iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** VI **Month of publication:** June 2023

DOI: https://doi.org/10.22214/ijraset.2023.50533

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VI Jun 2023- Available at www.ijraset.com

Communication Network with Secure, Reputable Mobile Agent Structure

Neera Yadav[1], Dharamveer Singh[2], Anil Kumar[3]
[1]*R D Engineering College Ghaziabad, U.P. India – 201206*
[2, 3]*Research Centre, Mata Rama Devi Trust, Modinagar, Ghaziabad, U.P. India – 201201*

Abstract: ***The proposed multiphase security model for mobile agents appears to address a critical issue in their mainstream deployment. The use of symmetric key cryptography is a sensible choice for securing mobile agent code and data, as it is more efficient than asymmetric key cryptography. The focus on defending against a meet-in-the-middle attack is also appropriate, as this is a common type of attack that mobile agents are vulnerable to. However, it would be helpful to provide more details about the specific authenticated encryption mechanism that will be used. This will help to evaluate the effectiveness of the proposed model and ensure that it is robust enough to defend against various types of attacks. It is also important to consider the energy consumption of the proposed security model, as mobile devices typically have limited battery life. Therefore, the model should be designed in a way that minimizes energy consumption while maintaining the required level of security. Overall, the proposed multiphase security model appears to be a step in the right direction for addressing the security issues associated with mobile agents. However, it is important to ensure that it is properly implemented and thoroughly tested to ensure its effectiveness and efficiency.***

I. INTRODUCTION TO MOBILE AGENTS

A. Theoretical Description

1) Mobile agent

It is a group of data and software that can independently go from one system to another while still operating on the final platform, according to computer science. A mobile agent in a distributed network As shown in Fig. 1.1, GiovenniVigna[1] Regarding the statement about mobile agents having advantages over conventional distributed approaches, this is generally true. Mobile agents can carry out tasks on behalf of a user or application, moving autonomously from one network node to another to perform their task. This can reduce network traffic and improve resource utilization since the agent can move closer to the data it needs to access or process, rather than transferring the data over the network to a centralized server. However, as mentioned earlier, security risks associated with mobile agents need to be addressed to ensure their safe deployment.

According to Greenberg et al. [2], a mobile agent is a software agent that is self-contained, social, learning, and—most importantly—moving. A process that can relocate its state from one environment to another while keeping its data and carrying on as usual in the new environment is referred to as a mobile agent. When and when this business is conducted will be decided by the agents. Movement is typically created using RPC approaches.

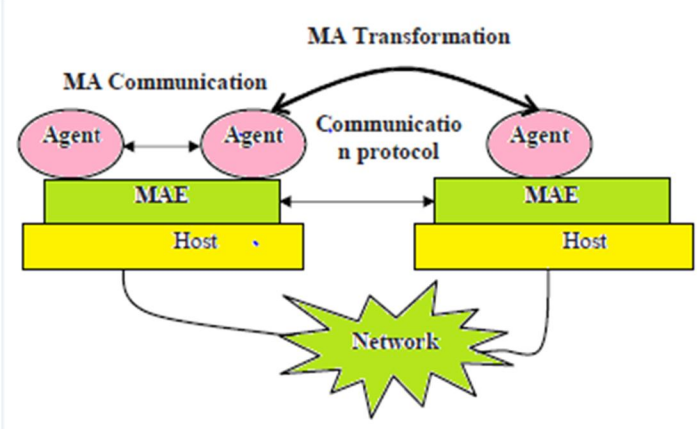


Fig1.1 Mobile Agent Framework

II. LITERATURE SURVEY

The security risks raised by mobile agents are well-known in the security community, and as a result, a lot of study is being done in this area. Many initiatives have been launched to address the threats that mobile agents confront, however the bulk of them only address a piece of the problem.

Collaborating agents as proposed by V. Roth and J.Peters[3], This approach can provide several benefits, such as fault tolerance, scalability, and adaptability to changing environments. In a collaborative agent system, agents can be dynamically added or removed as needed, and the system can continue to operate effectively even if some agents fail or are unavailable. Borselius[4], In the case of the shopping agent, deploying a large number of mobile agents, each travelling a separate route, can help to improve the chances of finding the best offer while also improving security. By having multiple agents travelling different routes and communicating their votes among themselves, the system can better resist attacks that attempt to manipulate the results by compromising a single agent.

According to N. Borselius[5], the security difficulties faced by non-mobile agents can be handled to a significant extent utilizing existing security technologies and protocols, at least in theory. In a large-scale multi-agent system, trust and delegation difficulties are challenging to handle. Agents must be able to reason and make decisions based on a number of security criteria, even though a public key infrastructure will almost definitely be a critical component of the solution.

Lee et. al[6]An intriguing method of mobile code protection is provided by Lee et al.'s technology, which enables an agent platform to run a programme that encapsulates an encrypted function without being able to decrypt the original function. This may aid in preventing hackers from gaining access to the original code and perhaps exploiting flaws or stealing confidential data."

Using cooperative agents, as proposed by Volker Roth [7], Overall, using cooperative agents can be an effective approach to improving the security and efficiency of mobile agent systems. However, it is important to ensure that the agents are properly designed, implemented, and secured to prevent attacks and ensure the safety and privacy of the system's users, for example, can achieve a similar scenario to that described with trustworthy nodes. One such strategy for ensuring that a mobile agent arrives safely at its destination is proposed by [8]. In the case of the shopping agent, a large number of mobile agents can be deployed, each travelling a separate route, and the agents communicate their votes among themselves before settling on the best offer.

Leeet. al [9], proposal of mixed-multiplicative homomorphism is an interesting approach to encrypting data without using keys or encryption techniques. Homomorphic encryption allows computations to be performed on encrypted data without needing to decrypt it, which can be useful in scenarios where sensitive data needs to be processed securely. According to M. Alfalayleh and L. Brankovic[10], The authors compare the security of mobile agents with agent platforms and conclude that securing mobile agents is more difficult due to their autonomous and mobile nature. Mobile agents need to be protected from malicious hosts and other potential security threats as they move between different hosts and networks. After three years, Fritz Hohl[11]Black box security works by encapsulating the mobile agent in a black box, which allows the agent to execute its code and perform its tasks without exposing its internal state or data to the host. The host is only able to interact with the agent through a limited set of interfaces, which are designed to prevent unauthorized access or modification of the agent's data. The cryptography history is presented by another significant legend, B. Scheiner[12]. Of course, confidentiality has always been necessary, but before the First World War, key advances were published in a more or less timely manner, and the topic proceeded in a similar way to other specialized areas. The first issue is the safety of the agent platform against malicious agents. Various generally accepted approaches, including access control, password security, and sandboxes, have already been presented to solve the first problem. Environmental key generation Ahila and Shunmugunathan[13], encrypted functions, obfuscation Uddin et. al [14], and tracing of path execution are the promptest solutions for malicious host problems so far. However, these systems have a number of flaws that necessitate a comprehensive preventive solution that balances security and overhead.In our current system, security against hostile hosts is a worry, and we've uncovered existing security flaws that must be corrected before this technology can be broadly deployed.

A. Aim Of The Research

1) The aim of the research is to develop a protection policyformobile agent and it'sitinerary from malicious host framework by authentication encryption to perform, integrity, confidentiality and authentication of each message without increasing the additional energy consumption.

2) To develop a secure mobile agent-based system, where computational code Jolly and Batra[15] is moved to a remote platform's data location, where it performs operations on resource data and returns the results to the agent owner.

3) To provide Security in mobile agent code.

4) To protect the data from malicious execution environment.

5) To facilitate integrity, authentication, confidentiality and malicious host problem.
6) Preventative steps are used to ensure confidentiality and authenticity, while detecting techniques are used to assure integrity.

B. Proposed Security System For Malicious Host

By using a malicious host method, we proposed a secure way to identify any prospective security vulnerabilities Wei et. al [16]. To ensure that mobile agent communication is secure, a partial result authentication code and sliding encryption are utilized. To participate in an agent-based transaction Mittal and Mishra [17], each host platform must first register with a trusted key server and get an identity certificate. This platform should have a public-private key pair and a random number K to encrypt the mobile agent. The 128-bit key is expanded into both the eight state variables and the eight counters in the AESsymmetric key methodChauhan and Purohit[18], resulting in a one-to-one link between the key and the initial state variables and counters.

III. SOFTWARE REQUIREMENT SPECIFICATION

A. Description

Mobile agents are object-oriented software program with adaptability, communication ability, responsiveness, autonomy, responsiveness, and intelligence that make them more useful than other network infrastructure mechanisms such as client servers. In a mobile agent, the following characteristics of flexibility and autonomy are evident.

B. Specific Requirements

The various specific requirements of the project work are as stated below:

1) Hardware Requirements
a) Intel Core i3 Processor, CD-ROM, 4GB RAM, 60 GB HDD,
b) Internet connection from client to server machine
c) TCP/IP network for communication between clients and server

2) Software Requirements
a) Operating System: Windows 10
b) Programming Tool: Java JDK 1.6
c) IDE: Net Beans

IV. DESIGNING AND FORMULATION OF THE PROBLEM

A. Architectural Strategy

Code security (tampering attack, code privacy, and code integrity) and data security are two types of mobile agent security (data integrity, data privacy, etc.). There are two techniques to migrate mobile agents.

1) Predefined Itinerary
2) Free Roaming

This form of itinerary, as seen in Fig.4.1, is referred to as a static itinerary, and it puts the mobile agent's adaptability to the test.

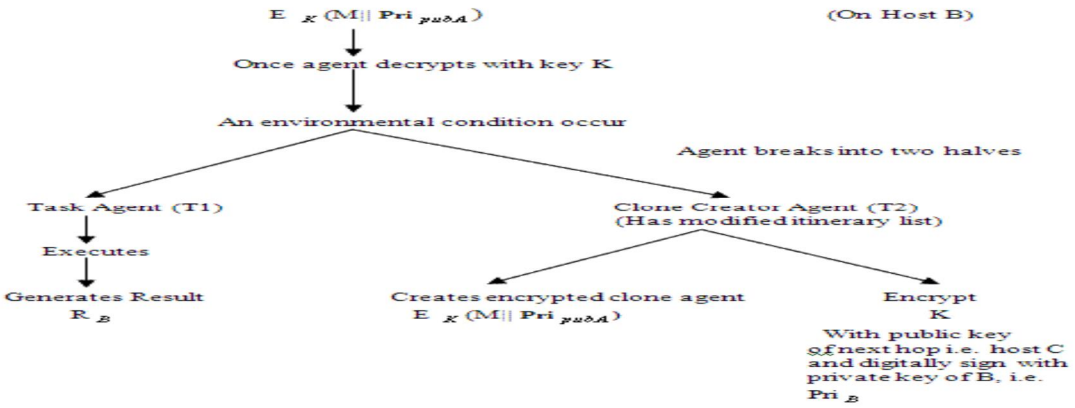


Fig. 4.1Mobile Agent's Itinerary Security

B. Process Flow Diagram For Node Registration

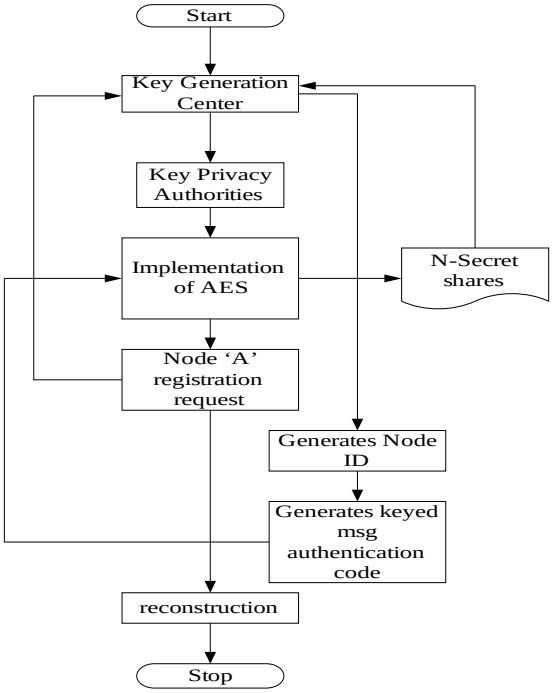


Fig. 4.7 Process Flow Diagram for Node Registration

C. Process Flow Diagram For Secure Key Issuing

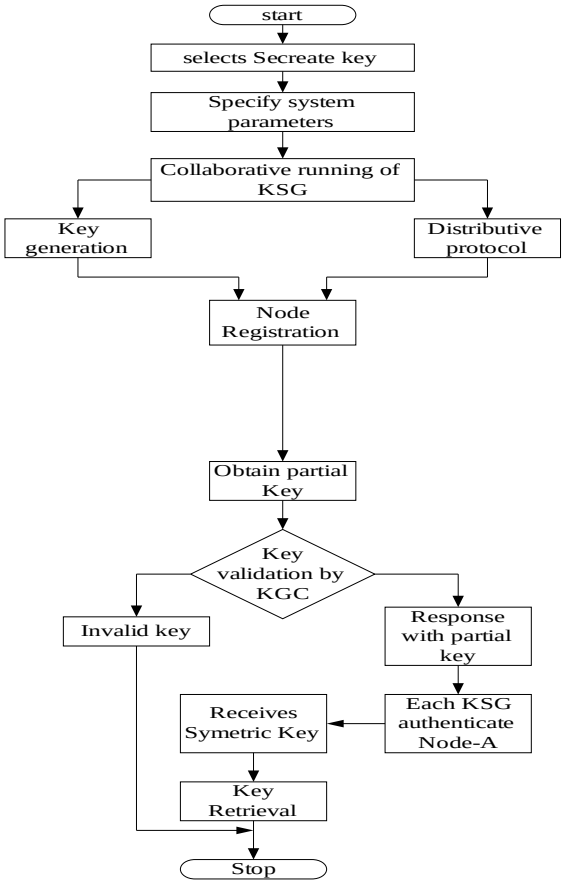


Fig.4.8 Process Flow Diagram for Security key Issuing

D. Sequence Diagram

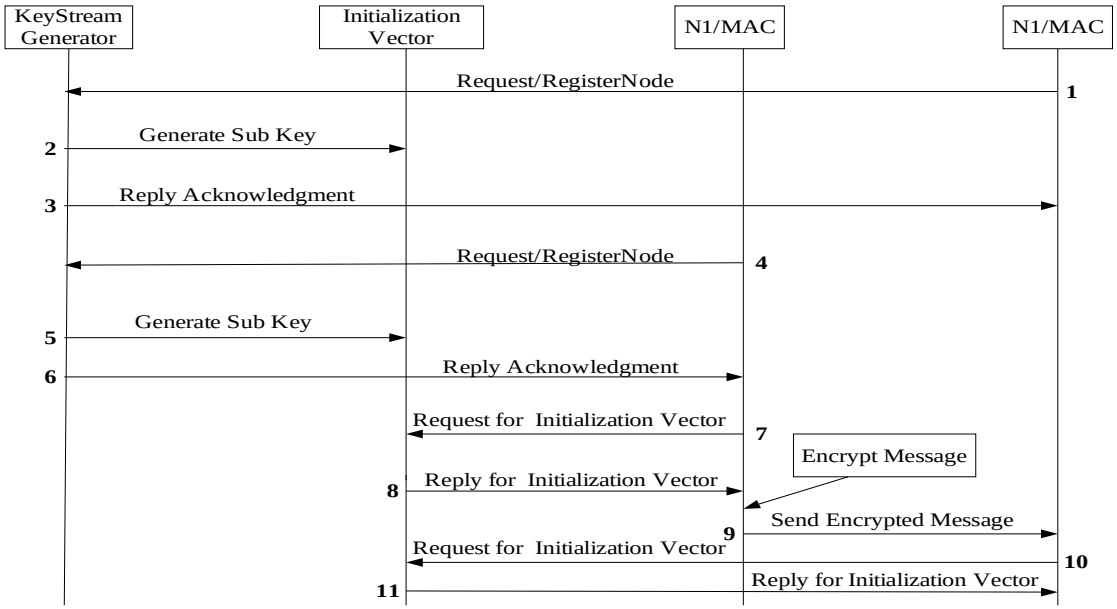


Fig. 4.9 Sequence Diagram

V. IMPLEMENTATION

A. Algorithm Used

Name: Algorithm for Node Registration
Input: Node Information configuration
Output: Successfully Node Registration
Step-1: Initialize Key Stream generator
Step-2: Initializing Vector of 256-bit Sub Keys
Step-3: Node “A” sends request to KSG
Step-4: KSG received the request, generates Node “A” identity (IDA)
Step-5: KSG generates Node “A” proof of registration (ProofA)
Step-6: Implement Rabbit-MAC scheme
Step-7: Distribute Secret shares to 32-bit n- Sub Keys
Step-8: WSN receives secret shares from KSG and sends them to Node “A”
Step-9: IDA and ProofAis reconstructed.

B. Approach Used For Implementation

1) Design Key Stream Generator
a) Status showing for keys generated
b) Status showing the key generation process
2) Design collaborative Initialization Vectors
3) Design WSN Node
a) Input for port number
b) Auto detection of Node IP address
c) Registration facility with KSG
d) Get keys from collaborative IV
e) Viewing of received message
f) Messaging privileged
- Destination IP settings

4) Design for Node registration

5) Design for securing key issuing

Encryption schemes are classified into two types. They are

a) Symmetric-key cryptography

b) Public key cryptography

C. Source Code Description

VI. RESULTS AND DISCUSSIONS

A. Overview

Testing is a crucial part of software development, as it helps to identify and fix errors or bugs in the system, ensuring that the software is of high quality and reliable. Various testing techniques can be used, such as functional testing, performance testing, security testing, and user acceptance testing. It is important to document the testing results, as they can be used as a reference for future updates or improvements to the system.

B. Test Plans

In this test plan all major activities are described below.

1) Unit Testing

Testing is a testing technique that focuses on individual modules or units of code to ensure that each unit works as expected and is error-free. The purpose of unit testing is to identify and fix bugs at an early stage of the development process. By testing each module separately, it's easier to isolate and fix any issues that arise, rather than trying to debug the entire program as a whole. Additionally, unit testing helps to ensure that changes made to a module don't affect the functionality of other modules.

Table 6.1 Unit Testing 1

Test Name: -	Testing of “WSN_Node.java
Testing Item: -	“MOBILE_Node.java”
Sample Input: -	Enter all the swing component accurately along with connectivity to other module
Output: -	Execution of program should yield operation of MOBILE_Node in the application of MOBILE.
Actual output: -	Same as expected output
Remarks: -	Testing is successful

Table 6.2 Unit Testing 2

Name of Test: -	Testing of “E_Key_1.java”
Testing Item: -	“E_Key_1.java”
Sample Input: -	Create a class and use thread and configure the swing component correctly.
Expected output: -	Execution of this program should yield the activation of initialization of vector
Expected output: -	Same as expected output
Remarks: -	Testing is successful

Table 6.3 Unit Testing 3

Name of Test: -	Testing of "I_Vectors.java"
Testing Item: -	"I_Vectors.java"
Sample Input: -	Create a class and method using key evoked due to performed action by node
Output: -	Generation of keys in Key stream generation center
Actual output: -	Same as expected output
Remarks: -	Testing is successful

Table 6.4 Unit Testing 4

Test Name: -	Testing of "receiver.java"
Testing Item: -	"receiver.java"
Sample Input: -	Initialize the port number to Zero and design a thread for this.
Output: -	The peer client should be able to receive message
Actual output: -	Same as expected output
Remarks: -	Testing is successful

2) Integration Testing

Data can be lost across an interface: one module's subsidiary functions may have a negative impact on another's, causing them to fail to deliver the expected principal function when combined; global data structures may cause problems. It is a technique for constructing the programme structure and testing for interface concerns at the same time. All components are merged in this step of testing. After then, the programme was put to the test.

Table 6.5 Integration Testing 1

Test Name: -	Integration of "MOBILE Node" and "Key stream generator"
Testing Item: -	Appropriateness in link creation between WSN Node and KSG
Sample Input: -	Provoke the MOBILE Node for requesting to join KSG
Expected output: -	The application instantly provokes the MOBILE Node to enter specific values
Output: -	Same as expected output.
Remarks: -	Testing is successful

Table 6.6 Integration Testing 2

Test Name: -	MAC and the KSG integration
Testing Item: -	Message Authentication Code
Sample Input: -	Start Node request till it comes in KSG
Expected output: -	KSG generate Node ID and keyed message which are combined to generate secret key
Output: -	Same as expected output
Remarks: -	Pass

3) System Testing

During system testing, the system as a whole was tested to ensure that it met the functional and non-functional requirements specified in the requirements specification. The testing included both positive and negative test cases to verify the correct functioning of the system under different conditions. The system was tested for performance, security, usability, and compatibility to ensure that it meets the expectations of the end-users. The results of the testing were analyzed, and any defects found were reported to the development team for rectification. After the defects were fixed, the testing was repeated to ensure that the system was functioning as expected. Finally, the system was validated by the customer to ensure that it met their requirements and expectations.

Table 6.7 System Testing 1

Test Name: -	Performance evaluation in terms of OS
• Testing Item: -	• Compatibility with different OS
Sample Input: -	Program execution in different versions of OS.
Expected output: -	Windows 10 as latest OS should deliver better performance
Actual output: -	Same as expected output
Remarks: -	Testing is successful

Table 6.8 System Testing 2

Test Name: -	Evaluation of performance in terms of time
Testing Item: -	Average time taken
Sample Input: -	Program to be executed in 10 passes
Expected output: -	Execution should be performed in 1-2 minutes
Actual output: -	Same as expected output.
Remarks: -	Testing is successful

Table 6.9 System Testing 3

Test Name: -	Processor type
Testing Item: -	Processor compatibility
Input: -	Execute the program in different versions of processor
Expected Output: -	Best performance in core i3
Actual output: -	Same as expected output.
Remarks: -	Testing is successful

Table 6.10 System Testing4

Test Name: -	IDE Version System Testing
Testing Item: -	Compatibility of NetBeans IDE version with programs
Input: -	Execution of programs in different versions of NetBeans
Expected Output: -	Nil issues in execution
Actual output: -	Same as expected output.
Remarks: -	Testing is successful

C. Snapshots Of Mobile Agent Security Development Model

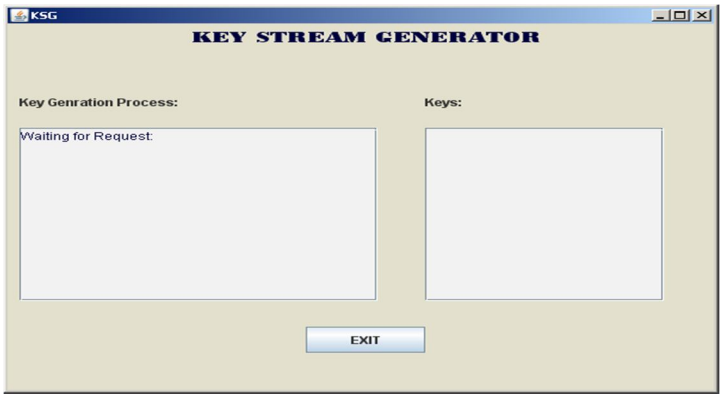


Fig. 6.1 Start the KSG Key Stream Generator GUI should be created

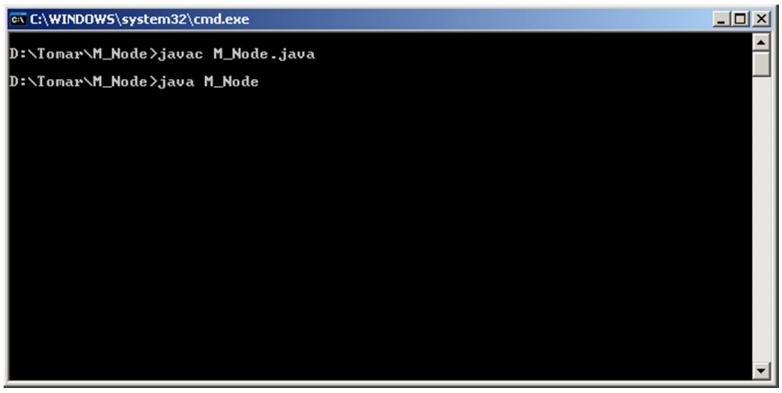


Fig 6.2Start the M_Node batch File

Fig. 6.4 Entering Port No

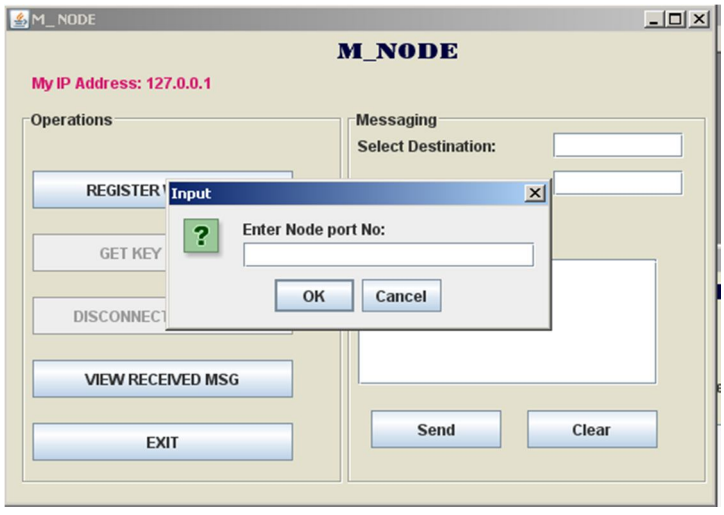


Fig. 6.3 Enter the Node Port no.

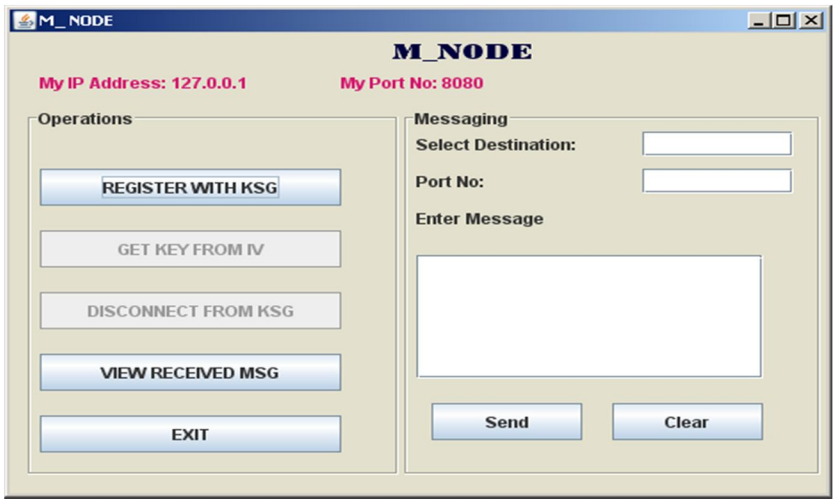


Fig. 6.5 Registering with KSG

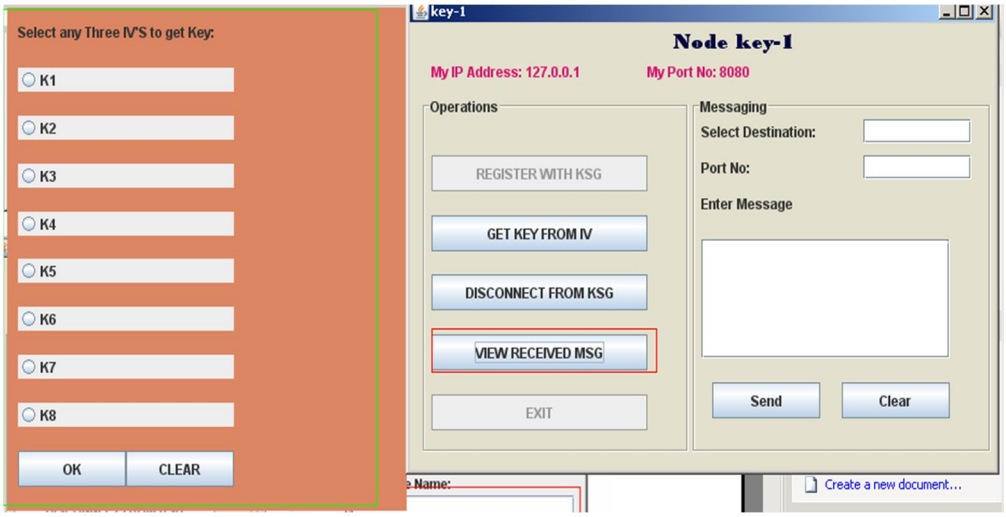


Fig. 6.6 Registering and Key Issuing Process

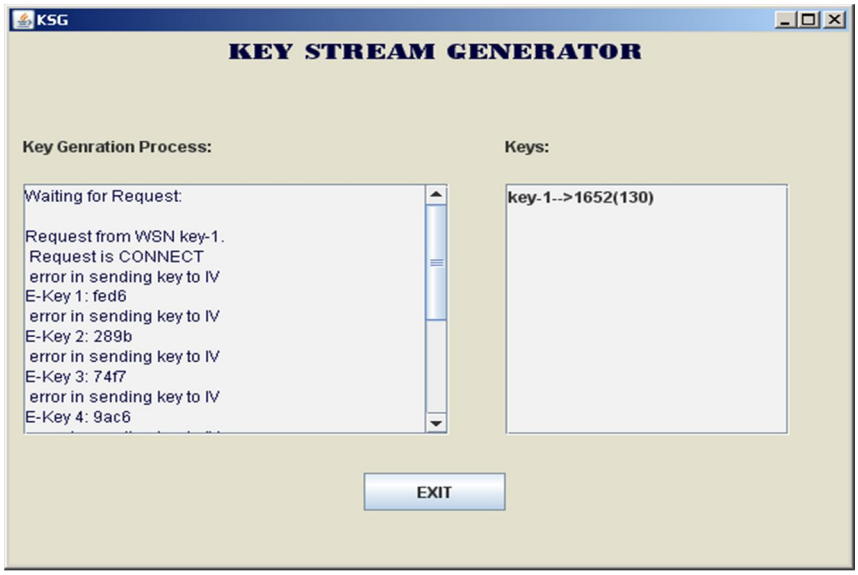


Fig. 6.7 Activities in KSG

VII. CONCLUSION AND FUTURE SCOPE

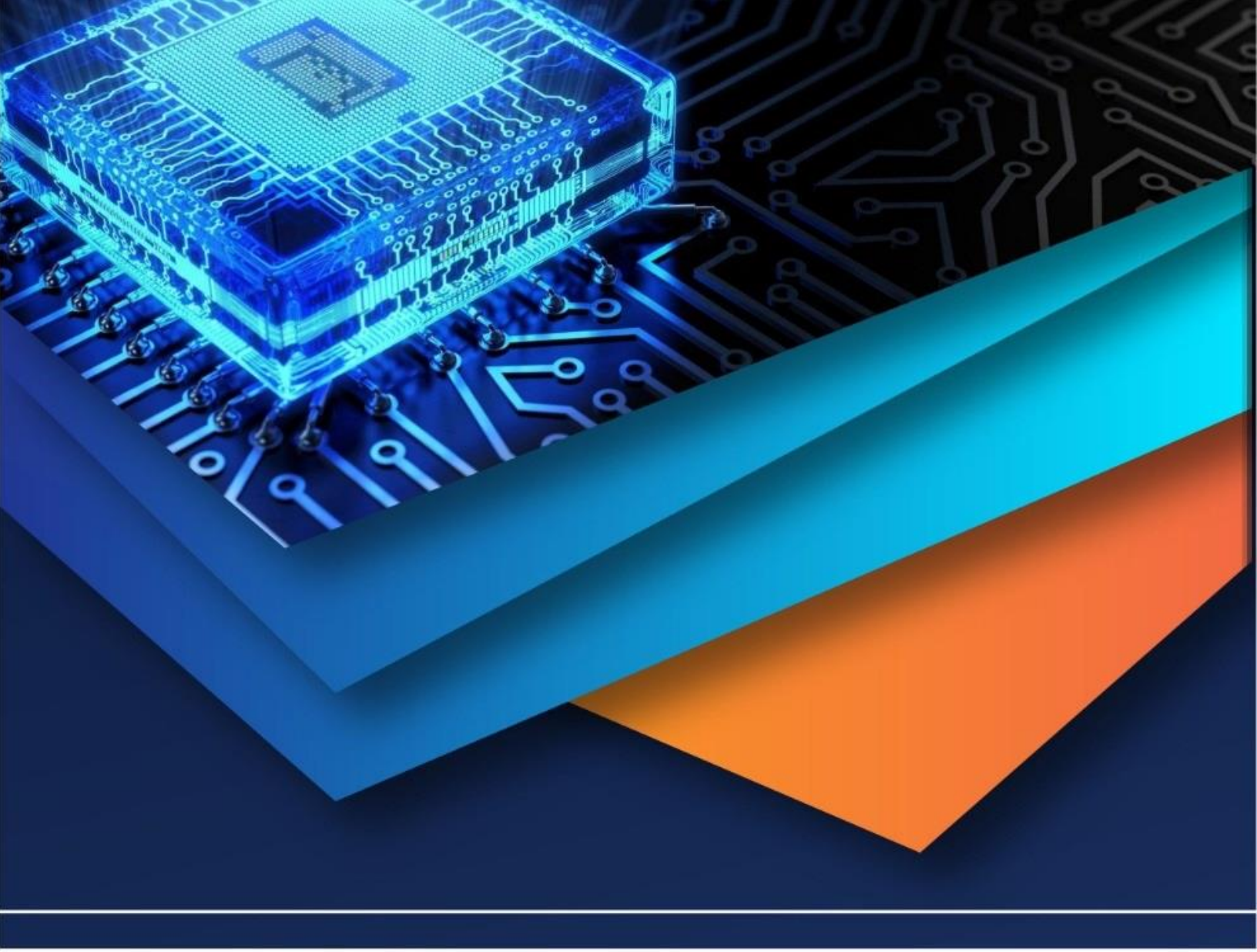
The most important security provided by the framework is that take into account the limitations of such devices. Additionally, the use of Machine Learning and Artificial Intelligence in mobile agent security can be explored further to provide more robust and adaptive security measures. Moreover, the integration of blockchain technology can also be considered to enhance the security of mobile agent systems. Overall, there is a lot of potential for further research and development in this area, and it is important to continue exploring new approaches to ensure the security of mobile agent systems in the face of evolving security threats.

List of Abbreviations

1) RPC: Remote Procedural Call
2) MAS: Mobile Agent System
3) TCP/IP: Transmission Control Protocol/ Internet Protocol
4) MAE: Mobile Agent Environment
5) SRS: Software Requirement Specification
6) WSN: Wireless Sensor Network
7) KSG: Key Stream Generator

REFERENCES

[1] Vigna G., "Protecting mobile agents through tracing," Proc. 3rd ECOOP Work. Mob. …, 1997, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.41.2081&rep=rep1&type=pdf.
[2] Iovane G., Bisogni C., De Maio L, and M. Nappi, "An Encryption Approach Using Information Fusion Techniques Involving Prime Numbers and Face Biometrics," IEEE Trans. Sustain. Comput., vol. 5, no. 2, pp. 260–267, 2020, doi: 10.1109/TSUSC.2018.2793466.
[3] Qadori H. Q, ZukarnainZ. A., HanapiZ. M., and SubramaniamS., "FuMAM: Fuzzy-Based Mobile Agent Migration Approach for Data Gathering in Wireless Sensor Networks," IEEE Access, vol. 6, no. c, pp. 15643–15652, 2018, doi: 10.1109/ACCESS.2018.2814064.
[4] Young A. and YungM., "Sliding encryption: A cryptographic tool for mobile agents," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 1267, pp. 230–241, 1997, doi: 10.1007/bfb0052350.
[5] RathM. and PattanayakB. K., "Security protocol with IDS framework using mobile agent in robotic MANET," Int. J. Inf. Secur. Priv., vol. 13, no. 1, pp. 46–58, 2019, doi: 10.4018/IJISP.2019010104.
[6] SrivastavaS. and NandiG. C., "Protection of mobile agent and its itinerary from malicious host," 2011 2nd Int. Conf. Comput. Commun. Technol. ICCCT-2011, pp. 405–411, 2011, doi: 10.1109/ICCCT.2011.6075189.
[7] Roth V. and PetersJ., "A scalable and secure global tracking service for mobile agents," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 2240, pp. 169–181, 2001, doi: 10.1007/3-540-45647-3_12.
[8] BorseliusN., "Mobile agent security," Electron. Commun. Eng. J., vol. 14, no. 5, pp. 211–218, 2002, doi: 10.1049/ecej:20020504.
[9] HyungjickL., Alves-FossJ., and HarrisonS., "The use of encrypted functions for mobile agent security," Proc. Hawaii Int. Conf. Syst. Sci., vol. 37, no. C, pp. 4757–4766, 2004, doi: 10.1109/hicss.2004.1265700.
[10] AlfalaylehM. and BrankovicL., "An overview of security issues and techniques in mobile agents," IFIP Adv. Inf. Commun. Technol., vol. 175, pp. 59–78, 2005, doi: 10.1007/0-387-24486-7_5.
[11] HohlF., "Time Limited Blackbox Security: Protecting Mobile Agents From Malicious Hosts," pp. 92–113, 1998, doi: 10.1007/3-540-68671-1_6.
[12] SarierN. D., "Multimodal biometric Identity Based Encryption," Futur. Gener. Comput. Syst., vol. 80, pp. 112–125, 2018, doi: 10.1016/j.future.2017.09.078.
[13] SrivastavaS. and NandiG. C., "Self-reliant mobile code: A new direction of agent security," J. Netw. Comput. Appl., vol. 37, no. 1, pp. 62–75, 2014, doi: 10.1016/j.jnca.2013.01.004.
[14] AhilaS. S., "Overview of Mobile Agent Security," no. 978, 2014.
[15] UddinM., MemonJ., AlsaqourR., ShahA., and RozanM. Z. A., "Mobile Agent based Multi-layer Security Framework for Cloud Data Centers," Indian J. Sci. Technol., vol. 8, no. 12, 2015, doi: 10.17485/ijst/2015/v8i12/52923.
[16] JollyP. K. and BatraS., "Security against Attacks and Malicious Code Execution in Mobile Agent Using IBF-CPABE Protocol," Wirel. Pers. Commun., vol. 107, no. 2, pp. 1155–1169, 2019, doi: 10.1007/s11277-019-06329-7.
[17] WeiC.-H., HwangM.-S., and ChinA. Y., "Security Analysis of an Enhanced Mobile Agent Device for RFID Privacy Protection," IETE Tech. Rev., vol. 32, no. 3, pp. 183–187, 2015, doi: 10.1080/02564602.2014.983190.
[18] Mittal P.and MishraM. K., Towards Extensible and Adaptable Methods in Computing. Springer Singapore, 2018.
[19] SinghR. and PurohitR., "Study of Mobile Agent Server Architectures for Homogeneous and Heterogeneous Distributed Systems," Int. J. Comput. Appl., vol. 156, no. 4, pp. 32–37, 2016, doi: 10.5120/ijca2016912420.
[20] Dharamveer, Samsher. Comparative analyses energy matrices and enviro-economics for active and passive solar still", materialstoday: proceedings, 2020; 45:6046-6052, https://doi.org/10.1016/j.matpr.2020.10.001.
[21] Dharamveer, Samsher, Kumar A.Performance analysis of N identical PVT-CPC collectors with an active single slope solar distiller and helically coiled heat exchanger using CuO nanoparticles, 2021, https://doi.org/10.2166/ws.2021.348.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY