UNIVERSITY OF PLYMOUTH

TSE CHI KIN
Email: chi.k.tse@students.plymouth.ac.uk

COMP3000 Computing Project
2023/2024

BSc (Hons) Cyber Security

SUPERVISORS: Dr Beta Yip

Dr Ivy Wong

Project Title:

ACS Mini System

Automation of Construction Data Monitoring with Secure and Safety System

Acknowledgements

I would like to express my deepest gratitude and appreciation to Professor Beta Yip and Ivy Wong for their invaluable guidance, support, and contributions throughout the development of this project. Their expertise, dedication, and unwavering commitment to excellence have played a significant role in the successful completion of this endeavor.

Professor Beta Yip, your mentorship and guidance have been instrumental in shaping the direction of this project. Your insightful feedback and constructive suggestions have pushed me to explore new horizons and strive for excellence. Your commitment to academic rigor and your passion for the subject matter have inspired me to dig deeper and challenge myself throughout the research process. Professor Ivy Wong, I am profoundly grateful for your invaluable contributions to this project. Your expertise in cyber security has been instrumental in shaping the research methodology and analyzing the data. This project would not have been possible without their guidance and encouragement, and for that, I am forever grateful.

I would also like to acknowledge the support and encouragement provided by my classmates. Their valuable insights, discussions, and collaboration have enriched the project and contributed to its success. I am grateful for the stimulating academic environment that the department has fostered, which has provided a platform for intellectual growth and the exchange of ideas.

Finally, I would like to extend my deepest appreciation to my family and friends for their unwavering support, understanding, and encouragement throughout this journey. Their belief in my abilities and constant motivation have been vital in overcoming challenges and staying focused.

Abstract

The construction industry is undergoing a digital transformation, with automation and robotics playing an increasingly significant role in enhancing safety standards and operational efficiency. However, the adoption of these technologies has been uneven, with small and medium-sized enterprises (SMEs) in the Hong Kong construction industry facing significant challenges due to limited resources, an aging workforce, and outsourcing patterns. Most SMEs still relied on manual monitoring methods and this manual approach increases the risk of errors and accidents. The implementation of automated monitoring systems in construction sites is not without challenges. The lack of stable electricity and the need for numerous Internet of Things (IoT) devices often strain the efficiency and reliability of these systems. Additionally, continuous monitoring, maintenance, and repairs require significant manpower, further increasing the resource constraints faced by SMEs.

The "ACS Mini System - Automation of Construction Data Monitoring with Secure and Safety System" project aims to address these challenges by developing an affordable and suitable solution tailored specifically for SMEs in the construction industry. The project focuses on evaluating and implementing data encryption algorithms for resource-constrained devices or embedded systems, ensuring secure data transmission and storage. This is a critical concern in the construction industry where sensitive information must be protected from unauthorized access or potential security breaches.

By leveraging lightweight data encryption algorithms optimized for resource-constrained environments, the ACS Mini System seeks to provide a secure and reliable solution for automating construction data monitoring. This approach not only enhances data security but also addresses the challenges of limited computing power and energy constraints often encountered in construction site environments.

To evaluate and compare the performance of different data encryption algorithms, the project will utilize Arduino-based simulations for data transfer with encryption. By testing and analyzing various encryption algorithms suitable for Arduino and similar resource-constrained platforms, the project aims to identify the most efficient and suitable encryption technique for automated instrumentation in the construction sector, specifically tailored to the needs and constraints of SMEs.

User-friendly interfaces and comprehensive documentation are prioritized to overcome adoption barriers and empower SMEs with limited technical expertise. The development process follows a secure development lifecycle, incorporating security considerations throughout the entire project lifecycle, including secure coding practices, data encryption, authentication mechanisms, and compliance with industry standards and regulatory requirements.

The successful implementation of the ACS Mini System has the potential to empower SMEs in the Hong Kong construction industry to embrace digital transformation and stay competitive in the rapidly evolving construction landscape. By providing an affordable, secure, and efficient solution for automating construction data monitoring, the project aims to improve safety standards, enhance operational efficiency, and foster a more sustainable and technologically advanced construction industry.

Table of Contents

Word Count

Word count: 10564

Code Repository

The developed solution’s code can be found at:
https://github.com/hkuspace-pu/ACSMini

1. Introduction

1.1 Background

The construction industry has historically relied on manual processes and labor-intensive methods, which often lead to inefficiencies, errors, and safety concerns. With the advent of digital technologies and automation, there is a growing demand for innovative solutions that can address these challenges and facilitate the modernization of construction operations. However, the adoption of these technologies has been uneven, with larger construction companies being at the forefront of embracing automation and digital transformation, while small and medium-sized enterprises (SMEs) face significant barriers.

In Hong Kong, the construction industry is a vital sector that plays a crucial role in the city's development and infrastructure projects. However, the industry is characterized by an aging workforce, outsourcing patterns, and limited resources among SMEs. These factors have contributed to the continued reliance on manual monitoring methods. The workers use field book to mark down construction data from working site, go back to office, input data to computer and calculate the result with excel, print out report and submit to client. Due to potential miscalculations or missed information, manual processing may increase the risk of errors and accidents.

While large-scale construction companies have recognized the benefits of automated monitoring systems and have invested in advanced technologies, SMEs often struggle to implement similar solutions due to various constraints. The big problem is the lack of stable electricity and suitable infrastructure on the working sites. Many construction sites operate in remote or temporary locations, making it difficult to establish reliable power sources and maintain complex monitoring systems.

Furthermore, the implementation and maintenance of automated monitoring systems requires significant financial resources and specialized technical expertise, which can be challenging for SMEs with limited budgets and personnel. The need for extensive manpower to monitor, repair, and maintain these systems further compounds the challenges faced by SMEs, limiting their ability to fully leverage the benefits of automation.

In response to these challenges, the "ACS Mini System - Automation of Construction Data Monitoring with Secure and Safety System" project aims to develop an affordable and suitable solution tailored specifically for SMEs in the construction industry. The project recognizes the unique needs and constraints faced by SMEs and seeks to address them through the integration of innovative technologies and practical approaches.

1.2 Aim and Objectives

The primary objective of the "ACS Mini System - Automation of Construction Data Monitoring with Secure and Safety System" project is to provide a comprehensive and affordable solution for SMEs in the construction industry, enabling them to automate data monitoring processes while ensuring secure data transmission and storage. By leveraging cutting-edge technologies and practical approaches, the project aims to address the unique challenges faced by SMEs, including limited resources, aging workforce, and site environment constraints.

One of the key focus areas of the project is the evaluation and implementation of data encryption algorithms for resource-constrained devices or embedded systems. Ensuring secure data transmission and storage is crucial in the construction industry, where sensitive information related to project designs, contractual agreements, and safety protocols must be protected from unauthorized access or potential security breaches. By leveraging lightweight data encryption algorithms optimized for resource-constrained environments, the ACS Mini System aims to provide a secure and reliable solution for automating construction data monitoring.

Furthermore, the project seeks to improve safety standards and enhance operational efficiency in the construction industry. By automating data collection and monitoring processes, the ACS Mini System aims to reduce the risk of errors and accidents associated with manual methods, ultimately contributing to a safer working environment for construction workers. Real-time data analysis and alerting mechanisms will be integrated into the system to identify potential safety hazards and enable prompt remedial actions.

Another objective of the project is to address the challenges of limited computing power and energy constraints often encountered in construction site environments. The integration of edge computing capabilities and cloud-based data management will play a crucial role in achieving this objective. Edge computing minimizes the need for constant data transmission, reducing power consumption and reliance on stable electricity sources. Meanwhile, the cloud-based architecture eliminates the need for extensive on-premises infrastructure, making the solution more accessible and cost-effective for SMEs with limited resources.

Additionally, the project will provide a user-friendly interface and reporting capabilities that enable SMEs to access and understand construction site data without requiring extensive technical expertise. This is vital for facilitating better decision-making and communication among stakeholders, ultimately leading to improved project management and collaboration within the construction industry.

To ensure the successful implementation and adoption of the ACS Mini System, the project will focus on developing a scalable and modular architecture. This will allow SMEs to start with a basic setup and gradually expand the system as their needs grow, ensuring a cost-effective and flexible solution that can adapt to varying project requirements and construction site complexities.

Finally, the project will foster collaboration and knowledge-sharing among SMEs in the construction industry. By providing a common platform for data monitoring and analysis, the ACS Mini System can facilitate the exchange of best practices, lessons learned, and innovative approaches, enabling SMEs to collectively address the challenges they face and drive continuous improvement within the industry.

1.3 Scope

The scope of the "ACS Mini System - Automation of Construction Data Monitoring with Secure and Safety System" project encompasses the development and implementation of a comprehensive solution tailored for SMEs in the construction industry, particularly in Hong Kong. The system will integrate various technologies and components to address the unique challenges faced by SMEs, including limited resources, aging workforce, and site environment constraints.

The project will focus on the design and implementation of low-power sensor nodes equipped with various sensors for collecting construction site data, such as temperature, humidity, vibration, and motion. These sensor nodes will be designed to operate on battery power or harvested energy sources, making them suitable for environments with limited or unstable electricity.

Additionally, the project will evaluate and implement lightweight data encryption algorithms optimized for resource-constrained devices or embedded systems. These algorithms will ensure secure data transmission from the sensor nodes to the central data management system, protecting sensitive construction site information from potential security breaches.

The ACS Mini System will incorporate edge computing capabilities to preprocess and filter sensor data at the edge, reducing the amount of data that needs to be transmitted to the central system. This approach will minimize power consumption and reduce the reliance on constant data transmission.

Furthermore, the project will develop a cloud-based data management and analytics platform for robust data storage, processing, and analysis. This platform will enable SMEs to leverage advanced analytics without the need for extensive on-premises infrastructure, making the solution more accessible and cost-effective.

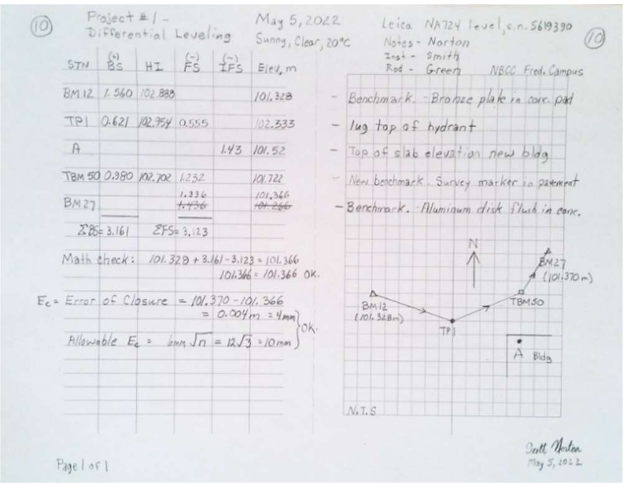
The project will also design and implement a user-friendly interface and reporting capabilities, allowing authorized personnel to access and visualize construction site data, generate reports, and receive alerts or notifications regarding potential safety concerns or operational issues.

While the primary focus of the project is on addressing the challenges faced by SMEs in the Hong Kong construction industry, the developed solution may have broader applications and potential for adaptation in other regions or industries facing similar resource constraints and monitoring requirements.

2. Literature review

2.1 Existing Systems

The construction industry has long been dominated by labor-intensive processes, manual data collection, and paper-based documentation. But traditional methods are time-consuming, labor-intensive, and prone to errors. Manual approaches often lead to increased risks of accidents and contribute to inefficiencies. Transitioning to more automated and digitized systems can help address these shortcomings. By leveraging innovative technologies, the construction industry can enhance data accuracy, improve safety, and streamline operations. This shift towards a more technologically driven approach presents an opportunity to increase productivity, reduce costs, and ultimately deliver better project outcomes for all stakeholders involved.



(10) Project #1 - Differential Leveling
May 5, 2022
Sunny, Clear, 20°C
Leica NA724 level, s.n. 5619390 (10)
Notes - Norton
Inst - Smith
Rod - Green
NBCC Fred. Campus

STN	(+) BS	HI	(−) FS	(−) IFS	Elev, m	
BM 12	1.560	102.888			101.328	- Benchmark. Bronze plate in conc. pad
TP1	0.621	102.954	0.555		102.333	- lug top of hydrant
A				1.43	101.52	- Top of slab elevation new bldg
TBM 50	0.980	102.702	1.232		101.722	- New benchmark. Survey marker in pavement
BM 27			1.336 ~~1.436~~		101.366 ~~101.266~~	- Benchmark. Aluminum disk flush in conc.
	ΣBS= 3.161		ΣFS= 3.123			

Math check: 101.328 + 3.161 − 3.123 = 101.366
101.366 = 101.366 OK.

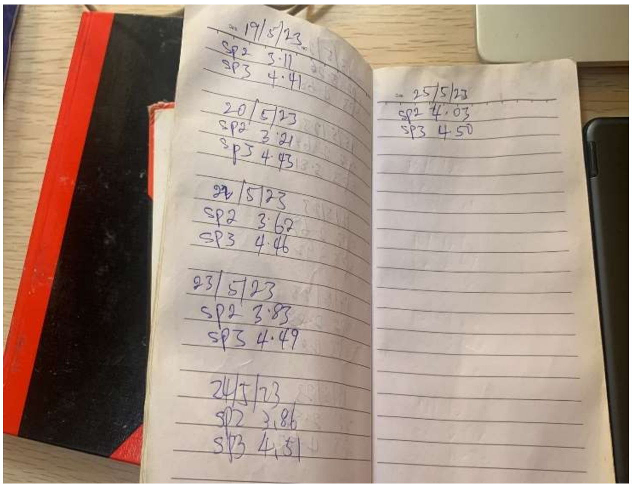
E_c = Error of Closure = 101.370 − 101.366
= 0.004 m = 4 mm
Allowable E_c = 6mm $\sqrt{n}$ = $12\sqrt{3}$ = 10 mm
OK.

N
BM27 (101.370 m)
TBM50
BM12 (101.328m)
TP1
A Bldg
N.T.S

Page 1 of 1

Scott Norton
May 5, 2022

Manual Survey



19/5/23
SP2 3.11
SP3 4.41

20/5/23
SP2 3.21
SP3 4.43

22/5/23
SP2 3.62
SP3 4.46

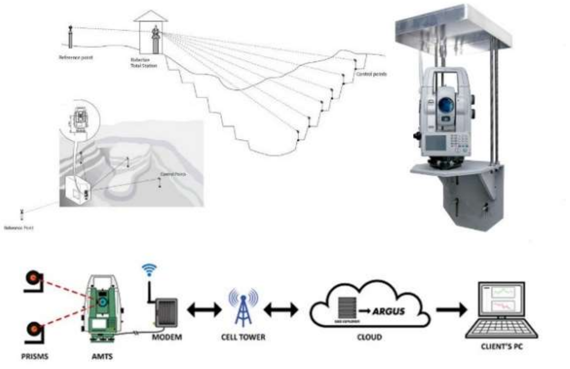
23/5/23
SP2 3.83
SP3 4.49

24/5/23
SP2 3.86
SP3 4.51

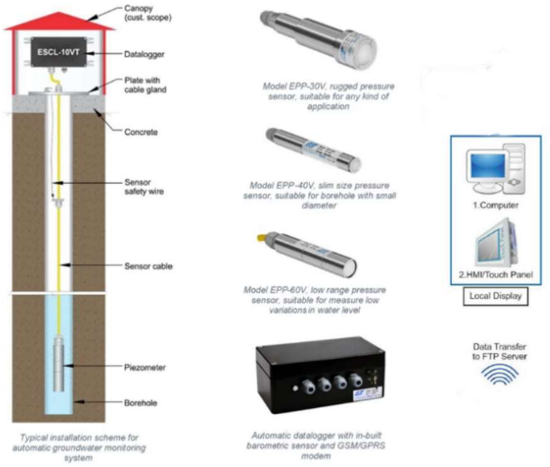
25/5/23
SP2 4.03
SP3 4.50

Water Level Records

In recent years, there has been a growing adoption of automated monitoring systems by large construction companies, leveraging advanced instruments and sensors to collect and analyze data in real-time.



Automated Motorized Total Station (AMTS) System


Canopy (cust. scope)
ESCL-10VT
Datalogger
Plate with cable gland
Concrete
Sensor safety wire
Sensor cable
Piezometer
Borehole
Typical installation scheme for automatic groundwater monitoring system
Model EPP-30V, rugged pressure sensor, suitable for any kind of application
Model EPP-40V, slim size pressure sensor, suitable for borehole with small diameter
Model EPP-60V, low range pressure sensor, suitable for measure low variations in water level
Automatic datalogger with in-built barometric sensor and GSM/GPRS modem
1.Computer
2.HMI/Touch Panel
Local Display
Data Transfer to FTP Server

Automatic Water Level Monitoring system

These automated monitoring systems typically employ various technologies, including Internet of Things (IoT) devices, wireless sensor networks (WSNs), and cloud computing platforms. IoT devices and WSNs enable the collection of data from multiple sources, such as structural health monitoring sensors, environmental sensors, and worker safety sensors. The collected data is then transmitted to cloud platforms or site servers for processing, analysis, and visualization. Several commercial and research-based solutions have been developed to address the need for automated construction data monitoring. For example, the Smart Site Safety System.

While automated monitoring systems offer significant benefits in terms of efficiency, safety, and decision-making, their adoption by SMEs in the construction industry has been hindered by several challenges, such as limited Resources. SMEs often face resource constraints, including limited budgets, lack of specialized IT personnel, and limited access to advanced technologies. The high initial investment required for implementing automated monitoring systems can be a big problem for many SMEs. Besides, existing automated monitoring systems are typically designed for large-scale construction projects and may lack the scalability and flexibility required by SMEs with varying project sizes and resource availability.

2.2 Challenges and Opportunities

This literature review explores the current state of research and industry practices related to the automation of construction data monitoring, secure data transmission, and the adoption of emerging technologies by small and medium-sized enterprises (SMEs) in the construction sector.

Automation of Construction Data Monitoring

The automation of construction data monitoring has been an active area of research, driven by the need for accurate, real-time data collection and analysis. Numerous studies have highlighted the benefits of automated monitoring systems, including improved safety, increased productivity, and enhanced decision-making capabilities.

Researchers have explored various technologies for automating construction data monitoring, such as wireless sensor networks (WSNs), Internet of Things (IoT) devices, and Building Information Modeling (BIM) integrated systems. WSNs have gained significant attention due to their ability to collect and transmit data from multiple sensors in real-time, enabling continuous monitoring of construction sites.

However, the implementation of automated monitoring systems in the construction industry has faced challenges related to the harsh and dynamic nature of construction sites, as well as the need for robust and reliable data transmission mechanisms. Researchers have proposed different approaches to address these challenges, including the use of edge computing and fog computing architectures to reduce data transmission loads and improve system resilience.

Secure Data Transmission in Construction

Ensuring secure data transmission is a critical aspect of automated construction data monitoring systems, as sensitive information related to project designs, contractual agreements, and safety protocols must be protected from unauthorized access or potential security breaches. Several studies have explored the application of data encryption algorithms and secure communication protocols in the construction sector.

Lightweight data encryption algorithms, optimized for resource-constrained devices or embedded systems, have been proposed as a viable solution for securing data transmission in construction sites. These algorithms aim to strike a balance between security and computational efficiency, enabling secure data transmission while minimizing the impact on system performance and energy consumption.

In addition to data encryption, research has focused on secure communication protocols and access control mechanisms for construction data monitoring systems. These protocols and mechanisms aim to ensure the authenticity, integrity, and confidentiality of transmitted data, mitigating the risk of unauthorized access or tampering.

Adoption of Emerging Technologies by SMEs in Construction

While the benefits of automation and digital technologies in the construction industry are widely recognized, the adoption of these technologies has been uneven, with SMEs facing significant barriers. Several studies have explored the challenges and factors influencing the adoption of emerging technologies by SMEs in the construction sector.

Limited financial resources, lack of technical expertise, and resistance to change have been identified as major obstacles hindering the adoption of new technologies by SMEs. Researchers have emphasized the need for affordable and user-friendly solutions tailored to the unique needs and constraints of SMEs, as well as the importance of providing training and support to facilitate the successful implementation of these technologies.

Furthermore, studies have highlighted the potential benefits of cloud-based solutions for SMEs in the construction industry, as they can provide access to advanced technologies and computing resources without the need for significant upfront investments in infrastructure.

2.3 Future Research Directions

The existing literature provides valuable insights into the automation of construction data monitoring, secure data transmission, and the adoption of emerging technologies by SMEs, there are still gaps and areas for further research:

1. Integration of Multiple Technologies: Many studies have focused on individual technologies or components, such as wireless sensor network or data encryption algorithms. However, there is a need for research exploring the integration and interoperability of multiple technologies to create comprehensive and scalable solutions for construction data monitoring and secure data transmission.

2. Energy Efficiency and Sustainability: Some studies have addressed energy efficiency concerns, further research is needed to develop sustainable solutions for automated construction data monitoring systems, particularly in resource-constrained environments.

3. Usability and User Acceptance: The technical aspects of automated monitoring systems have been extensively studied, there is a need for more research on usability, user experience, and user acceptance factors, especially for SMEs with limited technical expertise.

4. Regulatory and Policy Frameworks: As the adoption of automated monitoring systems and secure data transmission technologies in the construction industry grows, research is needed to investigate the regulatory and policy frameworks required to ensure compliance, data privacy, and ethical use of these technologies.

5. Implementation Strategies and Best Practices: The benefits of automation and emerging technologies are widely recognized, there is a need for research on effective implementation strategies and best practices tailored to the unique needs and constraints of SMEs in the construction industry.

By addressing these researches, the construction industry can unlock the full potential of automated data monitoring, secure data transmission, and the responsible adoption of emerging technologies, ultimately leading to improved safety, efficiency, and sustainability in construction operations.

3. Legal, Social, Ethical and Professional issues

3.1 Legal Issues:

Data Privacy and Protection: The ACS Mini System will collect and process sensitive construction site data like project designs, contractual agreements, and safety information. It is very important to make sure these sensitive data is kept private and protected according to follow the data protection laws and regulations, such as the Personal Data (Privacy) Ordinance (PDPO) in Hong Kong. Appropriate measures must be taken to protect the privacy and confidentiality of personal and sensitive data. This includes using strong data encryption and access control methods. Only authorized people can read it or access the data based on their permissions.

Intellectual Property Rights: The development of the ACS Mini System may involve the use of proprietary technologies, algorithms, or software components. Those may be owned by other companies or individuals. It is critical to respect intellectual property rights and get proper licenses or permissions before using anything proprietary. This can avoid infringement of intellectual property rights and legal disputes.

Construction Regulations and Standards: The construction industry has many regulations and standards in place related to safety, environmental protection, building codes, and more. The implementation of the ACS Mini System must comply with all relevant regulations and ensure that the automated monitoring processes of the system need to adhere to established construction standards and guidelines.

3.2 Social Considerations

Digital Divide and Accessibility: Implementing an advanced technology system like the ACS Mini System may create a situation where some small and medium construction companies can afford and utilize it, while others cannot. This will create a digital divide between different companies. It is essential to consider promoting equal accessibility and inclusion with the system's benefits. The advantages of using it should not be limited to only a certain group of companies.

Employment Impact: The automation of construction data monitoring processes may raise concerns that some jobs may be impacted or displaced by the technology. Or that current workers may need retraining on the new systems and processes. These are valid worries that must be proactively addressed. Providing training opportunities to upskill workers and help them adapt to new construction technologies and automated processes will be crucial.

Community Engagement: Construction projects often have a significant impact on local communities around the job sites in terms of noise, traffic, resource usage and more. The implementation of the ACS Mini System should involve community engagement and transparency, addressing potential concerns or misconceptions about the use of automated monitoring systems and their implications for the local environment and neighborhood.

3.3 Ethical Considerations

Algorithmic Bias and Fairness: The data encryption algorithms and analytics components of the ACS Mini System must be carefully designed and built to be completely fair and unbiased. There cannot be any discrimination. Appropriate measures should be put into place to prevent the system's algorithms from unfairly perpetuating any existing biases against certain groups based on race, gender, age, or other factors.

Privacy and Consent: While the main purpose of ACS is monitoring construction data, there may be some cases where personal data of workers or individuals also gets collected and processed by the system. Anytime personal data is involved, it is essential to get informed consent from people first. They must be clearly told what data will be collected, how it will be processed, where it will be stored, and explicitly consent to this.

Accountability and Transparency: All the decision-making processes and outcomes produced by the ACS Mini System need to be transparent and easily audited. There should be clear accountability over the system's recommendations and data analytics. Mechanisms must allow the decisions and insights from ACS to be explained, reviewed, and inspected when needed. This promotes trust and accountability with all stakeholders using the system.

3.4 Professional Considerations

Competence and Expertise: Properly implementing and maintaining the complex ACS Mini System requires a highly skilled and knowledgeable workforce. Construction companies will need access to staff with the required expertise in data, technology, security and more to effectively utilize ACS. Investing in professional training and continuing education opportunities is vital for building this competent workforce.

Professional Ethics and Conduct: The development and deployment of the ACS Mini System should adhere to established professional codes of conduct and ethical guidelines within the construction industry. This includes maintaining integrity, objectivity, and acting in the best interests of clients and stakeholders.

Continuous Improvement and Adaptation: The construction industry is constantly evolving, and the needs of companies may change over time. A culture of continuous improvement should be fostered, where the ACS Mini System gets frequently reviewed and updated based on new requirements, advances in technology, updated regulations, and current best practices. Sticking to an agile, adaptable mindset is key for the long-term success and relevance of ACS.

By carefully considering and addressing all these legal, social, ethical and professional issues, the "ACS Mini System - Automation of Construction Data Monitoring with Secure and Safety System" project can help enable the responsible, equitable and ethical adoption of new technology in the construction industry. It will require collaborative efforts from all stakeholders, including construction companies, government regulators, technology providers, professional organizations and more. Proactively navigating these challenges in a transparent manner will be crucial for delivering positive impacts with ACS Mini System.

4. Project Planning and Management

The "ACS Mini System - Automation of Construction Data Monitoring with Secure and Safety System" project aims to develop an affordable and suitable solution for small and medium-sized enterprises (SMEs) in the Hong Kong construction industry. To ensure the successful execution and delivery of the project, a comprehensive project plan is essential. This section outlines the key components of the project planning process, including scope definition, work breakdown structure, resource allocation, risk management, and project timeline.

The project will follow an agile development approach with iterative cycles. Security will be a core focus throughout the development of the lifecycle.

Project Phases
The ACS Mini System project will be executed across the following major phases:

Phase 1: Project Initiation (1 week)
- Formally launch the project and assemble the core team.
- Define project objectives, scope, and high-level requirements.
- Secure necessary resources, budget, and approvals.

Phase 2: Requirements Gathering and Analysis (2 weeks)
- Identify key construction site parameters to monitor (e.g. structural integrity, environment, safety).
- Analyze security needs of SME construction firms with limited IT resources.
- Define data privacy, integrity, authentication, and access control requirements.
- Develop security policies aligned with industry best practices and regulations.

Phase 3: System Design (3 weeks)
- Design a scalable, cost-effective architecture with minimal on-site hardware.
- Incorporate security controls like encryption, authentication, access management.
- Plan secure communication protocols for protecting data integrity.
- Design lightweight AI/ML models for efficient data processing and analysis.

Phase 4: Development (8 weeks)
- Apply secure coding practices and pre-built security libraries.
- Develop user-friendly data collection interfaces and integration components.
- Implement selected encryption algorithms and security mechanisms.
- Build monitoring dashboard with data visualization and analytics.
- Develop mobile app for on-site access and notification.
- Integrate lightweight AI/ML algorithms for construction data analysis.
- Set up secure cloud infrastructure for system hosting.

Phase 5: Testing and Quality Assurance (5 weeks)
- Conduct thorough testing (functional, performance, security, etc.).
- Perform vulnerability assessments and penetration testing.
- Test AI/ML model accuracy and computational performance.
- Collaborate with pilot SMEs for real-world testing and feedback.
- Fix any issues identified and optimize system performance.

Phase 6: Deployment and Implementation (1 week)
- Prepare production environment for system deployment.
- Migrate system components and data to production.
- Perform final integration and configuration checks.

Phase 7: User Training (1 week)
- Develop user guides, training materials and documentation.
- Conduct training sessions for pilot SME construction companies.

Phase 8: Documentation (1 week)
- Finalize system documentation, including architecture, deployment, and maintenance guides.
- Document security controls, policies, and procedures.

Phase 9: Project Review and Closure (1 week)
- Review project deliverables and ensure all requirements are met.
- Gather feedback and lessons learned from the project team.
- Transition the system to operational status and hand off to support team.
- Conduct project closing activities and prepare final reports.

Work Packages and Timelines

Work Package	Duration
Project Initiation	**1 weeks**
Requirements Gathering and Analysis	**2 weeks**
System Design	**3 weeks**
Development	**8 weeks**
Testing and Quality Assurance	**5 weeks**
Deployment and Implementation	**1 weeks**
User Training	**1 weeks**
Documentation	**1 weeks**
Project Review and Closure	**1 weeks**

Secure Development Lifecycle
The project will follow a secure development lifecycle, integrating security practices throughout all phases:

Design Phase:
- Incorporate security controls from the ground up.
- Plan for secure coding, data encryption, access controls, etc.
- Design secure system architecture and infrastructure.

Development Phase:
- Apply secure coding practices and utilize security libraries.
- Implement encryption, authentication, and authorization mechanisms.
- Conduct secure code reviews and static analysis.

Testing Phase:
- Perform security testing, including vulnerability assessments.
- Test authentication, access controls, and encryption implementation.
- Verify compliance with security policies and data privacy regulations.

Deployment Phase:
- Establish secure deployment practices and configuration guidelines.
- Implement secure update and patch management processes.
- Configure security monitoring and incident response procedures.

Operations Phase:
- Continuous security monitoring and threat detection.
- Timely application of security updates and patches.
- Incident response and recovery planning.

By integrating security throughout the entire lifecycle, the ACS Mini System will be designed, built, and operated with robust security controls to protect sensitive construction data.

Project Team and Resources
The core project team will include:

- Project Manager: Oversees planning, coordination and stakeholder management.
- Business Analysts: Gather requirements, map processes, manage change.
- Software Architects: System design, technology selection, architecture.
- Software Developers: Code, implement and integrate system components.
- Quality Assurance: Testing, validation, and compliance checks.
- Security Experts: Implement encryption, access controls and security practices.
- Cloud Engineers: Provision and manage cloud infrastructure.
- UI/UX Designers: Design user interfaces for dashboard and mobile app.
- Technical Writers: Develop documentation and training guides.

The team may also engage external consultants for specialized construction domain expertise.

Project Risks and Mitigation Strategies

The ACS Mini System project aims to develop a secure and automated solution for monitoring construction data at job sites for small and medium-sized construction companies (SMEs). Like any complex project, there are potential risks that need to be carefully managed to ensure successful delivery. This risk plan outlines some key risks and proposes mitigation strategies.

Risk 1: Technical Complexity
The project involves integrating advanced technologies like artificial intelligence (AI) and ensuring cybersecurity measures are in place. This technical complexity could lead to delays or unexpected issues during development.

Mitigation Strategy:
Before starting the project, the team will conduct a thorough feasibility study and technology assessment. This will help identify potential technical challenges early on. The project plan will allocate sufficient time for research and development activities. As the project progresses, the team will regularly review and update the project plan based on any emerging technical requirements.

Risk 2: Resource Constraints
SMEs often face limitations in terms of budget, availability of skilled workforce, and infrastructure hardware. These resource constraints can impact the project's execution.

Mitigation Strategy:
The team will conduct a comprehensive resource planning exercise involving representatives from participating SMEs. The goal is to identify and allocate necessary resources effectively. The team will explore cost-effective solutions that align with the budget constraints of SMEs. This may include leveraging cloud computing services, open-source software, or partnering with academic institutions for skilled workforce.

Risk 3: Data Security and Privacy
The ACS Mini System will handle sensitive construction data, which poses potential risks of data breaches and privacy violations. Any such incidents could have severe consequences for the participating SMEs.

Mitigation Strategy:
To mitigate this risk, the team will implement robust security measures, such as data encryption, access controls, and regular security audits. The system will comply with local privacy laws and industry regulations. The team will develop and implement appropriate risk strategies to ensure data security and privacy throughout the system's lifecycle.

Risk 4: Changes in Regulatory or Legal Requirements
Construction data management and cybersecurity are subject to various regulations and legal requirements. Changes in these regulations or legal requirements during the project's lifecycle could impact the system's development and implementation.

Mitigation Strategy:
The team will stay updated on relevant regulations, industry standards, and legal requirements related to construction data management and cybersecurity. Time and resources will be allocated for regular compliance reviews. If any changes are identified, the team will adapt the system accordingly to ensure adherence to the latest regulatory requirements.

Risk 5: Technical Complexity
The project involves integrating advanced artificial intelligence (AI) technologies and ensuring cybersecurity measures are in place. This technical complexity could lead to delays or unexpected issues during the development phase.

Mitigation Strategy:
Before starting the project, the team will conduct a thorough feasibility study and technology assessment to identify potential technical challenges early on. The project plan will allocate sufficient time for research and development activities related to AI and cybersecurity. Throughout the project, the team will regularly review and update the project plan based on any emerging technical requirements or challenges.

Risk 6: User Adoption and Change Management
The ACS Mini System introduces new technologies and processes to SMEs in the construction industry. There is a risk of resistance to change or difficulties in adopting the new system, which could hinder its successful implementation.

Mitigation Strategy:
The team will develop a comprehensive user adoption and change management plan. This plan will include extensive user training programs, clear communication about the system's benefits (such as improved safety, productivity, and cost savings), and user-friendly interfaces and documentation. The team will work closely with SMEs to understand their concerns and tailor the system to their specific needs.

Risk 7: Integration with Existing Systems and Data Sources
The ACS Mini System will need to integrate with various existing systems and data sources used by SMEs in the construction industry. Compatibility issues or data format inconsistencies could pose challenges during integration.

Mitigation Strategy:
The team will develop standardized data integration interfaces and adapters to simplify the integration process with diverse data sources and existing systems. Robust documentation and Software Development Kits (SDKs) will be provided to support third-party integration efforts. Extensive testing and validation of data integration components will be conducted to ensure reliability and compatibility.

Risk Monitoring and Control
Effective risk monitoring and control are crucial for the successful management of identified risks. The project team will implement the following mechanisms:

1. **Risk Register**: A comprehensive risk register will be maintained, documenting identified risks, their potential impact, likelihood, and assigned mitigation strategies.

2. **Risk Review Meetings**: Regular risk review meetings will be held with the project team, stakeholders, and subject matter experts. These meetings will discuss the status of identified risks, the effectiveness of mitigation strategies, and the identification of new potential risks.

3. **Risk Reports**: Periodic risk reports will be generated, summarizing the project's overall risk profile, high-priority risks, and the progress of mitigation efforts. These reports will be distributed to key stakeholders for transparency and informed decision-making.

4. **Risk Audits**: Periodic risk audits will be conducted to assess the effectiveness of the risk management process and ensure adherence to established policies and procedures.

By proactively identifying and addressing potential risks through this comprehensive risk plan, the ACS Mini System project can increase its chances of successful delivery while ensuring the security, privacy, and integrity of sensitive construction data. Regular risk monitoring, mitigation plan reviews, and continuous improvement of risk management practices are essential throughout the project lifecycle.

5. Project Methodology

The project "ACS Mini System - Automation of Construction Data Monitoring with Secure and Safety System" aims to develop a comprehensive solution for automating the collection, monitoring, and secure transmission of construction data. Therefore, data transmission is very important, and it takes a list of evaluation for encryption algorithms in resource constrained environments.

Encryption Algorithms

Encryption algorithms are mathematical techniques used to secure data by converting it into a coded format that can only be read by authorized parties with the correct decryption key. The research project evaluates several widely used and approved encryption algorithms to identify the most suitable options for resource-constrained environments like construction sites. The algorithms were selected to cover a broad range of encryption techniques and include those commonly used in secure communication protocols.

1. Advanced Encryption Standard (AES): AES is a popular symmetric encryption algorithm, which means the same key is used for both encryption and decryption. It is widely used in secure communication protocols like Transport Layer Security (TLS) and Secure Sockets Layer (SSL) to protect data transmitted over internet. AES comes in different variations based on the key size: AES-128, AES-192, and AES-256. The larger the key size, the stronger the encryption. For example, AES-128 uses a 128-bit key, while AES-256 uses a 256-bit key. AES works by dividing the plaintext (unencrypted data) into fixed-size blocks and applying a series of mathematical operations using the encryption key. These operations include substitution, permutation, and mixing of the data. The result is the ciphertext (encrypted data), which appears as a random sequence of bytes. AES also supports different modes of operation, such as Cipher Block Chaining (CBC) and Cipher Feedback (CFB), which add additional layers of security and improve efficiency. In this research, the intermediate key sizes of 128, 192, and 256 bits, along with the CFB mode, were chosen as they are approved by the National Institute of Standards and Technology (NIST).

AES Small is a lightweight variant of the AES algorithm designed for resource-constrained environments. It uses smaller key sizes (128 bits or less) and fewer rounds of encryption, making it more efficient in terms of memory and computational requirements while still providing adequate security. This makes AES Small particularly suitable for deployment in embedded systems and IoT devices, where power and space are limited. By balancing security and performance, AES Small offers a practical solution for encrypting data in construction site monitoring systems, ensuring the confidentiality and integrity of sensitive information without overburdening the underlying hardware.

2. BLAKE2b and BLAKE2s: BLAKE2b and BLAKE2s are cryptographic hash functions from the BLAKE2 family. Hash functions are used to ensure data integrity and authentication by producing a fixed-size message digest or hash value, which is a unique digital fingerprint of the input data. BLAKE2b is designed for 64-bit platforms and produces a 512-bit hash value, while BLAKE2s is designed for 32-bit platforms and produces a 256-bit hash value. Both algorithms are based on the BLAKE hash function but with improved performance, security, and simplicity. BLAKE2b and BLAKE2s are considered highly secure and resistant to various types of attacks, making them suitable for a wide range of applications, including digital signatures, file integrity verification, and password hashing.

3. ChaCha8, ChaCha12 and ChaCha20: ChaCha8, ChaCha12 and ChaCha20 are variants of the ChaCha stream cipher algorithm, with the numbers representing the number of rounds of encryption applied. ChaCha20 is the most widely used variant and is designed for high-performance encryption and decryption. It generates a keystream (a sequence of pseudorandom bytes) using a block function applied to a 256-bit key, a 96-bit nonce (a random value used only once), and a block counter. The plaintext is then XORed (combined using the Exclusive OR operation) with the keystream to produce the ciphertext.

ChaCha8 and ChaCha12 are lightweight variants of ChaCha20, with fewer rounds of encryption (8 and 12 rounds, respectively). They are designed for resource-constrained environments, such as embedded systems and IoT devices, where computational power and memory are limited.

4. ChaCha20-Poly1305: ChaCha20-Poly1305 is a combined authenticated encryption algorithm that uses ChaCha20 for encryption and Poly1305 for authentication. It provides both confidentiality and integrity protection for the data. In this combination, Poly1305 generates a one-time key and a nonce from a 32-byte key, which are then used by ChaCha20 to encrypt the plaintext. Poly1305 also generates a tag based on the ciphertext and additional data, which is used for authentication. ChaCha20-Poly1305 is widely used in secure communication protocols like TLS and DTLS due to its high performance and security properties.

Poly1305 is a message authentication code (MAC) algorithm used for data integrity and authentication. It is designed to generate a unique code (called a tag) that is sent along with the encrypted data. Poly1305 works by processing the encrypted data using a separate authentication key and a nonce (a random value used only once). The resulting tag is then used by the recipient to verify the authenticity and integrity of the data.

5. Secure Hash Algorithm (SHA-256): SHA-256 is a widely used cryptographic hash function from the Secure Hash Algorithm (SHA) family. Hash functions are used to ensure data integrity and authentication by producing a fixed-size message digest or hash value, which is a unique digital fingerprint of the input data. SHA-256 processes the input data in 512-bit blocks and produces a 256-bit hash value. It is commonly used in various security protocols, digital signatures, and file integrity verification. SHA-256 is considered secure and resistant to various types of attacks, making it a reliable choice for many applications.

SHA3-256 is a hash function from the SHA-3 family, which was designed to be more secure and resistant to certain types of attacks, such as collisions and preimage attacks, compared to the previous SHA-2 family (which includes SHA-256). SHA3-256 uses a different internal structure and a larger block size of 1088 bits, while still producing a 256-bit hash value. It is considered more secure and efficient than SHA-256, making it a suitable alternative for applications that require a high level of security.

SHAKE256 is an extendable-output function from the SHA-3 family. Unlike the fixed-length hash functions, SHAKE256 allows for the generation of hash values of arbitrary length, making it more flexible for diverse applications. This feature is particularly useful in scenarios where different output lengths are required, such as key derivation, random number generation, and applications that need to handle variable-length inputs or outputs. The security properties of SHAKE256, including its resistance to various types of attacks, make it a suitable choice for applications that demand a high level of cryptographic strength. Its ability to produce hash values of customizable lengths enables it to be tailored to the specific requirements of different systems and use cases.

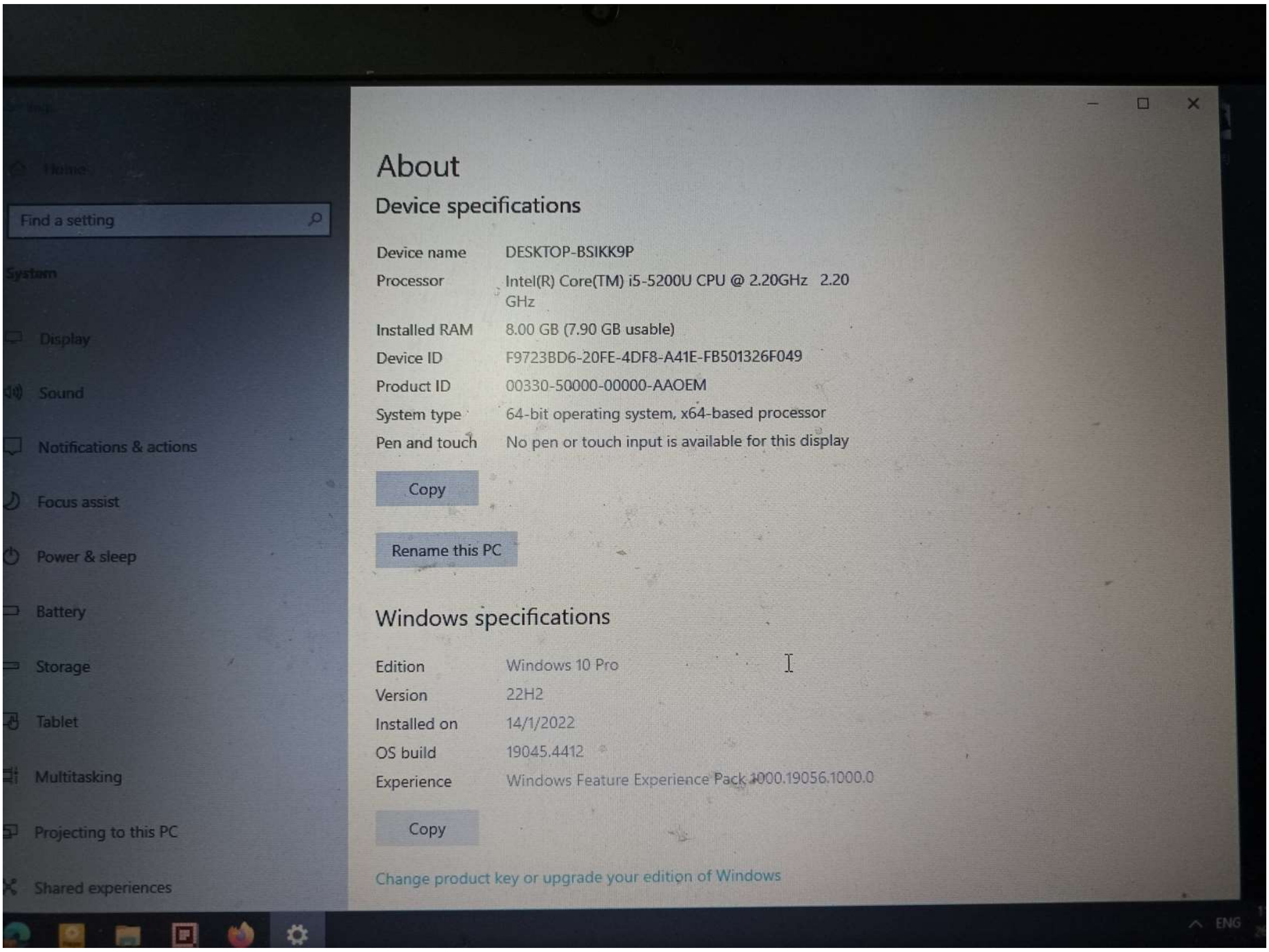
The research project evaluates the performance and security properties of these encryption algorithms to identify the most suitable options for secure and efficient data monitoring in the construction industry, particularly for small and medium-sized enterprises (SMEs) with limited resources. The selection of algorithms covers a wide range of techniques, including symmetric encryption, hash functions, authenticated encryption, and lightweight variants, to cater to different requirements and constraints.

6. Project Implementation

The ACS Mini System project aims to automate construction data monitoring processes while ensuring secure and safe data handling through the implementation of encryption algorithms on resource-constrained devices. The project implementation can be divided into several phases, including hardware setup, software development, and performance testing. This write-up will provide an overview of the project implementation process, focusing on the testing environment, equipment used, and the steps involved in evaluating and selecting the most suitable encryption algorithm for the resource-constrained Arduino Uno board.

Testing Environment: Windows 10 System

The testing environment for the ACS Mini System project will be a Windows 10 operating system. This choice is made based on the widespread availability and compatibility of Windows 10 with various development tools and software required for the project.



About

Device specifications

Device name DESKTOP-BSIKK9P
Processor Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.20 GHz
Installed RAM 8.00 GB (7.90 GB usable)
Device ID F9723BD6-20FE-4DF8-A41E-FB501326F049
Product ID 00330-50000-00000-AAOEM
System type 64-bit operating system, x64-based processor
Pen and touch No pen or touch input is available for this display

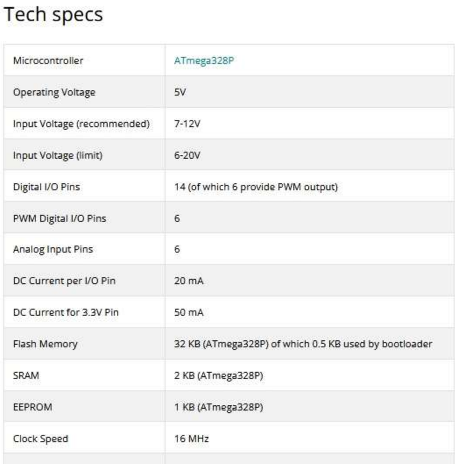
Windows specifications

Edition Windows 10 Pro
Version 22H2
Installed on 14/1/2022
OS build 19045.4412
Experience Windows Feature Experience Pack 1000.19056.1000.0

Change product key or upgrade your edition of Windows

Equipment: Arduino Uno

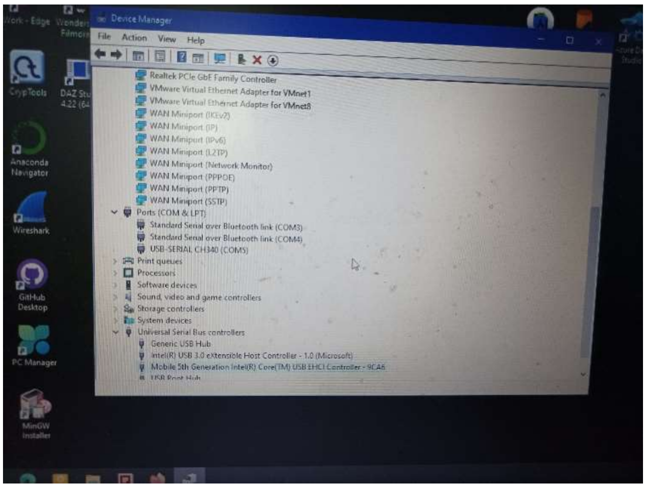
The Arduino Uno board will be the primary hardware component used in the project. This microcontroller board is widely adopted for prototyping and testing purposes due to its affordability, ease of use, and extensive community support. The Arduino Uno features an ATmega328P microcontroller, which has 32 KB of flash memory, 2 KB of SRAM, and 1 KB of EEPROM. These limited resources make it an ideal platform for testing the performance of lightweight encryption algorithms suitable for resource-constrained environments.



Tech specs

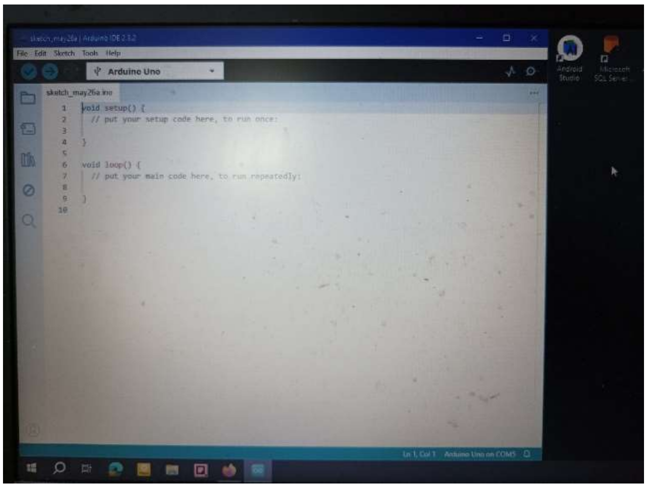
Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz

The implementation process begins by connecting the Arduino Uno board to a laptop computer through the USB port and installing the CH340 driver.



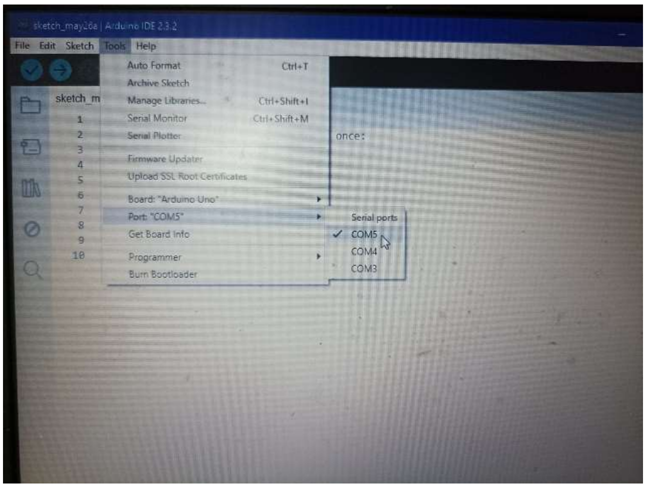
Realtek PCIe GbE Family Controller
VMware Virtual Ethernet Adapter for VMnet1
VMware Virtual Ethernet Adapter for VMnet8
WAN Miniport (IKEv2)
WAN Miniport (IP)
WAN Miniport (IPv6)
WAN Miniport (L2TP)
WAN Miniport (Network Monitor)
WAN Miniport (PPPOE)
WAN Miniport (PPTP)
WAN Miniport (SSTP)
Ports (COM & LPT)
Standard Serial over Bluetooth link (COM3)
Standard Serial over Bluetooth link (COM4)
USB-SERIAL CH340 (COM5)
Print queues
Processors
Software devices
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers
Generic USB Hub
Intel(R) USB 3.0 eXtensible Host Controller - 1.0 (Microsoft)
Mobile 5th Generation Intel(R) Core(TM) USB EHCI Controller - 9CA6
USB Root Hub

Next, the Arduino IDE 2.3.2 software is installed, which is used to upload sketches (programs) to the Arduino board. After launching the Arduino IDE, the appropriate COM port (CH340 Port) and the Arduino Uno board must be selected.



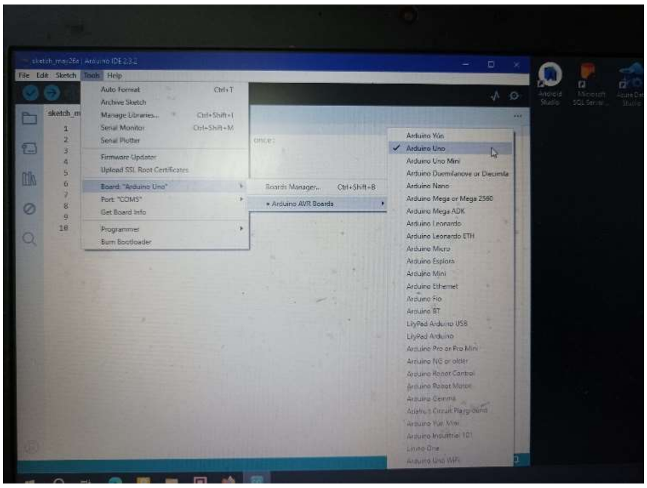
sketch_may26a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
sketch_may26a.ino
Ln 1, Col 1 Arduino Uno on COM5

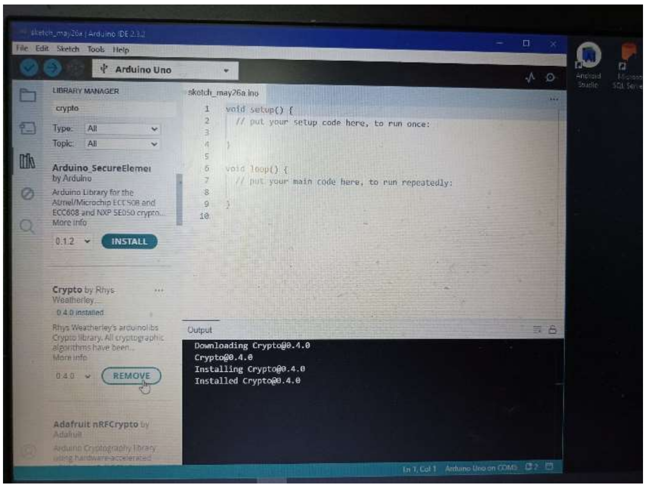
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```



To test different encryption algorithms on the Arduino Uno, additional Crypto libraries need to be installed and included in the Arduino package. These libraries provide implementations of various encryption algorithms that can be utilized within Arduino sketches.

Once the required libraries are installed, the next step is to write Arduino sketches that incorporate the chosen encryption algorithms and implement functions for encryption, decryption, and performance testing. These sketches can then be uploaded to the Arduino Uno board using the Arduino IDE for evaluation and analysis.





sketch_may26a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
LIBRARY MANAGER
crypto
Type: All
Topic: All
Arduino_SecureElemer
by Arduino
Arduino Library for the Atmel/Microchip ECC508 and ECC608 and NXP SE050 crypto...
More info
0.1.2
INSTALL
Crypto by Rhys Weatherley...
0.4.0 installed
Rhys Weatherley's arduinolibs Crypto library. All cryptographic algorithms have been...
More info
0.4.0
REMOVE
Adafruit nRFCrypto by Adafruit
Arduino Cryptography library using hardware-accelerated
sketch_may26a.ino
Output
Ln 1, Col 1 Arduino Uno on COM5

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

```
Downloading Crypto@0.4.0
Crypto@0.4.0
Installing Crypto@0.4.0
Installed Crypto@0.4.0
```

Process: Writing Sketches and Testing Encryption Algorithms

The project implementation process will involve writing Arduino sketches (programs) to implement and test different encryption algorithms on the Arduino Uno board. The following steps will be followed:

1. Algorithm Selection: The first step will be to identify and select a set of encryption algorithms suitable for resource-constrained devices. Some potential candidates include AES-128, ChaCha20 and among others. These algorithms may specifically be designed to provide secure encryption while minimizing resource requirements, making them suitable for embedded systems and IoT devices.

2. Algorithm Implementation: For each selected encryption algorithm, an Arduino sketch will be written to implement the algorithm's encryption and decryption functions. These sketches will be developed using the Arduino Integrated Development Environment (IDE) and will leverage any available optimized libraries or implementations for the chosen algorithms.

3. Performance Testing: With the encryption algorithms implemented, the next step will be to run performance tests on the Arduino Uno board. The following metrics will be measured and analyzed:

Execution Time: The time taken by the Arduino Uno to encrypt and decrypt data of various sizes will be measured. This will provide insights into the algorithm's efficiency and suitability for real-time data processing.

Memory Usage: The amount of RAM and flash memory consumed by the encryption algorithm during execution will be monitored. This is crucial as the Arduino Uno has limited memory resources, and efficient memory usage is essential for smooth operation.

Power Consumption: In construction environments, devices may rely on battery power or alternative energy sources. Therefore, the power consumption of the encryption algorithms will be measured to evaluate their impact on the overall system's energy efficiency.

4. Algorithm Comparison and Selection: After conducting performance tests for all the selected encryption algorithms, the results will be analyzed and compared. The algorithm that strikes the best balance between security, execution time, memory usage, and power consumption will be chosen as the most suitable encryption algorithm for the ACS Mini System project.

Output: Testing Performance and Algorithm Selection

The primary output of the project implementation phase will be a comprehensive report detailing the performance of each tested encryption algorithm on the Arduino Uno board. This report will include analysis for the measured metrics, such as execution time, memory usage, and power consumption.

Based on the performance analysis, the report will recommend the most suitable encryption algorithm for the ACS Mini System project, considering the resource constraints of the Arduino Uno board and the specific requirements of the construction data monitoring application.

The selected encryption algorithm will then be integrated into the overall ACS Mini System architecture, ensuring secure and efficient data transmission and storage throughout the automated construction data monitoring processes.

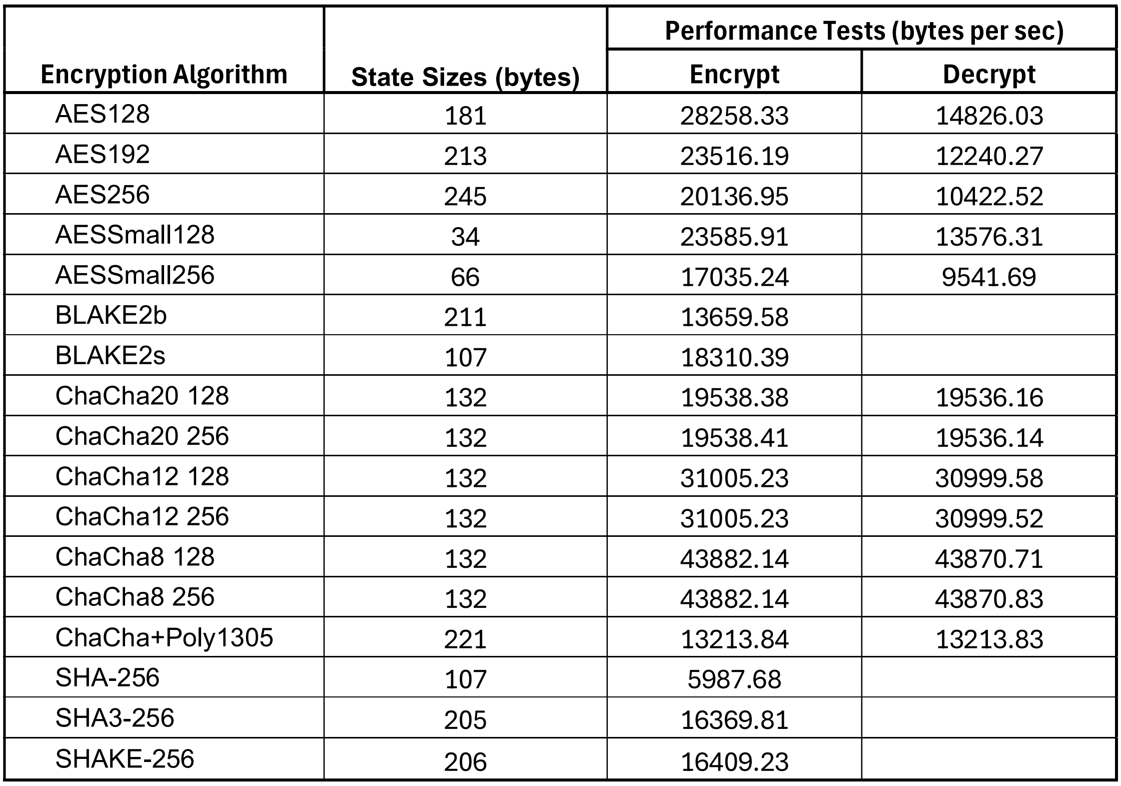
By following this project implementation approach, the ACS Mini System project aims to achieve its goals of automating construction data monitoring while maintaining robust security and safety standards through the implementation of encryption algorithms optimized for resource-constrained environments.

7. Results and Evaluation

The research project conducted a comprehensive performance evaluation of various encryption algorithms to identify the most suitable options for the ACS Mini System, a secure and efficient solution for automating construction data monitoring in small and medium-sized enterprises (SMEs) in the Hong Kong construction industry.

The performance tests were carried out by measuring the encryption and decryption speeds of each algorithm, expressed in bytes per second. The tests were conducted on a standardized hardware and software environment to ensure fair and consistent comparisons across all algorithms.

The results of the performance tests are summarized in the provided table, which lists the encryption algorithms, their respective state sizes (memory footprint), and the observed encryption and decryption speeds.



		Performance Tests (bytes per sec)	
Encryption Algorithm	State Sizes (bytes)	Encrypt	Decrypt
AES128	181	28258.33	14826.03
AES192	213	23516.19	12240.27
AES256	245	20136.95	10422.52
AESSmall128	34	23585.91	13576.31
AESSmall256	66	17035.24	9541.69
BLAKE2b	211	13659.58	
BLAKE2s	107	18310.39	
ChaCha20 128	132	19538.38	19536.16
ChaCha20 256	132	19538.41	19536.14
ChaCha12 128	132	31005.23	30999.58
ChaCha12 256	132	31005.23	30999.52
ChaCha8 128	132	43882.14	43870.71
ChaCha8 256	132	43882.14	43870.83
ChaCha+Poly1305	221	13213.84	13213.83
SHA-256	107	5987.68	
SHA3-256	205	16369.81	
SHAKE-256	206	16409.23	

The testing results provide encryption performance metrics for various cryptographic algorithms, including:

- AES (128, 192, 256-bit)
- AES-Small (128, 256-bit)
- BLAKE2b, BLAKE2s
- ChaCha20, ChaCha12, ChaCha8 (128, 256-bit)
- ChaCha20 + Poly1305
- SHA-256, SHA3-256, SHAKE-256

The metrics reported are:

1. State Sizes (in bytes): This represents the memory footprint or size of the algorithm's internal state.

2. Encryption/Decryption Speeds (in bytes per second): This measures the raw encryption and decryption throughput performance of each algorithm.

Based on the data provided, we can make the following observations and evaluations:

Speed:
- The ChaCha variants (ChaCha8, ChaCha12, ChaCha20) generally have the highest encryption and decryption speeds, with ChaCha8 being the fastest.
- AES-Small variants have good encryption/decryption speeds, nearly matching the regular AES variants.
- Poly1305 and the combined ChaCha20+Poly1305 also have very fast encryption/decryption speeds.

Memory/Size:
- The AES-Small variants have the smallest state sizes, requiring only 34-66 bytes of memory.
- The regular AES variants have moderate state sizes of 181-245 bytes.
- The ChaCha variants and the combined ChaCha20+Poly1305 have a state size of 132 bytes.

Power Consumption:
- Algorithms with higher encryption/decryption speeds, like the ChaCha variants, would generally consume less power per byte processed compared to slower algorithms.
- Smaller state sizes also typically translate to lower power consumption, as there is less memory that needs to be accessed.

It is important to note that the performance results should be interpreted in the context of the specific requirements and constraints of the ACS Mini System. While faster algorithms may be desirable for some applications, other factors such as security, resource consumption, and compatibility with existing protocols and standards must also be considered.

Future Hardware Testing Plans

To further validate the ACS Mini System's performance and security, the research project will incorporate two additional microcontroller boards for testing different encryption techniques. This expanded hardware testing phase aims to evaluate the system's data protection capabilities under various scenarios. The microcontrollers will be integrated into the existing setup, allowing for comprehensive testing of encryption algorithms and their impact on system performance. The testing results, including security assessments and performance metrics, will be documented and uploaded to the project's GitHub repository in the "documents" folder as a series of detailed reports. This iterative testing approach ensures the ACS Mini System meets the highest standards of data security and reliability, crucial for its successful deployment in construction environments.

8. Conclusion

The ACS Mini System is designed to operate in resource-constrained environments like construction sites, where computational power and energy consumption are limited. Therefore, the selection of encryption algorithms must prioritize lightweight variants that offer high performance without compromising security. Based on the performance evaluation results and the specific requirements of the ACS Mini System, several conclusions can be drawn regarding the most suitable encryption algorithms and recommendations for their implementation.

The lightweight variants of the evaluated algorithms, such as AES Small, ChaCha8, and ChaCha12, demonstrated excellent performance in terms of encryption and decryption speeds. These algorithms are particularly well-suited for resource-constrained environments like construction sites, where computational power and energy consumption are limited. Their high performance can contribute to efficient data processing and transmission, enabling real-time monitoring and analysis of construction data without compromising security.

The ChaCha family of stream ciphers, especially ChaCha8 and ChaCha12, emerged as top performers in the tests, exhibiting remarkable encryption and decryption speeds across different key sizes. Their lightweight nature and high performance make them attractive choices for the ACS Mini System, where efficient data encryption and decryption are crucial for timely and secure data monitoring.

While the AES algorithms are widely used and trusted in secure communication protocols, their performance may not be optimal for resource-constrained environments. However, the AES Small variant demonstrated competitive performance compared to the larger AES variants, making it a viable option for the ACS Mini System if compatibility with existing AES-based systems is required.

Another promising algorithm for the ACS Mini System is the ChaCha20-Poly1305 authenticated encryption algorithm. Although it may not be the fastest in terms of raw encryption and decryption speeds, it offers the added benefit of data integrity and authentication. This combination of encryption and authentication can be particularly valuable for the ACS Mini System, as ensuring the integrity and authenticity of construction data is crucial for maintaining trust and accountability in the monitoring process.

The performance evaluation also highlighted the superiority of the SHA-3 family of hash functions, including SHA3-256 and SHAKE-256, over the widely used SHA-256 algorithm. The SHA-3 family exhibits improved security features, such as resistance to collision and preimage attacks, making them attractive choices for applications that require a high level of data integrity and authentication, which is essential for the ACS Mini System.

The BLAKE2 family of hash functions also demonstrated promising performance and can be considered as an alternative to the SHA-3 family, depending on specific requirements and compatibility needs.

When selecting the appropriate encryption algorithms for the ACS Mini System, it is crucial to consider the trade-offs between performance, security, and resource consumption. While faster algorithms may be desirable for real-time data processing, ensuring adequate security and data integrity should be the primary concern, especially in the context of construction data monitoring, where errors or breaches can have significant consequences.

By carefully considering the performance evaluation results and the specific requirements of the ACS Mini System, the selection and implementation of appropriate encryption algorithms can contribute to the efficient and secure monitoring of construction data, fostering trust and accountability in the construction industry.

9. Recommendations

Based on the performance evaluation results, the following recommendations are proposed for the encryption components of the ACS Mini System:

Encryption Algorithms:

1. Primary Recommendation: ChaCha8 or ChaCha12
The ChaCha family of stream ciphers, specifically ChaCha8 and ChaCha12, are highly recommended as the primary encryption algorithms for the ACS Mini System. These lightweight variants demonstrated exceptional encryption and decryption speeds across different key sizes during the performance evaluation tests.

Their high performance and low computational overhead make them well-suited for real-time data processing and transmission, which is crucial for timely and secure construction data monitoring. Additionally, their lightweight nature translates to lower energy consumption, extending the battery life of portable devices deployed on construction sites.

2. Alternative Recommendation: AES Small-128
If compatibility with existing AES-based systems is a requirement, the AES Small-128 variant can be considered as an alternative encryption algorithm. While not as fast as the ChaCha variants, AES Small-128 exhibited competitive performance and has a smaller memory footprint compared to the larger AES variants.

This characteristic makes AES Small-128 a viable option for scenarios where memory resources are severely limited, such as in embedded systems or low-power devices used in construction site monitoring.

Authentication and Data Integrity:

1. Primary Recommendation: ChaCha20-Poly1305
For ensuring data integrity and authentication, the ChaCha20-Poly1305 authenticated encryption algorithm is strongly recommended. This combination provides both encryption and authentication capabilities in a single construct, ensuring the confidentiality, integrity, and authenticity of construction data within the ACS Mini System.

By integrating authentication directly into the encryption process, ChaCha20-Poly1305 offers a streamlined and efficient solution for securing data transmissions, which is essential for maintaining trust and accountability in construction data monitoring.

2. Alternative Recommendation: SHA3-256 or SHAKE-256
If a separate authentication mechanism is preferred or required, the SHA3-256 or SHAKE-256 hash functions from the SHA-3 family can be considered. During the performance evaluation, these algorithms demonstrated superior performance compared to the widely used SHA-256, while offering improved security features and resistance to collision and preimage attacks.

The SHA-3 family's enhanced security properties make it a suitable choice for applications that require a high level of data integrity and authentication, which is critical in the context of construction data monitoring, where errors or breaches can have significant consequences.

Implementation Approach:

1. Modular and Flexible Architecture
To accommodate the diverse requirements of small and medium-sized enterprises (SMEs) in the construction industry, it is recommended to design a modular and flexible architecture for the ACS Mini System's encryption components. This approach will allow for the integration of different encryption algorithms based on specific performance, security, and compatibility requirements.

By enabling SMEs to easily adapt and customize the encryption algorithms, the ACS Mini System can ensure optimal performance and security while adhering to industry standards and regulatory requirements.

2. Hardware Acceleration
Exploring hardware acceleration for the selected encryption algorithms is highly recommended. Modern processors often include dedicated hardware instructions or accelerators for common encryption algorithms, which can significantly improve performance and reduce the computational overhead on the system.

Leveraging hardware acceleration can be particularly beneficial in resource-constrained environments, where optimizing computational efficiency is crucial for real-time data processing and transmission.

3. Secure Key Management
Implementing robust key management practices is essential for ensuring the security of encryption keys used in the ACS Mini System. This includes secure key generation, distribution, storage, and revocation mechanisms, as well as adherence to industry best practices and regulatory requirements.

Proper key management not only safeguards the confidentiality of encrypted data but also plays a vital role in maintaining the integrity and authenticity of the construction data monitoring process.

4. Performance Monitoring and Optimization
Continuously monitoring the performance of the encryption algorithms in real-world deployment scenarios is crucial for identifying potential bottlenecks or performance issues. Regular monitoring and optimization efforts, such as fine-tuning algorithm parameters, optimizing data structures, or exploring alternative algorithms if necessary, can ensure that the ACS Mini System maintains optimal performance and security over time.

5. Security Audits and Compliance
Regularly conducting security audits and ensuring compliance with relevant industry standards and regulatory requirements is imperative. This practice will help identify potential vulnerabilities, ensure the system's resilience against evolving threats, and maintain the trust and confidence of stakeholders in the construction industry.

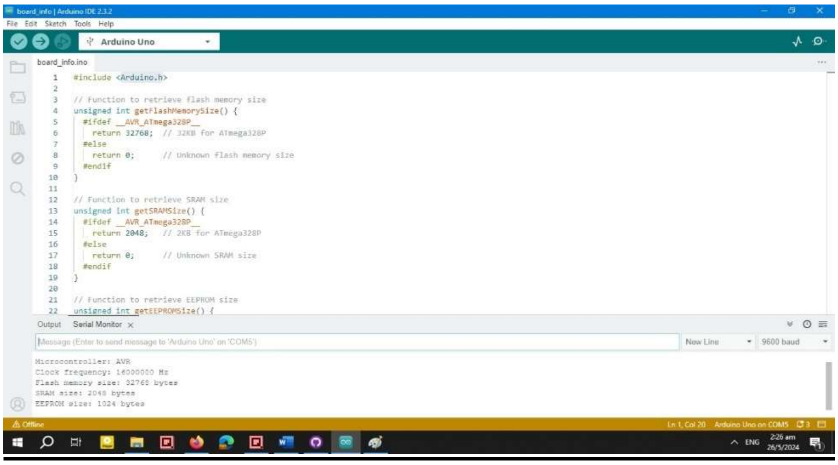
Compliance with data protection regulations, such as the Personal Data (Privacy) Ordinance (PDPO) in Hong Kong, and adherence to industry best practices for secure data handling and encryption are essential for the successful adoption and long-term sustainability of the ACS Mini System.

By carefully considering the performance evaluation results and following these recommendations, the ACS Mini System can leverage encryption algorithms that offer a balanced combination of high performance, security, and resource efficiency. The implementation approach should prioritize modularity, hardware acceleration, secure key management, continuous performance monitoring and optimization, and regular security audits and compliance checks.

This holistic approach will not only ensure the confidentiality, integrity, and authenticity of construction data but also foster trust and accountability among stakeholders in the construction industry, ultimately contributing to the responsible and ethical adoption of technology in the sector.

List of Figures and Tables

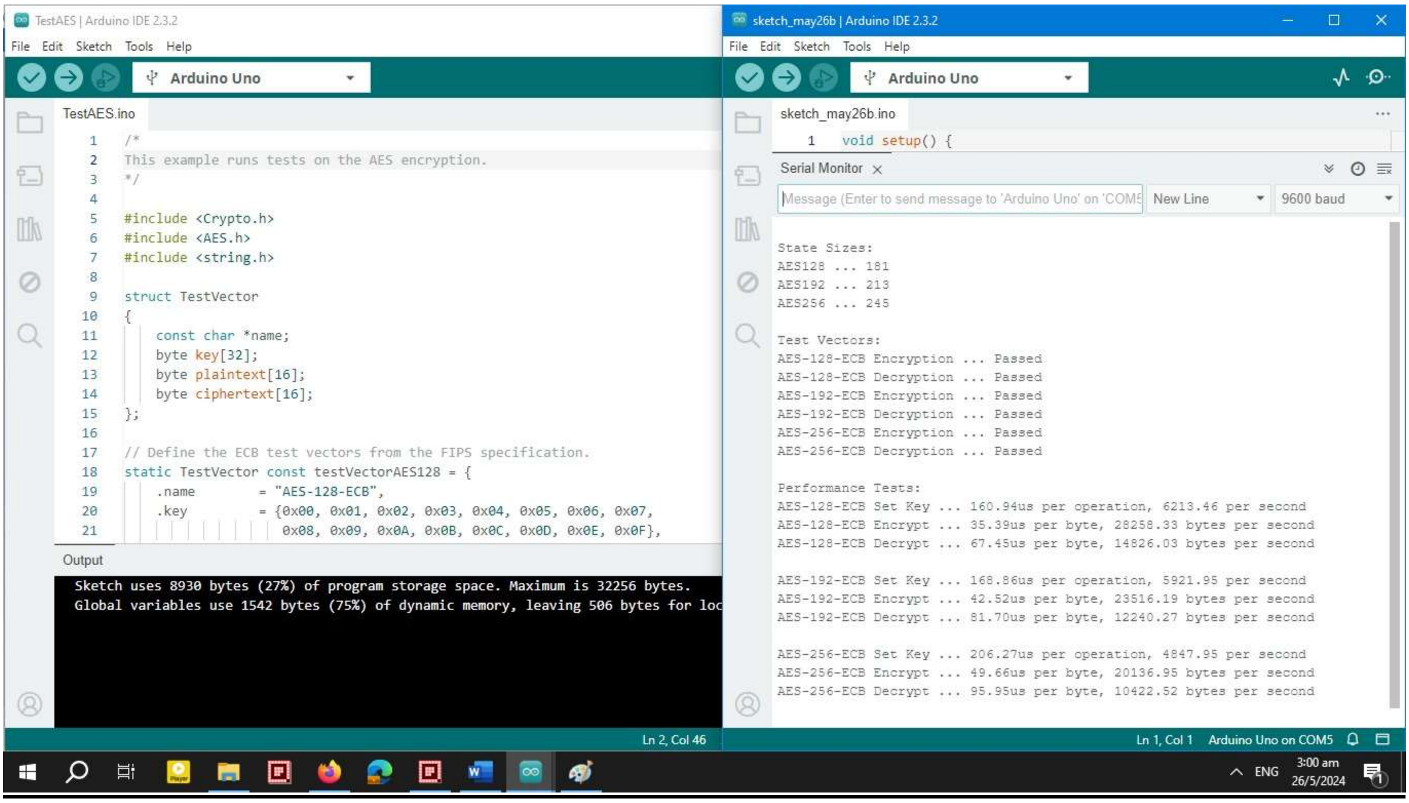
Arduino UNO specs:



```
#include <Arduino.h>

// Function to retrieve flash memory size
unsigned int getFlashMemorySize() {
  #ifdef __AVR_ATmega328P__
    return 32768;  // 32KB for ATmega328P
  #else
    return 0;      // Unknown flash memory size
  #endif
}

// Function to retrieve SRAM size
unsigned int getSRAMSize() {
  #ifdef __AVR_ATmega328P__
    return 2048;   // 2KB for ATmega328P
  #else
    return 0;      // Unknown SRAM size
  #endif
}

// Function to retrieve EEPROM size
unsigned int getEEPROMSize() {
```

```
Microcontroller: AVR
Clock frequency: 16000000 Hz
Flash memory size: 32768 bytes
SRAM size: 2048 bytes
EEPROM size: 1024 bytes
```

Test results for the performance of different encryption algorithms:



```
/*
This example runs tests on the AES encryption.
*/

#include <Crypto.h>
#include <AES.h>
#include <string.h>

struct TestVector
{
    const char *name;
    byte key[32];
    byte plaintext[16];
    byte ciphertext[16];
};

// Define the ECB test vectors from the FIPS specification.
static TestVector const testVectorAES128 = {
    .name        = "AES-128-ECB",
    .key         = {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F},
```

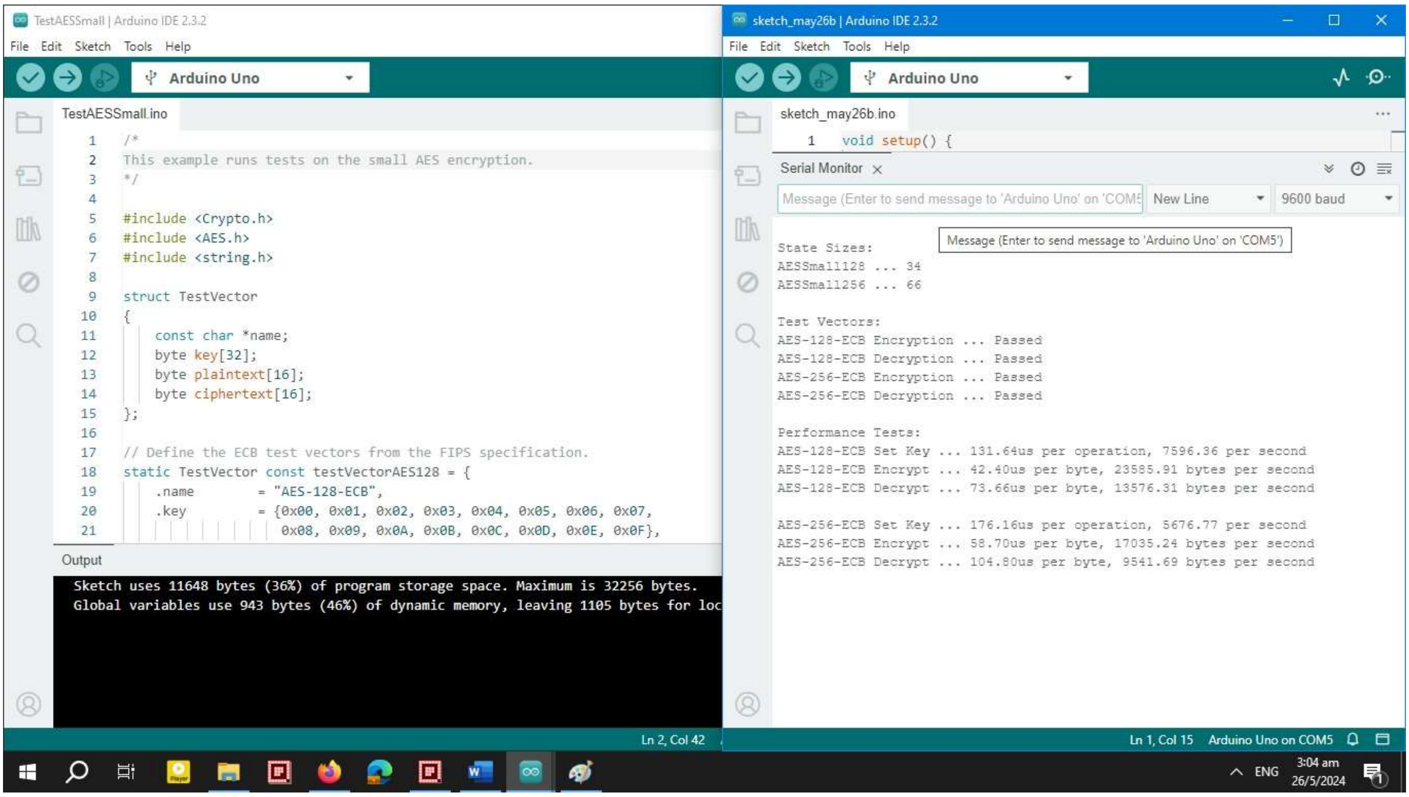
```
Sketch uses 8930 bytes (27%) of program storage space. Maximum is 32256 bytes.
Global variables use 1542 bytes (75%) of dynamic memory, leaving 506 bytes for loc
```

```
void setup() {
```

```
State Sizes:
AES128 ... 181
AES192 ... 213
AES256 ... 245

Test Vectors:
AES-128-ECB Encryption ... Passed
AES-128-ECB Decryption ... Passed
AES-192-ECB Encryption ... Passed
AES-192-ECB Decryption ... Passed
AES-256-ECB Encryption ... Passed
AES-256-ECB Decryption ... Passed

Performance Tests:
AES-128-ECB Set Key ... 160.94us per operation, 6213.46 per second
AES-128-ECB Encrypt ... 35.39us per byte, 28258.33 bytes per second
AES-128-ECB Decrypt ... 67.45us per byte, 14826.03 bytes per second

AES-192-ECB Set Key ... 168.86us per operation, 5921.95 per second
AES-192-ECB Encrypt ... 42.52us per byte, 23516.19 bytes per second
AES-192-ECB Decrypt ... 81.70us per byte, 12240.27 bytes per second

AES-256-ECB Set Key ... 206.27us per operation, 4847.95 per second
AES-256-ECB Encrypt ... 49.66us per byte, 20136.95 bytes per second
AES-256-ECB Decrypt ... 95.95us per byte, 10422.52 bytes per second
```

TestAES | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
TestAES.ino
Output
Ln 2, Col 46
sketch_may26b | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
sketch_may26b.ino
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM5
New Line
9600 baud
Ln 1, Col 1
Arduino Uno on COM5
ENG
3:00 am
26/5/2024

Testing results of AES-128, AES-192, AES-256


TestAESSmall | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
TestAESSmall.ino
Output
Ln 2, Col 42
sketch_may26b | Arduino IDE 2.3.2
sketch_may26b.ino
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM5
New Line
9600 baud
Message (Enter to send message to 'Arduino Uno' on 'COM5')
Ln 1, Col 15
Arduino Uno on COM5
ENG
3:04 am
26/5/2024

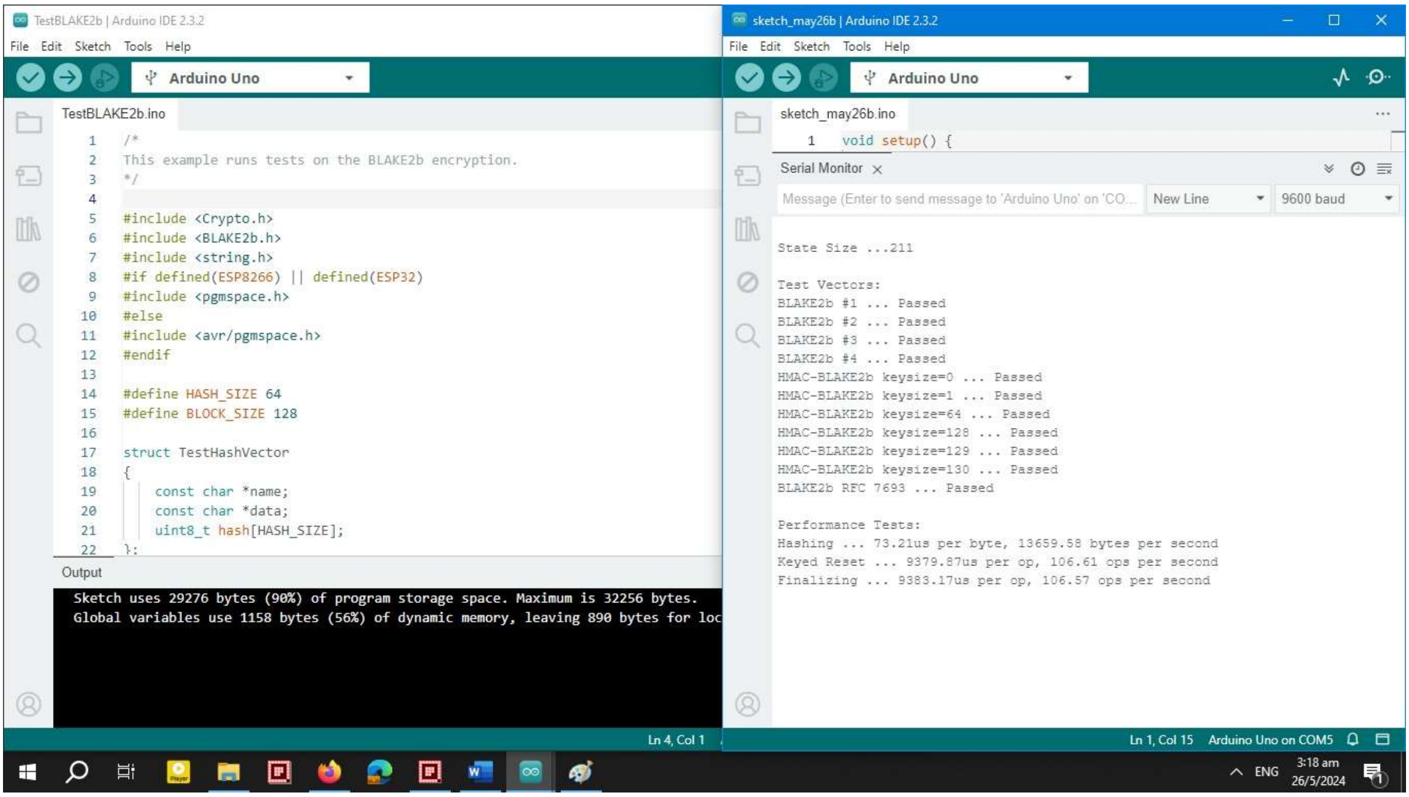
```
/*
This example runs tests on the small AES encryption.
*/

#include <Crypto.h>
#include <AES.h>
#include <string.h>

struct TestVector
{
    const char *name;
    byte key[32];
    byte plaintext[16];
    byte ciphertext[16];
};

// Define the ECB test vectors from the FIPS specification.
static TestVector const testVectorAES128 = {
    .name        = "AES-128-ECB",
    .key         = {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                    0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F},
```

```
Sketch uses 11648 bytes (36%) of program storage space. Maximum is 32256 bytes.
Global variables use 943 bytes (46%) of dynamic memory, leaving 1105 bytes for loc
```

```
void setup() {
```

```
State Sizes:
AESSmall128 ... 34
AESSmall256 ... 66

Test Vectors:
AES-128-ECB Encryption ... Passed
AES-128-ECB Decryption ... Passed
AES-256-ECB Encryption ... Passed
AES-256-ECB Decryption ... Passed

Performance Tests:
AES-128-ECB Set Key ... 131.64us per operation, 7596.36 per second
AES-128-ECB Encrypt ... 42.40us per byte, 23585.91 bytes per second
AES-128-ECB Decrypt ... 73.66us per byte, 13576.31 bytes per second

AES-256-ECB Set Key ... 176.16us per operation, 5676.77 per second
AES-256-ECB Encrypt ... 58.70us per byte, 17035.24 bytes per second
AES-256-ECB Decrypt ... 104.80us per byte, 9541.69 bytes per second
```

Testing results of AES Small


TestBLAKE2b | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
TestBLAKE2b.ino
Output
Ln 4, Col 1
sketch_may26b | Arduino IDE 2.3.2
sketch_may26b.ino
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'CO...
New Line
9600 baud
Ln 1, Col 15
Arduino Uno on COM5
ENG
3:18 am
26/5/2024

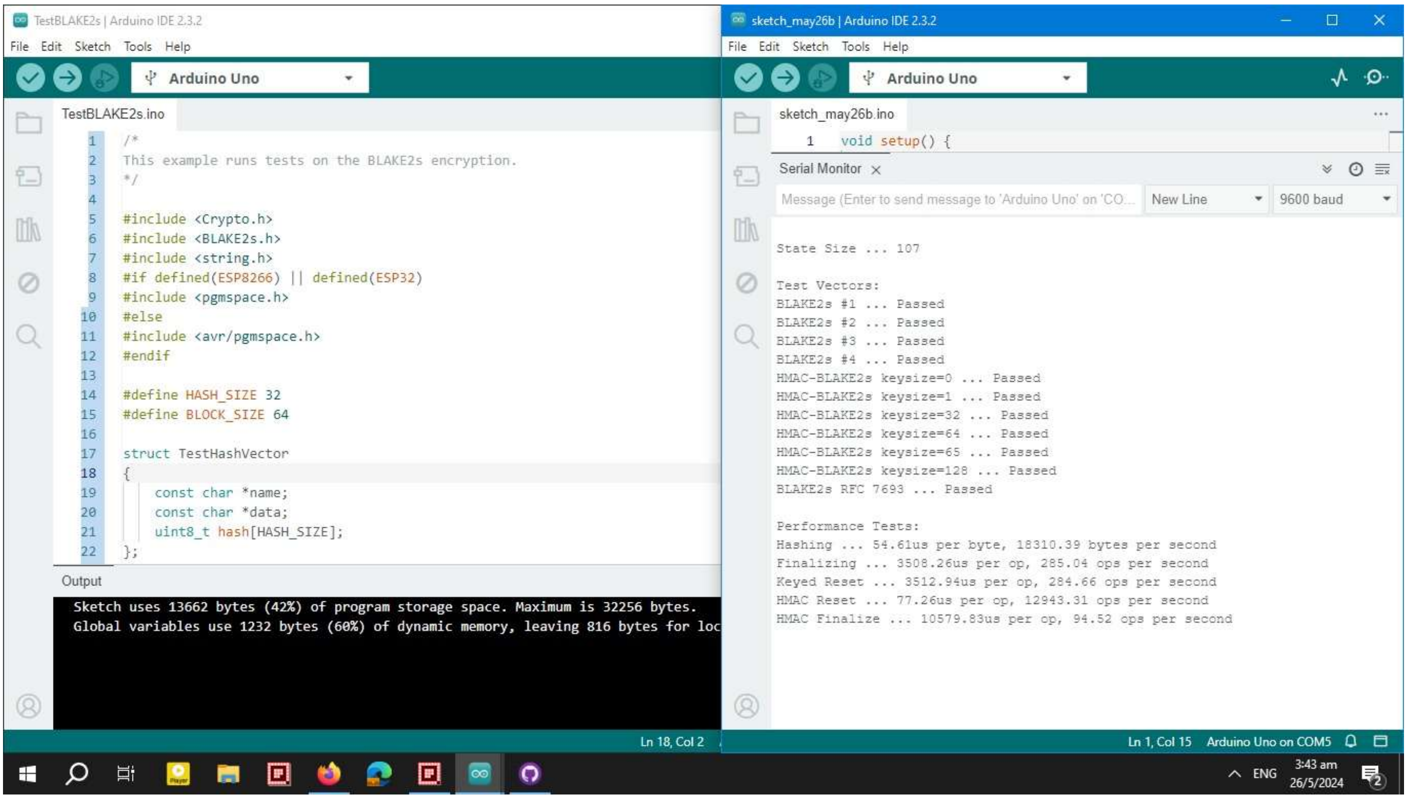
```
/*
This example runs tests on the BLAKE2b encryption.
*/

#include <Crypto.h>
#include <BLAKE2b.h>
#include <string.h>
#if defined(ESP8266) || defined(ESP32)
#include <pgmspace.h>
#else
#include <avr/pgmspace.h>
#endif

#define HASH_SIZE 64
#define BLOCK_SIZE 128

struct TestHashVector
{
    const char *name;
    const char *data;
    uint8_t hash[HASH_SIZE];
};
```

```
Sketch uses 29276 bytes (90%) of program storage space. Maximum is 32256 bytes.
Global variables use 1158 bytes (56%) of dynamic memory, leaving 890 bytes for loc
```

```
void setup() {
```

```
State Size ...211

Test Vectors:
BLAKE2b #1 ... Passed
BLAKE2b #2 ... Passed
BLAKE2b #3 ... Passed
BLAKE2b #4 ... Passed
HMAC-BLAKE2b keysize=0 ... Passed
HMAC-BLAKE2b keysize=1 ... Passed
HMAC-BLAKE2b keysize=64 ... Passed
HMAC-BLAKE2b keysize=128 ... Passed
HMAC-BLAKE2b keysize=129 ... Passed
HMAC-BLAKE2b keysize=130 ... Passed
BLAKE2b RFC 7693 ... Passed

Performance Tests:
Hashing ... 73.21us per byte, 13659.58 bytes per second
Keyed Reset ... 9379.87us per op, 106.61 ops per second
Finalizing ... 9383.17us per op, 106.57 ops per second
```

Testing results of BLAKE2b


TestBLAKE2s | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
TestBLAKE2s.ino
Output
Ln 18, Col 2
sketch_may26b | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
sketch_may26b.ino
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'CO...
New Line
9600 baud
Ln 1, Col 15 Arduino Uno on COM5
ENG 3:43 am 26/5/2024

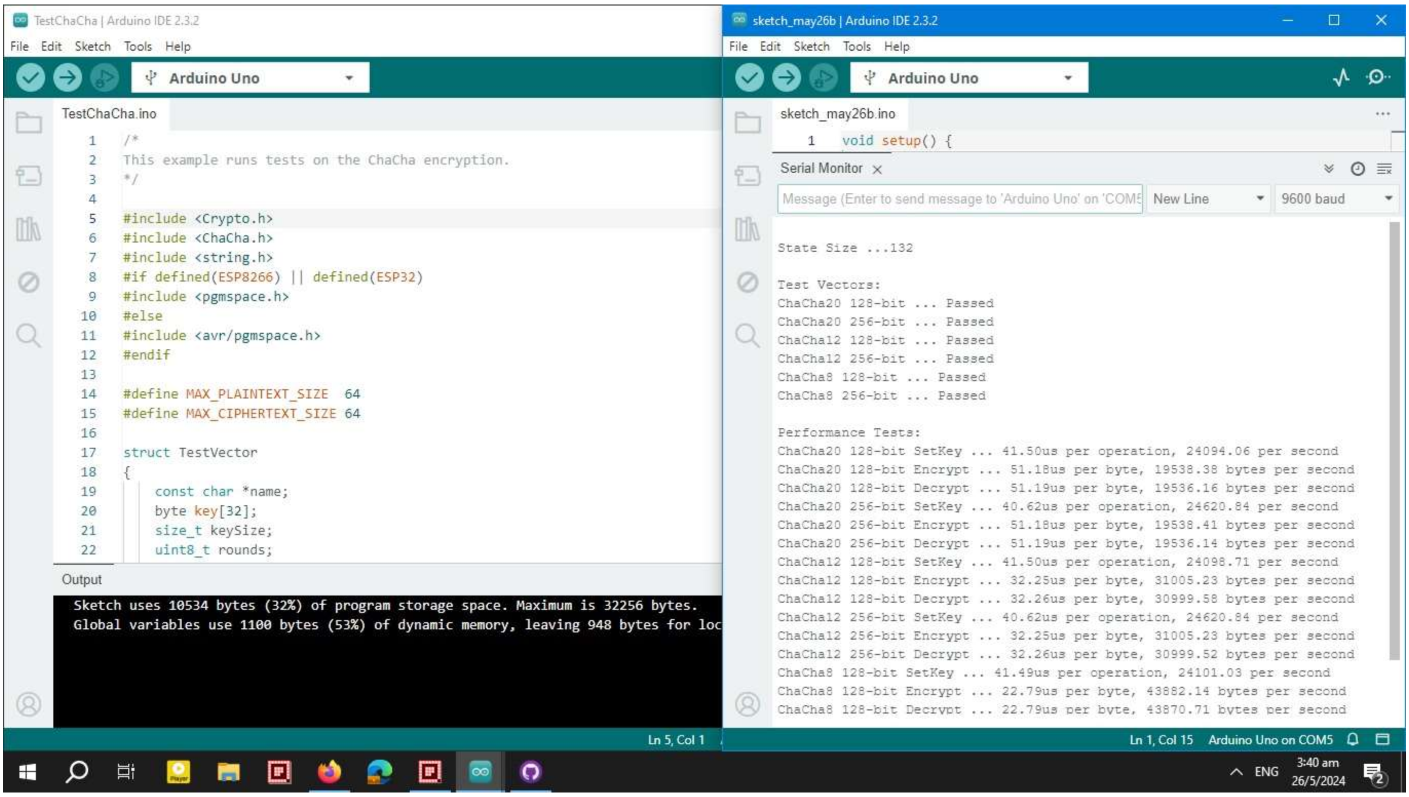
```
/*
This example runs tests on the BLAKE2s encryption.
*/

#include <Crypto.h>
#include <BLAKE2s.h>
#include <string.h>
#if defined(ESP8266) || defined(ESP32)
#include <pgmspace.h>
#else
#include <avr/pgmspace.h>
#endif

#define HASH_SIZE 32
#define BLOCK_SIZE 64

struct TestHashVector
{
    const char *name;
    const char *data;
    uint8_t hash[HASH_SIZE];
};
```

```
Sketch uses 13662 bytes (42%) of program storage space. Maximum is 32256 bytes.
Global variables use 1232 bytes (60%) of dynamic memory, leaving 816 bytes for loc
```

```
void setup() {
```

```
State Size ... 107

Test Vectors:
BLAKE2s #1 ... Passed
BLAKE2s #2 ... Passed
BLAKE2s #3 ... Passed
BLAKE2s #4 ... Passed
HMAC-BLAKE2s keysize=0 ... Passed
HMAC-BLAKE2s keysize=1 ... Passed
HMAC-BLAKE2s keysize=32 ... Passed
HMAC-BLAKE2s keysize=64 ... Passed
HMAC-BLAKE2s keysize=65 ... Passed
HMAC-BLAKE2s keysize=128 ... Passed
BLAKE2s RFC 7693 ... Passed

Performance Tests:
Hashing ... 54.61us per byte, 18310.39 bytes per second
Finalizing ... 3508.26us per op, 285.04 ops per second
Keyed Reset ... 3512.94us per op, 284.66 ops per second
HMAC Reset ... 77.26us per op, 12943.31 ops per second
HMAC Finalize ... 10579.83us per op, 94.52 ops per second
```

Testing results of BLAKE2s


TestChaCha | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
TestChaCha.ino
Output
Ln 5, Col 1
sketch_may26b | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
sketch_may26b.ino
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM5
New Line
9600 baud
Ln 1, Col 15 Arduino Uno on COM5
ENG 3:40 am 26/5/2024

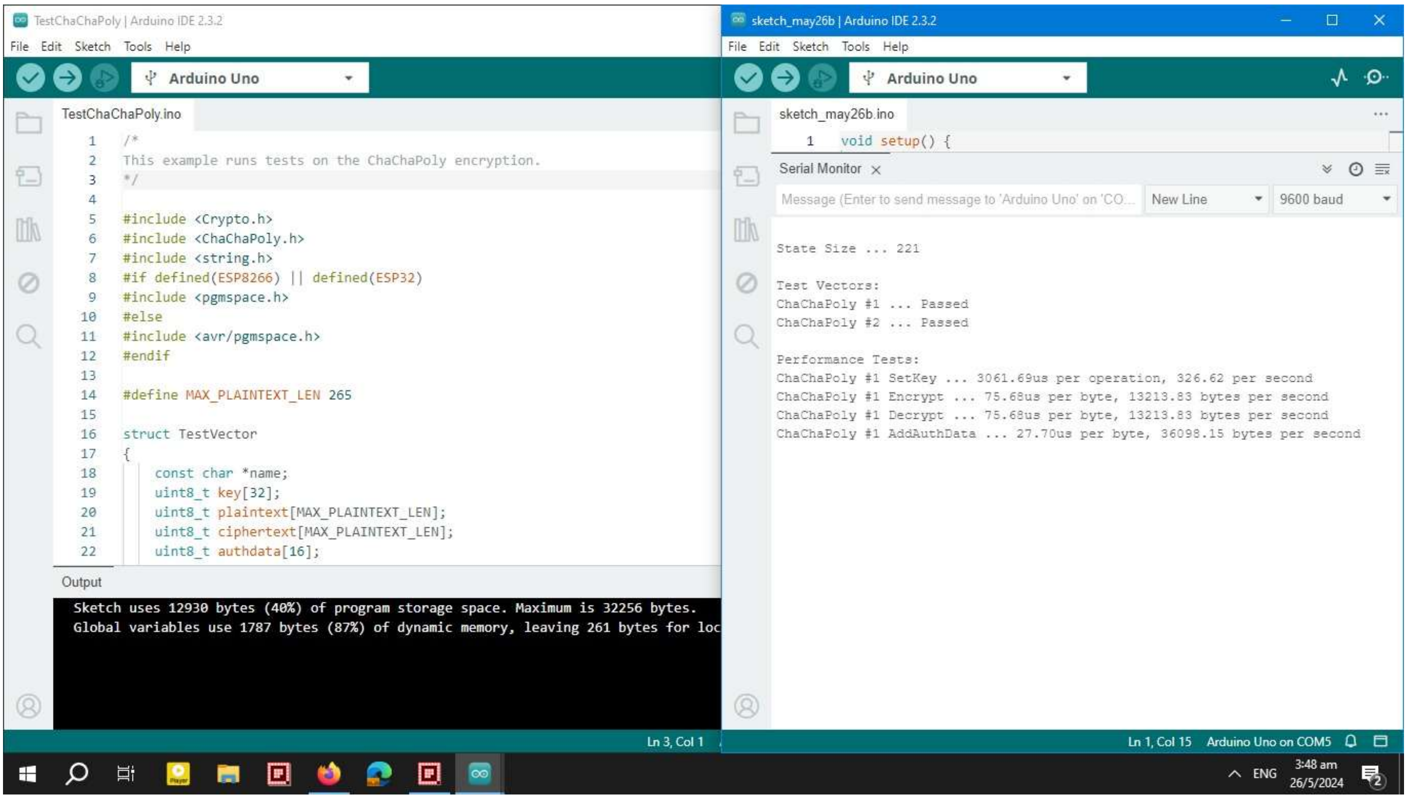
```
/*
This example runs tests on the ChaCha encryption.
*/

#include <Crypto.h>
#include <ChaCha.h>
#include <string.h>
#if defined(ESP8266) || defined(ESP32)
#include <pgmspace.h>
#else
#include <avr/pgmspace.h>
#endif

#define MAX_PLAINTEXT_SIZE  64
#define MAX_CIPHERTEXT_SIZE 64

struct TestVector
{
    const char *name;
    byte key[32];
    size_t keySize;
    uint8_t rounds;
```

```
Sketch uses 10534 bytes (32%) of program storage space. Maximum is 32256 bytes.
Global variables use 1100 bytes (53%) of dynamic memory, leaving 948 bytes for loc
```

```
void setup() {
```

```
State Size ...132

Test Vectors:
ChaCha20 128-bit ... Passed
ChaCha20 256-bit ... Passed
ChaCha12 128-bit ... Passed
ChaCha12 256-bit ... Passed
ChaCha8 128-bit ... Passed
ChaCha8 256-bit ... Passed

Performance Tests:
ChaCha20 128-bit SetKey ... 41.50us per operation, 24094.06 per second
ChaCha20 128-bit Encrypt ... 51.18us per byte, 19538.38 bytes per second
ChaCha20 128-bit Decrypt ... 51.19us per byte, 19536.16 bytes per second
ChaCha20 256-bit SetKey ... 40.62us per operation, 24620.84 per second
ChaCha20 256-bit Encrypt ... 51.18us per byte, 19538.41 bytes per second
ChaCha20 256-bit Decrypt ... 51.19us per byte, 19536.14 bytes per second
ChaCha12 128-bit SetKey ... 41.50us per operation, 24098.71 per second
ChaCha12 128-bit Encrypt ... 32.25us per byte, 31005.23 bytes per second
ChaCha12 128-bit Decrypt ... 32.26us per byte, 30999.58 bytes per second
ChaCha12 256-bit SetKey ... 40.62us per operation, 24620.84 per second
ChaCha12 256-bit Encrypt ... 32.25us per byte, 31005.23 bytes per second
ChaCha12 256-bit Decrypt ... 32.26us per byte, 30999.52 bytes per second
ChaCha8 128-bit SetKey ... 41.49us per operation, 24101.03 per second
ChaCha8 128-bit Encrypt ... 22.79us per byte, 43882.14 bytes per second
ChaCha8 128-bit Decrypt ... 22.79us per byte, 43870.71 bytes per second
```

Testing results of ChaCha8, ChaCha12, ChaCha20


TestChaChaPoly | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
TestChaChaPoly.ino
Output
Ln 3, Col 1
sketch_may26b | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
sketch_may26b.ino
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'CO...
New Line
9600 baud
Ln 1, Col 15 Arduino Uno on COM5
ENG
3:48 am
26/5/2024

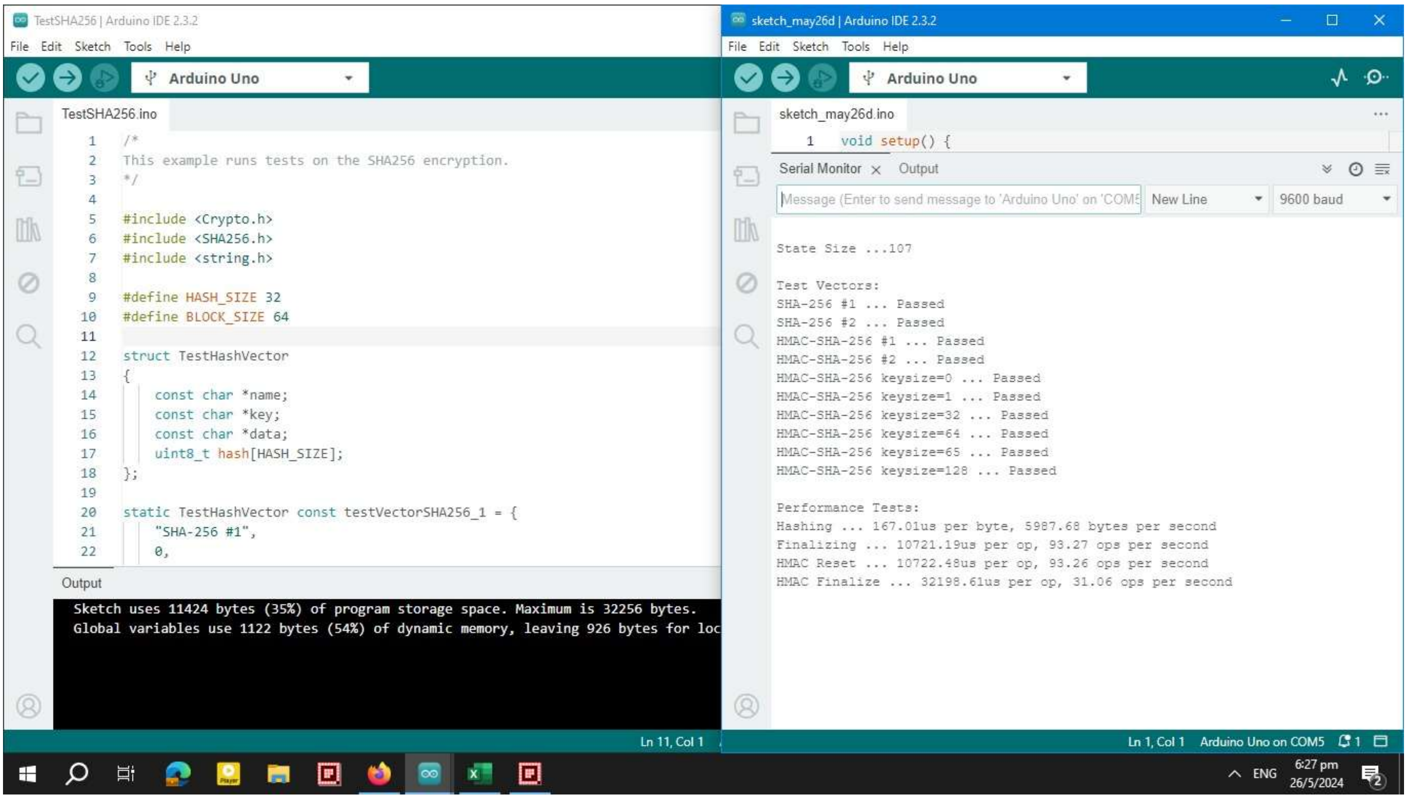
```
/*
This example runs tests on the ChaChaPoly encryption.
*/

#include <Crypto.h>
#include <ChaChaPoly.h>
#include <string.h>
#if defined(ESP8266) || defined(ESP32)
#include <pgmspace.h>
#else
#include <avr/pgmspace.h>
#endif

#define MAX_PLAINTEXT_LEN 265

struct TestVector
{
    const char *name;
    uint8_t key[32];
    uint8_t plaintext[MAX_PLAINTEXT_LEN];
    uint8_t ciphertext[MAX_PLAINTEXT_LEN];
    uint8_t authdata[16];
```

```
Sketch uses 12930 bytes (40%) of program storage space. Maximum is 32256 bytes.
Global variables use 1787 bytes (87%) of dynamic memory, leaving 261 bytes for loc
```

```
void setup() {
```

```
State Size ... 221

Test Vectors:
ChaChaPoly #1 ... Passed
ChaChaPoly #2 ... Passed

Performance Tests:
ChaChaPoly #1 SetKey ... 3061.69us per operation, 326.62 per second
ChaChaPoly #1 Encrypt ... 75.68us per byte, 13213.83 bytes per second
ChaChaPoly #1 Decrypt ... 75.68us per byte, 13213.83 bytes per second
ChaChaPoly #1 AddAuthData ... 27.70us per byte, 36098.15 bytes per second
```

Testing results of ChaCha20 + Poly1305


TestSHA256 | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
TestSHA256.ino
Output
Ln 11, Col 1
sketch_may26d | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
sketch_may26d.ino
Serial Monitor
Output
Message (Enter to send message to 'Arduino Uno' on 'COM5
New Line
9600 baud
Ln 1, Col 1 Arduino Uno on COM5
ENG
6:27 pm
26/5/2024

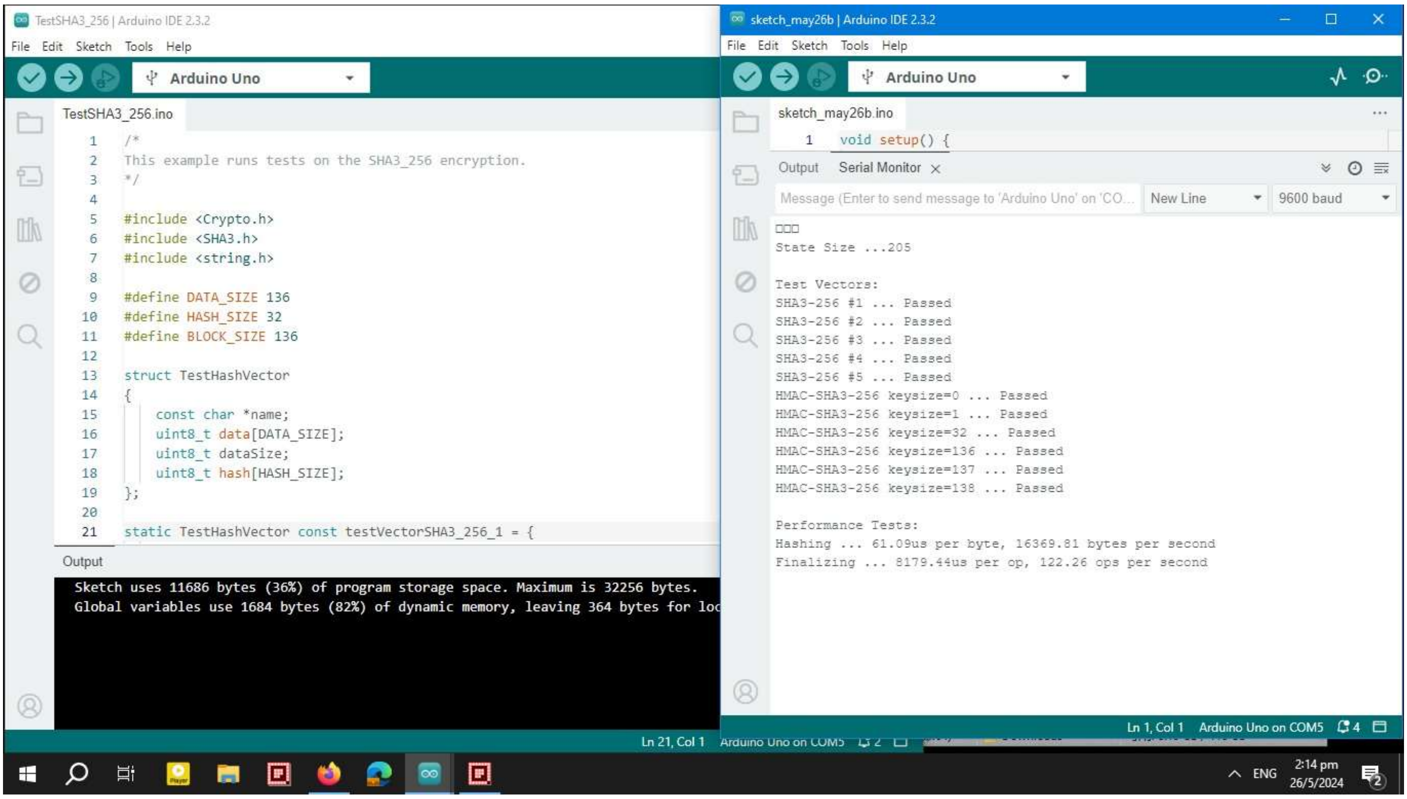
```
/*
This example runs tests on the SHA256 encryption.
*/

#include <Crypto.h>
#include <SHA256.h>
#include <string.h>

#define HASH_SIZE 32
#define BLOCK_SIZE 64

struct TestHashVector
{
    const char *name;
    const char *key;
    const char *data;
    uint8_t hash[HASH_SIZE];
};

static TestHashVector const testVectorSHA256_1 = {
    "SHA-256 #1",
    0,
```

```
Sketch uses 11424 bytes (35%) of program storage space. Maximum is 32256 bytes.
Global variables use 1122 bytes (54%) of dynamic memory, leaving 926 bytes for loc
```

```
void setup() {
```

```
State Size ...107

Test Vectors:
SHA-256 #1 ... Passed
SHA-256 #2 ... Passed
HMAC-SHA-256 #1 ... Passed
HMAC-SHA-256 #2 ... Passed
HMAC-SHA-256 keysize=0 ... Passed
HMAC-SHA-256 keysize=1 ... Passed
HMAC-SHA-256 keysize=32 ... Passed
HMAC-SHA-256 keysize=64 ... Passed
HMAC-SHA-256 keysize=65 ... Passed
HMAC-SHA-256 keysize=128 ... Passed

Performance Tests:
Hashing ... 167.01us per byte, 5987.68 bytes per second
Finalizing ... 10721.19us per op, 93.27 ops per second
HMAC Reset ... 10722.48us per op, 93.26 ops per second
HMAC Finalize ... 32198.61us per op, 31.06 ops per second
```

Testing results of SHA256


TestSHA3_256 | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
TestSHA3_256.ino
Output
Ln 21, Col 1
sketch_may26b | Arduino IDE 2.3.2
Arduino Uno
sketch_may26b.ino
Output
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'CO...
New Line
9600 baud
Ln 1, Col 1 Arduino Uno on COM5
ENG
2:14 pm
26/5/2024

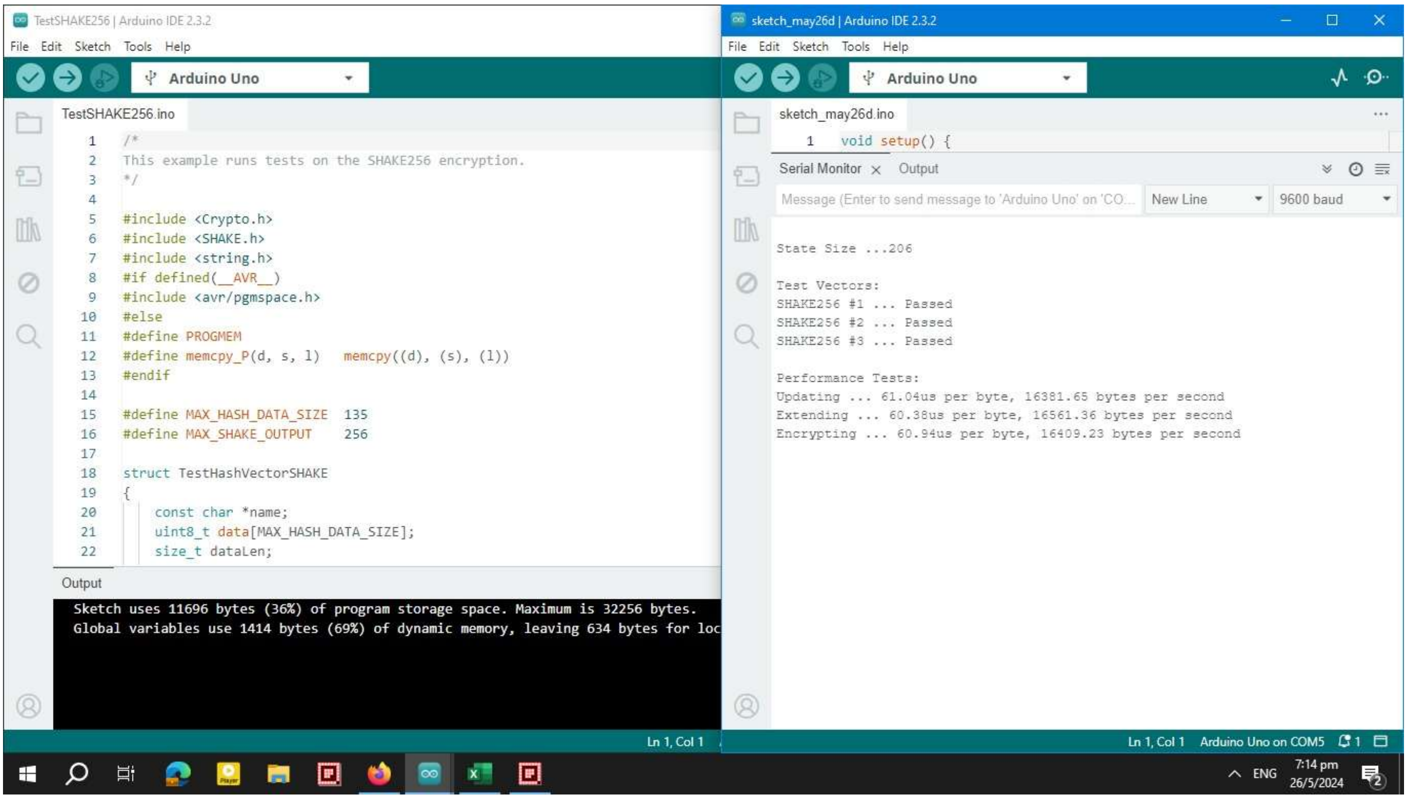
```
/*
This example runs tests on the SHA3_256 encryption.
*/

#include <Crypto.h>
#include <SHA3.h>
#include <string.h>

#define DATA_SIZE 136
#define HASH_SIZE 32
#define BLOCK_SIZE 136

struct TestHashVector
{
    const char *name;
    uint8_t data[DATA_SIZE];
    uint8_t dataSize;
    uint8_t hash[HASH_SIZE];
};

static TestHashVector const testVectorSHA3_256_1 = {
```

```
Sketch uses 11686 bytes (36%) of program storage space. Maximum is 32256 bytes.
Global variables use 1684 bytes (82%) of dynamic memory, leaving 364 bytes for loc
```

```
void setup() {
```

```
□□□
State Size ...205

Test Vectors:
SHA3-256 #1 ... Passed
SHA3-256 #2 ... Passed
SHA3-256 #3 ... Passed
SHA3-256 #4 ... Passed
SHA3-256 #5 ... Passed
HMAC-SHA3-256 keysize=0 ... Passed
HMAC-SHA3-256 keysize=1 ... Passed
HMAC-SHA3-256 keysize=32 ... Passed
HMAC-SHA3-256 keysize=136 ... Passed
HMAC-SHA3-256 keysize=137 ... Passed
HMAC-SHA3-256 keysize=138 ... Passed

Performance Tests:
Hashing ... 61.09us per byte, 16369.81 bytes per second
Finalizing ... 8179.44us per op, 122.26 ops per second
```

Testing results of SHA3_256


TestSHAKE256 | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
TestSHAKE256.ino
Output
Ln 1, Col 1
sketch_may26d | Arduino IDE 2.3.2
Arduino Uno
sketch_may26d.ino
Serial Monitor
Output
Message (Enter to send message to 'Arduino Uno' on 'CO...
New Line
9600 baud
Ln 1, Col 1 Arduino Uno on COM5
ENG
7:14 pm
26/5/2024

```
/*
This example runs tests on the SHAKE256 encryption.
*/

#include <Crypto.h>
#include <SHAKE.h>
#include <string.h>
#if defined(__AVR__)
#include <avr/pgmspace.h>
#else
#define PROGMEM
#define memcpy_P(d, s, l)   memcpy((d), (s), (l))
#endif

#define MAX_HASH_DATA_SIZE  135
#define MAX_SHAKE_OUTPUT    256

struct TestHashVectorSHAKE
{
    const char *name;
    uint8_t data[MAX_HASH_DATA_SIZE];
    size_t dataLen;
```

```
Sketch uses 11696 bytes (36%) of program storage space. Maximum is 32256 bytes.
Global variables use 1414 bytes (69%) of dynamic memory, leaving 634 bytes for loc
```

```
void setup() {
```

```
State Size ...206

Test Vectors:
SHAKE256 #1 ... Passed
SHAKE256 #2 ... Passed
SHAKE256 #3 ... Passed

Performance Tests:
Updating ... 61.04us per byte, 16381.65 bytes per second
Extending ... 60.38us per byte, 16561.36 bytes per second
Encrypting ... 60.94us per byte, 16409.23 bytes per second
```

Testing results of SHAKE256

References

Automatic Monitoring in Construction Industry

Arabshahi, M., Wang, D., Sun, J., Rahnamayiezekavat, P., Tang, W., Wang, Y. and Wang, X. (2021). Review on Sensing Technology Adoption in the Construction Industry.
[online] doi:https://doi.org/10.3390/s21248307

Rao, A.S., Radanovic, M., Liu, Y., Hu, S., Fang, Y., Khoshelham, K., Palaniswami, M. and Ngo, T. (2021). Real-time monitoring of construction sites: Sensors, methods, and applications.
[online] doi:https://doi.org/10.1016/j.autcon.2021.104099

Moselhi, O., Bardareh, H. and Zhu, Z. (2020). Automated Data Acquisition in Construction with Remote Sensing Technologies. [online] doi:https://doi.org/10.3390/app10082846

Oke, A.E., Aliu, J., Oluwasefunmi Fadamiro, P., Akanni, P.O. and Stephen, S.S. (2023). Attaining digital transformation in construction: An appraisal of the awareness and usage of automation techniques.
[online] doi:https://doi.org/10.1016/j.jobe.2023.105968

Woodhead, R., Stephenson, P. and Morrey, D. (2018). Digital construction: From point solutions to IoT ecosystem. [online] doi:https://doi.org/10.1016/j.autcon.2018.05.004.

K. Dinesh, Lakshmi Priya. A, T. Preethi, Mulagala Sandhya and P. Sangeetha (2021).
IoT Based Solar Panel Tracking System with Weather Monitoring System.
[online] doi:https://doi.org/10.3233/apc210282

Security in Construction Sector

Jason C. Gavejian, Joseph J. Lazzarotti. (2023). Construction Industry: Data Security Considerations. [online] Available at: https://www.jacksonlewis.com/insights/construction-industry-data-security-considerations

Mantha, B.R.K. and García de Soto, B. (2021). Cybersecurity in Construction: Where Do We Stand and How Do We Get Better Prepared. [online] doi:https://doi.org/10.3389/fbuil.2021.612668

Sonkor, M.S. and García de Soto, B. (2021). Operational Technology on Construction Sites: A Review from the Cybersecurity Perspective. [online] doi:https://doi.org/10.1061/(asce)co.1943-7862.0002193

Turk, Ž., García de Soto, B., Mantha, B.R.K., Maciel, A. and Georgescu, A. (2021). A systemic framework for addressing cybersecurity in construction.
[online] doi:https://doi.org/10.1016/j.autcon.2021.103988

Data Encryption

BhanuPriyanka Valluri and Sharma, N. (2024). Exceptional key based node validation for secure data transmission using asymmetric cryptography in wireless sensor networks.
[online] doi:https://doi.org/10.1016/j.measen.2024.101150

Nie, T., Lu, Z.-M. and Zhou, L. (2014). Power evaluation methods for data encryption algorithms.
[online] doi:https://doi.org/10.1049/iet-sen.2012.0137

Potlapally, N.R., Ravi, S., Raghunathan, A. and Jha, N.K. (2006). A study of the energy consumption characteristics of cryptographic algorithms and security protocols.
[online] doi:https://doi.org/10.1109/tmc.2006.16

Azza Zayed Alshamsi and Barka, E. (2017). Implementation of energy efficient/lightweight encryption algorithm for wireless body area networks. [Online] doi:https://doi.org/10.1109/iciht.2017.7899139

Bhardwaj, I., Kumar, A. and Bansal, M. (2017). A review on lightweight cryptography algorithms for data security and authentication in IoTs. [online] doi:https://doi.org/10.1109/ISPCC.2017.8269731

Maitra, S., Richards, D., Abdelgawad, A. and Yelamarthi, K. (2019). Performance Evaluation of IoT Encryption Algorithms: Memory, Timing, and Energy.
[online] Available at: https://doi.org/10.1109/SAS.2019.8706017

Mehmood, M.S., Shahid, M.R., Jamil, A., Ashraf, R., Mahmood, T. and Mehmood, A. (2019). A Comprehensive Literature Review of Data Encryption Techniques in Cloud Computing and IoT Environment.
[online] doi:https://doi.org/10.1109/icict47744.2019.9001945

Panahi, P., Bayılmış, C., Çavuşoğlu, U. and Kaçar, S. (2021). Performance Evaluation of Lightweight Encryption Algorithms for IoT-Based Applications. [online] doi:https://doi.org/10.1007/s13369-021-05358-4

Mahdi, M.S., Hassan, N.F. and Abdul-Majeed, G.H. (2021). An improved chacha algorithm for securing data on IoT devices. [online] doi:https://doi.org/10.1007/s42452-021-04425-7

Arduino and Security

www.arduino.cc. (n.d.). Arduino Tutorials.
[online] Available at: https://www.arduino.cc/en/Tutorial/HomePage

www.arduino.cc. (July 28, 2023.). Security at Arduino.
[online] Available at: https://www.arduino.cc/en/security

Julham, Fachrizal, F., Adam, H.A., Fatmi, Y. and Lubis, A.R. (2017). Security of data communications between embedded arduino systems with substitution encryption. [online] doi:https://doi.org/10.1109/IAC.2017.8280578

Dwi Novazrianto, Rini Wisnu Wardhani and Naufal Hafiz Syahidan (2021). Present-80 Encryption Algorithm Implementation on GPRS Arduino Mega-2560 Cyber Physical Tracking System.
[online] Available at: https://ieeexplore.ieee.org/document/9527519

Abdullah, S., Noor, N.M., Khalid, N.A., Kasiran, Z. and Hisham, A.J.K. (2022). IOT Security: Data Encryption for Arduino-based IOT Devices.
[online] Available at: https://mail.journalppw.com/index.php/jpsp/article/view/5116

Reference Books

Park, C., Farzad Pour Rahimian, Dawood, N., Pedro, A., Dongmin, L., Hussain, R. and Soltani, M. (2023). Digitalization in Construction. Taylor & Francis.

Houtan Jebelli, Mahmoud Habibnezhad, Shayan Shayesteh, Asadi, S., Lee, S. and Springerlink. (2022). Automation and Robotics in the Architecture, Engineering, and Construction Industry. Cham: Springer International Publishing, Imprint Springer.

Rodrigues, H. (2021). Sustainability and automation in smart constructions: proceedings of the International Conference on Automation Innovation in Construction (CIAC-2019), Leiria, Portugal. Cham, Switzerland: Springer.

Banday, M.T. (2019). Cryptographic security solutions for the Internet of Things. Hershey, PA: Information Science Reference, an imprint of IGI Global.

Sharma, S.K., Bhushan, B. and Debnath, N.C. (2021). Security and Privacy Issues in IoT Devices and Sensor Networks. London, United Kingdom: Academic Press, an imprint of Elsevier.

Zhou, J., Lejla Batina, Li, Z., Lin, J., Losiouk, E., Majumdar, S., Daisuke Mashima, Meng, W., Picek, S., Mohammad Ashiqur Rahman, Shao, J., Masaki Shimaoka, Soremekun, E., Su, C., Je Sen Teh, Aleksei Udovenko, Wang, C., Zhang, L. and Yury Zhauniarovich (2023). Applied Cryptography and Network Security Workshops. Springer Nature.

Merli, D. (2024). Engineering Secure Devices. No Starch Press.
Jean-Philippe Aumasson (2024). Serious Cryptography, 2nd Edition: A Practical Introduction to Modern Encryption. No Starch Press.

Seneviratne, P. (2015). Internet of Things with Arduino Blueprints. Packt Publishing Ltd.