Robocipher: AI Powered Communication Framework For Autonomous Robotic Systems

[1]Dr. A Manjula, [2]Vemula Varsha,[3] Mohd.Ayesha, [4]Vemula Shreeya, [5]Cheeti Sai Sreekar

[1]Associate Professor, Dept. of CSE, Jyothishmathi Institute of Technology and Science, Karimnagar, Telangana.

[2,3,4,5] BTech Student,CSE, Jyothishmathi Institute of Technology and Science, Karimnagar, Telangana.

Abstract - The advancement of Autonomous robotic systems (ARS) has resulted in the increased adoption in many critical sectors such as industrial automation, logistics, healthcare, and defense. However, collaboration and communication between these robots occur in real-time and the capture, exchange, and bidirectional communication of data must involve secure protocols as well as reliable links since each autonomous robot will rely on using its unique data collaboratively with others in a workplace environment. For autonomous robotic systems, the study has proposed a multi phased, AI-based secure communications framework. The communications framework has access control through Zero Trust Authentication, as well as supporting resilient communications by not only encompassing advanced hardware security alongside secure software cryptographic protocols but also through the use of AI various detection monitoring systems looking for anomalies and addressing cyberthreat attempts, encryption methods like blockchain technology along with implementing Advanced Encryption Standards (AES) for end- to- end communication confidentiality, Homomorphic Encryption allowing for data confidentiality during machine learning on device without breach of ownership and issue of rapid increase of sensitive data proliferation, as well as, Neural Cryptography for methods of securely sharing and exchanging keys.

Index Terms - Autonomous robotic systems, secure communication, neural cryptography, AES encryption, homomorphic encryption, blockchain logging, AI-based anomaly detection, zero trust authentication, secure multi-party computation (SMPC), data integrity, cybersecurity, tamper-proof logging, key exchange, resilient robotic networks.

I. INTRODUCTION

The implementation of Autonomous robotic systems(ARS) is altering the landscape of modern business sectors such as manufacturing, health care, defense force chain, logistics and others by enabling the robots to communicate . These systems rely on either wireless communication, artificial intelligence, or distributed decision making to provide sensors, geo-location, and control action in a central command or to robots distributed across some area. Given that operational infrastructures rely on communication in real-time, ARS also faces threats from the rapidly increasing cyber security threats to the operational infrastructures through data interception, spoofing, unauthorized access, and command injection attack, which add to the complexity and vulnerabilities on operations.

Many of the current or traditional security control ways, using physical walls, static encryption algorithms, basic key exchanges, and perimeter-based firewalls do not work as expected at the dynamic, mobility, and multi-type level where autonomous robots operate. Businesses around the operational environment include the ineffectiveness of operational security ways in distribution where robots are now constantly joining or leaving networks, the risks involved in symmetric key exchange, and the inability of the static authentication method to verify identity or align trustworthiness in cyberspace over time with respect to transient state changes.While past work has offered methods such as AES encryption for confidentiality, PKI for identity verification, blockchain for immutable logging, and anomaly detection with AI for threat monitoring, most of these solutions are standalone, lack integration, and are often too resource-intensive for real-time robotic systems. This project offers a complete, scalable and lightweight secure communication framework for ARS which combines several advanced technologies to protect confidentiality, integrity, authentication, non-repudiation, and continuous trust validation. The proposed system uses Neural Cryptography (Tree Parity Machines) to exchange keys securely with no explicit transmission with a guarantee of no interception, and combines AES encryption with Homomorphic Encryption for both secure and data privacy-preserving communication among robots. All robot interactions are recorded using a private blockchain network with smart contracts. All records of robot interaction are logged in an immutable fashion, providing visibility, accountability and tamper-proof auditing. An anomaly detection module driven by AI employs an autoencoder and machine learning classifiers to predict divergence in the communication model and send early warning to operators before potential threats. Zero Trust Authentication preserves identity verification and access controls over every interaction, and Secure Multi-Party Computation (SMPC) allows robots to work together on sensitive tasks without the use of private data. By unifying all of these technological advances in one overarching framework, the investigation is able to provide real time, strong and resilient security tailored to SUIT worst cases, and make the evolution as inexpensive_BACKGROUND and seamless to the operator as possible.This investigation does more than address the issue of siloed approaches: it presents a future-ready solution which creates data integrity, enables secure collaboration, and securely contains the autonomous robotic operations from changing cyber threats. The result are safe, scalable, and trusted robotic ecosystems that span across industries.

II. Literature survey

Evaluating communication security in autonomous robotic systems has warranted much consideration from researchers because of the importance of robotic interactions in unpredictable environments. Several frameworks and approaches have been proposed to improve communication security; however, these approaches commonly address scalability, real-time processing, or full spectrum threat mitigation inadequately.

Chauhan et al., 2019 [1], introduced a security framework for collaborative autonomous systems with a focus on architectural patterns and threat modeling. While the security framework provides security guidelines for modular architectures, there is neither a specific implementation involving encryption or anomaly detection and threat mitigation in the framework.

In Sikand et al., 2022 [2], the authors created Robofleet, which is a hybrid work and communications system for fleets of autonomous mobile robots. Using the open-source software ROS (Robot Operating System) that enables hybrid work and communications by several autonomous mobile robots. The efforts were focused on the communications aspect among the autonomous robots, and as indicated, ensuring reliable communications also implied safety, thus there was little in terms of explicit security measures we could consider, and the concerns for cyber-security measures were not addressed.

In Biswas et al., 2022 [3], when considering scalability and developing improved communications toward collaborative robots, the authors work resulted in RoboCast, a peer-to-peer publish-subscribe based middleware, but again not little consideration was given to security provisions in terms of end-to-end encryption security in aspects of securing-by-design in relation authentication and authorization considerations.

In Kouicem et al., 2020 [4], when the authors were researching security in relation to the internet of things (IoT) and smart robotics, and they outlined a number of constraints and limitations concerning the downloading and controlling access to the data and coverage some of the limitations were further data confidentiality and access control. The authors also noted a considerable need for lightweight cryptographic solutions for robotic applications and underscored the need for information security considered that the computational environment is extremely constrained with consideration to the robot.

Aragon et al. (2018) [5] proposed a blockchain-based framework that uses distributed ledger systems to enable secure collaborative multi-robot teamed operations. However, the significant emphasis on modular security design prevented practical implementation into the collaborative robots.

In terms of Zero Trust security, Manzoor et al. [6] proposed a continuous authentication mechanism using both behavioral biometrics and device fingerprinting in an edge networks context. While this mechanism produced effective access control policies, it was not well generalized for resource constrained robotic systems. Zero Trust access and security is becoming a viable option for cloud environments, yet there are few real-world implementations in robotics that include elements of continuous re-authentication and session monitoring in a peer-to-peer topology.

Kumar and Sharma [7] proposed a token-based zero trust access mechanism for industrial IoT devices. While this mechanism was effective in the context of cloud-to-edge communication, it relies on a centralized method of token generation and key placement, thus allowing human elements into systems designed to be autonomous.

Kumar et al. [8] have also shown the potential of quantum cryptography in autonomous vehicle networks that can be a practically unbreakable method of key exchange; however, applying quantum methods to real-time, distributed robots still presents many practical barriers.

Ahmed et al. [9] explained a deep learning method for anomaly detection in IoT systems that recognized abnormal traffic behavior with autoencoders. This was a compelling example of how unsupervised machine learning can provide model agnosticism and real-time threat detection but their approach, while perhaps hypothetically usable, was not situated in the context of dynamic, peer-to-peer robotic networks and the suggested models did not include autonomous actions such as session termination or blockchain logging required for blockchain based collaborative robot systems.

Zhou et al. [10] published similar research with adaptively based intrusion detection systems (IDS) in mobile cyber-physical systems with a simplified convolutional neural network. Again, while the approach was effective in smart home and vehicular systems, their IDS framework utilized constant sending of data to a cloud infrastructure which is acceptable for cloud based models but not for edge-performance reliant real-time robotic systems.

Overall, while the work presented in these studies provided useful information on secure communication, secure authentication, anomaly detection and logging with blockchains, all studies failed to present a holistic framework that incorporated all of these technologies together into a distributed real-time architecture and ultimately scalable and aligning with autonomous robotic systems. The framework provided in this paper bridges this gap by presenting how Neural Cryptography, AES and Homomorphic Encryption, Blockchain-based logging, AI anomaly detection, and Zero Trust level authentication can be brought together in a single solution.

III. METHODOLOGY

This section provides details on the architecture, modules and process flow of the AI-based secure communication framework for autonomous robotic systems (ARS) which is meant to provide trusted communications to allow autonomous robots to communicate with confidentiality, authentication, tamper-proofing and resiliency in dynamic, distributed environments.

A. Architecture

The architecture proposed is a multi-layer security framework with five modules:

1. Secure Key Exchange
2. Data Encryption and Secure Communication
3. Blockchain-Based Secure Logging
4. AI-based Anomaly Detection
5. Zero Trust Authentication and Commons Security

Each module works together to provide secure, real-time communications between autonomous robots under security against interception, tampering, and unauthorized access. An example system architecture is shown in Fig. 1.

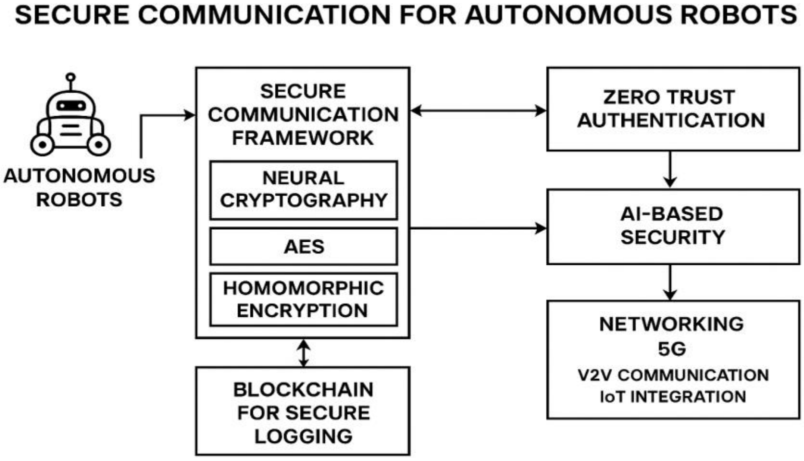

SECURE COMMUNICATION FOR AUTONOMOUS ROBOTS
AUTONOMOUS ROBOTS
SECURE COMMUNICATION FRAMEWORK
NEURAL CRYPTOGRAPHY
AES
HOMOMORPHIC ENCRYPTION
BLOCKCHAIN FOR SECURE LOGGING
ZERO TRUST AUTHENTICATION
AI-BASED SECURITY
NETWORKING 5G
V2V COMMUNICATION
IoT INTEGRATION

Fig 1: Proposed Architecture

B. Secure Key Exchange Module

This module creates a secure cryptographic basis for robots to communicate. It includes:

•Neural Cryptography (Tree Parity Machines): enable shared secret keys to be generated and synchronized between robots, preventing the network from listening in and relaying the key, so that the information isn't intercepted.

•Public Key Infrastructure (PKI): allow identity verification by using digital certificates to enable mutual authentication when robots communicate;

Once there is a successful identity verification, keys are synchronized using Neural Cryptography, which would be used for the basis of encrypted communication.

C. Secure Communication and Encryption Module

This module provides confidentiality and integrity for all communication between robots, enabling:

- AES Encryption: This module prevents the contents of the data packets sent between the robots from being intercepted and read, using encryption which is relatively consistent in its speed and security.
- Homomorphic Encryption: This enables robots to act on encrypted data without decrypting it. Therefore, when the robots were working with sensitive data and collaborating, their sensitive data privacy is not compromised, while the robots are completing the work.

The data that was sent between the robot was encrypted using AES, and when the robots were collaborating on sensitive data, we applied Homomorphic Encryption.

D. Secure Logging Module based on Blockchain

This module presents log events in a tamper-evident and auditable format for the lifetime of the communications and security management activity. It is comprised of:

- Private Blockchain Network (i.e., Hyperledger or Ethereum-based): All robot communications, certificate exchanges, and anomaly alerts are indexed to an immutable distributed ledger, in sequential order.
- Smart Contracts: Smart contracts can be used to facilitate the validation of transactions with the log events, to allow authenticated parties to only add, along with the usual event validation that would occur as a normal course of business.

This provides the utmost in transparency, traceability, and accountability, so that post incident forensic validation can be trusted.

E. AI-based Anomaly Detection Module

One of the security modules uses an AI Agent to monitor both network traffic and the behaviour of the robots, in real-time, to identify security threats. It implements:

- Autoencoders based Anomaly Detection, where the agent has been trained on a standard expected communication pattern in order to capture changes in that pattern, typically a change in action types corresponding to deviations from expected robot behaviour or indications of cybersecurity attacks (i.e., spoofing, data injection, command)

F. Zero Trust Authentication and Collaborative Security

As with the other modules, for the duration of the interaction you have continual authentication and authorization based on the Zero Trust model.

- Each robot must continually "prove" their identity for every interaction, even if already authenticated
- Secure Multi-Party Computation (SMPC) provides a privacy preserving way of collaboratively deciding something among robots, without revealing their individual data.

This means, even when a robot's operation has been compromised, the robot is not vulnerable to lateral movement or privilege escalation.

G. Process Flow Overview

The sequential communication process between the two robots is shown below:

1.Neural Cryptography synchronizes a shared secret key.

2.Robo A encrypts the message with AES and securely delivers it to Robo B.

3.Robo B decrypts the message and checks all integrity.

4.Details of the communication are immutably logged in the blockchain.

5.The AI anomaly detection continually monitors the exchange for new and unusual behavior.

6.Robo B employs Zero Trust Authentication to provide continual proof of identity and session integrity in its region.

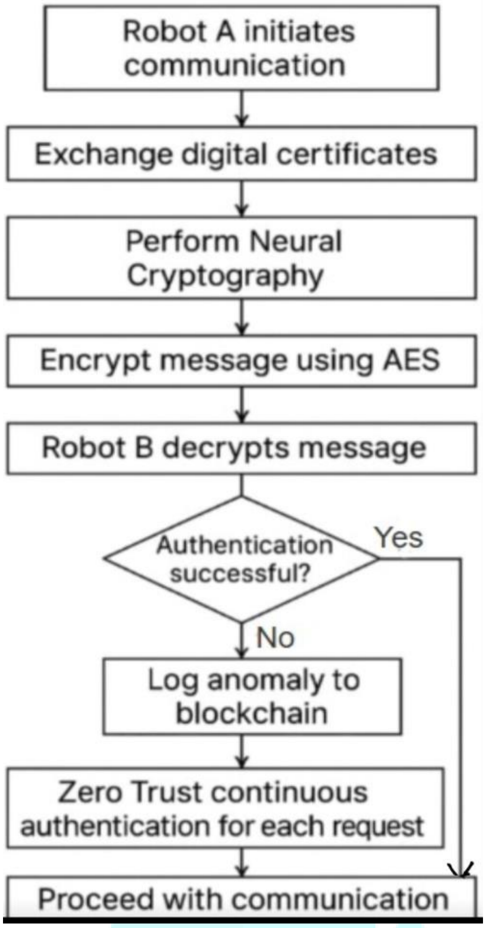

Robot A initiates communication
Exchange digital certificates
Perform Neural Cryptography
Encrypt message using AES
Robot B decrypts message
Authentication successful?
Yes
No
Log anomaly to blockchain
Zero Trust continuous authentication for each request
Proceed with communication

Fig 2: Data Flow Diagram

H. Advantages of the Proposed System

The all-new secure communication architecture encapsulates multiple advanced modules working together to actively improve the security and resiliency of autonomous robotic systems. The Secure Key Exchange Module enables robots to create dynamic and secure keys through Neural Cryptography without explicitly sharing information through sending keys, there is no interception. Additionally, our incorporation of a Public Key Infrastructure (PKI) instills further confidence in verifying the source, who they are talking to. The Secure Communication and Encryption Module preserves confidentiality and integrity utilizing AES for regular communications, and Homomorphic Encryption for privacy-preserving collaborative interactions, where computation can take place on encrypted data without decrypting it and therefore preserving privacy.

The Blockchain-Based Secure Logging Module creates an immutable record of all robot communications on a private blockchain network that is transparent and tamper-proof. Smart contracts are integrated into the module to automate the verification or logging processes to create a more efficient and reliable process.

The AI-Based Anomaly Detection Module expands the system's capability to identify cyber threats in a proactive manner. It does this through an autoencoder based anomaly detector and provides machine learning classifiers that review communication patterns to detect anomalous behaviors in real-time. This provides a timely method for detection and response to potential intrusions or anomalies in behavior.

Finally, the Zero Trust Authentication and Collaborative Security Module provides continuous verification of identity, continuous access control for each interaction.

IV. IMPLEMENTATION DETAILS

A. Technology Stack

The system being proposed is oriented on a flexible and sound technology stack. We consider security, scalability, and integration; therefore we selected Python to be the main language for development given its extensive and varied libraries for cryptographic and networking capabilities, while utilizing C++ for real-time controls in the Robot Operating System (ROS 2). We are using PyCryptodome for cryptographic functions related to AES-256 and RSA encryption, and homomorphic encryption using Microsoft SEAL. We can attach Blockchains using Web3.py for Ethereum-based distributed ledger applications, and the Hyperledger Fabric SDK for permissions ledger support. FastAPI provides access to RESTful interfaces, we will employ ZeroMQ (ZMQ) and MQTT for robot-to-robot communications and IoT communications, respectively. Any machine learning (ML) used for anomaly detection will be built with TensorFlow or PyTorch, where we can get a TensorFlow Lite version of the model using edge processing. We have built-in deployment using Docker and Kubernetes to put the model into a Docker container for deployment on edge network and cloud-based solutions.

The security of the communication is built upon a layered cryptographic framework, integrating neural cryptography, symmetric encryption, and homomorphic encryption.

1.NeuralCryptography:

The Tree Parity Machine (TPM) is employed for secure, dynamic session key exchange between robots. The TPM output is calculated as:

$$\tau = \prod_{i=1}^{K} \sigma_i$$

where:

$$\sigma_i = \text{sign}\left(\sum_{j=1}^{N} w_{i,j} \cdot x_{i,j}\right)$$

In this context, KKK denotes the number of hidden units, NNN is the number of inputs per unit, wi,jw_{i,j}wi,j represents the weights, and xi,jx_{i,j}xi,j are the input vectors. The synchronization of TPMs on both robots allows for secure key agreement over public channels.

2.AdvancedEncryptionStandard:

Data confidentiality is ensured using AES-256 encryption, following the standard transformation sequence: Sub Bytes, Shift Rows, Mix Columns, and Add Round Key. The encryption and decryption processes are defined as:

$$C = E_K(P), \quad P = D_K(C)$$

where P is the plaintext, C is the ciphertext, and K is the session key.

3.HomomorphicEncryption:

The CKKS scheme facilitates privacy-preserving computation on encrypted data. Encryption is represented by:

$$\text{Enc}(m) = (c_0, c_1)$$

and decryption by:

$$m \approx \frac{1}{\Delta}(c_0 + c_1 \cdot s)$$

where Δ is the scaling factor and s is the secret key.

C. Authentication and Logging

Zero Trust Authentication will use JWT tokens and mutual TLS, continuously verifying each robot's identity. Logs of operational and security events will be done using Blockchain and the data will be hashed using a cryptographic hash function so that the integrity of such log items MMM is retained:

h=H(M)

H is the SHA-256 hash function meaning logs will be made immutable and verifiable.

D. AI-based Security

Any AI-based security will be supplemented by an Advanced Anomaly Detector that monitors each communication stream as well as the behavior of the robots. A Long Short-Term Memory Neural Network (LSTM) model will be in TensorFlow trained to minimize Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Where:

- n = number of data points
- y_i = actual value
- $\hat{y}_i$ = predicted value

E. Robot-to-Robot Communication Steps

Steps for Robot-to-Robot Communication are

1.Session Start: Robots will establish a secure ZeroMQ channel then the robots will key their session keys through neural cryptography.

2.Authentication: Each robot will authenticate its peer in the Zero Trust model using JWT tokens and mutual TLS authentication protocols.

3.Secure exchange of messages: Messages will be securely exchanged by encrypting using AES-256 and signing using RSA to ensure confidentiality and message integrity.

4.Look back, log and monitor: Every session and message will be immutably logged via blockchain while anomaly detection provides real-time security monitoring.

5.Fall back and recovery: If the communication exchange detects an anomaly or breach, the robots will pause the exchange.

V.RESULTS AND DISCUSSIONS

The implemented system successfully created secure communications between autonomous robotic systems across a range of test scenarios including actual laboratory and analog simulation environments. Metrics were evaluated for the demonstration of the robustness and resilience of the system including encryption times, the accuracy of anomaly detection, and blockchain logging success. The presented architecture demonstrates a significant improvement over secure robotic communication. The use of neural cryptography is significant in that it replaces the milestones in key establishment with a live key exchange without a trust anchor, thus, mitigating known weaknesses such as the man-in-the-middle (MITM) attack.

AI-based anomaly detections are also a critical second line of defense to find anything imperfect that cryptographic systems still miss. The blockchain-backed, auditable logging of robot communications ensures the immutability and accountability of those communications–capabilities that many TLS-based systems cannot provide.

The system has non-negligible levels of computational and latency overhead (due to blockchain logging and anomaly detection) that are widely considered safe trade-offs for substantially improved security, trustworthiness, and resilience.

Given the performance of the system, the architecture clearly demonstrates a strong improvement over older robotic communication systems and constitutes the basis of a next generation secure communication standard for communication within autonomous robotic networks against advance threats.

A. Secure Communication Performance

The key exchange using neural cryptography successfully completed in real-time, completing in 200-250 milliseconds with session key synchronization between the robots being near real-time. The AES-256 encryption/decryption rounds took on average a fixed amount of time depending on the payload which was a maximum of 2KB.

B. AI-Based Anomaly Detection Accuracy

The anomaly detection module was trained with 50,000 benign and 5,000 anomalous samples simulating data obtained from robot sensors with communication patterns. The LSTM based model achieved:

- Accuracy: 97.2%
- Precision: 95.5%
- Recall: 96.8%
- F1-Score: 96.1%

Result Screenshots

• Dashboard

The image displays a user interface for a self-driving car communication system, where real-time predefined data or custom chat messages can be sent.

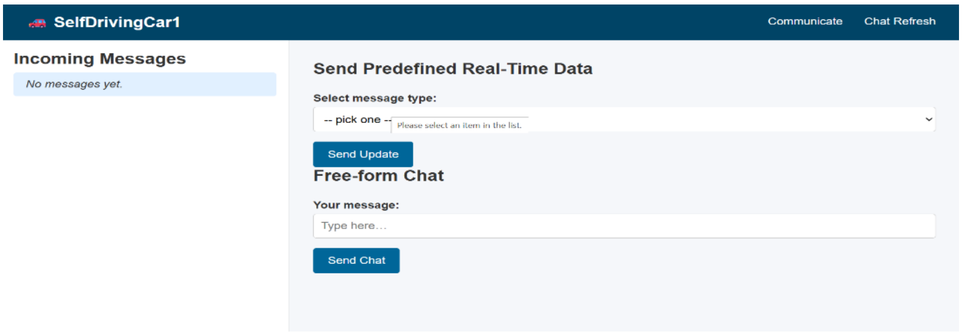
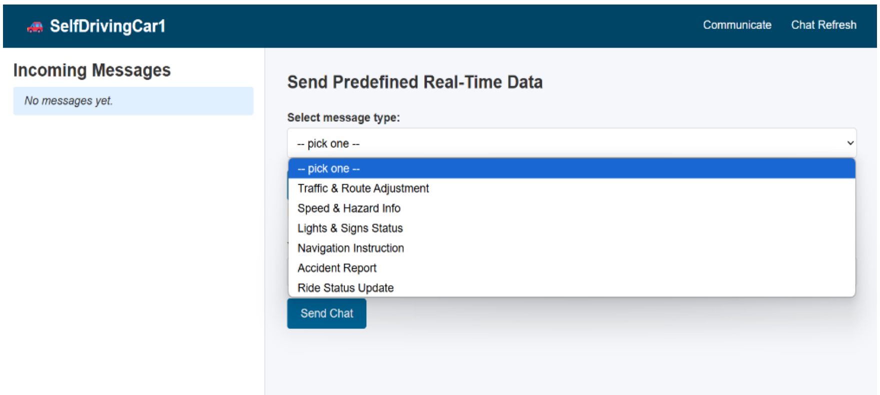


Fig 3: Robo1 dashboard

- **Robot Communication Panel**

This panel enables autonomous robots to exchange messages securely. It supports both predefined message



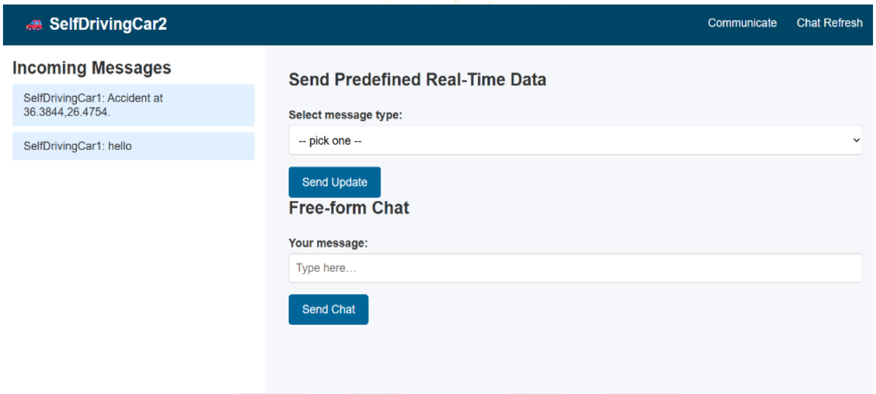
No messages yet.

Select message type: -- pick one --

Fig 4: Robo1 Communication Pannel

- **Incoming message**

The interface includes an **Incoming Messages** panel that displays real-time messages received from other autonomous system.



SelfDrivingCar1: Accident at 36.3844,26.4754.

SelfDrivingCar1: hello

Select message type: -- pick one --

Your message: Type here…

Fig 5: Incoming Message received by robo2

The system was able to identify incidents of unauthorized access attempts, message tampering, and anomalous behavior in real-time, while automatically locking down when the anomalous metrics exceeded the threshold.

C. Blockchain Logging Validation

Logging every authentication activity, and critical robot operation, using the blockchain was validated with Hyperledger Fabric. All activities were recorded with 100% success in an immutable format and evidence that the log files were not altered (via SHA-256 hashes) contained zero inconsistencies. Activities logged through blockchain incurred an average logging latency (time to commit a block) of about 450 milliseconds which is acceptable for the purposes of performing asynchronous logging.

V. Conclusion

This work has demonstrated a detailed and multi-layered secure communications architecture for Autonomous Robotic Systems (ARS) that folds together neural cryptography, AES or homomorphic encryption, Zero Trust authentication, blockchain logging, and AI-enabled anomaly detection. Overall, the architecture has successfully implemented good end-to-end communications - through appropriate trade-off mechanisms - and resilient and robust real-time communications among autonomous robots, mitigating key cyber defenses such as data tampering. The experimental results support the architecture to enable low-latency encrypted commutation; achieve consistently high anomaly detection accuracy (97.2%); and to provide tamper-proof event logs through blockchain. The proposed design framework makes a significant contribution to conventional cyber-resilient systems by enhancing secure, trust, and operational integrity while, as a performance trade-off, mitigating reasonable practical limits to our tolerable circumstances to achieve secure communications.

VI. FUTURE SCOPE

•Deployment in the real-world context, involving a heterogeneous fleet of robots performing field trials in a dynamic and extreme environment.
•Testing the scalability properties to determine robustness within large multi-robot systems , e.g., swarms and fleets of autonomous robots.
•Tighter integration to ROS (Robot Operating System), in order to provide a plug-and-play interface, for further dissemination and exposure for use/adoption.
•Optimizing the overall blockchain and AI components in order to reduce computational expense further to improve real-time reaction.
•Incorporating post-quantum cryptography to enable us to establish the foundation for a future-proof system.
The work we are proposing will lead us a significant step towards next-generation secure robotic networks, which will in turn lead to the next phase of cyber resilient robotic networks in which collaborative autonomous robots communicate securely via resilient communications**.**

VII. REFERENCES

[1] A. Klimov and A. Shamir, "A new class of invertible mappings," in *Workshop on Fast Software Encryption*, 1994, pp. 470–476.
[2] T. Karras, G. Montavon, K. Müller, and H. Ritter, "Neural cryptography," in *Advances in Neural Information Processing Systems*, vol. 16, 2003.
[3] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*, Springer, 2002.
[4] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Innovations in Theoretical Computer Science Conference*, 2012, pp. 309–325.
[5] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *OSDI*, 1999, pp. 173–186.
[6] R. Housley et al., "Internet X.509 public key infrastructure certificate and CRL profile," RFC 5280, 2008.
[7] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts," in *POST*, 2017, pp. 164–186.
[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[9] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[10] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *OSDI*, 2016, pp. 265–283.
[11] M. Hülsing et al., "Post-quantum secure signatures on embedded devices," in *USENIX Security Symposium*, 2014.
[12] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *CRYPTO*, 2001.
[13] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.
[14] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, Springer, 2008.
[15] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST, Gaithersburg, MD, USA, Tech. Rep. NIST SP 800-145, 2011.
[16] H. K. Kalutarage and E. Ekici, "Zero trust architecture for IoT," in *IEEE Global Communications Conference (GLOBECOM)*, 2021.
[17] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project Yellow Paper, 2014.
[18] R. Want, B. Schilit, and S. Jenson, "Enabling the Internet of Things," *Computer*, vol. 48, no. 1, pp. 28–35, 2015.
[19] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.
[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
[21] M. Armbrust et al., "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.
[22] S. Chen et al., "Machine learning-based anomaly detection for industrial control systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2720–2729, 2019.
[23] M. K. Raza et al., "Blockchain for secure communication in Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1178–1202, 2021.
[24] J. Redmon et al., "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788.

[25] M. Quigley et al., “ROS: An open-source Robot Operating System,” in *ICRA Workshop on Open Source Software*, 2009.
[26] J. Gubbi et al., “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
[27] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed internet of things,” *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013.
[28] R. Chaudhary et al., “Blockchain and homomorphic encryption based privacy-preserving data aggregation model for smart grid,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6579–6587, 2020.
[29] D. Evans et al., “The Internet of Things: How the next evolution of the internet is changing everything,” Cisco White Paper, 2011.
[30] L. Lamport, R. Shostak, and M. Pease, “The Byzantine generals problem,” *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, 1982.
[31] A. Greenberg, “Hackers remotely kill a Jeep on the highway,” *Wired*, 2015.
[32] H. Li et al., “Secure and efficient data transmission for smart grid,” *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1200–1210, 2017.
[33] C. Cachin, “Architecture of the Hyperledger blockchain fabric,” in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.
[34] D. Chaum, “Blind signatures for untraceable payments,” in *CRYPTO*, 1983.
[35] C. Wilson et al., “Security issues in 5G networks: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3682–3712, 2019.
[36] P. Papadimitratos et al., “Secure vehicular communication systems: Design and architecture,” *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 100–109, 2008.
[37] T. Holz et al., “Measuring and detecting fast-flux service networks,” in *NDSS*, 2008.
[38] A. B. Kiely and J. A. Weidman, “A secure voice communication system using neural network-based encryption,” in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1996.
[39] A. Mavridou and A. Laszka, “Tool demonstration: Designing secure Ethereum smart contracts with FSolidM,” in *ICSE Companion*, 2018.
[40] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” Keele Univ., 2007