one morphism and leads to n morphism. The direct attack object under security category leads to only morphism whereas direct threat object also extends only morphism. The exploitation object causes morphism, includes one morphism and includes up to n morphism. The services object leads to morphism and tends to one morphism. The direct data object includes morphism and causes one morphism. The direct method object includes only morphism whereas direct damage object does not undergoes any morphism. The direct risk object under Compliance category supports morphism and compatible to one morphism. The legality and law object extends morphism and tends to one morphism whereas maintenance object tends to morphism and supports one morphism. The enforcement object includes morphism and one morphism. The development object under compliance category leads to morphism and includes one morphism. The design and architecture object leads to morphism and supports up to one morphism. The security governance, security management, security compliance and security regulation objects under compliance category supports only morphism. The CORAS tool was used to identify the assets, the various types of security requirements as categorical objects and their mappings. The requirements along with their multiple morphisms are documented which are to be delivered to the succeeding design phase. The serious limitation in the said work is that the algorithm may not be more suitable to reduce the reachability and complexity of the list of security requirements.

The case study of security requirement engineering on connected car navigation system is not completed since the cyber security requirements are highly scalable. The requirement analysis is performed to identify all possible security relations across objects in different categories and the model can be very well applied to similar distributed systems like wireless sensor network and for a grid of connected medical devices.

The software requirement output document is described by

General Requirements (GR)
The general requirements undergoes zero morphism whereas the security requirement includes :
GR 1: Security Regulation Policy.
GR 2: Direct damage assessmernent policy.

Defensive Requirements (DR)
The defensive requirement undergoes one morphism whereas the security requirements are
DR 1: Direct attack techniques
DR 2: Direct threat model
DR 3: Direct risk avoidance method
DR 4: Security governance policy
DR 5: Security management techniques
DR 6: Security compliance standards.

Critical Requirements (CR)
The critical requirement undergoes 2 morphism whereas the security requirements includes:
CR 1: Secured Infrastructure facilities
CR 2: Service Identification techniques
CR 3: Direct vulnerable data source
CR 4: Direct risk mitigation
CR 5: Maintenance Regulatory policy
CR 6: Compliance Enforcement rules
CR 7 : Design and architecture pattern.

Offensive Requirements (OR)
The offensive requirement undergoes 3 morphism and the security requirements are:
OR 1: Platform utilities
OR 2: Vulnerability detection methods
OR 3: Exploitation methods.

Assurance Requirements (AR)
The assurance requirement undergoes n morphism and the security requirements are :
AR 1: Legality and law policy
AR 2 : Development policy

Table 1: Categorical theory for software requirements Output Document

Category	Objects	Morphism	One morphism	n-morphism
S	Infrastructure	*Supports*	*Compatible*	-
S	Platform	*Leads to*	*Extends*	*Includes*
S	Vulnerability	*Tends to*	*Causes*	*Leads to*
S	Direct attack	*Leads to*	-	-
S	Direct threat	*Extends*	-	-
S	Exploitation	*Causes*	*Includes*	*Includes*
S	Services	*Leads to*	*Tends to*	-
S	Direct data	*Includes*	*Causes*	-
S	Direct method	*Includes*	-	-
S	Direct damage	-	-	-
C	Direct risk	*Supports*	*Compatible*	-
C	Legality and law	*Extends*	*Tends to*	*Compatible*
C	Maintenance	*Tends to*	*Supports*	-
C	Enforcement	*Includes*	*Includes*	-
C	Development	*Leads to*	*Includes*	*Supports*
C	Design and architecture	*Leads to*	*Supports*	-
C	Security governance	*Supports*	-	-
C	Security management	*Supports*	-	-
C	Security compliance	*Includes*	-	-
C	Security regulations	-	-	-

VI. CONCLUSION

The software security requirements are classified based on their respective levels of execution as infrastructure security, platform security and application level security requirements focusing all possible deployments. The software requirement engineering is being carried out so as to satisfy the domain security requirements in various modes of operations and compliant with the recent security standards for minimum level of risks.

The distributed software security requirements were collected and processed as inception, elicitation and elaboration phases and finally organized with priorities. The requirement analysis for such a distributed software system is done with categorical theory and morphisms in a mathematical way to identify the functional or type mapping between security objects. The composition and identity operations along with functors across a numbers of security related categories are analyzed with functors and product and coproducts of security objects. The categorical products and sum of the security requirements can be achieved by pairing of object types and declaring functional objects between morphisms. The categorical theory is very much used in understanding the functional mapping across the security types with all possible relations among them. The security objects are mapped and in the same way the compliances are mapped as morphisms between any types of requirement objects. A case study on cyber connected cars navigation system was considered to apply the proposed 'n category and n morphism' model and the cyber use case diagram was drawn for the cyber security requirements for the said software system. The various mappings in the connected cars case study are applied to reduce the risk of non-compliances with the help of trusted server.

REFERENCES

1. Paul Soulier, Depeng Li et.al "A Survey of Language-Based Approaches to Cyber-Physical and Embedded System Development" Tsinghua Science And Technology, ISSN 1007-0214 02/11 pp130-141, Volume 20, Number 2, April 2015.
2. Lin Chen, Xingshu Chen et.al "Research and Practice of Dynamic Network Security Architecture for IaaS Platforms" Tsinghua Science And Technology, ISSN 1007-0214 09/13 pp496-507, Volume 19, Number 5, October 2014.
3. Zhen Chen, Wenyu Dong et.al "Collaborative Network Security in Multi-Tenant Data Center for Cloud Computing "Tsinghua Science And Technology , ISSN 1007-0214 09/10 pp82-94 ,Volume 19, Number 1, February 2014.
4. Hong Zhong, Jingyu Wen et.al "Efficient Conditional Privacy-Preserving and Authentication Scheme
5. for Secure Service Provision in VANET" Tsinghua Science And Technology, ISSNll1007-0214ll04/09llpp620-629 ,Volume 21, Number 6, December 2016.
6. S. M. Riazul Islam et.al "The Internet of Things for Health Care: A Comprehensive Survey" Received April 4, 2015, accepted May 8, 2015, date of publication June 1, 2015, date of current version June 4, 2015.
7. Tobias Dehling et.al "Information Security and Privacy of Patient-Centered Health IT Services: What needs to be done" 47th Hawaii International Conference on System Science, 2014.
8. Daniel Mellado et.al "A systematic review of security requirements engineering" Computer Standards & Interfaces 32 (2010) 153–165.
9. Charles B. Haley et.al "A Framework for Security Requirements Engineering" SESS'06, May 20–21, 2006, Shanghai, China.
10. Muthu Ramachandran "Software Security Requirements Engineering: State of the Art" School of Computing, Creative Technologies and Engineering Leeds Beckett University, Springer-Verlag Berlin Heidelberg 2011.
11. R. Saranya "Survey on Security Measures of Software Requirement Engineering" International Journal of Computer Applications (0975 – 8887) ,Volume 90 – No 17, March 2014.
12. Vladimir A. Khlevnoy et.al "A Formal Approach to Distributed System Security Test Generation" International Journal of Computer Trends and Technology (IJCTT) – volume 16 number3–Oct2014.
13. Manoj Kumar et.al "Analysis of Different Security Issues and Attacks in Distributed System A-Review" International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 3, Issue 4, April 2013.
14. Charles B. Haley, Robin Laney et.al "Security Requirements Engineering: A Framework for Representation and Analysis" IEEE Transactions On Software Engineering, Vol. 34, No. 1, January/February 2008.
15. Daniel Mellado et.al "A common criteria based security requirements engineering process for the development of secure information systems" Computer Standards & Interfaces 29 (2007) 244–253 , 2006 Elsevier B.V.
16. Mohammed F. H. Abulamddi "A Survey of Approaches Reconciling between Safety and Security Requirements Engineering for Cyber-Physical Systems" Journal of Computer and Communications, 2017.
17. S. Ravikumar, S. Chandrasekaran, S. Ramesh (2016), "Safety Assessment of Distributed Automotive Software System Model with Design for Traceability," Asian Journal of Information Technology,15(11): pp.1799-1815,2016. DOI: 10.36478/ajit.2016.1799.1815
18. S.Ravikumar, Chandrasekaran Subramaniam (2016), "A Survey on Different Software Safety Hazard Analysis and Techniques in Safety Critical Systems," Middle East Journal of Scientific Research, 24, (Special Issue on Innovations in Information, Embedded and Communication Systems) DOI:10.5829/idosi.mejsr.2016.24.IIECS.23145, © IDOSI Publications, 2016 ISSN 1990-9233,pp.90-97.
19. D.Kavitha., S.Chandrasekaran., M.Vigilson Prem (2016). Software Security Risk Assessment of Data Communication Network Through Attack Decomposition Using Fuzzy Rough Sets, Asian Journal of Information Technology, Volume: 15, Issue: 11 , Page No.: 1758-1775. DOI: 10.36478/ajit.2016.1758.1775
20. D.Kavitha & S.Chandrasekaran (2017). Security Threat Management by Software Obfuscation for Privacy in Internet of Medical Thing (IoMT) Application, Journal of Computational and Theoretical Nanoscience, Volume 14, Number 7, July 2017, pp. 3100-3114(15).
21. S. Chandrasekaran, R. S. Vijayravikumaran and et.al, (2011). CAR Based Safety Model in Automotive Software Engineering, SEPADS' 11 Proceedings of the 10th WSEAS international conference on Software Engineering, parallel and distributed systems pp 206-211, ISBN: 978-960-474-277-6.
22. S Ravikumar, D Kavitha , (2020), IoT based home monitoring system with secure data storage by Keccak–Chaotic sequence in cloud server, Journal of Ambient Intelligence and Humanized Computing, Springer Berlin Heidelberg, DOI: https://doi.org/10.1007/s12652-020-02424-x.
23. D Kavitha, S Ravikumar. (2020), Designing an IoT based autonomous vehicle meant for detecting speed bumps and lanes on roads, Journal of Ambient Intelligence and Humanized Computing, Springer Berlin Heidelberg, DOI: https://doi.org/10.1007/s12652-020-02419-8
24. D. Kavitha ,S.Chandrasekaran, S. K. Rani, (2016) ,HDTCV: Hybrid Detection Technique for Clickjacking Vulnerability, Artificial Intelligence and Evolutionary Computations in Engineering Systems. vol 394. pp 607-620, Springer, New Delhi. https://doi.org/10.1007/978-81-322-2656-7_56.
25. D. Kavitha S. Ravikumar "IOT and context-aware learning-based optimal neural network model for real-time health monitoring" in Transactions on Emerging Telecommunications Technologies, First published: 05 October 2020, https://doi.org/10.1002/ett.4132, Online ISSN:2161-3915, John Wiley & Sons Ltd..
26. Ravikumar, S., Kavitha, D. IOT based autonomous car driver scheme based on ANFIS and black widow optimization. J Ambient Intell Human Comput (2021). https://doi.org/10.1007/s12652-020-02725-1. © 2021 Springer Nature Switzerland AG.

AUTHORS PROFILE

Dr.D.Kavitha, is currently working as an Assistant Professor in Department of Computer Science and Engineering at SRM Valliammai Engineering College, Chennai, Tamil Nadu, India. She has completed Ph.D and Master of Engineering in Computer Science and Engineering from Anna University, Chennai, Her research interests are mainly focused on Software Security, Internet of Things and Cyber Security. She has an overall teaching experience of 12 years and has more than 25 papers to her credit in reputed international journals and conferences.

Dr. S.Ravikumar, is currently working as an Assistant Professor in the Department of Information Technology at SRM Valliammai Engineering College, Chennai, Tamil Nadu, India. He has completed Ph.D in Computer Science and Engineering from Anna University, Chennai and Master of Engineering in Computer Science Engineering from Madurai Kamaraj University. He has 20 years of teaching experience and his research interests are focused on Autonomous Vehicle and Connected cars, Software Safety and Reliability, Cloud Computing and IOT.