De Montfort University

User Trust Score FIS Report

IMAT 3406: Fuzzy Logic and Knowledge-Based Systems

by Harrison Charlesworth (P2662654)

Contents

Keywords: Fuzzy Logic, Fuzzy Inference System (FIS), Trust Assessment, Threshold Cryptography, Trust Management, Trust Score, Security Systems, Cybersecurity, Authentication, Policy Compliance

Abstract

In the dynamic field of cybersecurity, the combination of fuzzy logic and user trust evaluation highlights itself as an exciting possibility, particularly within the context of threshold cryptography. This report showcases the process taken to produce an intuitive fuzzy inference system, designed to enhance the security frameworks by assessing the nature of a user's trust. The system utilises three input variables, Authentication success, policy compliance and regularity of contributions to assess the complexity of a user's trust. The output is a trust score, a precise expression of user reliability, which moves on from the simple yes and no structure of traditional trust assessments.

The initial iteration of the system was designed after an extensive literature review showcasing the gaps in current systems. This initial design was then put through large amounts of testing to try and perfect the systems membership functions, rule base and accuracy. The systems final design presents itself as an adaptable design that suggests its future applications is far beyond what is explored in this report.

This report not only showcases the effective use of fuzzy logic for trust evaluation but also anticipates its possible expansion. It positions the system as a foundation for future developments, wherein fuzzy logic could become key in the complex domain of digital trust management, further enhancing security in the field of threshold cryptographic and cyber security as a whole.

Introduction

Threshold cryptography is an approach to secure systems that requires a certain number of parts, or 'threshold', of the whole group to agree before an action, such as decrypting information or signing a document, is authorised. This method improves security because it eliminates single points of failure, if one part of the system is compromised, the entire system doesn't necessarily fail. These types of systems are used in cryptocurrency platforms for transaction authorisation, the cryptographic 'key' necessary for a transaction is divided among several participants. In this case, the transaction is only authorised when the threshold is met.

Fuzzy logic provides a different approach to conventional binary logic by handling degrees of truth rather than strict true or false values. While its applications are diverse, its role in cryptography systems specifically threshold cryptography is not extensively researched. This report explores the possibility of using fuzzy logic in the world of cryptography to further enhance security.

This report outlines the development of a Fuzzy Inference System (FIS) using the fuzzy logic toolbox within MATLAB to produce a system designed to assess the degree of trust associated with each user within a threshold cryptographic system. Using these tools allowed me to design, test and develop a trust score FIS along with reflecting on the strengths and weaknesses of the system along with assessing its usability within real-world applications.

Literature Review

Background Areas

The foundational principles of fuzzy logic, which extend the binary nature of classical logic to introduce a range of possibilities, were first introduced by Lotfi A. Zadeh in his 1965 paper on fuzzy sets (Zadeh, L.A, 2004). This framework has been significant in modelling the complexities of human reasoning, as further highlighted by resources such as (MathWorks (A), no date), which details the application and implications of fuzzy logic in various fields. This approach is a refined way of handling information, which accommodates for the uncertainties in real-world scenarios. The Mamdani fuzzy inference model is known for its ability to closely align with human decision-making, highlighting how it is an essential tool in fields requiring complex judgment such as those in threshold cryptography (MathWorks (B), no date).

The integration of fuzzy logic into systems that manage trust aligns with current trends in digital security, as it offers a practical approach to evaluate and authenticate user interactions, crucial for maintaining integrity in threshold cryptography systems (Scott, G, 2023). The literature indicates that while the concept of fuzzy logic is well-established, its implementation within threshold cryptography is still an area ripe for exploration.

Threshold cryptography is a fundamental concept in secure digital communication. It stems from the idea that cryptographic operations can be divided and shared among multiple people, ensuring that no single person holds complete control (Yvo G Desmedt, 1994). This approach enhances security by requiring an agreement between a subset (threshold) of users therefore mitigating the risks associated with a single point of failure. As noted by (Shamir, A 1979) , threshold cryptography not only strengthens security protocols in various applications but also highlights the importance of maintaining the integrity of a system even when parts are compromised.

Fuzzy Logic in Threshold Cryptography

This is why the possibility of integrating fuzzy logic into threshold cryptography systems presents a compelling advancement. Alpos, O. and Cachin, C. (1970) proposed the idea of distributed cryptography that sways away from the conventional threshold constraints, advocating for adaptable and expressive trust models. The work by Odyurt, U. (2014) explores the idea of Fuzzy identity-based encryption, showcasing that fuzzy logic is suitable for the creation of error-tolerant cryptographic systems that adapt to outputs rather than simply shutting a user out after one mistake. This closely aligns with the idea of a trust score FIS that can interpret varying degrees of trustworthiness, alerting the nature of user interactions depending upon their respective trust levels.

The work by B. Kazemian, H. and Ma, Y. (no date) showcases the integration of fuzzy logic into Public Key Encryption and provides a framework for incorporating fuzzy logic into cryptographic systems which compost security. Similarly, the study by Sahu, R. and Pradhan, T. (2023) illustrates the practical application of fuzzy logic in refining capabilities within encrypted databases. These advancements highlight the relevance of fuzzy logic in cryptographic environments, as explored by Vasani, V. and Chudasama, V. (2018), which showcases that fuzzy-based trust models can be effectively used within cloud computing to decide which centres are the most trusted and suggests that this can be transferred to threshold cryptography, offering a more dynamic and context-sensitive assessment of trust. Together, these studies underline a growing yet promising area for research, pointing towards integrating FIS in cryptographic protocols to achieve a higher degree of security and tailored trust management.

Real-World Application

Despite the ever-growing advancements of fuzzy logic in the majority of fields the use of fuzzy logic in threshold cryptography is absent. Despite this trust score FIS is present in real-world systems.

An example of this, is the integration of fuzzy logic into a trust evaluation system within cloud computing that offers a practical example that can be used as a foundation to transfer this system into threshold cryptography systems. In a study by **Rathi, P. (2017)**, the use of fuzzy logic in cloud computing is explored for determining user privileges based on their trust scores. This approach addresses the complexity of determining user access management. The methodology developed in this study for assessing trust and managing user access privileges is particularly relevant to threshold cryptography, where the trustworthiness of users is a critical factor in granting access to sensitive data.

In threshold cryptography, similar to cloud computing, the challenge is establishing a reliable system to evaluate the trust levels of various users. The application of fuzzy logic, as demonstrated in the cloud computing study, offers a practical solution. It enables a system where access rights and privileges are not purely binary but are based on the varying degrees of trust assigned to each user. This approach allows for a more flexible and secure management of cryptographic tasks, adapting to the changing trust levels among users.

This past literature highlights what is needed in order to assess a user's trust within many different systems, some key aspects that stood out are how often users contributed to actions within a system such as attempting to gain access to data that they shouldn't, authorisation Success, and how often users comply to security policies are crucial elements. The importance of monitoring regular contribution is highlighted by a Ponemon Institute report, which reveals the risks associated with excessive user access to sensitive data (Ponemon Institute, 2023). Meanwhile, a

high authorisation success rate is indicative of effective access control and authentication mechanisms, playing a vital role in preventing unauthorised access (SecurityScorecard, 2023). Furthermore, adherence to security policies is pivotal in creating a trustworthy environment (Ethico, 2023). Collectively, these aspects provide a comprehensive view of user behaviour and reliability within a secure system framework.

Adopting the fuzzy logic-based trust evaluation system from cloud computing and assessing the necessary aspects of a user within a system can be transferred to threshold cryptography which could revolutionise the way access control and user privileges are managed, leading to enhanced security and efficiency. It would result in the potential for a more adaptive, responsive, and secure cryptographic environment, where user privileges are aligned with their demonstrated trustworthiness.

Challenges

There is a range of challenges that appear when considering the possibility of integrating fuzzy logic into threshold cryptography. A significant challenge in traditional threshold cryptography is the equal distribution of power among all parties, regardless of their trustworthiness. In such systems, every participant has the same level of authority, which can cause problems if certain users are less reliable or pose higher security risks.

A solution to this problem suggested by Lysyanskaya, A. (2000) introduces new measures of security for threshold schemes, particularly focusing on adaptively secure threshold protocols focusing on adaptively secure threshold protocols that can be efficiently implemented for various cryptographic applications. The key part of this solution is its adaptability to user trust levels. By constructing threshold cryptosystems based on the Cramer-Shoup scheme, the study proposes systems that are secure against adaptive chosen ciphertext attacks. This adaptability means that the system can adjust the cryptographic power and privileges of each user based on their trustworthiness, assessed through the protocol.

Another problem that stands out is the complexity of the system. Ensuring that both the outputs and inputs of the system are straightforward and clear ensuring that the user is able to identify with ease the trustworthiness of each individual user whilst still providing a clear verdict to what degree a user can be trusted. A solution to this problem would be ensuring the inputs to the system are relatable and can be easily inputted whilst still providing an accurate and detailed result on a user's level of trust.

The challenges highlighted provide a framework for the fuzzy inference system to be built on, ensuring users only have a level of power that matches their level of trust.

Future Trends & Conclusion

The future of fuzzy logic within threshold cryptography is an area set for significant advancements. The trend of this collaboration is moving forward to producing intelligent systems that not only dynamically adapt to varying trust levels of users but into many varying areas of cryptography. Further enhancing the security of cryptographic systems and the assessment of users that use these systems. These coming advancements are set to revolutionise the field, making cryptographic systems not only more secure but also more responsive to the complexities of user behaviour and trust dynamics.

This literature review highlighted the need for developing a system that balances detailed fuzzy logic algorithms with user trust evaluations, ensuring both efficiency and strong security measures. It highlights the importance of combining advanced methods such as fuzzy logic and threshold

cryptography, to produce a reliable cryptographic system tailored to individual trust dynamics further enhancing the security and efficiency of the system.

System Overview

Approach to the Problem

There is no denying that trusting someone with sensitive information or the power to control such information can be a challenge. In addressing this problem within the world of threshold cryptography, my approach is to use a fuzzy logic inference system to produce a trust value for each user. Providing a way to decide who can be trusted more than others. Recognising the limitations of traditional binary logic systems, where trust is often a simplistic yes/no result, my system introduces a multi-faceted trust score, depicted from a range of behavioural indicators that were identified during my literature review.

This approach allows for an assessment to be made of each user's trust level, allowing for a trust score to be assigned directly to each user's level of authority within a threshold cryptographic system. This ensures that more power is granted only to those users deemed most trustworthy, safeguarding the system against the risks posed by uniformly distributed privileges. Such that the threshold may be adaptive and increase/decrease depending on the trust score of each user. Having a threshold made up of users who have been deemed untrustworthy in comparison to others users within the system can pose a huge security risk and therefore my FIS will allow for an assessment to be made of which users can be deemed trustworthy enough to meet the threshold requirement and therefore allow the cryptographic action to take place.

System Layout

My system, fisTrust (See appendix 1), makes use of three inputs; Authentication Success Ratio, Policy Compliance Score, Regularity of Contributions, as highlight in my literature review, to produce an output variable; Trust Score. Each of these inputs were chosen due to the literature highlighted previously in this report. These three inputs were deemed key in determining how much a user can be trusted. These inputs will allow for a user's trust to be comprehensively assessed within a threshold cryptographic system:

1. Authentication Success Ratio: Selected for its direct reflection of a user's security judgement, this input measures successful system access attempts, highlighting user reliability.
2. Policy Compliance Score: This input evaluates adherence to system rules, crucial for identifying users who consistently maintain system integrity.
3. Regularity of Contributions: Chosen to gauge active involvement in cryptographic tasks, frequent contributions imply a user's commitment and dependability.

Design Considerations

During the initial design of my system, I experimented with using a range of different graphs to represent the membership functions. I initially started by using only trapezoidal membership functions (trapmf), (See Appendix 3) along with also experimenting with using only triangular membership functions, (trimf), (See Appendix 4). I found that while 'trimf' offered sharp distinctions between trust levels, I also found that 'trapmf' stood out to me as it was able to smoothly capture the gradual transitions, crucial for trust evaluation. Therefore, I decided to experiment with combining both of these types to produce a system that utilises both types (see appendix 5). I also

considered the possibility of using the Gaussian membership function (Gaussmf), (see Appendix 6) however this did not represent the membership functions distinctively. Therefore, I chose to go with the combination of both 'trapmf' and 'trimf'.

During the initial construction of my system, I decided to implement a base set of rules consisting of 27 rules as calculated by multiplying all three membership functions of the three inputs (3 * 3 *3 = 27), see Appendix 2. The base set of rules provides a starting point for the system to evolve from during testing up to a maximum of 135 rules achieved through multiplying all membership functions of both all the inputs and the output (3*3*3*5 = 135). Expanding on this rule base depending on the test rules will allow for the system to be refined in its accuracy and performance.

Despite these preliminary choices, I acknowledge the system's evolving nature. Post-testing adjustments, guided by example real-world data, will fine-tune the system's design to enhance the accuracy of its outputs. Therefore, the system is still subject to change in many areas such as membership functions, graph representations used, and the rule base all dependent on the results highlighted by the real-world testing later in this report.

Technical Description

Trust Score FIS

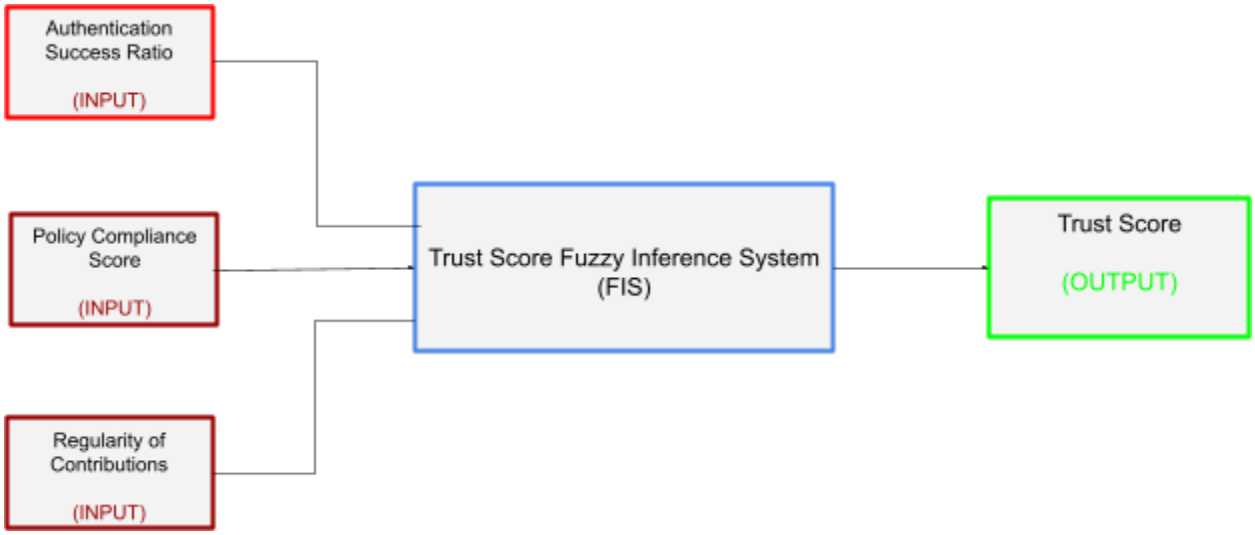


Figure 1: Diaragam showcasing the Trust Score FIS

Description of the System

My Trust Score FIS will utilise three inputs (highlighted in the red boxes) to produce one output (highlighted in the green box) the trust score. The selection and structure of this system reflects the key components of evaluating the amount you can trust a user.

Variable	Type of Variable	Range	Intervals
Authentication Success Ratio	INPUT	0-100%	Low, Medium, High
Policy Compliance Score	INPUT	0-100%	Poor, Average, Excellent
Regularity of Contributions	INPUT	0-30	Infrequent, Occasional. Frequent
Trust Score	OUTPUT	0-1	Untrustworthy, Somewhat Trustworthy, Average Trust, Highly Trustworthy, Very Trustworthy

Figure 2: A table showcasing the variables and their types along with their chosen ranges and intervals.

Justification for the chosen parameters

Authentication Success Ratio: The 0-100% range captures the entirety of user authentication attempts, accounting for occasional slip-ups and access issues. The intervals reflect varying levels of user reliability and security compliance.

Policy Compliance Score: Also expressed as a percentage, this input accurately represents the extent of user compliance with system policies. The intervals allow for differentiation between users who rarely follow rules, those who generally comply, and those who consistently observe policies.

Regularity of Contributions: Measured over a scale of 0-30, this reflects the frequency of user engagement in key system operations over the space of a month. The intervals indicate the user's degree of active participation, which is critical in trust assessment.

Trust Score: The output is a continuous scale from 0-1, where nuanced intervals from Untrustworthy to Very Trustworthy provide a granular view of trust. This scale allows for a more detailed and graduated assessment of trustworthiness, crucial for informed decision-making within the system.

Future Direction of the Trust Score FIS:

I recognise that the use of only 3 inputs to determine how much a user can be trusted will be limited in the accuracy of the output. Therefore, I have considered the possibility of expanding my system beyond 1 system made up of 3 inputs and 1 output to create the idea of using 3 systems utlising a total of 6 inputs and 3 systems to produce a trust score which will produce a more accurate output due to the more in depth assessment of a users trust by assessing 6 inputs rather than only 3. However due to time constraints and the idea only being theoretical at the time of this report and time constraints, I decided to use the design of 1 system that utlises 3 inputs and 1 output as this will allow me to build a foundation of system before choosing to further expand the system.

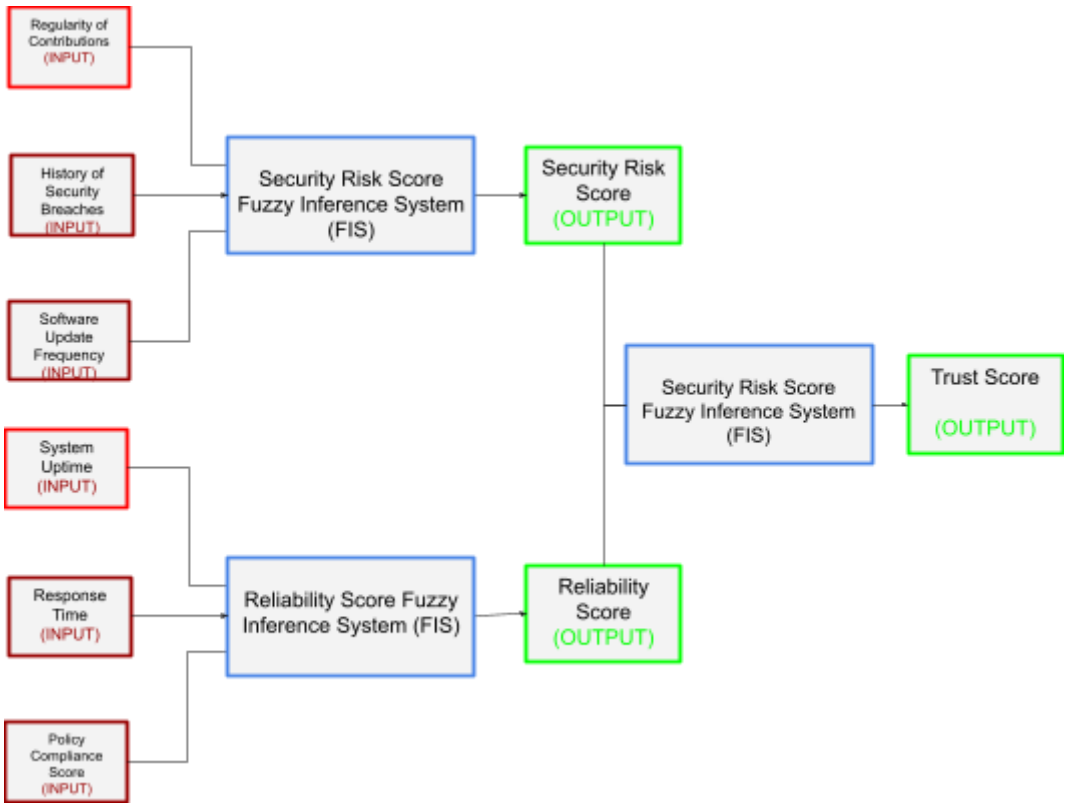


Figure 3: A diagram showcasing the possibility of expanding the FIS to produce a more accurate result.

Variable	Type of Variable	Range	Intervals
Regularity of Contributions	INPUT	0-30	Infrequent, Occasional. Frequent
History of Security Breaches	INPUT	0-100	None, Few, Many
Software update Frequency	INPUT	0-365	Rarely, Sometimes, Often
Security Risk Score	OUTPUT	0-1	Low Risk, Moderate Risk, High-Risk
System Uptime	INPUT	0-100%	Low, Moderate, High
Response Time	INPUT	0-24	Slow, Average, Fast
Policy Compliance Score	INPUT	0-100%	Poor, Average, Excellent
Reliability Score	OUTPUT	0-1	Unreliable, Fairly Reliable, Reliable
Trust Score	OUTPUT	0-1	Untrustworthy, Somewhat Trustworthy, Average Trust, Highly Trustworthy, Very Trustworthy

Figure 4: A diagram showcasing expansions of new variables and their associated ranges and intervals.

Justification for the chosen parameters

Inputs:

1. **Regularity of Contributions:** Scaled to a month's activity, it accurately reflects user engagement in cryptographic tasks.
2. **History of Security Breaches:** A higher value indicates more breaches, Helps to assess the risk a user might bring to the system.
3. **Software Update Frequency:** Encourages regular updates, a critical behaviour for cybersecurity, with the range allowing for daily to annual update frequencies.
4. **System Uptime:** Higher percentages are indicative of stable usage, correlating with user reliability and system stability.
5. **Response Time:** A critical measure in time-sensitive operations, with lower times, indicating more efficient user interactions.
6. **Policy Compliance Score:** Directly correlates with the user's adherence to system policies, essential for maintaining system security.

Outputs:

1. **Security Risk Score:** Provides a nuanced assessment of user risk, influencing trustworthiness evaluations and access control decisions.
2. **Reliability Score:** Distinctly rates user reliability, important for delegating responsibilities within the system.
3. **Trust Score:** Offers a spectrum of trust, from 'Very Low Trust' to 'Very High Trust', crucial for nuanced access control based on trustworthiness.

Experimental Design & Evaluation

General Notes

In the initial design phase, a variety of graphical representations were considered to best model the fuzzy logic system. The decision was made to use a combination of trapezoidal (trapmf) and triangular (trimf) membership functions, as they appeared most promising in the initial system design. However, the exclusive use of trapezoidal graphs was also a standout option and remains a consideration. The foundational rule base consists of 27 rules, with the possibility to increase up to 135 to include additional refinements identified during the testing phase. The overall aim of the testing process is to refine the system, expanding or modifying the rule base and membership functions as needed, to enhance the accuracy and reliability of the system's outputs.

Data and Test Subjects:

I decided to focus on 3 different demographics for my test data. I decided to cover professionals with varying amounts of experience, as shown in full in Appendix 7 & X. Overall I expect to see that the professionals will less years of experience would have a lower trust score and those with the most years of experience would have higher trust scores.

Whilst finding research and example data about varying demographics who are part of threshold cryptographic system posed a challenge to find relevant examples. However, I was able to find some example data such in other areas that I could interpret, Naser Alraja, M. (2023) conducted extensive research on policy compliance across a global setting. Overson,J.(2019) work showcases how authorisation success rate can affect someone's credentials. M, K (2021)'s work allowed me to gain an insight into how humans contribution to systems can be both harmful and helpful. Using these

sources, I was able to produce a set of synthetic data which was used to test the system throughout multiple tests see Appendix 8 for the full data sets.

Test 1

Goal of Test 1

The goal of the test 1 is to assess the preliminary configuration of the system by exposing it to real-world synthetic data (See appendix 8). It will evaluate how the system operates and how the effectiveness of the current membership functions and rule base. To do this I have created some expected outcomes to compare the results of my test too (Appendix 9). I have decided to test 5 different defuzzification methods (LOM, SOM, MOM, Bisector & Centroid) during my testing stage until 1 method stands out as the most compatible.

Results & Analysis of Test 1

The overall results of this test were not what was expected of the system highlighting the need for adjustments to the setup of the system (results can be seen in full through appendix 10). When comparing the results against my expected outcomes (See Appendix 11) I noticed it showcased a clear biased towards outputting higher trust scores than what was expected. Overall, the results of the test showcased a large amount of high trust score suggesting that the input membership functions are too broad.

Test 2

Changes Made

The primary change I decided to make to the system was to use only trapezoidal membership functions to represent all my inputs and output. During the initial testing I believe using a combination of trapezoidal and triangular membership functions suited my system best however due to the results of test 1 I implemented this change. As it improved how my membership functions were presented. I also expanded the rule base as the results of test 1 showcased some outputs of 0 indicating no rules were firing and outputs of 1 indicating incorrect rules. See appendix 12 for all changes. Expected outcomes were also changed after viewing the correlation of data from the previous test (See Appendix 13 for updated expected outcomes)

Results & Analysis of Test 2

The results of test 2 (see appendix 14), showcased a major improvement to the systems functionality as the outputs were more diverse and spread out of the expected outcomes. Comparing the results of this test against my expected outcomes further showcased the systems improvement using exclusively trapezoidal membership functions (see appendix 15). The defuzzification methods that best represented my system's outputs for this test was SOM, Bisector and Centroid as it presented an accurate distribution of outputs more accurate than other defuzzification methods.

Test 3

Changes Made

After reviewing the data and outputs from the last test it allowed me to gain a better insight into what I expect to see from my system therefore the expected outcomes for this test were altered (See appendix 17). The results of test 3 highlighted a large amount of highly/very trustworthy outputs therefore I decided to make adjustments to the spread of these membership functions to reduce the amount of these outputs. Additional rules were also added to attempt to solve some of the previous false outputs from test 2. (See appendix 16 for all changes)

Results & Analysis of Test 3

Overall, the results of test 3 showcased a major improvement to the system's output accuracy. Increasing the number of outputs matching the expected outcomes of the test. Once again, SOM, Bisector and Centroid continued to stand out as the best defuzzification method for my system's outputs. However, some results especially for the higher outputs such as highly and very trustworthy were outputting the same result. (See Appendix 18 & 19 for results and comparison). Overall, this was the best the system has looked so far.

Test 4

Changes Made

Approaching this test, the goal was to further improve the system's accuracy and configuration, in an attempt to improve on the results of the previous test. The changes made in order to try and achieve this was to add additional rules for the highly and very trustworthy outputs to make it so all outputs were no longer the same, along with adding rules to further increase the amount of average trust outputs (See appendix 20). These changes aimed to make it so the outputs better aligned with the new expected outcomes (appendix 21).

Results & Analysis of Test 4

During this test I only test the output defuzzification methods of MOM, Bisector and Centroid as these methods have been highlighted as the best methods throughout all previous testing. The results of this test did not show any improvement in comparison to the results of test 3, as the system was unable to differentiate between highly and very trustworthy outputs. Overall, this was not an improvement on test 3 therefore the system used for test 3 is presented as the best configuration. See appendix 22 & 23 for the results and comparisons of this test.

Final System Configuration

The final configuration of my Trust Score fuzzy inference system is the configuration used in test 3 because when using the smallest of maximum (SOM) defuzzification method the system presented itself and its outputs as the most accurate and representative of what was expected of the system's outcomes.

The full system configuration can be found in **Appendix 24.**

Critical Reflection

The development of the fuzzy inference system designed to produce user trut scores for users within a threshold cryptography system has been a process marked by notable strengths and weaknesses. The foundation for the system and its idea was set out by the literature review despite the limiting amount of literature on this topic. Reviewing literature in related areas played a crucial role in understanding what inputs were best suited for the system in order to produce accurate results that reflected the amount of trust associated with a user.

An in-depth amount of testing was conducted on the system's initial design allowing me to gain an insight into its initial functionality. Major changes were made to the system especially in the way its membership functions were represented. These changes were implemented between tests 1 & 2 and showed a significant improvement in the system's functionality. The changes made for test 3 also showed a significant improvement in the outputs produced by the system, however in an attempt to further improve the system for test 4, the changes implemented unfortunately did not add to the accuracy of the system. It became clear that further testing, if not bounded by time constraints, would be crucial in refining the system's accuracy and reliability.

The final choice of defuzzification method chosen was Smallest of Maximum (SOM), as this method throughout all testing showcased the most accurate and evenly spread outputs. However, it was made clear there was some imperfections in choosing this defuzzification method. Which potentially highlights the need for a specific defuzzification method designed with the user trust score system in mind, to further improve the system's functionality, outputs and accuracy.

Moreover, the current rule base does not cover all data input scenarios. With additional time, expanded testing could resulted in an extension the rule base, broadening the system's application to all areas. Covering all possible scenarios is crucial in ensuring the system's ability to be applied to the real world, as any untested data could reveal gaps in the system's decision-making.

The implementation of the system and its idea promises to add an additional layer of security to threshold cryptography systems. As it assesses the level of trust between the users who make up the threshold, ensuring that the threshold is made up of a set of users who you can trust. Furthermore, as time goes on more literature and insights to data related to both fuzzy logic in threshold cryptography and threshold cryptography itself will become more apparent. Therefore, the possibility of returning to this project in the future when more literature and data is available could result in further developing the system and functionality. The opportunity of implementing this fuzzy inference system into my development-based project also highlights its self as an exciting opportunity to combine two areas of research with real world application.

In summary, the idea and concept behind the idea provides a foundation of a path to success in implementing fuzzy logic into the world of threshold cryptography. With continued development the system could significantly advance in its functionality and accuracy, making it a crucial tool in the field of threshold cryptography due to its ability to further enhance security. This initial project lays the foundation for what could become a key component in the ever-evolving landscape of cybersecurity.

Conclusion

In concluding this venture of exploring the possibility of implementing fuzzy logic into the world of threshold cryptography, it has provided me an understanding of the theory and practical areas of fuzzy logic, The entire process has been documented in detail in this report, showcasing the evolution of taking the idea behind the project and making it into a reality.

The findings are indicative of the system's potential to serve as a foundation within the world of threshold cryptography, suggesting that a well-tuned fuzzy inference system could significantly improve trust management. While the current system is not without its imperfections, it stands as a testament to the viability of fuzzy systems in complex security systems.

The implications of integrating such a system into threshold cryptography are far-reaching. Not only does it have the ability to enhance security measures, but it also introduces a dynamic and adaptable approach to trust assessment, which is critical in the ever-evolving landscape of cyber threats. This also suggest that the idea for the future direction of this project presented in the technical description section of this report highlights how the system can be expanded to incorporate more inputs assessing both the security risk and reliability of users making the trust scores more accurate. This is an area which I would be excited to explore in the future, further perfecting the system.

In the future further research is set to refine the system’s capabilities. Future work on this project should aim to expand the rules base, optimise the defuzzification process and explore the implementation of expanding the system beyond its final configuration. The theoretical concept behind the idea is still presented as a much-needed area within the field of threshold cryptography. Whilst this system represents a big step forward in further enhancing the security of threshold cryptography, it is only the first step of long journey towards a more secure and trustworthy world of threshold cryptography.

References:

Mehta, H. (no date) Cyber security bizz - everything about hacking!: Nas.io communities, Nas.io is the All-in-One tool you need to turn your community into a business. Available at: https://nas.io/cybersecurity (Accessed: 02 December 2023).

MathWorks (A). (no date) *What is Fuzzy Logic?*, *MATLAB & simulink - mathworks United Kingdom*. Available at: https://uk.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html (Accessed: 30 November 2023).

MathWorks (B). (no date) *Mamdani and Sugeno Fuzzy Inference Systems*, *Mamdani and Sugeno Fuzzy Inference Systems - MATLAB & Simulink - MathWorks United Kingdom*. Available at: https://uk.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html (Accessed: 30 November 2023).

Alpos, O. and Cachin, C. (1970) *Do not trust in numbers: Practical distributed cryptography with general trust*, *Cryptology ePrint Archive*. Available at: https://eprint.iacr.org/2022/1767 (Accessed: 30 November 2023).

B.Kazemian, H. and Ma, Y. (no date) *Fuzzy Logic Application to searchable cryptography*, *London Met Repository*. Available at: https://repository.londonmet.ac.uk/5863/1/Camera%20Ready%20Version%20for%20Fuzzy%20Logic%20Application%20to%20Searchable%20Cryptography.pdf (Accessed: 30 November 2023).

Lysyanskaya , A. (2000) *Adaptively Secure Threshold Cryptography: Introducing Concurrency, Removing Erasures*, *Springer Link*. Available at: https://link.springer.com/chapter/10.1007/3-540-45539-6_16 (Accessed: 30 November 2023).

Odyurt, U. (2014) *Application of Fuzzy Logic in Identity-Based Cryptography*, *Linnaeus University*. Available at: https://www.diva-portal.org/smash/get/diva2:725347/FULLTEXT03.

Rathi, P. (2017) *Rule based trust evaluation using fuzzy logic in cloud computing*, *IEEE Xplore*. Available at: https://ieeexplore.ieee.org/document/8342481 (Accessed: 30 November 2023).

Sahu, R. and Pradhan, T. (2023) *COMPREHENSIVE REVIEW OF CRYPTOGRAPHY BASED ON FUZZY LOGIC* , *EuroBull*. Available at: https://www.eurchembull.com/uploads/paper/1a9194080eff8f4eb7c021c03d02ecfe.pdf (Accessed: 30 November 2023).

Scott, G. (2023) *Fuzzy logic: Definition, meaning, examples, and history*, *Investopedia*. Available at: https://www.investopedia.com/terms/f/fuzzy-logic.asp#:~:text=Fuzzy%20logic%20is%20an%20approach,an%20array%20of%20accurate%20conclusions (Accessed: 30 November 2023).

Shamir, A. (1979) *How to share a secret - MIT*, *Massachusetts Institute of Technology*. Available at:

https://web.mit.edu/6.857/OldStuff/Fall03/ref/Shamir-HowToShareASecret.pdf (Accessed: 30 November 2023).

Vasani, V. and Chudasama, V. (2018) *A Trust Rating Model Using Fuzzy Logic in Cloud*, *Springer Link*. Available at: https://link.springer.com/chapter/10.1007/978-981-13-2354-6_36 (Accessed: 30 November 2023).

Yvo G Desmedt (1994) *Threshold cryptography - desmedt - 1994 - wiley online library*, *Wiley Online Library*. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4460050407 (Accessed: 30 November 2023).

Zadeh, L.A. (2004) *Fuzzy sets*, *Information and Control*. Available at: https://www.sciencedirect.com/science/article/pii/S001999586590241X?ref=pdf_download&fr=RR-2&rr=82e57a978d153691 (Accessed: 30 November 2023).

https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4460050407

Naser Alraja, M. (2023) *Information security policies compliance in a global setting: An employee's perspective*, *Computers & Security*. Available at: https://www.sciencedirect.com/science/article/pii/S0167404823001189#:~:text=The%20Unified%20Model%20of%20Information,was%20assessed%20in%20two%20phases. (Accessed: 28 November 2023).

Overson,J.(2019) What your login success rate says about your credential , Medium. Available at: https://jsoverson.medium.com/what-your-login-success-rate-says-about-your-credential-stuffing-threat-1f10bc20eaee (Accessed: 28 November 2023).

M, K. (2021) What role do humans play in ensuring cybersecurity?, ISACA. Available at: https://www.isaca.org/resources/isaca-journal/issues/2021/volume-5/what-role-do-humans-play-in-ensuring-cybersecurity (Accessed: 28 November 2023).

Prince, B. (2014) *Excessive employee access privileges expose corporate data to risk*, *SecurityWeek*. Available at: https://www.securityweek.com/excessive-employee-access-privileges-expose-corporate-data-risk-survey/ (Accessed: 20 November 2023).

Scorecard, S. (2023) *22 Cybersecurity Metrics & Kpis to track in 2023*, *SecurityScorecard*. Available at: https://securityscorecard.com/blog/9-cybersecurity-metrics-kpis-to-track/ (Accessed: 20 November 2023).

Ethico (2022) *How creating a culture of compliance leads to trust*, *Ethico, LLC*. Available at: https://ethico.com/how-creating-a-culture-of-compliance-leads-to-trust/ (Accessed: 20 November 2023).

Appendices

Pre-testing System Design (Matlab):

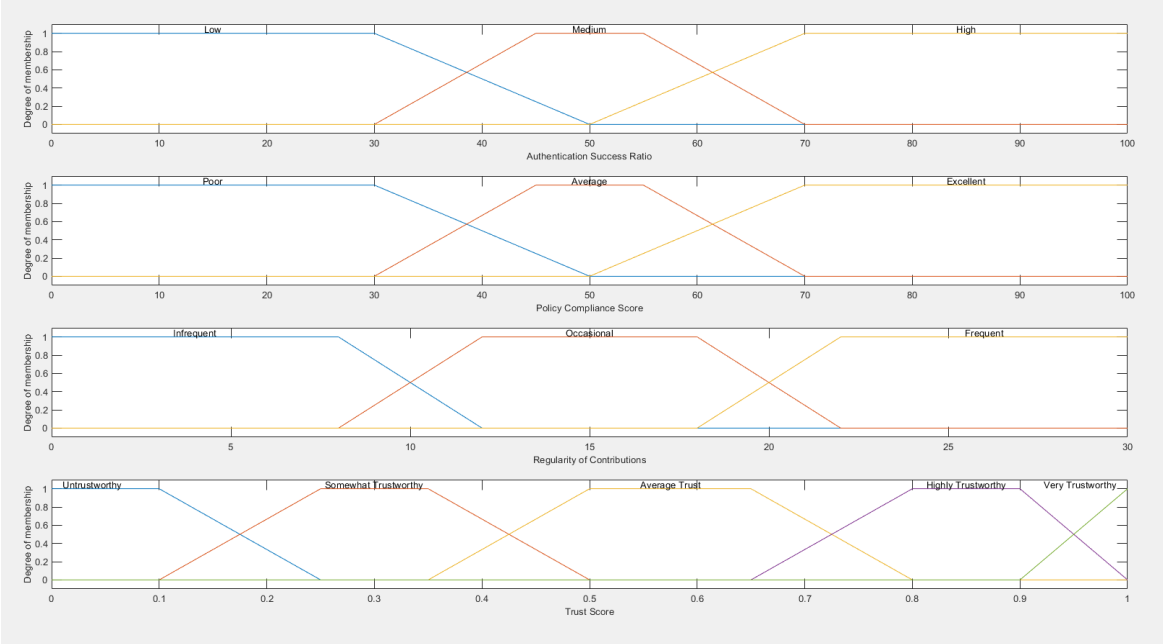
Appendix 1: Variable Declarations

```
% A declaration of new FIS
a = newfis('Trust Score');

% Declaring new variables - these are INPUTS
a=addvar(a,'input','AuthenticationSuccessRatio',[0 100]);
a=addvar(a,'input','PolicyComplianceScore',[0 100]);
a=addvar(a,'input','RegularityofContributions',[0 30]);

% Intervals the Authentication Success variable with membership functions
a=addmf(a,'input',1,'Low','trapmf',[0 0 30 50]);
a=addmf(a,'input',1,'Medium','trimf',[30 50 70]);
a=addmf(a,'input',1,'High','trapmf',[50 70 100 100]);

% Intervals for the Policy Compliance input variable with membership functions
a=addmf(a,'input',2,'Poor','trapmf',[0 0 30 50]);
a=addmf(a,'input',2,'Average','trimf',[30 50 70]);
a=addmf(a,'input',2,'Excellent','trapmf',[50 70 100 100]);

% Intervals the Regualrity of Contributions input variable with membership functions
a=addmf(a,'input',3,'Infrequent','trapmf',[0 0 8 12]);
a=addmf(a,'input',3,'Occasional','trimf',[8 15 22]);
a=addmf(a,'input',3,'Frequent','trapmf',[18 24 30 30]);

% Declaring a new variable - this is the OUTPUT
a=addvar(a,'output','Trust Score',[0 1]);

% Populating the output variable with membership functions
a=addmf(a,'output',1,'Untrustworthy','trapmf',[0 0 0.1 0.25]);
a=addmf(a,'output',1,'Somewhat Trustworthy','trimf',[0.2 0.35 0.5]);
a=addmf(a,'output',1,'Average Trust','trimf',[0.4 0.5 0.6]);
a=addmf(a,'output',1,'Highly Trustworthy','trimf',[0.5 0.65 0.8]);
a=addmf(a,'output',1,'Very Trustworthy','trapmf',[0.75 0.9 1 1]);
```

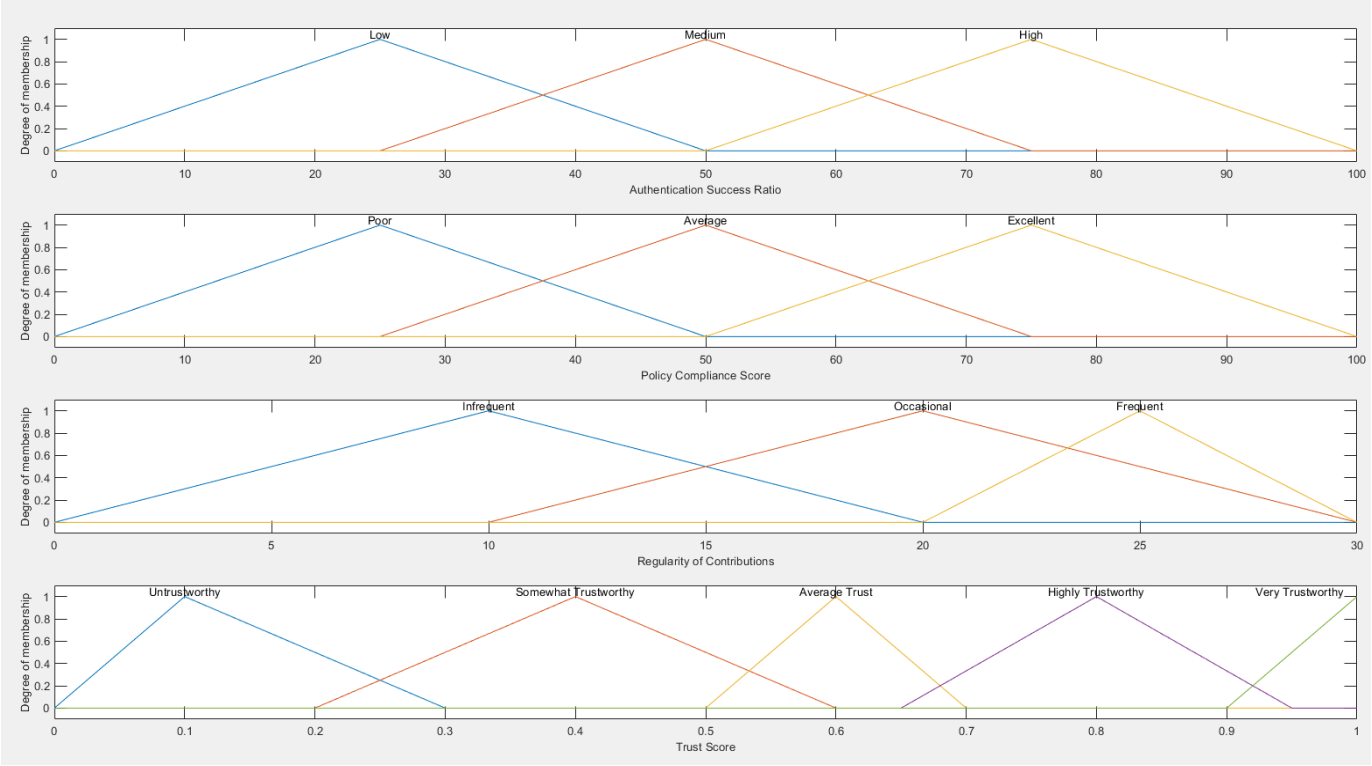
Appendix 2: Pre-testing System Rule Base

```
% Rule Base
rule1 = [1 1 1 1 1 1];
rule2 = [1 1 2 2 1 1];
rule3 = [1 1 3 3 1 1];
rule4 = [1 2 1 2 1 1];
rule5 = [1 2 2 3 1 1];
rule6 = [1 2 3 4 1 1];
rule7 = [1 3 1 3 1 1];
rule8 = [1 3 2 4 1 1];
rule9 = [1 3 3 5 1 1];
rule10 = [2 1 1 2 1 1];
rule11 = [2 1 2 3 1 1];
rule12 = [2 1 3 4 1 1];
rule13 = [2 2 1 3 1 1];
rule14 = [2 2 2 4 1 1];
rule15 = [2 2 3 5 1 1];
rule16 = [2 3 1 4 1 1];
rule17 = [2 3 2 5 1 1];
rule18 = [2 3 3 5 1 1];
rule19 = [3 1 1 3 1 1];
rule20 = [3 1 2 4 1 1];
rule21 = [3 1 3 5 1 1];
rule22 = [3 2 1 4 1 1];
rule23 = [3 2 2 5 1 1];
rule24 = [3 2 3 5 1 1];
rule25 = [3 3 1 5 1 1];
rule26 = [3 3 2 5 1 1];
rule27 = [3 3 3 5 1 1];

% The ruleList of all defined rules.
ruleList = [rule1; rule2; rule3; rule4; rule5; rule6; rule7; rule8;
    rule9; rule10; rule11; rule12; rule13; rule14; rule15; rule16;
    rule17; rule18; rule19; rule20; rule21; rule22; rule23; rule24;
    rule25; rule26; rule27
];
```

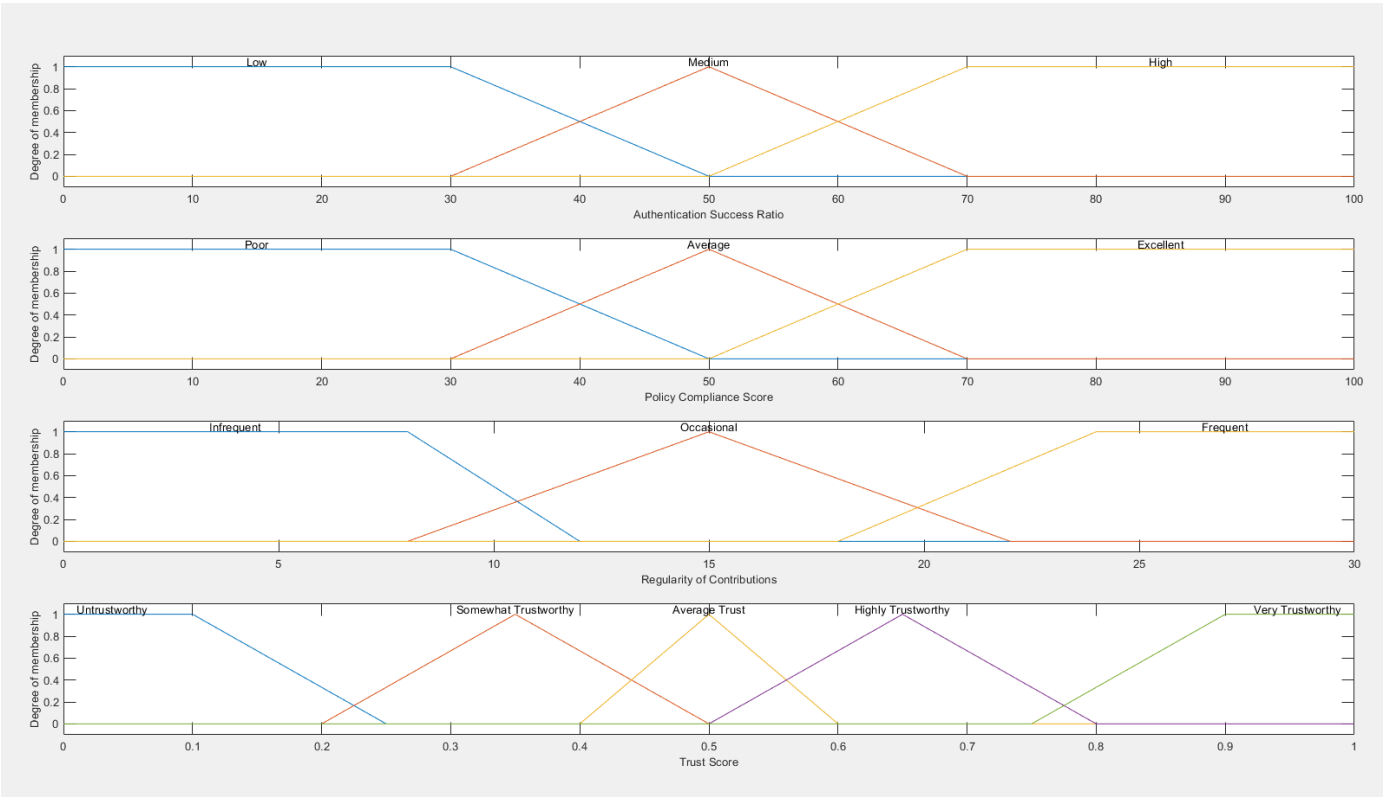
Appendix 3: Pre-testing System Fuzzy Set Distribution Graphs (trapmf)



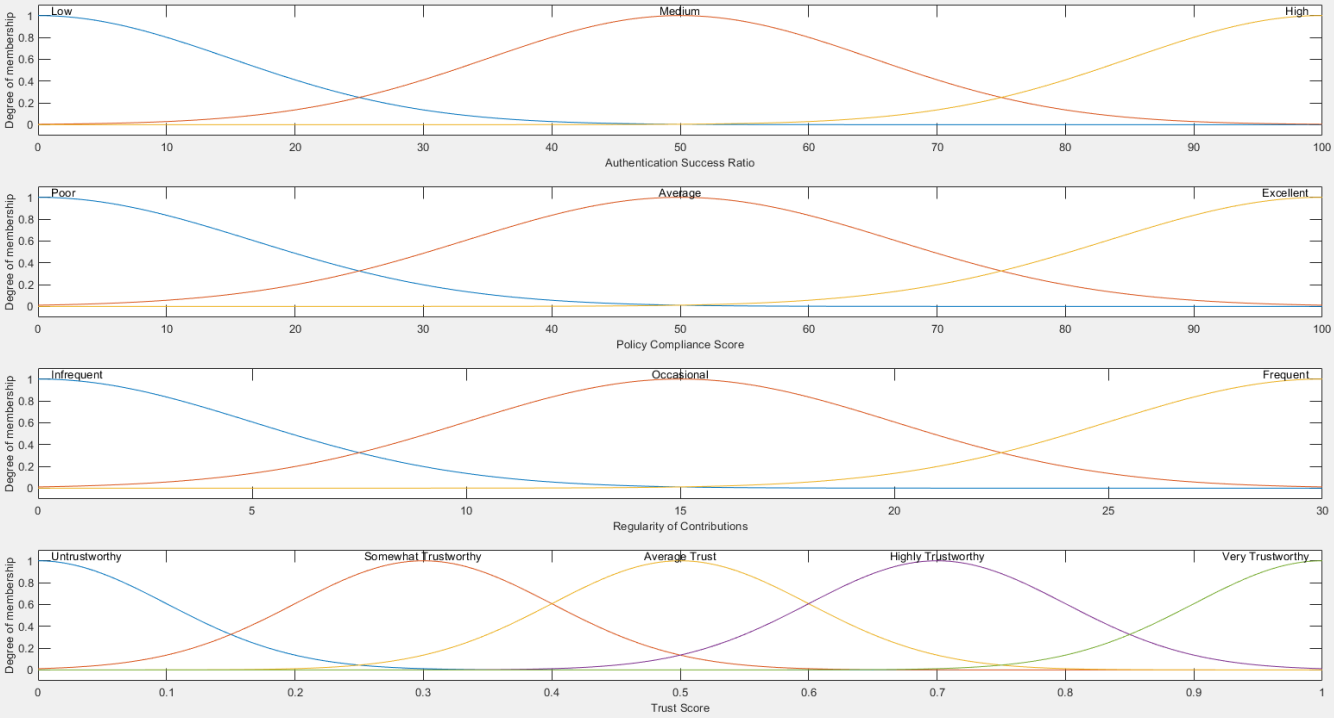
Appendix 4: Pre-testing System Fuzzy Set Distribution Graphs (trimf)



Appendix 5: Pre-testing System Fuzzy Set Distribution Graphs (trapmf & trimf)



Appendix 6: Pre-testing System Fuzzy Set Distribution Graphs (gausmf)



Testing:

Appendix 7: Test Data Demographics

Subject Group 1: Novice Professionals (0-5 Years of Experience)

This age demographic is adaptable but output results may be affected by their minimal experience in threshold cryptography systems.

1. **Authentication Success Ratio:** Likely high due to being familiar with technology and adaptive use of security measures.
2. **Policy Compliance Score:** May vary as this group could be more open to experimenting with new approaches, sometimes at the expense of strict policy adherence.
3. **Regularity of Contributions:** Expected to be high since younger professionals are often more involved in daily operations and decision-making processes.

Subject Group 2: Experienced Professionals (6-15 Years of Experience)

Experience professionals usually have a significant understanding of the role they play in security systems due to their often positions of greater responsibility.

1. **Authentication Success Ratio:** Expected to be high as this group generally has established routines and a clear understanding of the importance of secure logins.
2. **Policy Compliance Score:** Likely very high, as middle-aged professionals have had time to integrate compliance into their regular practices.
3. **Regularity of Contributions:** Expected to be consistent and reliable, reflecting their ongoing active role in critical operations.

Subject Group 3: Veteran Professionals (16+ Years of Experience)

This group comprises heavily experienced professionals. The majority will hold senior management roles making their decisions different to other demographics.

1. **Authentication Success Ratio:** This is expected to vary as highly experienced users will have high ratios, but those less comfortable with rapid technology changes may face challenges.
2. **Policy Compliance Score:** Generally high, as this group values established protocols and understands the consequences of non-compliance.
3. **Regularity of Contributions:** Potentially less frequent, as senior professionals may only take part in selected operational tasks but will make significant contributions when involved.

Appendix 8: Sample Test Data:

Test Number	Group	Authentication Success Ratio	Policy Compliance Score	Regularity of Contributions
1	Novice	45%	30%	30
2	Novice	50%	55%	15
3	Novice	55%	60%	20
4	Novice	60%	40%	25
5	Novice	65%	50%	18
6	Experienced	85%	75%	22
7	Experienced	80%	80%	20
8	Experienced	90%	85%	24
9	Experienced	88%	77%	19
10	Experienced	82%	79%	21
11	Veteran	95%	90%	8
12	Veteran	97%	92%	7
13	Veteran	97%	94%	6
14	Veteran	98%	93%	5
15	Veteran	99%	91%	9

Test 1:
Appendix 9: Expected Outcomes

Test Number	Expected Output Result
1 (Group 1)	Untrustworthy
2	Somewhat Trustworthy
3	Somewhat Trustworthy
4	Untrustworthy
5	Somewhat Trustworthy
6 (Group 2)	Average Trust
7	Highly Trustworthy
8	Highly Trustworthy
9	Average Trust
10	Average Trust
11 (Group 3)	Highly Trustworthy
12	Very Trustworthy
13	Very Trustworthy
14	Highly Trustworthy
15	Very Trustworthy

Appendix 10: Results of Test 1

LOM vs SOM vs MOM (Test 1)

	SOM	MOM	LOM
Test Number	Trust Score Output	Trust Score Output	Trust Score Output
1	0.62	0.65	0.68
2	0.62	0.65	0.68
3	0.80	0.90	1.00
4	0.58	0.79	1.00
5	0.84	0.92	1.00
6	0.86	0.93	1.00
7	0.80	0.90	1.00
8	0.90	0.95	1.00
9	0.82	0.91	1.00
10	0.83	0.92	1.00
11	0.90	0.95	1.00
12	0.90	0.95	1.00
13	0.90	0.95	1.00
14	0.90	0.95	1.00
15	0.87	0.94	1.00

Centroid vs Bisector Defuzzification Values (Test 1)

	Bisector	Centroid
Test Number	Trust Score Output	Trust Score Output
1	0.63	0.62
2	0.69	0.72
3	0.78	0.77
4	0.77	0.77
5	0.85	0.81
6	0.90	0.90
7	0.89	0.89
8	0.91	0.91
9	0.89	0.89
10	0.90	0.90
11	0.91	0.91
12	0.91	0.91
13	0.91	0.91
14	0.91	0.91
15	0.91	0.90

Appendix 11: Expected Outcomes vs Actual Outcomes (Test 1)

(SOM vs MOM vs LOM):

	SOM	MOM	LOM
Test Number	Trust Score Output	Trust Score Output	Trust Score Output
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
5	False	False	False
6	False	False	False
7	False	False	False
8	False	False	False
9	False	False	False
10	False	False	False
11	False	False	False
12	True	True	True
13	True	True	True
14	False	False	False
15	True	True	True

Expected Outcomes vs Actual Outcomes (Bisector vs Centroid)

Test Number	Bisector	Centroid
1	False	False
2	False	False
3	False	False
4	Flase	False
5	False	False
6	False	False
7	False	False
8	False	False
9	False	False
10	False	False
11	False	False
12	True	True
13	True	True
14	False	False
15	True	True

Test 2:

Appendix 12: Changes To system for Test 2

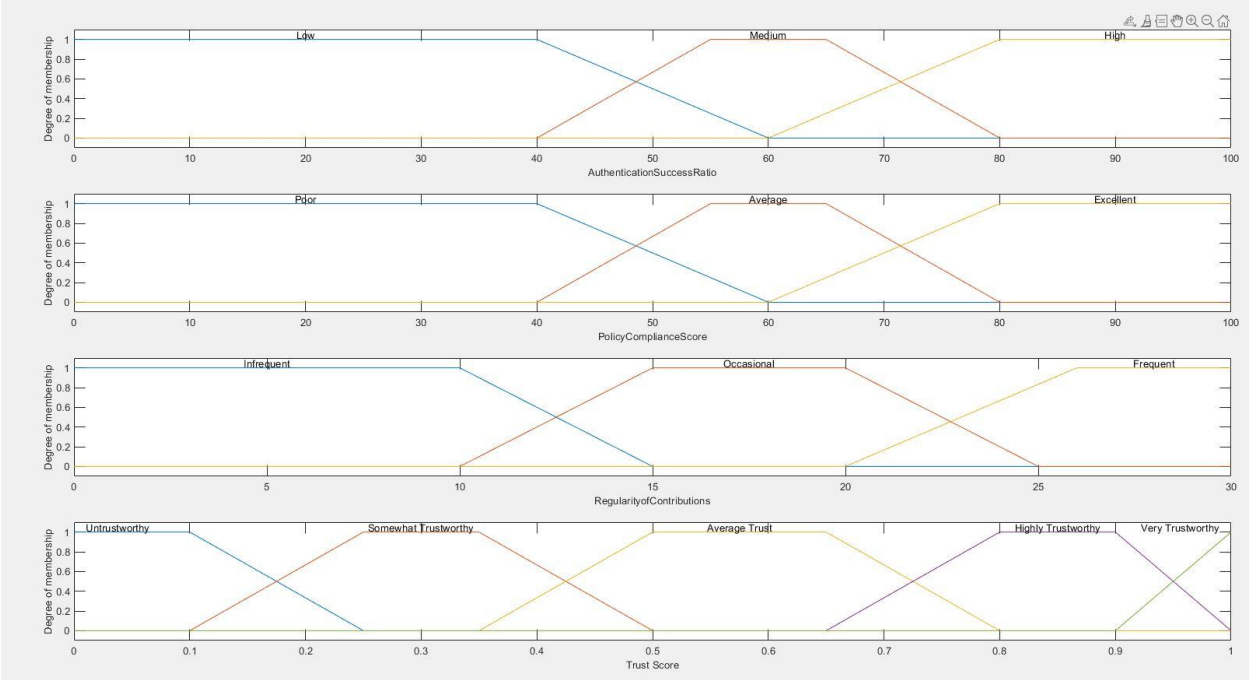
Membership Functions

```
% A declaration of new FIS
a = newfis('Trust Score', 'DefuzzificationMethod', 'lom');

% Declaring new variables - these are INPUTS
a=addvar(a,'input','AuthenticationSuccessRatio',[0 100]);
a=addvar(a,'input','PolicyComplianceScore',[0 100]);
a=addvar(a,'input','RegularityofContributions',[0 30]);

% Intervals the Authentication Success variable with membership functions
a=addmf(a,'input',1,'Low','trapmf',[0 0 40 60]);
a=addmf(a,'input',1,'Medium','trapmf',[40 55 65 80]);
a=addmf(a,'input',1,'High','trapmf',[60 80 100 100]);

% Intervals for the Policy Compliance input variable with membership functions
a=addmf(a,'input',2,'Poor','trapmf',[0 0 40 60]);
a=addmf(a,'input',2,'Average','trapmf',[40 55 65 80]);
a=addmf(a,'input',2,'Excellent','trapmf',[60 80 100 100]);

% Intervals the Regualrity of Contributions input variable with membership functions
a=addmf(a,'input',3,'Infrequent','trapmf',[0 0 10 15]);
a=addmf(a,'input',3,'Occasional','trapmf',[10 15 20 25]);
a=addmf(a,'input',3,'Frequent','trapmf',[20 26 30 30]);

% Declaring a new variable - this is the OUTPUT
a=addvar(a,'output','Trust Score',[0 1]);

% Populating the output variable with membership functions
a=addmf(a,'output',1,'Untrustworthy','trapmf',[0 0 0.1 0.25]);
a=addmf(a,'output',1,'Somewhat Trustworthy','trapmf',[0.1 0.25 0.35 0.5]);
a=addmf(a,'output',1,'Average Trust','trapmf',[0.35 0.5 0.65 0.8]);
a=addmf(a,'output',1,'Highly Trustworthy','trapmf',[0.65 0.8 0.9 1]);
a=addmf(a,'output',1,'Very Trustworthy','trapmf',[0.9 1 1 1]);
```

Changes to Rule base and List

```
% Rule Base
rule1 = [2 2 2 1 1 1];
rule2 = [2 2 3 2 1 1];
rule3 = [2 3 2 2 1 1];
rule4 = [3 2 2 3 1 1];
rule5 = [3 3 2 4 1 1];
rule6 = [3 2 3 4 1 1];
rule7 = [3 3 3 5 1 1];
rule8 = [1 1 1 1 1 1];
rule9 = [1 1 2 1 1 1];
rule10 = [1 1 3 2 1 1];
rule11 = [1 2 1 2 1 1];
rule12 = [1 2 2 2 1 1];
rule13 = [1 2 3 3 1 1];
rule14 = [1 3 1 2 1 1];
rule15 = [1 3 2 3 1 1];
rule16 = [1 3 3 4 1 1];
rule17 = [2 1 1 2 1 1];
rule18 = [2 1 2 2 1 1];
rule19 = [2 1 3 3 1 1];
rule20 = [2 2 1 3 1 1];
rule21 = [2 3 1 3 1 1];
rule22 = [2 3 2 4 1 1];
rule23 = [2 3 3 4 1 1];
rule24 = [3 1 1 3 1 1];
rule25 = [3 1 2 3 1 1];
rule26 = [3 1 3 4 1 1];
rule27 = [3 2 1 4 1 1];

% New Rules
rule28 = [3 3 2 5 1 1];
rule29 = [3 2 3 5 1 1];
rule30 = [3 3 1 4 1 1];
rule31 = [2 3 3 4 1 1];
rule32 = [3 3 3 5 1 1];
rule33 = [2 2 3 4 1 1];
rule34 = [3 2 2 4 1 1];
```

Updated Graphs for Test 2



Appendix 13: Expected Outcomes (Test 2)

Test Number	Expected Output Result
1 (Group 1)	Untrustworthy
2	Somewhat Trustworthy
3	Somewhat Trustworthy
4	Somewhat Untrustworthy
5	Somewhat Trustworthy
6 (Group 2)	Average Trust
7	Highly Trustworthy
8	Highly Trustworthy
9	Average Trust
10	Average Trust
11 (Group 3)	Highly Trustworthy
12	Very Trustworthy
13	Highly Trustworthy
14	Very Trustworthy
15	Very Trustworthy

Appendix 14: Test 2 Results

LOM vs SOM vs MOM (Test 2)

	SOM	MOM	LOM
Test Number	Trust Score Output	Trust Score Output	Trust Score Output
1	0.22	0.30	0.38
2	0.00	0.08	0.15
3	0.00	0.05	0.10
4	0.48	0.58	0.67
5	0.00	0.08	0.15
6	0.75	0.87	1.00
7	0.80	0.86	1.00
8	0.97	0.99	1.00
9	0.78	0.86	1.00
10	0.78	0.86	1.00
11	0.80	0.85	0.90
12	0.80	0.85	0.90
13	0.80	0.85	0.90
14	0.80	0.85	0.90
15	0.80	0.85	0.90

Centroid vs Bisector Defuzzification Values (Test 2)

	Bisector	Centroid
Test Number	Trust Score Output	Trust Score Output
1	0.37	0.40
2	0.20	0.21
3	0.12	0.16
4	0.57	0.58
5	0.33	0.38
6	0.77	0.73
7	0.85	0.85
8	0.91	0.88
9	0.82	0.78
10	0.84	0.82
11	0.84	0.84
12	0.84	0.84
13	0.84	0.84
14	0.84	0.84
15	0.84	0.84

Appendix 15: Expected Outcomes vs Actual Outcomes of Test 2

(SOM vs MOM vs LOM Test 2)

	SOM	MOM	LOM
Test Number	Trust Score Output	Trust Score Output	Trust Score Output
1	True	False	False
2	False	False	True
3	False	False	False
4	True	False	False
5	False	False	True
6	True	False	False
7	True	True	False
8	True	True	False
9	True	False	False
10	True	False	False
11	True	True	True
12	False	False	False
13	True	True	True
14	False	False	False
15	False	False	False

Expected Outcomes vs Actual Outcomes (Bisector vs Centroid Test 2)

	Bisector	Centroid
Test Number	Trust Score Output	Trust Score Output
1	False	False
2	False	False
3	True	True
4	False	False
5	True	False
6	True	True
7	True	True
8	False	True
9	False	True
10	False	False
11	True	True
12	False	False
13	True	True
14	False	False
15	False	False

Test 3:

Appendix 16: Changes to System for Test 3

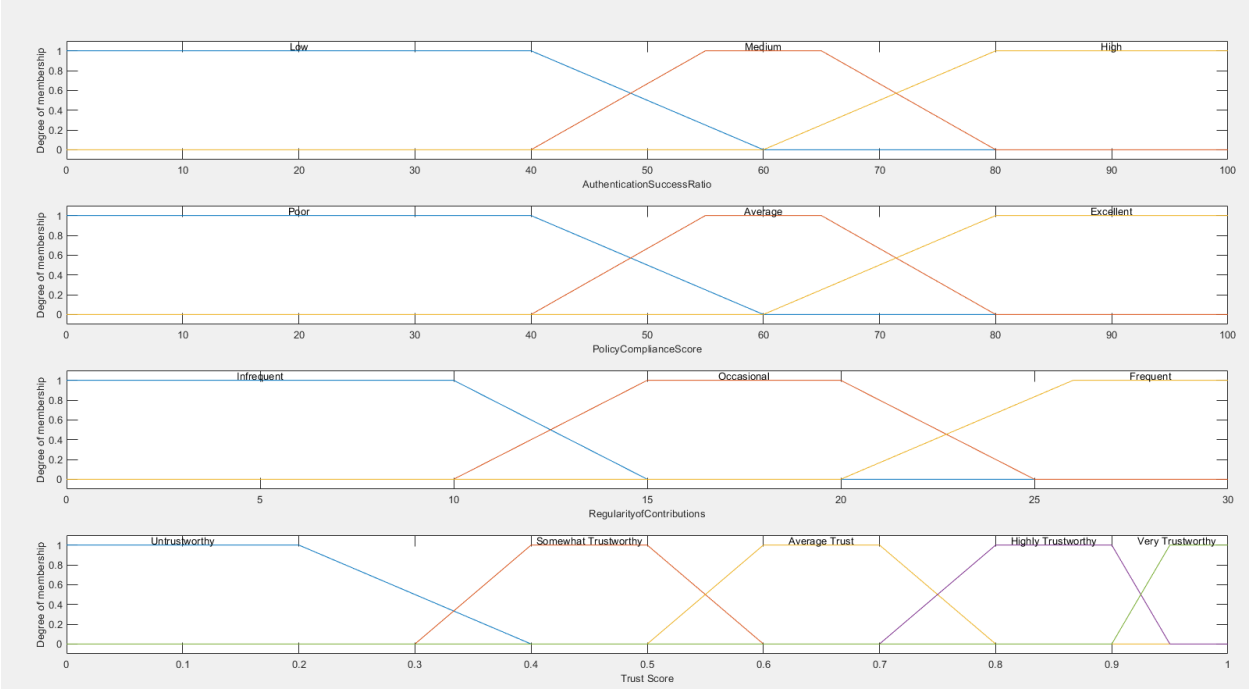
Changes to membership functions (Test 3)

```
% Declaring new variables - these are INPUTS
a=addvar(a,'input','AuthenticationSuccessRatio',[0 100]);
a=addvar(a,'input','PolicyComplianceScore',[0 100]);
a=addvar(a,'input','RegularityofContributions',[0 30]);

% Intervals the Authentication Success variable with membership functions
a=addmf(a,'input',1,'Low','trapmf',[0 0 40 60]);
a=addmf(a,'input',1,'Medium','trapmf',[40 55 65 80]);
a=addmf(a,'input',1,'High','trapmf',[60 80 100 100]);

% Intervals for the Policy Compliance input variable with membership functions
a=addmf(a,'input',2,'Poor','trapmf',[0 0 40 60]);
a=addmf(a,'input',2,'Average','trapmf',[40 55 65 80]);
a=addmf(a,'input',2,'Excellent','trapmf',[60 80 100 100]);

% Intervals the Regualrity of Contributions input variable with membership functions
a=addmf(a,'input',3,'Infrequent','trapmf',[0 0 10 15]);
a=addmf(a,'input',3,'Occasional','trapmf',[10 15 20 25]);
a=addmf(a,'input',3,'Frequent','trapmf',[20 26 30 30]);

% Declaring a new variable - this is the OUTPUT
a=addvar(a,'output','Trust Score',[0 1]);

% Populating the output variable with membership functions
a=addmf(a,'output',1,'Untrustworthy','trapmf',[0 0 0.2 0.4]);
a=addmf(a,'output',1,'Somewhat Trustworthy','trapmf',[0.3 0.4 0.5 0.6]);
a=addmf(a,'output',1,'Average Trust','trapmf',[0.5 0.6 0.7 0.8]);
a=addmf(a,'output',1,'Highly Trustworthy','trapmf',[0.7 0.8 0.9 0.95]);
a=addmf(a,'output',1,'Very Trustworthy','trapmf',[0.9 0.95 1 1]);;

% Rule Base
```

Changes to rule base (Test 3)

```
% Rule Base
rule1 = [2 2 2 1 1 1];
rule2 = [2 2 3 2 1 1];
rule3 = [2 3 2 2 1 1];
rule4 = [3 2 2 3 1 1];
rule5 = [3 3 2 4 1 1];
rule6 = [3 2 3 4 1 1];
rule7 = [3 3 3 5 1 1];
rule8 = [1 1 1 1 1 1];
rule9 = [1 1 2 1 1 1];
rule10 = [1 1 3 2 1 1];
rule11 = [1 2 1 2 1 1];
rule12 = [1 2 2 2 1 1];
rule13 = [1 2 3 3 1 1];
rule14 = [1 3 1 2 1 1];
rule15 = [1 3 2 3 1 1];
rule16 = [1 3 3 4 1 1];
rule17 = [2 1 1 2 1 1];
rule18 = [2 1 2 2 1 1];
rule19 = [2 1 3 3 1 1];
rule20 = [2 2 1 3 1 1];
rule21 = [2 3 1 3 1 1];
rule22 = [2 3 2 4 1 1];
rule23 = [2 3 3 4 1 1];
rule24 = [3 1 1 3 1 1];
rule25 = [3 1 2 3 1 1];
rule26 = [3 1 3 4 1 1];
rule27 = [3 2 1 4 1 1];
rule28 = [3 3 2 5 1 1];
rule29 = [3 2 3 5 1 1];
rule30 = [3 3 1 4 1 1];
rule31 = [2 3 3 4 1 1];
rule32 = [3 3 3 5 1 1];
rule33 = [2 2 3 4 1 1];
rule34 = [3 2 2 4 1 1];

%New rules for Test 3
rule35 = [1 1 2 2 1 1];
rule36 = [1 2 2 1 1 1];
rule37 = [2 2 1 2 1 1];
rule38 = [2 3 2 3 1 1];
rule39 = [3 2 2 3 1 1];
rule40 = [3 3 1 4 1 1];
rule41 = [3 3 1 5 1 1];
```

New Graphs (Test 3)



Appendix 17: Expected Outcomes (Test 3)

Test Number	Expected Output Result
1 (Group 1)	Somewhat Trustworthy
2	Untrustworthy
3	Untrustworthy
4	Average Trust
5	Untrustworthy
6 (Group 2)	Average Trust
7	Highly Trustworthy
8	Average Trust
9	Highly Trustworthy
10	Average Trust
11 (Group 3)	Highly Trustworthy
12	Highly Trustworthy
13	Very Trustworthy
14	Highly Trustworthy
15	Very Trustworthy

Appendix 18: Test 3 Results

LOM vs SOM vs MOM (Test 3)

	SOM	MOM	LOM
Test Number	Trust Score Output	Trust Score Output	Trust Score Output
1	0.38	0.45	0.52
2	0.00	0.13	0.26
3	0.00	0.10	0.20
4	0.59	0.65	0.71
5	0.00	0.13	0.26
6	0.77	0.88	1.00
7	0.80	0.89	1.00
8	0.94	0.97	1.00
9	0.79	0.88	1.00
10	0.78	0.88	1.00
11	0.80	0.89	1.00
12	0.80	0.89	1.00
13	0.80	0.89	1.00
14	0.80	0.89	1.00
15	0.80	0.89	1.00

Centroid vs Bisector Defuzzification Values (Test 3)

	Bisector	Centroid
Test Number	Trust Score Output	Trust Score Output
1	0.49	0.52
2	0.25	0.26
3	0.17	0.20
4	0.65	0.65
5	0.32	0.37
6	0.81	0.80
7	0.87	0.87
8	0.93	0.90
9	0.85	0.83
10	0.86	0.85
11	0.87	0.87
12	0.87	0.87
13	0.87	0.87
14	0.87	0.87
15	0.87	0.87

Appendix 19: Expected Outcomes vs Actual Outcomes of Test 3

(SOM vs MOM vs LOM Test 3)

	SOM	MOM	LOM
Test Number	Trust Score Output	Trust Score Output	Trust Score Output
1	True	True	True
2	True	True	True
3	True	True	True
4	True	True	True
5	True	True	True
6	True	False	False
7	True	True	False
8	Flase	False	False
9	True	True	False
10	True	False	False
11	True	True	False
12	True	True	False
13	False	False	True
14	True	True	False
15	False	False	True

Expected Outcomes vs Actual Outcomes (Bisector vs Centroid Test 3)

	Bisector	Centroid
Test Number	Trust Score Output	Trust Score Output
1	True	True
2	True	True
3	True	True
4	True	True
5	True	True
6	Flase	False
7	True	True
8	Flase	Flase
9	True	True
10	False	False
11	True	True
12	True	True
13	False	False
14	True	True
15	False	False

Test 4:

Appendix 20: Changes for Test 4

Additon to rule base (Test 4)

```
%New rules for Test 4
rule42 = [3 3 3 4 1 1];
rule43 = [3 3 2 5 1 1];
rule44 = [3 2 3 4 1 1];
rule45 = [2 3 3 4 1 1];
rule46 = [3 2 2 4 1 1];
rule47 = [3 1 3 4 1 1];
rule48 = [3 3 3 5 1 1];

% The ruleList of all defined rules.
ruleList = [rule1; rule2; rule3; rule4; rule5; rule6; rule7; rule8;
    rule9; rule10; rule11; rule12; rule13; rule14; rule15; rule16;
    rule17; rule18; rule19; rule20; rule21; rule22; rule23; rule24;
    rule25; rule26; rule27; rule28; rule29; rule30; rule31; rule32;
    rule33; rule34; rule35; rule36; rule37; rule38; rule39; rule40;
    rule41; rule42; rule43; rule44; rule45; rule46; rule47; rule48;
];
```

Appendix 21: Expected Outcomes (Test 4):

Test Number	Expected Output Result
1 (Group 1)	Somewhat Trustworthy
2	Untrustworthy
3	Untrustworthy
4	Average Trust
5	Untrustworthy
6 (Group 2)	Average Trust
7	Highly Trustworthy
8	Highly Trust
9	Highly Trustworthy
10	Average Trust
11 (Group 3)	Highly Trustworthy
12	Highly Trustworthy
13	Very Trustworthy
14	Highly Trustworthy

15	Very Trustworthy

Appendix 22: Results of Test 4

MOM vs Bisector vs Centroid (Test 4):

	MOM	Bisector	Centroid
Test Number	Trust Score Output	Trust Score Output	Trust Score Output
1	0.38	0.49	0.52
2	0.00	0.25	0.26
3	0.00	0.17	0.20
4	0.59	0.65	0.65
5	0.00	0.32	0.37
6	0.77	0.81	0.80
7	0.80	0.87	0.87
8	0.77	0.87	0.87
9	0.79	0.85	0.83
10	0.78	0.86	0.85
11	0.80	0.87	0.87
12	0.80	0.87	0.87
13	0.80	0.87	0.87
14	0.80	0.87	0.87
15	0.80	0.87	0.87

Appendix 23: Expected Outcomes vs Actual Outcomes of Test 4

(MOM vs Bisector vs Centroid Test 4):

	MOM	Bisector	Centroid
Test Number	Trust Score Output	Trust Score Output	Trust Score Output
1	True	True	True
2	True	True	True
3	True	True	True
4	True	True	True
5	True	True	True
6	False	False	False
7	True	True	True
8	False	False	False
9	True	True	True
10	False	False	False
11	True	True	True
12	True	True	True
13	False	False	False
14	True	True	True
15	False	False	False

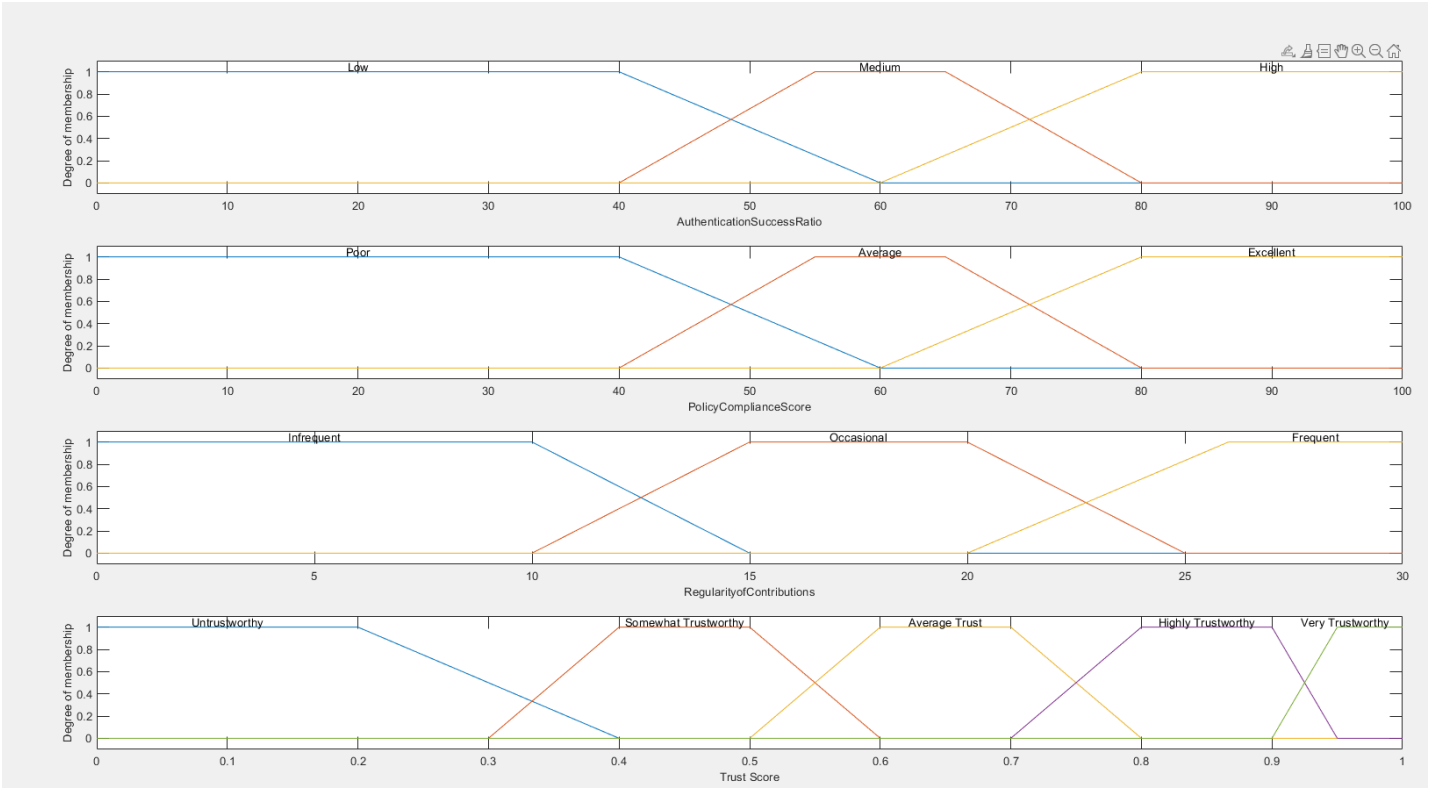
Appendix 24: Final System Configuration

```
% A declaration of new FIS
a = newfis('Trust Score', 'DefuzzificationMethod', 'Som');

% Declaring new variables - these are INPUTS
a=addvar(a,'input','AuthenticationSuccessRatio',[0 100]);
a=addvar(a,'input','PolicyComplianceScore',[0 100]);
a=addvar(a,'input','RegularityofContributions',[0 30]);

% Intervals the Authentication Success variable with membership functions
a=addmf(a,'input',1,'Low','trapmf',[0 0 40 60]);
a=addmf(a,'input',1,'Medium','trapmf',[40 55 65 80]);
a=addmf(a,'input',1,'High','trapmf',[60 80 100 100]);

% Intervals for the Policy Compliance input variable with membership functions
a=addmf(a,'input',2,'Poor','trapmf',[0 0 40 60]);
a=addmf(a,'input',2,'Average','trapmf',[40 55 65 80]);
a=addmf(a,'input',2,'Excellent','trapmf',[60 80 100 100]);

% Intervals the Regualrity of Contributions input variable with membership functions
a=addmf(a,'input',3,'Infrequent','trapmf',[0 0 10 15]);
a=addmf(a,'input',3,'Occasional','trapmf',[10 15 20 25]);
a=addmf(a,'input',3,'Frequent','trapmf',[20 26 30 30]);

% Declaring a new variable - this is the OUTPUT
a=addvar(a,'output','Trust Score',[0 1]);

% Populating the output variable with membership functions
a=addmf(a,'output',1,'Untrustworthy','trapmf',[0 0 0.2 0.4]);
a=addmf(a,'output',1,'Somewhat Trustworthy','trapmf',[0.3 0.4 0.5 0.6]);
a=addmf(a,'output',1,'Average Trust','trapmf',[0.5 0.6 0.7 0.8]);
a=addmf(a,'output',1,'Highly Trustworthy','trapmf',[0.7 0.8 0.9 0.95]);
a=addmf(a,'output',1,'Very Trustworthy','trapmf',[0.9 0.95 1 1]);
```

```
% Rule Base
rule1 = [2 2 2 1 1 1];
rule2 = [2 2 3 2 1 1];
rule3 = [2 3 2 2 1 1];
rule4 = [3 2 2 3 1 1];
rule5 = [3 3 2 4 1 1];
rule6 = [3 2 3 4 1 1];
rule7 = [3 3 3 5 1 1];
rule8 = [1 1 1 1 1 1];
rule9 = [1 1 2 1 1 1];
rule10 = [1 1 3 2 1 1];
rule11 = [1 2 1 2 1 1];
rule12 = [1 2 2 2 1 1];
rule13 = [1 2 3 3 1 1];
rule14 = [1 3 1 2 1 1];
rule15 = [1 3 2 3 1 1];
rule16 = [1 3 3 4 1 1];
rule17 = [2 1 1 2 1 1];
rule18 = [2 1 2 2 1 1];
rule19 = [2 1 3 3 1 1];
rule20 = [2 2 1 3 1 1];
```

```
rule21 = [2 3 1 3 1 1];
rule22 = [2 3 2 4 1 1];
rule23 = [2 3 3 4 1 1];
rule24 = [3 1 1 3 1 1];
rule25 = [3 1 2 3 1 1];
rule26 = [3 1 3 4 1 1];
rule27 = [3 2 1 4 1 1];
rule28 = [3 3 2 5 1 1];
rule29 = [3 2 3 5 1 1];
rule30 = [3 3 1 4 1 1];
rule31 = [2 3 3 4 1 1];
rule32 = [3 3 3 5 1 1];
rule33 = [2 2 3 4 1 1];
rule34 = [3 2 2 4 1 1];
rule35 = [1 1 2 2 1 1];
rule36 = [1 2 2 1 1 1];
rule37 = [2 2 1 2 1 1];
rule38 = [2 3 2 3 1 1];
rule39 = [3 2 2 3 1 1];
rule40 = [3 3 1 4 1 1];
rule41 = [3 3 1 5 1 1];
```

```
% The ruleList of all defined rules.
ruleList = [rule1; rule2; rule3; rule4; rule5; rule6; rule7; rule8;
    rule9; rule10; rule11; rule12; rule13; rule14; rule15; rule16;
    rule17; rule18; rule19; rule20; rule21; rule22; rule23; rule24;
    rule25; rule26; rule27; rule28; rule29; rule30; rule31; rule32;
    rule33; rule34; rule35; rule36; rule37; rule38; rule39; rule40;
    rule41;
];

a = addrule(a, ruleList);

AuthenticationSuccessRatioInput = 45;
PolicyComplianceScoreInput = 30;
RegularityofContributionsInput = 30;
format long;
evalA = evalfis([AuthenticationSuccessRatioInput, PolicyComplianceScoreInput, RegularityofContributionsInput], a);
disp(['Trust Score', num2str(evalA)]);

% Visualization (you would typically run this in MATLAB to see the plots)
subplot(4,1,1),plotmf(a, 'input', 1);
subplot(4,1,2),plotmf(a, 'input', 2);
subplot(4,1,3),plotmf(a, 'input', 3);
subplot(4,1,4),plotmf(a, 'output', 1);
```


Low
Medium
High
Degree of membership
AuthenticationSuccessRatio
Poor
Average
Excellent
PolicyComplianceScore
Infrequent
Occasional
Frequent
RegularityofContributions
Untrustworthy
Somewhat Trustworthy
Average Trust
Highly Trustworthy
Very Trustworthy
Trust Score