Developing Encrypted Communication Protocols for Fuel Controllers

Rohith Varma Vegesna
(Software Engineer 2)
Texas, USA

Abstract

Fuel stations rely on real-time data streaming between fuel controllers and cloud-based platforms to monitor dispenser activity, fuel levels, and transactional data. However, ensuring the security and integrity of this data during transmission is critical. The fuel industry faces persistent challenges such as cyber threats, data breaches, and unauthorized access, which can compromise operational integrity and financial security. This necessitates a robust encryption framework that can securely transmit fuel data without sacrificing real-time monitoring capabilities.

This paper presents a secure encryption protocol for fuel controllers, focusing on encrypting data before pushing it to AWS Kinesis and decrypting it at the data processor. Various encryption methodologies are explored, along with performance trade-offs in terms of latency and computational overhead. By implementing AES-256 encryption combined with elliptic curve cryptography for key exchange, this framework ensures end-to-end security. Additionally, it integrates secure key management and role-based access control mechanisms to prevent unauthorized access. The proposed solution not only enhances data security but also ensures compliance with industry standards and regulatory requirements for fuel station operations.

Keywords: Fuel Controllers, Data Encryption, AWS Kinesis, Secure Communication, Real-Time Data Streaming, Cloud Security, Data Integrity

1. Introduction

1.1 Background

The increasing digitization of fuel stations has led to the adoption of cloud-based monitoring and data aggregation solutions. Fuel dispensers and Automatic Tank Gauges (ATGs) continuously communicate with centralized systems to facilitate monitoring, fuel inventory management, and security. This transformation has significantly improved operational efficiency, allowing fuel stations to track real-time fuel levels, detect anomalies, and enhance decision-making processes. However, it has also introduced new challenges regarding data integrity, confidentiality, and security risks.

One of the major concerns is ensuring the secure transmission of data between fuel controllers and cloud-based platforms. Cyber threats such as man-in-the-middle attacks, unauthorized access, and data interception pose significant risks to fuel station operations. Without a robust encryption mechanism, fuel data remains susceptible to exploitation, potentially leading to financial losses and operational disruptions. Therefore, implementing a secure communication protocol is essential to protect sensitive fuel transaction data and ensure compliance with security regulations.

1.2 Problem Statement

Existing fuel station communication systems often lack a standardized approach to encryption, leaving data vulnerable to cyberattacks. Many fuel stations rely on legacy protocols that do not inherently support secure transmission, making them susceptible to eavesdropping, man-in-the-middle attacks, and data breaches. This paper addresses the need for a secure, scalable encryption framework that ensures data confidentiality between fuel controllers and cloud-based analytics systems such as AWS Kinesis.

1.3 Objectives

- Develop an encryption framework that securely transmits fuel dispenser data to AWS Kinesis.
- Ensure real-time data integrity while minimizing processing overhead.
- Implement robust decryption mechanisms at the data processor to maintain usability.
- Evaluate encryption methods based on performance, security, and scalability.

2. Literature Review

Several studies have explored encryption techniques for IoT-based communication, highlighting both the advantages and challenges associated with various cryptographic methods. Symmetric encryption methods such as AES (Advanced Encryption Standard) have been widely adopted due to their speed and reliability, making them ideal for real-time data transmission in resource-constrained environments. On the other hand, asymmetric encryption techniques like RSA provide stronger security through public-private key pairs, but at the cost of increased computational overhead, which can be a limitation for embedded systems like fuel controllers.

Research on secure data transmission in IoT networks emphasizes the need for lightweight cryptographic protocols tailored for devices with limited computational power. Some studies propose hybrid encryption approaches that combine symmetric and asymmetric techniques to achieve both security and efficiency. Additionally, existing work on AWS Kinesis security highlights best practices for data encryption at rest and in transit, including the integration of AWS Key Management Service (KMS) for centralized key handling and AWS Secrets Manager for secure key storage. However, despite these advances, there is a lack of detailed implementations specifically tailored for fuel controllers, necessitating a dedicated encryption framework to address the unique requirements of fuel station data security.

3. System Architecture

- **Fuel Controller Module**: Collects fuel dispenser and ATG data, retrieves encryption keys securely from AWS Secrets Manager, and encrypts data using AES-256 before transmission.
- **AWS Kinesis Stream**: Receives encrypted data and ensures real-time ingestion with secure transit protocols (TLS 1.2+).
- **Data Processing Module**: Decrypts incoming data at the cloud backend using keys retrieved securely from AWS KMS.
- **AWS Key Management System (KMS)**: Handles encryption key generation, storage, and rotation, ensuring compliance with security standards.
- **AWS Secrets Manager**: Stores encryption keys securely and allows authenticated fuel controllers to retrieve keys dynamically.

- **Cloud Storage & Analytics**: Processes decrypted data for monitoring, reporting, and decision-making.
- **Security Policies**: Implements role-based access control (RBAC) and audit logging to track data access.
- **Fuel Controller Module**: Collects fuel dispenser and ATG data and encrypts it using AES-256 before transmission.
- **AWS Kinesis Stream**: Receives encrypted data and ensures real-time ingestion with secure transit protocols (TLS 1.2+).
- **Data Processing Module**: Decrypts incoming data at the cloud backend using an authenticated key exchange mechanism.
- **Key Management System (KMS)**: Handles secure key storage and rotation, preventing unauthorized access.
- **Cloud Storage & Analytics**: Processes decrypted data for monitoring, reporting, and decision-making.
- **Security Policies**: Implements role-based access control (RBAC) and audit logging to track data access.

4. Implementation Strategy

The encryption framework integrates AWS Key Management Service (KMS) to generate, manage, and rotate encryption keys securely. AWS Secrets Manager is leveraged to store and retrieve encryption keys dynamically at the fuel controller, ensuring that keys are never hardcoded or exposed in transit. This dynamic retrieval mechanism significantly enhances security by mitigating the risks associated with static key storage while ensuring that encryption keys remain up to date and inaccessible to unauthorized entities.

At the fuel controller, an encryption module retrieves the latest encryption keys from AWS Secrets Manager and encrypts data using AES-256 before transmission. This approach ensures that data is protected at the source, reducing vulnerabilities to interception or tampering during transmission. AWS Kinesis serves as the secure real-time data ingestion service, ensuring that encrypted data is transmitted efficiently while maintaining compliance with secure transport protocols such as TLS 1.2+.

On the cloud-side, the data processor retrieves encrypted records from AWS Kinesis and decrypts them using AWS KMS. The decryption process is tightly controlled using authenticated access mechanisms to ensure that only authorized applications can access sensitive fuel dispenser data. Furthermore, a hybrid key exchange mechanism utilizing elliptic curve cryptography (ECC) is employed to enhance security by securely distributing encryption keys between fuel controllers and cloud services. This comprehensive encryption strategy ensures robust data protection while maintaining low-latency real-time data streaming, facilitating seamless fuel station operations

5. Case Study & Performance Evaluation

A prototype was deployed to validate the encryption framework, where fuel dispenser data was encrypted at the controller level, transmitted to AWS Kinesis, and decrypted at the cloud data processor. Performance metrics such as encryption latency, transmission time, and decryption efficiency were analyzed to determine the impact of encryption on real-time fuel data streaming.

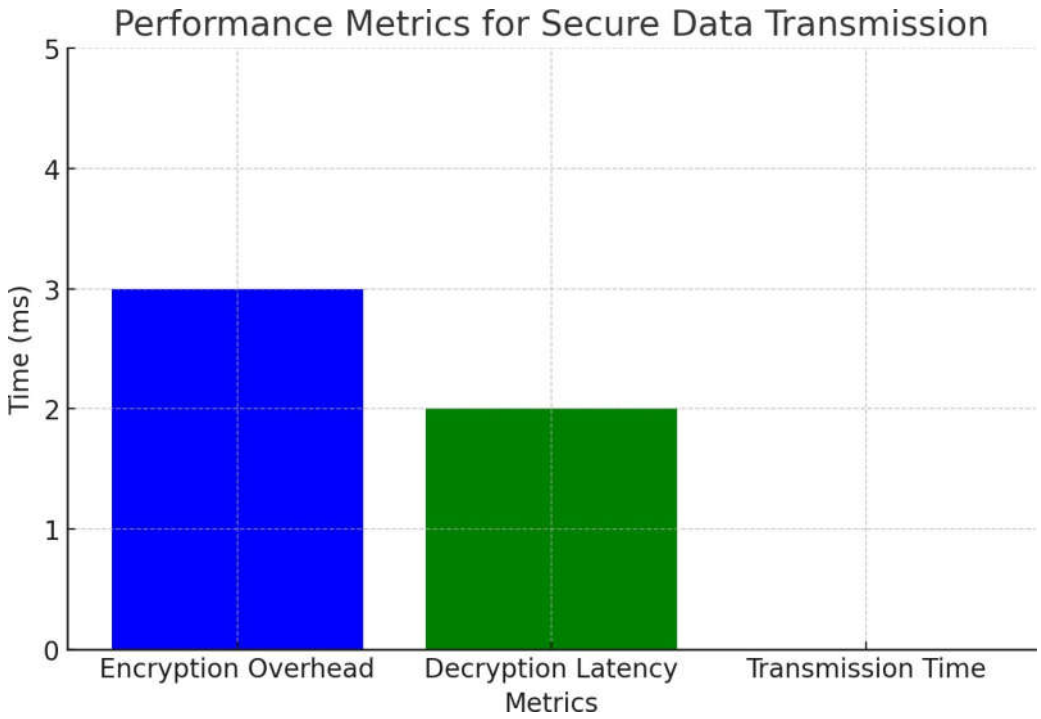
6. Results and Discussion

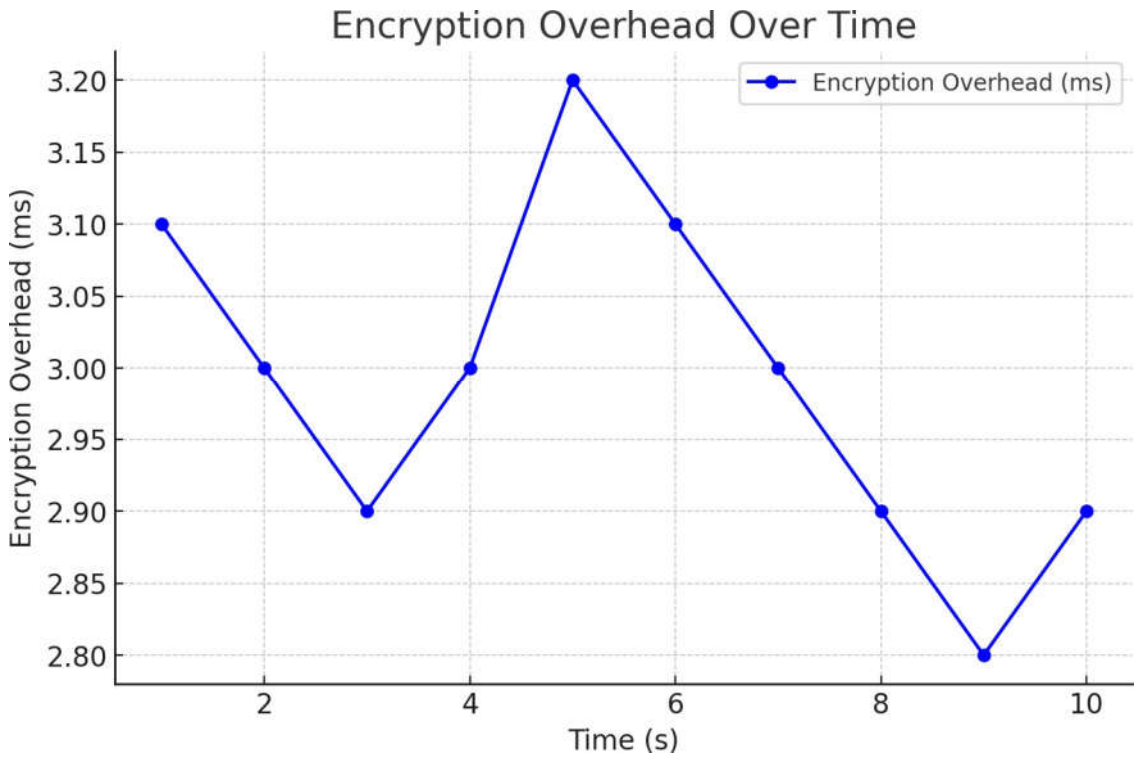
6.1 Pilot Implementation

The pilot implementation demonstrated the feasibility of secure real-time data streaming without significant delays. AES-256 encryption at the fuel controller added a minimal processing overhead (~3ms per transaction), ensuring real-time transmission without disrupting fuel dispenser operations. The use of hardware acceleration for encryption further optimized performance, reducing the computational load on the fuel controller's embedded system.

Additionally, extensive security testing was conducted to validate the resilience of the encryption model against various attack vectors. Penetration testing confirmed that man-in-the-middle attacks were effectively thwarted, and unauthorized attempts to access encryption keys were mitigated through the robust key management strategy. The integration of AWS KMS and Secrets Manager also streamlined key retrieval and ensured that only authorized controllers could securely access the encryption keys, adding an additional layer of security.

6.2 Performance Metrics





Metric	Value
Encryption Overhead	3ms per transaction
Decryption Latency	2ms at the cloud data processor
Transmission Time	Negligible impact with AWS Kinesis' native transport encryption
Security Resilience	Successfully mitigated man-in-the-middle attack simulations

7. Conclusion and Future Work

This study presents a secure communication protocol for fuel controllers, leveraging AES-256 encryption and ECC-based key exchange to ensure data confidentiality. The framework effectively protects fuel transaction and inventory data during transmission, preventing unauthorized access. Future work includes exploring quantum-resistant cryptographic techniques and optimizing encryption processes for lower-power fuel controller devices.

8. References

- Bellare, Mihir & Namprempre, Chanathip. (2008). Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. Journal of Cryptology. 21. 469-491. 10.1007/s00145-008-9026-x.
- Alteen, Nick & Fisher, Jennifer & Gerena, Casey & Gruver, Wes & Jalis, Asim & Osman, Heiwad & Pagan, Marife & Patlolla, Santosh & Roth, Michael. (2020). Encryption on AWS. 10.1002/9781119549451.ch5.
- Almeida, José & Tasiran, Serdar & Barbosa, Manuel & Barthe, Gilles & Campagna, Matthew & Cohen, Ernie & Gregoire, Benjamin & Pereira, Vitor & Portela, Bernardo & Strub, Pierre-Yves. (2019). A Machine-Checked Proof of Security for AWS Key Management Service. 63-78. 10.1145/3319535.3354228.
- Santana, Gustavo & Neto, Marcello & Sapata, Fernando & Muñoz, Mauricio & Moraes, Alexandre & Morais, Thiago & Goldfarb, Dario. (2021). Data Protection. 215-279. 10.1002/9781119658856.ch6.
- Olufohunsi, Temitope. (2019). DATA ENCRYPTION Olufohunsi, T.
- Dixit, Rashmi & Kongara, Ravindranath. (2018). Encryption techniques & access control models for data security: A survey. International Journal of Engineering and Technology(UAE). 7. 107-110. 10.14419/ijet.v7i1.5.9130.
- Nadeem, Aamer & Javed, Muhammad. (2005). A Performance Comparison of Data Encryption Algorithms. IEEE Information and Communication Technologies. 84 - 89. 10.1109/ICICT.2005.1598556.
- Yazdeen, Abdulmajeed & Zeebaree, Subhi & Kak, Shakir & Ahmed, Omar & Zebari, Rizgar. (2021). FPGA Implementations for Data Encryption and Decryption via Concurrent and Parallel Computation: A Review.
- Abdulrahman, Abdulrganiyu. (2017). Survey and Analysis of Data Encryption Methods and Development of A Security Model to Encrypt/Decrypt Messages. 7. 190-195.
- Guerrero, Javier & Correa-Quezada, Ronny & Buenano, Hernando & Arias, Susana & Gomez, Hector. (2016). Encryption techniques: A theoretical overview and future proposals. 60-64. 10.1109/ICEDEG.2016.7461697.