Fuzzy Logic based Decision Support System for Component Security Evaluation

Shah Nazir[1], Sara Shahzad[1], Saeed Mahfooz[1], and Muhammad Nazir[2]
[1]Department of Computer Science, University of Peshawar, Pakistan
[2]Institute of Business and Management Sciences, University of Agriculture, Pakistan

Abstract: *Software components are imperative parts of a system which play a fundamental role in the overall function of a system. A component is said to be secure if it has a towering scope of security. Security is a shield for unauthorized use as unauthorized users may informally access and modify components within a system. Such accessing and modifications ultimately affect the functionality and efficiency of a system. With an increase in software development activities security of software components is becoming an important issue. In this study, a fuzzy logic based model is presented to handle ISO/IEC 18028-2 security attributes for component security evaluation. For this purpose an eight input, single output model based on the Mamdani fuzzy inference system has been proposed. This component security evaluation model helps software engineers during component selection in conditions of uncertainty and ambiguity.*

Keywords: *Software component, component security, fuzzy logic.*

Received May 1, 2015; accepted November 29, 2015

1. Introduction

A component is a processing unit of a system which performs a key responsibility in the functionality of the system. Component Based Software Engineering (CBSE) reduces development time for new systems. Components available for reuse are already tested, experimented, and debugged. A new system developed using existing components turns out to be less expensive and almost pre-tested and debugged. Maximum software reuse is suggested in many cases, as reuse saves overall development time and also carries a code that is error free, already used and tested in many systems [15, 18]. An individual software component is a software suite that provides established functionality. These separate software components' functions are combined based on the principle of CBSE. Many such software systems are structured and organized from a collection of different components. The components should be explicitly clear from all perspectives for the development of the software system. A component may be replaced by another component if the decedent component has the criteria of the antecedent component. Due to the increased demand of the development of large and complex software systems, CBSE is becoming more and more common, to save time, cost of development, and to use already tested code. Study shows that in recent years such component based software development is performed for nearly half of the total developed software systems [17].

When choosing composition of components for a new system it is necessary to consider functional, non-functional, and system appraisal requirements. Along with that security is one of the highest obligations of a system. Software security is the protection of the system from unauthorized access and modification. When software is designed from diverse components, it might go through elevated security threats to the intended software [10]. These threats have an effect on the functionality and efficiency of the system. The fundamental concern of component security is how to put together a secure component based system.

The main objective of the proposed research is to put forward a methodology for evaluating the security of software components. Here, a Fuzzy Logic (FL) approach is modelled to evaluate the security of components. Fuzziness is beneficial in situations of vagueness and uncertainty. The proposed method incorporates attributes of ISO/IEC 18028-2 defined for security of components [20].

The organization of the paper is as follow: In section 2 related works is presented. Section 3 provides details of fuzzy logic technique, security attributes, and model for evaluating security of components. Section 4 presents results of the model, implementation, and discussion. Section 5 is the conclusion of the paper.

2. Related Work

Related research and security methodologies are presented and used by researchers. Khan and Han [9] differentiate the security characteristics and appropriateness of components. For software component characteristics representation and comparison their method involves logic programming. Gandotra *et al.* [5] presented a Secure Software System (SSS) for evolving the security of system using fuzzy logic. This method helps to evolve the mid stage between failed and safe state for security goal. Their proposed system consists of different phases like elicitation of security requirements, categorization, prioritization, mapping of threats into security requirements, and monitoring security level.

Ghosh and McGraw [6] defined an approach for documentation to test security properties of component. Black and white box testing techniques are used to test and verify the security of software component. Liao *et al.* [12] developed an approach to collect different forensic information. They proposed fuzzy logic and expert system to examine computer crime in network and the technique makes digital evidences automatically. The experimental results of the proposed method with other methods are also compared and show that the system classifies most attacks and provide understandable information for forensic experts. Engina *et al.* [3] proposed a fuzzy approach for Attribute Control Charts (ACC) and is solved by Greedy Algorithm (GAs). Two main parameters, which are sample size and acceptance number, are determined for every stage by the GA. The technique is applied on engine valve manufacturing firm. Siadat *et al.* [19] proposed the grid security improvement by new trust management system. New domain addition to grid system and selecting services provider are the advantages of the proposed approach.

Khan *et al.* [8] classified the properties of security into functional and non-functional security. Functional security is outer protection of components while non-functional security component are fixed with component functionality. Lee *et al.* [11] used component specification technique and described some definitions of component. The operators defined are, component version, functional requirements, nonfunctional requirements, and cooperating component. Z scheme is used for the specification of component. Cai *et al.* [2] proposed quality assurance for both component and system of the component. ComPARE is used to assess real life component. Moriconi *et al.* [14] suggested a method in which various representations of software architecture and required security at architecture level are described. The method is demonstrated with the help of open distribution transaction processing reference architecture.

3. Fuzzy Logic for Evaluating Security of Component

In the proposed research work, fuzzy logic is used for evaluation of component security. This approach is quite beneficial in situations of uncertainty. It has a variety of applications [1, 4, 7, 16].

Component security evaluation is the process of determining the security of a component. A high quality component evaluation system fulfills the required criteria of security. Details of the proposed method are given below.

3.1. Fuzzy Logic

Fuzzy set theory has been used for solving different problems in diverse fields. It is mostly used in the field of engineering as it resolves the problems of imprecision and vagueness. This tool helps providing solution for problems which are difficult to model. Detail of fuzzy logic concepts is given in Lofti [13]. It consists of different inputs and Membership Function (MF) and on the basis of MF a model of fuzzy rules is designed. Here, Fuzzy logic approach is tried to model and evaluate the security of components.

The different MFs which are used for inputs are: for access control, MF:(no access control, medium access control and full access control), for authentication, MF: (low authentication, medium authentication and high authentication), for non-repudiation, MF:(no non-repudiation, medium non-repudiation and high non-repudiation), for data confidentiality, MF:(no data confidentiality, medium data confidentiality and high data confidentiality), for communication flow, MF:(no communication flow, medium communication flow and high communication flow), for data integrity, MF:(no data integrity, medium data integrity and high data integrity), for availability, MF:(no availability, medium availability and high availability), for privacy, MF:(no privacy, medium privacy and high privacy).

Membership functions are mathematically described as;

$$m_A(x)=\{1\ x\in A,\ 0\ x\notin A\} \quad m_A(x)\in\{0,1\} \tag{1}$$

And can be plotted as shown in figure 1;

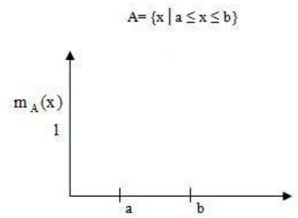

Figure 1. Membership function (MF).

The proposed model is shown in Figure 2. It consists of different MF, fuzzy rules and the rules are stored in the database. The process clearly elaborates the steps involve in the designing of membership functions and the rules from these membership functions. The results of the model are analyzed and a decision is made after the final results. In the last step approval of the most secure component should be taken from the competent authority and the most secure component is delivered to the designer of the system.

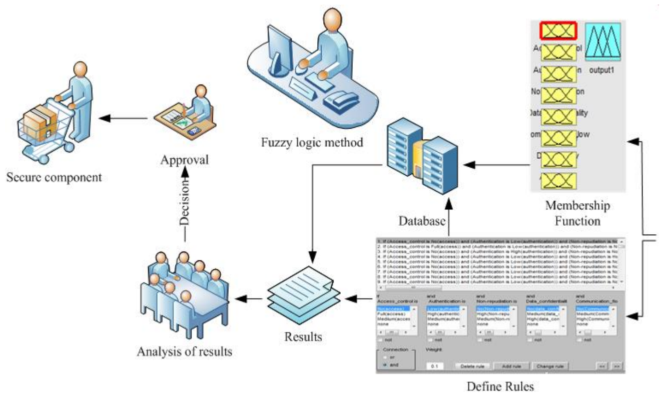

Figure 2. Process of the proposed model.

The membership functions are given below, in Figures 3-10. Figure 3 shows the different membership functions that are; No (access), Medium (access) and Full (access) for the input labelled as access control. The degree of membership functions is plotted as; No (access) is between 0-0.29, Medium (access) is 0.30-0.60 and Full(access) is 0.61-1.

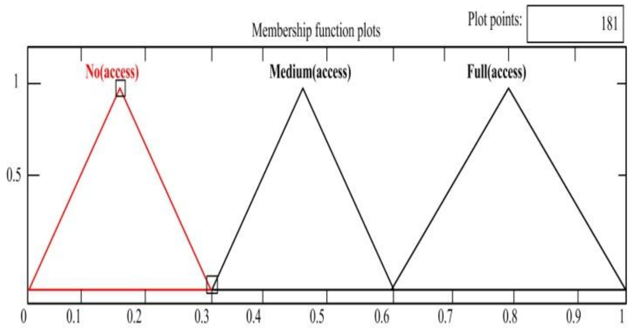

Figure 3. MFs for input access control.

Figure 4 shows the different membership functions that are; Low (authentication), medium (authentication) and High (authentication) for the input labelled as authentication. The degree of membership functions is plotted as; Low (authentication) is between 0-0.30, medium (authentication) is 0.31-0.59 and High (authentication) is 0.6-1.

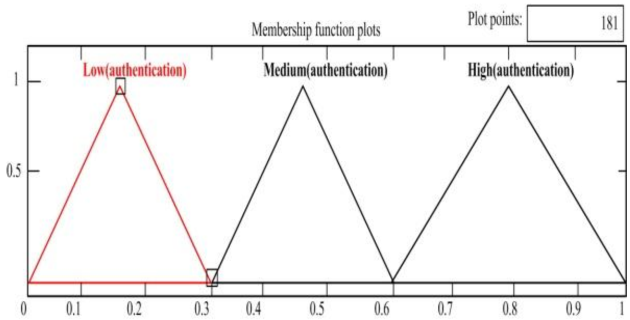

Figure 4. MFs for input authentication.

Figure 5 shows different membership functions that are; No (non-repudiation), Medium (non-repudiation) and High (non-repudiation) for the input labelled as non-repudiation. The degree of membership functions is plotted as; No (non-repudiation) is between 0-0.30, medium (non-repudiation) is 0.31-0.59 and High (non-repudiation) is 0.60-1.

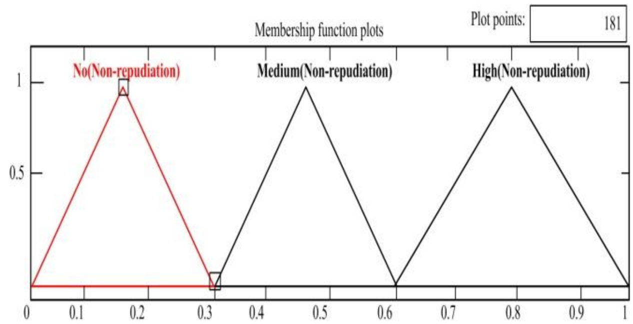

Figure 5. MFs for input non-repudiation.

Figure 6 shows the different membership functions that are; No (data confidentiality), Medium (data confidentiality) and High (data confidentiality) for the input labelled as data confidentiality. The degree of membership functions is plotted as; No (data confidentiality) is between 0-0.30, medium (data confidentiality) is 0.31-0.61 and High (data confidentiality) is 0.62-1.

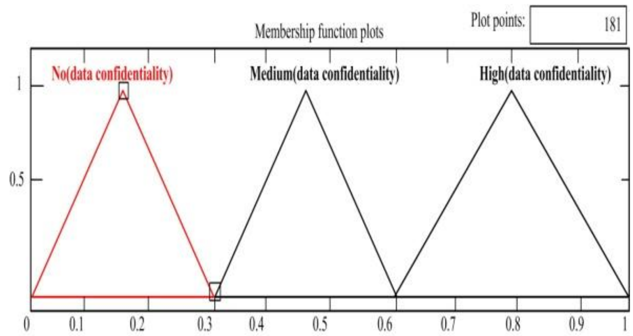

Figure 6. MFs for input data confidentiality.

Figure 7 shows the different membership functions that are; No (communication), Medium (communication) and High (communication) for the input labelled as communication. The degree of membership functions are plotted as; No(communication) is between 0-0.30, medium(communication) is 0.31-0.60 and High(communication) is 0.61-1.

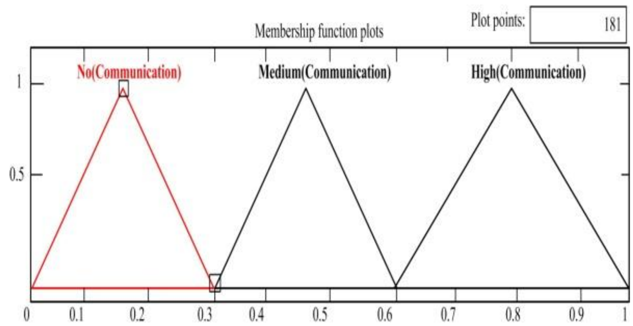

Figure 7. MFs for input communication.

Figure 8 shows the different membership functions that are; No (data integrity), Medium (data integrity) and High (data integrity) for the input labelled as data integrity. The degrees of membership functions are plotted as; No (data integrity) is between 0-0.30, medium (data integrity) is 0.31-0.60 and High (data integrity) is 0.61-1.

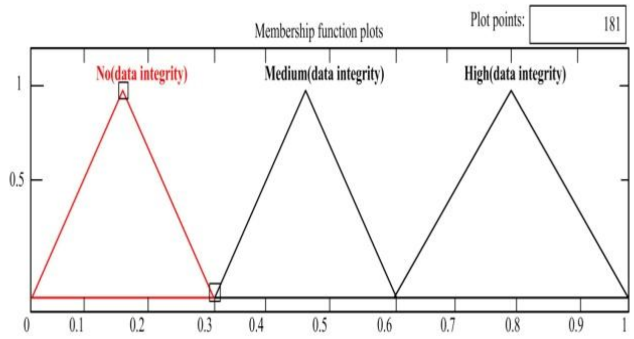

Figure 8. MFs for input data integrity.

Figure 9 shows the different membership functions that are; No (availability), Medium (availability) and High (availability) for the input labelled as availability. The degrees of membership functions are plotted as; No (availability) is between 0-0.30, medium (availability) is 0.31-0.60 and High (availability) is 0.61-1.

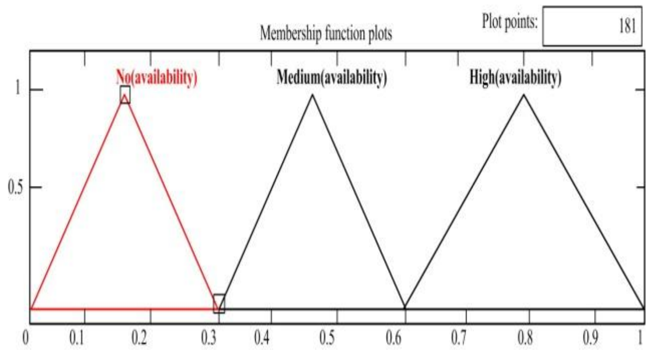

Figure 9. MFs for input availability.

Figure 10 shows the different membership functions that are; No (privacy), Medium (privacy) and High (privacy) for the input labelled as privacy.

The degree of membership functions are plotted as; No (privacy) is between 0-0.30, medium (privacy) is 0.31-0.61 and High(privacy) is 0.62-1.

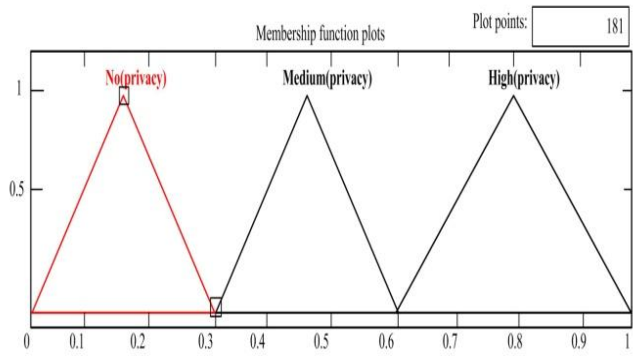

Figure 10. MFs for input privacy.

3.1. Security Attributes

The proposed method is applied on ISO/IEC 18028-2 [20] to design a model which evaluates the security of components according to the following specifications.

3.1.1. Access Control

The access control attribute of security provides authorization to use the component of the software. Access control can certify only allowed users to access the information of the available software component(s). The range of MF for access control is given below.

$$\mu_{(Access\ control)}=\{\ 0<x\leq 0.29=low,\ 0.30<x\leq 0.60=medium,\ 0.61<x\leq 1=high\} \tag{2}$$

3.1.2. Authentication

Authentication is the process of determining validity of information provided for some specific purpose. This attribute of security corroborates the identity of user to software components. It confirms the validity of user when accessing the available information of software components. Mathematically it can be shown as follows:

$$\mu_{(Authentication)}=\{\ 0<x\leq 0.30=low,\ 0.31<x\leq 0.59=medium,\ 0.60<x\leq 1=high\} \tag{3}$$

3.1.3. Non-repudiation

The non-repudiation security attribute facilitate technological means to prevent a user from denying performing a specific action about the component. This helps in confirming the availability of data to a third party as a proof that some events have taken place. Non-repudiation is mathematically shown by the equation:

$$\mu_{(Non\text{-}\ repudiation)}=\{\ 0<x\leq 0.30=low,\ 0.31<x\leq 0.59=medium,\ 0.60<x\leq 1=high\} \tag{4}$$

3.1.4. Data Confidentiality

This attribute of security protects data attached with software components from illegal or unauthorized access. Some data confidentiality methods and algorithms are used to secure data of available software components. MF for safety is given below in equation:

$$\mu_{(Data\ confidentiality)}=\{\ 0<x\leq 0.30=low,\ 0.31<x\leq 0.61=medium,\ 0.62<x\leq 1=high\} \tag{5}$$

3.1.5. Communication Flow

Communication flow confirms that sharing of information related to different software components is made only with authorized persons. The components are protected from illegal access. It is an ethical principle associated with component(s). Communication exists between software component(s) and user. Confidentiality can be privileged and cannot be accessed or modified by any illegal user. Mathematically communication flow is shown below in the following equation.

$$\mu_{(Communication\ flow)}=\{\ 0<x\leq 0.30= low,\ 0.31<x\leq 0.60= medium,\ 0.61<x\leq 1= high\} \quad (6)$$

3.1.6. Data Integrity

Data integrity provides accuracy and correctness of the information related to software components. All data and information of the components is protected from unauthorized modifications, creation, and replication. Data integrity is mathematically shown below using the equation:

$$\mu_{(Data\ integrity)}=\{\ 0<x\leq 0.30= low,\ 0.31<x\leq 0.60= medium,\ 0.61<x\leq 1= high\} \quad (7)$$

3.1.7. Availability

Availability of a component denotes the fraction of time for which it is working and functional. It can be affected by component load, errors, and malicious attacks. So the data and information related to component should be available to authorized access whenever it need. There should be no denial of service to authorized access. Availability of components is mathematically shown by the following equation.

$$\mu_{(Availability)}=\{\ 0<x\leq 0.30= low,\ 0.31<x\leq 0.60= medium,\ 0.61<x\leq 1= high\} \quad (8)$$

3.1.8. Privacy

This attribute of security provides the safety of data and information to software components. This attribute presents the needed protection and control the data and also the information related to software components. Privacy is mathematically shown in the equation below.

$$\mu_{(Privacy)}=\{\ 0<x\leq 0.30= low,\ 0.31<x\leq 0.61= medium,\ 0.62<x\leq 1= high\} \quad (9)$$

3.2. Design of Fuzzy Inference System

The proposed model is designed using the fuzzy tool box. It consists of five basic GUI tools including FIS editor, membership function editor, rule editor, rule viewer, and surface viewer. Figure 11 describes the proposed model using the Mamdani fuzzy inference system. In the figure, the top-left part represents the basic model of inputs and outputs. The top-right part describes the rule editor in which the fuzzy rules are designed. The bottom-left part of the figure shows the membership function editor and the bottom-right part is the surface viewer of rules.

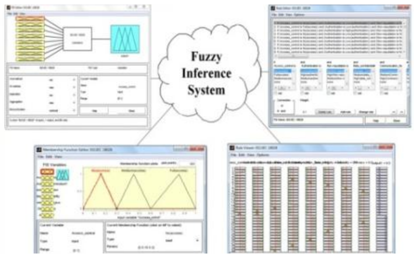

Figure 11. Process of the proposed fuzzy logic approach.

Based on the proposed method (section 3), the three membership functions and the eight inputs (mentioned in Figures 3-10), the fuzzy rules are obtained. These rules are in the form as follows:

- *Rule 1*. If (Access_control is No(access)) and (Authentication is Low(authentication)) and (Non-repudiation is No(Non-repudiation)) and (Data_confidentiality is No(data_confidentionality)) and (Communication_flow is No(Communication_flow)) and (Data_integrity is No(Data_integrity)) and (Availability is No(availibility)) and (Privacy is No(privacy)) then (output1 is Low_Secure) (0.1)
- *Rule 2*. If (Access_control is Full(access)) and (Authentication is High(authentication)) and (Non-repudiation is High(Non-repudiation)) and (Data_confidentiality is High(data_confidentionality)) and (Communication_flow is High(Communication_flow)) and (Data_integrity is High(Data_integrity)) and (Availability is High(availibility)) and (Privacy is No(privacy)) then (output1 is Ver_High_Secure) (1)
- *Rule 3*. If (Access_control is Mediium(access)) and (Authentication is Medium(authentication)) and (Non-repudiation is Medium(Non-repudiation)) and (Data_confidentiality is Medium(data_confidentionality)) and (Communication_flow is Medium(Communication_flow)) and (Data_integrity is Medium(Data_integrity)) and (Availability is Medium(availibility)) and (Privacy is High(privacy)) then (output1 is Medium_Secure) (0.5)
- *Rule 4*. If (Access_control is No(access)) and (Authentication is High(authentication)) and (Non-repudiation is High(Non-repudiation)) and (Data_confidentiality is Medium(data_confidentionality)) and (Communication_flow is High(Communication_flow)) and (Data_integrity is High(Data_integrity)) and (Availability is Medium(availibility)) and (Privacy is Medium(privacy)) then (output1 is very_Low_Secure) (0.8)

. . . . . .
and so on.

4. Results and Discussion

On the basis of the designed rules and model, the security of components can be evaluated. Table 1 shows input/output details and the membership functions used. Following is the configuration of the model as implemented in fuzzy logic.

Name= “component security evaluation model”
Type = “mamdani”
Version = “2.0”
Num Inputs = “8”
Num Outputs = “1”
And Method = “min”
Or Method = “max”
Imp Method = “min”
Agg Method = “max”
Defuzz Method = “centroid”

Inputs are given according to domain expert opinion in command interface of the designed model, for example,

```
a= readfis ('FIS Model')
a = name: 'FIS Model'
       type: 'mamdani'
       and Method: 'min'
       or Method: 'max'
       defuzz Method: 'centroid'
       imp Method: 'min'
       agg Method: 'max'
       input: [1x8 struct]
      output: [1x1 struct]
```

Eight inputs, one relating to each ISO/IEC 18028-2 attribute, are given to the designed model.

For example,

```
out=evalfis ([0.8 0.7 0.7 0.7 0.8 0.8 0.9 0.8], fismat)
out =  0.900
out=evalfis ([0.1 0.2 0.3 0.1 0.2 0.3 0.3 0.2], fismat)
out =    0.500
out=evalfis ([0.6 0.1 0.6 0.1 0.3 0.3 0.1 0.2], fismat)
out =    0.500
out=evalfis ([0.7 0.8 0.9 0.6 0.7 0.8 0.9 0.7], fismat)
out =  0.800
```

In table 2, 0.900, 0.500, 0.500 and 0.800 are the outputs according to given inputs for four components and from this results user can make decision about the most secure component.

Table 1. Inputs and output and their membership function.

[Input1] = [Access control]	Range = [0 1], Num MFs=3 MF 1= 'No(access control)': 'trimf', [0 0.16 .29] MF 2= 'Full(access control)': 'trimf', [0.6 0.85 1] MF 3= 'Medium(access control)': 'trimf', [0.29 0.45 0.6]
[Input2] = [Authentication]	Range= [0 1], Num MFs=3 MF 1= 'Low(authentication)': 'trimf', [0 0.16 0.30] MF 2= 'High(authentication)': 'trimf', [0.6 0.85 1] MF 3= 'Medium(authentication)': 'trimf', [0.30 0.45 0.59]
[Input3] = [Non-repudiation]	Range= [0 1], Num MFs=3 MF 1= 'No(Non-repudiation)': 'trimf', [0 0.16 0.30] MF 2= 'High(Non-repudiation)': 'trimf', [0.6 0.85 1] MF 3= 'Medium(Non-repudiation)' :'trimf', [0.29 0.45 0.6]
[Input4] = [Data confidentiality]	Range= [0 1], Num MFs=3 MF 1= 'No(data_confidentionality)': 'trimf', [0 0.16 0.30] MF 2= 'Medium(data_confidentionality)': 'trimf', [0.31 0.45 0.6] MF 3= 'High(data_confidentionality)': 'trimf', [0.6 0.85 1]
[Input5] = [Communication flow]	Range= [0 1], Num MFs=3 MF 1= 'No(Communication_flow)': 'trimf', [0 0.16 0.29] MF 2= 'Medium(Communication_flow)': 'trimf', [0.31 0.45 0.6] MF 3= 'High(Communication_flow)': 'trimf', [0.6 0.85 1]
[Input6] = [Data integrity]	Range= [0 1], Num MFs=3 MF 1= 'No(Data_integrity)':'trimf', [0 0.16 0.30] MF 2= 'Medium(Data_integrity)': 'trimf', [0.31 0.45 0.6] MF 3= 'High(Data_integrity)': 'trimf', [0.6 0.85 1]
[Input7] = [Availability]	Range= [0 1], Num MFs=3 MF 1= 'No(availability)':'trimf', [0 0.16 0.30] MF 2= 'Medium(availability)': 'trimf', [0.31 0.45 0.61] MF 3= 'High(availability)': 'trimf', [0.6 0.85 1]
[Input8] = [Privacy]	Range= [0 1], Num MFs=3 MF 1= 'No(privacy)': 'trimf', [0 0.16 0.3] MF 2= 'Medium(privacy)': 'trimf', [0.3 0.45 0.6] MF 3= 'High(privacy)': 'trimf', [0.6 0.85 1]
[Output1] = [output1]	Range= [0 1], Num MFs=5 MF 1= 'Low_Secure': 'trimf', [0.2 0.3 0.4] MF 2= 'Medium_Secure': 'trimf', [0.4 0.5 0.6] MF 3= 'High_Secure': 'trimf', [0.6 0.7 0.8] Ble 1MF 4= 'very_Low_Secure': 'trimf', [0 0.1 0.2] MF 5= 'Ver_High_Secure': 'trimf', [0.8 0.9 1]

Table 2. Inputs and output.

Component ID	Access control	Authentication	Non-repudiation	Data confidentiality	Communication flow	Data integrity	Availability	Privacy	Output
C1	0.8	0.7	0.7	0.7	0.8	0.8	0.9	0.8	0.9
C2	0.1	0.2	0.3	0.1	0.2	0.3	0.3	0.2	0.5
C3	0.6	0.1	0.6	0.1	0.3	0.3	0.1	0.2	0.5
C4	0.7	0.8	0.9	0.6	0.7	0.8	0.9	0.7	0.8

From the results provided in Table 2, the output for component C1 is highest (that is, 0.9), which shows that it is the most secure component according to the attributes defined in ISO/IEC 18028-2. This output guides the decision(s) to be made for the selection of secure component.

5. Conclusions

A flawless system is needed to evaluate the security of software components. Security is an important factor of a component to be considered while selecting software components for component based software engineering. There is a huge library of software components which are available off-the-shelf, but most of them fail or do not maintain the satisfactory level working due to the lack of availability of security details. This research shows that the proposed methodology, which is based on ISO/IEC 18028-2 security attributes, is useful in situations of uncertainty and ambiguity, thus helping to select the most secure software component.

References

[1] Bilal M., Hussain A., Jaffar M., Choi T., and Mirza A., "Estimation and Optimization Based ill-posed Inverse Restoration using Fuzzy Logic," *Multimedia Tools and Applications*, vol. 69, no. 3, pp. 1067-1087, 2014.

[2] Cai X., Lyu R., and Wong K., "Component-Based Embedded Software Engineering: Development Framework, Quality Assurance and A Generic Assessment Environment," *International Journal of Software Engineering and Knowledge Engineering*, vol. 12, no. 2, pp. 107-133, 2002.

[3] Engina O., Çelika A., and Kaya İ., "A fuzzy Approach to Define Sample Size for Attributes Control Chart in Multistage Processes: An Application in Engine Valve Manufacturing Process," *Applied Soft Computing*, vol. 8, no. 4, pp. 1654-1663, 2008.

[4] Fredrick T. and Radhamani G., "The Fuzzy Logic Based ECA Rule Processing for XML Databases," *The International Arab Journal of Information Technology*, vol. 12, no. 6A, pp. 635-641, 2015.

[5] Gandotra V., Singhal A., and Bedi P., "A Step Towards Secure Software System using Fuzzy logic," *in proceedings of 2nd International Conference on Computer Engineering and Technology*, Chengdu, pp. 427-432, 2010.

[6] Ghosh A. and McGraw G., "An Approach for Certifying Security in Software Components," *in Proceedings of 21st National Information Systems Security Conference, National Institute, Standards and Technology*, pp. 82-86, 1998.

[7] Jeon G., Park S., Fang Y., Lee R., and Jeong J., "Application for Deinterlacing Method using Edge Direction Classification and Fuzzy Inference System," *Multimedia Tools and Applications*, vol. 59, no. 1, pp. 149-168, 2012.

[8] Khan K., Han J., and Zheng Y., "Security Properties of Software Components," *in Proceedings of International Workshop on Information Security*, Kuala Lumpur, pp. 52-56, 1999.

[9] Khan K. and Han J., "A Security Characterisation Framework for Trustworthy Component Based Software Systems," *in Proceedings of the 27th Annual International Computer Software and Applications Conference*, Dallas, pp. 164 - 169, 2003.

[10] Khan K., Han J., and Zheng Y., "A Scenario Based Security Characterisation of Software Components," *in Proceedings of the 3rd Australasian Workshop on Software and System Architectures*, Sydney, pp. 55-63, 2000.

[11] Lee J., Yoo C., and Chang O., "Component Contract-Based Interface Specification Technique using Z," *International Journal of Software Engineering and Knowledge Engineering*, vol. 12, no. 4, pp. 453-469, 2002.

[12] Liao N., Tian S., and Wang T., "Network Forensics based on Fuzzy Logic and Expert System," *Computer Communications*, vol. 32, no. 17, pp. 1881-1892, 2009.

[13] Lofti A., "Fuzzy Logic," *Computer*, vol. 21, no. 4, pp. 83-93, 1988.

[14] Moriconi M., Qian X., Riemenschneider R., and Gong L., "Secure Software Architectures," *IEEE Symposium on Security and Privacy*, CA, pp. 84-93, 1997.

[15] Nazir S., Khan M., Anwar S., Khan H., and Nazir M., "A Novel Fuzzy Logic Based Software Component Selection Modeling," *in Proceedings of International Conference on Information Science and Applications*, Suwon, pp. 1-6, 2012.

[16] Nazir S., Shahzad S., Khan S., Alias N., and Anwar S., "A Novel Rules Based Approach for Estimating Software Birthmark," *The Scientific World Journal*, vol. 2015, pp. 1-8, 2015.

[17] Rawashdeh A. and Matalkah B., "A New Software Quality Model for Evaluating COTS Components," *Journal of Computer Science*, vol. 2, no. 4, pp. 373-381, 2006.

[18] Sandhu P. and Singh H., "A neuro-fuzzy based Software Reusability Evaluation System with Optimized Rule Selection," *in Proceedings of International Conference on Emerging Technologies*, Peshawar, pp. 664-669, 2006.

[19] Siadat S., Rahmani A., and Mohsenzadeh M., "Proposed Platform for Improving Grid Security by Trust Management System," *Computer Science and Information Security*, vol. 6, no.1, pp. 143-148, 2009.

[20] Sabnis S., Chandrashekhar U., and Bastry F., "Challenges of Securing an Enterprise and Meeting Regulatory Mandates," *in Proceedings of the 12th International Telecommunications Network Strategy and Planning Symposium*, New Delhi, pp. 1-6, 2006.

Shah Nazir did PhD in Computer Science with specialization in Software Engineering from University of Peshawar. He has more than 20 research publications in well reputed international Journals and conference proceedings. He is serving at the University of Peshawar, Pakistan.

Sara Shahzad has a Ph.D. in Agile Software Development Processes with an interest towards Software Process Improvement. She is running Software Engineering research group at the department of Computer Science, University of Peshawar. Currently, she is working in the areas of software quality, reverse engineering, and empirical Software Engineering research with a focus on Software Engineering Education.

Saeed Mahfooz has done his Ph.D. from Liverpool John Moore University, Liverpool, UK in Distributed Multimedia Systems in 2001. Before that he has done MS from WIU Arizona State, USA in 1990. His research interest includes QoS Architectures, QoS Routing, Network Protocols, IPv6, Cloud Computing, Wireless Networks, MANETs, future Internet architecture and Next Generation Networks. He is also heading the Computer Networks Research Group at Department of Computer Science, University of Peshawar. He is also member of IEEE and currently he is head of the Computer Science Department, University of Peshawar.

Muhammad Nazir did his MSc in Computer Science from University of Peshawar. Currently he is enrolled in MS Computer Science program with specialization in the field of databases.