Article

Enhancing security and privacy in educational environments: A secure grade distribution scheme with Moodle integration

Ismaila Idris Sinan[1], Valerie Viet Triem Tong[2], Vivian Nwoacha[1], Jules Degila[3], Adebukola Onashoga[4], Idemudia Edetalehn Oaihimire[5], Kingsley Eghonghon Ukhurebor[6,*]

[1] Africa Centre of Excellence on Technology Enhanced Learning (ACETEL), National Open University of Nigeria (NOUN), Abuja 900001, Nigeria
[2] CentraleSupélec, Campus de Rennes, Av. de la Boulaie, 35510 Cesson-Sévigné, France
[3] African Center of Excellence on Mathematical Sciences, Computer Science and Applications, University of Abomey-Calavi (UAC) Campus d'Abomey-Calavi, 01 BP 526 Cotonou, Benin
[4] Department of Computer Science, Federal University of Agriculture-Abeokuta, Abeokuta 111101, Nigeria
[5] Faculty of Law, Edo State University-Uzairue, P.M.B. 04, Auchi 312001, Nigeria
[6] Department of Physics, Edo State University-Uzairue, P.M.B. 04, Auchi 312001, Nigeria
*** Corresponding author:** Kingsley Eghonghon Ukhurebor, ukeghonghon@gmail.com, ukhurebor.kingsley@edouniversity.edu.ng

CITATION

Sinan II, Tong VVT, Nwoacha V, et al. (2024). Enhancing security and privacy in educational environments: A secure grade distribution scheme with Moodle integration. Journal of Infrastructure, Policy and Development. 8(7): 3737. https://doi.org/10.24294/jipd.v8i7.3737

ARTICLE INFO

Received: 18 December 2023
Accepted: 29 January 2024
Available online: 1 August 2024

Abstract: In today's digital education landscape, safeguarding the privacy and security of educational data, particularly the distribution of grades, is paramount. This research presents the "secure grade distribution scheme (SGDS)", a comprehensive solution designed to address critical aspects of key management, encryption, secure communication, and data privacy. The scheme's heart lies in its careful key management strategy, offering a structured approach to key generation, rotation, and secure storage. Hardware security modules (HSMs) are central to fortifying encryption keys and ensuring the highest security standards. The advanced encryption standard (AES) is employed to encrypt graded data, guaranteeing the confidentiality and integrity of information during transmission and storage. The scheme integrates the Diffie-Hellman key exchange protocol to establish secure communication, enabling users to securely exchange encryption keys without vulnerability to eavesdropping or interception. Secure communication channels further fortify graded data protection, ensuring data integrity in transit. The research findings underscore the SGDS's efficacy in achieving the goals of secure grade distribution and data privacy. The scheme provides a holistic approach to safeguarding educational data, ensuring the confidentiality of sensitive information, and protecting against unauthorized access. Future research opportunities may centre on enhancing the scheme's robustness and scalability in diverse educational settings.

Keywords: education; communication; data privacy; data security; secure grade distribution

1. Introduction

Educational institutions continuously utilize technology in the digital era to enhance teaching and learning processes (Fink et al., 2020; Liu and Wang, 2001; Zeng, 2010). Online learning environments like Moodle and others are quickly becoming indispensable tools for disseminating educational content to a global audience (Mudiyanselage and Pan, 2020; Sahoo et al., 2020). Because of these platforms' adaptability, accessibility, and engagement, educational institutions employ them more frequently. However, data security has become a concern because of this technological shift, particularly when handling sensitive student information like grades (Mudiyanselage and Pan, 2020; Pérez et al., 2017).

As more educational institutions transition to online learning, ensuring student grades and other academic data remain accurate and private is imperative. Unauthorized access, manipulation, or data breaches might compromise the educational process and erode trust in these networks (Buja, 2021). The delicate nature of educational data highlights how critical it is to resolve these problems immediately. For instance, the Family Educational Rights and Privacy Act (FERPA) of the United States places legal and ethical responsibilities on educational institutions to ensure student privacy and preserve student information (FERPA, 2023). As a result, it is critical to develop robust data security solutions tailored particularly for educational contexts.

Hence, this article provides a comprehensive approach to enhancing data security in educational environments, particularly emphasizing the secure distribution of student grades. The study proposes a "secure grade distribution scheme (SGDS)" using tested cryptographic techniques and hardware security modules (HSMs). The study presents the SGDS, which relies on HSMs for encryption key storage and retrieval, AES (for encrypting and decrypting information), Diffie-Hellman key exchange (for initiating secure communication between parties), and message integrity code (for ensuring data integrity), to create a proof-of-concept secure student grade distribution system on Moodle, an open-source learning management system developed using PHP. This scheme provides a more simple and efficient way to easily integrate the Moodle platform for students, teachers, and administrators. The article positions the need for developing the SGDS based on reviewed literature focusing on Moodle's grade distribution system. Taking into account the contributions and shortcomings of past work focusing on the open-source learning management system, the study developed a comprehensive solution designed to address critical aspects of securing students' grades within Moodle.

However, it is worthy to acknowledge that the SGDS's approach is an implementation or an adaptation of a commonly used encryption system. Supposedly, this study serves as a valuable combination of security and privacy practices to address a key practice in modern educational technology environments: digital grade transmissions that would address some of the major data security and privacy issues affecting students' grades. Practically, this study will be useful to academic researchers, industry professionals, and regulatory institution policymakers on the latest Moodle plug-ins, integrations, and local configurations.

The rest of the paper is organized as section 2, related work, which meticulously reviews existing research in the field, outlining the objectives, contributions, and limitations of prior studies. The core of the paper, section 3, delves into the proposed scheme, breaking down its components such as key management, the advanced encryption standard (AES) implementation, the Diffie-Hellman protocol, MIC verification, and more. Section 4 justifies each element within the scheme, detailing their roles and significance in enhancing security. A practical case scenario (section 5) demonstrates the application of the scheme for transmitting grades securely. The proof of concept (section 6) assesses the feasibility of the scheme through testing, validation, and user feedback in a controlled lab environment. Concluding with section 7, the paper suggests avenues for future research.

2. Related work

Plyer et al. (2022) created a unique method for grading chemistry examinations in Moodle. Their plugin can properly grade chemistry tests, and the mark is safely maintained in Moodle. Pérez et al. (2017) suggested a method for detecting any modification of Moodle grades and alerting the users in charge to maintain the grades' security. The article focuses on SQL injection, a code injection attack that targets data-driven systems that introduce malicious SQL statements into a field for execution. The suggested solution may detect student grade changes and inform the instructor. It was created using PHP. However, the research was limited to detecting SQL injection and did not include prevention methods. In the study of Abdelsalam et al. (2023), researchers aimed to enhance the security of Moodle's grade distribution system. They proposed a new encryption scheme to protect graded data during transmission. The research introduces cryptographic techniques to safeguard sensitive information. Another related work (Cyoy, 2022) focuses on implementing two-factor authentication in Moodle to ensure secure access to grade-related information. The study explores methods to add an extra layer of protection to prevent unauthorized access to student grades. Korać et al. (2022) investigated the vulnerabilities of Moodle's gradebook and proposed strategies to strengthen its security. The study delves into potential threats and provides recommendations to address weaknesses in the Moodle platform's grade management system (Elmaghrabi and Eljack, 2019).

A comprehensive review of existing security measures in Moodle's grade distribution is presented in this research, where the strengths and weaknesses of current methods and suggestions for improvements to enhance overall system security are analysed, as summarized in **Table 1**.

Table 1. Summary of related work.

Reference	Aim	Contribution	Limitations
Pérez et al., 2017	The objective of the research is to prevent changes in student grades in the Moodle platform.	The study suggested a solution that will detect any change in a student's status and inform the instructor of it.	The research has limitations in detecting SQL injection and did not include prevention methods. The research only provides means of detecting changes in grades, not preventing them.
Plyer et al., 2022	Providing a new grading method for chemistry exams and safe grade storage inside the Moodle platform is the primary objective of the work.	The study developed and installed a Moodle plugin for grading chemistry examinations.	The research did not develop any security technique for preventing data breaches in grades in the Moodle platform.
Abdelsalam et al., 2023	Enhancing the security of Moodle's grade distribution system using a new encryption scheme.	Introduction of cryptographic techniques to safeguard sensitive information.	The research did not provide a secured way of sharing grades with staff and students.
Cyoy, 2022	Implementing two-factor authentication in Moodle to secure access to grade-related information.	Exploration of methods to add an extra layer of protection.	The research did not provide encryption for student's grades.
Korać et al., 2022	Investigating vulnerabilities in Moodle's gradebook and proposing strategies for improvement.	In-depth analysis of potential threats and recommendations.	Limited information on the practical implementation of suggested strategies.
Elmaghrabi and Eljack, 2019	Reviewing existing security measures in Moodle's grade distribution.	Analysis of strengths and weaknesses, suggestions for improvements.	Lack of empirical testing for proposed enhancements.

3. The proposed scheme

The SGDS will increase the security and privacy of grade data on the Moodle platform, and its effective deployment and thorough assessment will yield significant outcomes. The research's main findings and conclusions are highlighted in this section.

3.1. Key management and HSMs

Key management is essential to this technique as a fundamental component of data security. HSMs, specialized equipment made to generate and secure cryptographic keys, are used in the scheme. Grade data is encrypted and decrypted using these keys. Key management gains additional security and reliability with the integration of HSMs.

3.2. Key generation

Mathematically, key generation within the scheme can be represented and expressed as follows:

Let K be the set of cryptographic keys used within the scheme, where $K = \{K_1, K_2, \dots, K_n\}$. For each encryption session, a unique cryptographic key, K_i, is generated using HSMs:

$$K_i = \text{HSM.GenerateKey}(),$$

3.2.1. Key storage

The generated cryptographic keys are securely stored within the HSMs. This can be mathematically represented and expressed as:

HSMs ensure the tamper-resistant storage of keys, which can be denoted as:

$$\text{HSM.StoreKey}(K_i)$$

3.2.2. Key retrieval

The cryptographic keys are safely retrieved from the HSMs for encryption or decryption. Mathematically, this can be represented and expressed as:

To obtain a specific key for an encryption or decryption session:

$$K_i = \text{HSM.RetrieveKey}()$$

3.2.3. Nonce integration

Nonces, arbitrary integers created for every encryption session, are key to improving security and preventing replay attacks. They are made securely and integrated into the generation of keys. The representation of nonce integration mathematically is expressed as follows:

Let N be the set of nonces used within the scheme, where $N = \{N_1, N_2, ..., N_n\}$. For each encryption session, a unique nonce, N_i, is generated:

$$N_i = \text{HSM.GenerateNonce}(),$$

The nonce is securely combined with the cryptographic key to create a session-specific key, denoted as Ki nonce, ensuring unique keys for each session:

$$K_{i_{nonce}} = K_i \text{XOR} N_i$$

3.3. AES implementation

One of the scheme's main components for maintaining data security and secrecy is using the AES. The AES is a well-known symmetric encryption technique known for its high security. To secure graded data against unwanted access, our system carefully integrates AES for encryption and decryption.

AES encryption process

AES operates on data in fixed-size blocks, applying a series of transformation rounds using a specific encryption key. In the context of the SGDS, we employ AES-256, which operates a 256-bit encryption key for maximum security.

Mathematical representation:

- Data division: Grade data, denoted as G, is divided into fixed-size blocks, represented as G_1, G_2, ..., G_n.
- AES encryption rounds: AES performs a series of transformation rounds using the encryption key, K. The number of rounds depends on the critical size, with AES-256 using 14 rounds.
- Block encryption: AES encrypts each data block, G_i, using the encryption key, K, and the respective round, R_i. Mathematically, this can be represented as
$$C_i = \text{AES\_Encrypt}(G_i, K, R_i)$$
- Ciphertext concatenation: The ciphertext blocks, C_i, are concatenated to form the complete ciphertext, C.

3.4. AES decryption process

On the recipient's end, AES decryption is applied to retrieve the original grade data from the ciphertext. The decryption process is the reverse of encryption and is mathematically represented and expressed.

- Ciphertext division: The ciphertext, C, is divided into blocks, represented as C_1, C_2, ..., C_n.
- AES decryption rounds: AES decryption employs the same number of rounds and the decryption key, K, denoted as R_n, R_{n-1}, ..., R_1, where n is the number of rounds.
- Block decryption: Each ciphertext block, C_i, is decrypted using the decryption key, K, and the corresponding round, R_i, yielding the original data block, G_i. Mathematically: $G_i = \text{AES\_Decrypt}(C_i, K, R_i)$.
- Data concatenation: The decrypted data blocks, G_i, are concatenated to obtain the original grade data, G.

3.5. Diffie-Hellman key exchange protocol

The Diffie-Hellman key exchange protocol is a critical component of this scheme that enables secure key exchange between users, notably instructors and students. This protocol helps create a secure channel within the Moodle platform, allowing secure communication without requiring pre-shared keys.

Diffie-Hellman key exchange process

The Diffie-Hellman protocol allows two parties, in this case, instructors and students, to generate a shared secret key over an unsecured channel without explicitly sharing it. This process can be mathematically represented as follows:

- Parameter setup: A set of parameters, including a large prime number (p) and a primitive root (g), is chosen, and made publicly available. Both instructors and students use these parameters.
- Key generation (instructors): Instructors generate their private keys (InstructorPrivateKey) and corresponding public keys (InstructorPublicKey) using the chosen parameters;
  - ✓ InstructorPrivateKey = a (a randomly chosen secret integer)
  - ✓ InstructorPublicKey = $g^a \text{mod}\, p$
- KeY GEneration (students): Similarly, students generate their private keys (StudentPrivateKey) and corresponding public keys (StudentPublicKey) using the same parameters;
  - ✓ StudentPrivateKey = b (b randomly chosen secret integer)
  - ✓ StudentPublicKey = $g^b \text{mod}\, p$
- Key exchange:
  - ✓ Instructors and students exchange their public keys (InstructorPublicKey and StudentPublicKey) over the unsecured channel.
- Shared secret key: Both parties independently compute the shared secret key (SharedSecretKey) using the received public keys and their own private keys. Mathematically;
  - ✓ Instructors calculate: SharedSecretKey = $\text{studentPublicKey}^a \bmod p$
  - ✓ Students calculate: SharedSecretKey = $\text{studentPublicKey}^b \bmod p$

3.6. Message integrity code (MIC) verification

To guarantee the integrity of grade data throughout transmission, the scheme includes MIC checking as a crucial security mechanism. It provides an effective way to find any unauthorized changes or tampering with the grade data.

3.6.1. MIC generation process

Mathematically, the MIC generation process can be represented and expressed as follows:

1) MIC generation (instructors): When an instructor prepares to distribute grade data, a unique MIC is generated for each data packet (MIC_Instructor1, MIC_Instructor2, etc.). This is achieved by hashing the grade data and a secret key known only to the instructor.
2) Mathematically; MIC_Instructor_i = Hash (GradeData_i + InstructorSecretKey).
3) MIC generation (students): When students receive the data packets, they generate their own MICs for the received data. This ensures that they can verify data integrity and detect any unauthorized changes; MIC_Student_i = Hash (ReceivedData_i + StudentSecretKey).

3.7. Key metrics

In the context of key management, the scheme necessitates the generation and distribution of cryptographic keys for encryption and decryption processes. The key metrics include:

- Instructor keys:

    - ✓ Private key: Each instructor possesses a private key generated securely within the Moodle environment.
    - ✓ Public key: The corresponding public key is derived from the private key using the Diffie-Hellman key exchange protocol.
- Student keys:
    - ✓ Private key: Each student has a unique private key generated within the Moodle environment.
    - ✓ Public key: Similarly, the student's public key is generated through the Diffie-Hellman key exchange protocol.
- Staff keys:
    - ✓ Private key: Staff members also have a private key generated securely within Moodle.
    - ✓ Public key: The public key for staff is generated using the same Diffie-Hellman key exchange process.

Number of keys calculation

The number of keys required can be calculated based on the number of participants within the Moodle system. If there are '*n*' instructors, '*m*' staff members, and '*p*' students, the total number of keys can be expressed as:

$$\text{Total keys} = n(\text{Instructor keys}) + m(\text{Staff keys}) + p(\text{Student keys})$$

This formula or expression accounts for the unique keys associated with each role within the educational environment. The integration ensures that each participant has the necessary cryptographic keys to engage in secure grade distribution.

4. Justification of each element

This section presents a detailed justification for each element incorporated into the SGDS based on the results of experiments conducted to assess their effectiveness.

4.1. Key management and HSMs

- Security enhancement: The integration of HSMs is justified by their ability to securely generate, store, and retrieve cryptographic keys. HSMs provide a dedicated and tamper-resistant environment, enhancing the overall security of the key management process.
- Reliability: The secure storage of cryptographic keys within HSMs ensures their reliability and protection against unauthorized access. This reliability is crucial for maintaining the confidentiality and integrity of grade data.

4.2. AES implementation

- High-level security: The use of the AES, specifically AES-256, is justified by its reputation for providing a high level of security. AES is widely recognized for its resistance to various cryptographic attacks, making it suitable for safeguarding sensitive grade data.
- Symmetric encryption efficiency: AES's symmetric encryption approach is efficient for bulk data encryption and decryption, ensuring that the process is both secure and computationally feasible within the Moodle environment.

4.3. Diffie-Hellman key exchange protocol

- Secure key exchange: The Diffie-Hellman key exchange protocol is justified by its ability to facilitate secure key exchange between instructors, staff, and students. It eliminates the need for pre-shared keys, enhancing the overall security of communication channels within Moodle.
- Public and private key generation: The use of public and private keys in the Diffie-Hellman protocol allows entities to securely share public keys while maintaining the confidentiality of their private keys. This ensures a secure and efficient key exchange process.

4.4. MIC verification

- Tamper detection: MIC verification is crucial for detecting any unauthorized changes or tampering with grade data during transmission. This element ensures the integrity of the data, preventing malicious alterations.
- Hashing for integrity: The use of hash functions for MIC generation provides a reliable and efficient method for verifying data integrity. Hashing ensures that even minor changes to the data result in significantly different MIC values.

5. Case scenario

In this scenario, we will examine the processes and steps an instructor takes to transmit grades to staff and students safely. The instructor broadcasts the ciphertexts and MICs during transmission via a public channel, allowing staff and students to view the encrypted grades.

Step 1: Instructor's initial setup:

- Diffie-Hellman parameter setup:
  The instructor selects a large prime number, p.
  The instructor selects a primitive root, g (where g is a primitive root modulo p).
- Key generation (instructor):
  The instructor generates a private key:
  InstructorPrivateKey (a randomly chosen secret integer).
  The instructor computes the corresponding public key:

InstructorPublicKey using the equation: InstructorPublicKey =

$$g^{\text{InstructorPrivateKey}} \bmod p$$

Step 2: Student and staff setup:

- Key generation (student and staff):

Students generate their private keys, StudentPrivateKey (randomly chosen secret integers).

Students compute their public keys, StudentPublicKey, using the equation: StudentPublicKey = $g^{\text{StudentPrivateKey}} \bmod p$

Staff generate their private keys, StaffPrivateKey (randomly chosen secret integers).

Staff compute their public keys, StaffPublicKey, using the equation:

$$\text{StaffPublicKey} = g^{\text{StaffPrivateKey}} \bmod p$$

Step 3: Diffie-Hellman key exchange:

- Key exchange:

Instructor and students exchange their public keys (InstructorPublicKey and StudentPublicKey).

Instructor and staff exchange their public keys (InstructorPublicKey and StaffPublicKey).

- Shared secret key calculation:

The instructor calculates SharedSecretKey using the equation: $\mathrm{SharedSecretKey} = \mathrm{StudentPublickey}^{\mathrm{InstructorPrivateKey}} \bmod p$

The instructor calculates SharedSecretKey using the equation: $\mathrm{SharedSecretKey} = \mathrm{StaffPublickey}^{\mathrm{InstructorPrivateKey}} \bmod p$

Students and staff calculate SharedSecretKey similarly. Both parties derive the same SharedSecretKey.

Step 4: Grade encryption and MIC generation:

- Grade data: The instructor has grade data, GradeData.
- Nonce generation: A unique nonce, *Ni*, is generated for this session.
- Session-specific key creation:

The instructor creates a session-specific key by combining the SharedSecretKey with the nonce: *Ki*_instructor = SharedSecretKey XOR *Ni*.

- AES encryption: The instructor encrypts the grade data (GradeData) using the session-specific key (*Ki*_instructor) and obtains the Ciphertext Ciphertext.
- MIC generation: The instructor calculates a MIC for the encrypted grades using a cryptographic hash function: MIC = Hash(*Ki*_instructor || Ciphertext).

Step 5: Publication on a public channel

- Public Channel Transmission: The instructor publishes the Ciphertext and MIC on a public channel accessible to staff and students.

Step 6: Decryption and MIC verification (student and staff):

- Decryption: Students and staff retrieve the ciphertext and MIC from the public channel.

Session-specific key creation (student):

The student calculates the session-specific key: *Ki*_student = SharedSecretKey XOR *Ni*.

The staff calculates the session-specific key: *Ki*_staff = SharedSecretKey XOR *Ni*.

- AES decryption:

The student decrypts the Ciphertext using *Ki*_student and retrieves the grade data (GradeData).

The staff decrypts the Ciphertext using *Ki*_staff and retrieves the grade data (GradeData).

- MIC verification (student and staff):

The student calculates a MIC using the received encrypted data (Ciphertext) and the shared key (*Ki*_student) and compares it to the received MIC.

The staff calculates a MIC using the received encrypted data (ciphertext) and the shared key (*Ki*_staff) and compares it to the received MIC.

Step 7: User feedback:

- User feedback: Both students and staff provide feedback on the grade distribution experience, security, and any issues or suggestions for improvement.

6. Proof of concept

The feasibility and practicality of the proposed scheme for enhancing security and privacy in educational environments, particularly within the context of Moodle integration, were assessed through a proof of concept. This section outlines the steps involved in developing, implementing, and evaluating the scheme within a controlled lab environment. The proof of concept, however, does not employ HSM; instead, all cryptographic keys are kept in a file inside Moodle.

6.1. Lab environment setup

A virtualized lab environment was created, consisting of three distinct virtual machines (VMs) to represent instructors, staff, and students. These VMs were configured to operate on the same network, allowing for seamless communication.

6.2. Moodle installation and configuration

Moodle, the widely used open-source learning management system, was installed on each of the VMs, mimicking a real educational environment. The installation and configuration encompassed the web server setup, database creation, and Moodle initialization. A shared folder was established on the instructor's VM for resource sharing.

6.3. Scheme plugin development

A custom scheme plugin was developed to implement secure grade distribution features within Moodle. The plugin integrated AES encryption, Diffie-Hellman key exchange, and MIC verification. The development environment included PHP tools and a code editor.

6.4. Plugin installation and activation

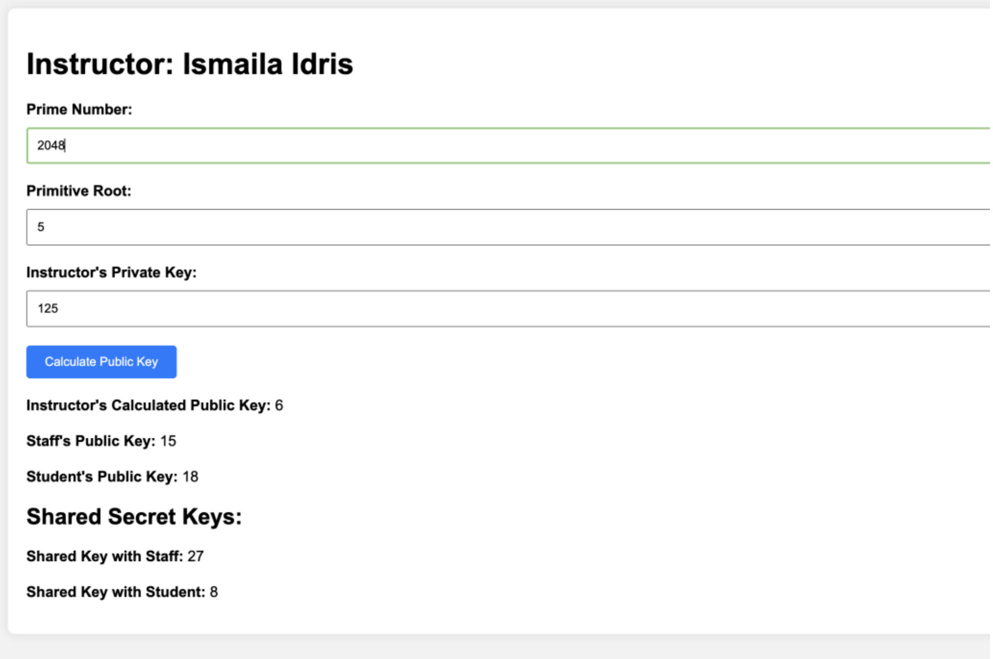
The developed scheme plugin was uploaded and activated on each of the Moodle instances representing instructors, staff, and students. This enabled the secure grade distribution features across the roles.

6.5. Testing and validation

A series of tests were conducted to verify the functionality of the scheme plugin:

6.5.1. Key exchange

The Diffie-Hellman key exchange was initiated between the instructor and each student to establish a shared secret key (see **Figure 1**).



Instructor: Ismaila Idris

Prime Number: 2048

Primitive Root: 5

Instructor's Private Key: 125

Instructor's Calculated Public Key: 6

Staff's Public Key: 15

Student's Public Key: 18

Shared Secret Keys:

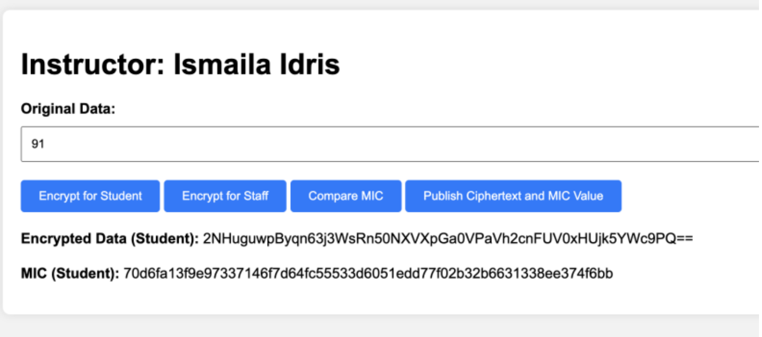
Shared Key with Staff: 27

Shared Key with Student: 8

Figure 1. Key exchange using Diffie-Hellman.

6.5.2. AES encryption and decryption

Instructors encrypted grades and shared them with staff and students. Recipients successfully decrypted the ciphertext using the shared secret key (see **Figure 2**).



Instructor: Ismaila Idris

Original Data: 91

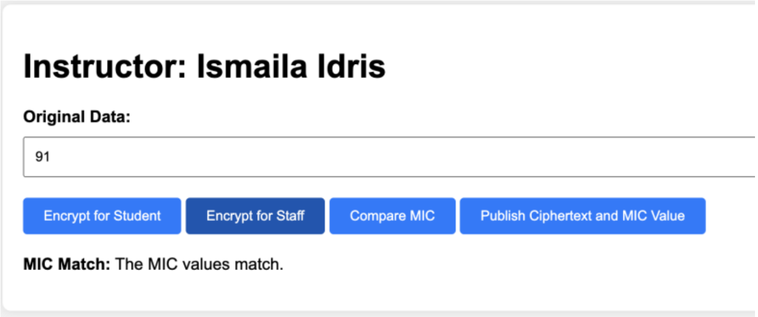
Encrypted Data (Student): 2NHuguwpByqn63j3WsRn50NXVXpGa0VPaVh2cnFUV0xHUjk5YWc9PQ==

MIC (Student): 70d6fa13f9e97337146f7d64fc55533d6051edd77f02b32b6631338ee374f6bb

Figure 2. AES Encryption and MIC generation.

6.5.3. MIC verification

Recipients verified the integrity of the received grades using MIC validation (see **Figure 3**).



Instructor: Ismaila Idris

Original Data: 91

MIC Match: The MIC values match.

Figure 3. MIC verification

6.6. Evaluation and feedback

Feedback was gathered from participants who represented the roles of instructors, staff, and students in the lab environment. The feedback aimed to evaluate the ease of use, security, and effectiveness of the scheme.

6.7. Discussion

Integrating a secure grade distribution mechanism inside the Moodle platform is essential to guaranteeing the security and privacy of sensitive academic data. The system provides a reliable solution for graded data security by integrating cutting-edge cryptographic methods, including Diffie-Hellman key exchange, AES encryption, and MIC verification. Using the Diffie-Hellman key exchange protocol is one of this system's advantages.

Additionally, via the rapid and safe establishment of encryption keys made possible by this protocol, staff, students, and instructors may securely interact without directly trading sensitive information. The data is well safeguarded during transmission because of robust session-specific keys derived from shared secrets and nonces. Another feature of the system is its use of AES for high-grade data encryption. Since AES is a well-used and reliable encryption technology, its applicability for protecting educational data is demonstrated by its mathematical form. Users are reassured about the system's security by the openness with which the encryption and decryption procedures are explained. Adding MIC checking improves the system's security. It guarantees the received data's integrity, enabling the detection of any unauthorized alteration. To protect cryptographic keys, a crucial step is the adoption of HSMs for secure key management. By offering a safe and specialized setting for key storage and cryptographic operations, HSMs lower the possibility of key compromise. As stated earlier, it is worthy to acknowledge that the SGDS's approach is an implementation or an adaptation of a commonly used encryption system.

7. Conclusion and directions for future research

In conclusion, the secure grade distribution system provides a comprehensive and robust solution for securing academic data within the Moodle platform. Combining cryptographic techniques, including Diffie-Hellman key exchange, AES encryption, and MIC verification, offers a secure and efficient means of protecting sensitive grade data. Including HSMs further strengthens the system's security, ensuring the confidentiality of cryptographic keys. This system can serve as a model for educational institutions and platforms seeking to enhance data security and privacy. The ongoing development and research in the field will be crucial to adapting to evolving threats and vulnerabilities. User feedback and involvement will be central to the system's continued improvement. Maintaining academic integrity and ensuring students' records' privacy in an increasingly digital education landscape is essential.

Despite the strengths of the current system, the following are suggested areas for potential future research and development:

- Scalability and performance optimization: Investigate methods for improving the system's scalability and performance to ensure efficient operation, mainly as educational institutions grow.

- Blockchain integration: Explore the integration of blockchain technology for enhanced data integrity and transparency.
- User-centric security: Develop user-controlled key management and additional layers of access control, allowing students and staff to play an active role in data security.
- Enhanced feedback mechanisms: Improve user feedback mechanisms to enhance the user experience and security.
- Consideration of any vulnerabilities or shortcomings to the SGDS, including potential private key compromise through crash dump attacks.
- Orientation of the SGDS as a supplementary approach from a defence-in-depth strategy to contextualize properly the motivation for pursuing this method.

Author contributions: Conceptualization, IIS; methodology, IIS, VVTT, VN, JD, AO, IEO and KEU; validation, IIS, VVTT, VN and KEU; formal analysis, IIS, VVTT, VN and KEU; investigation, IIS, VVTT, VN and KEU; resources, IIS, VVTT, VN and KEU; data curation, IIS, VVTT, VN and KEU; writing—original draft preparation, IIS, VVTT, VN and KEU; writing—review and editing, IIS, VVTT, VN, JD, AO, IEO, and KEU; visualization, IIS, VN and KEU; supervision, VN and KEU; project administration, IIS, VN and KEU; funding acquisition, IIS, VVTT, VN, JD, AO, IEO, and KEU. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: The authors appreciate the authors and publishers, whose articles were used as guides for this study. Also, the authors express gratitude to their respective institutions and the Africa Centre of Excellence on Technology Enhanced Learning (ACETEL), National Open University of Nigeria, Abuja as well as the Digital Science and Technology Network (DSTN) and Confidentiality, Integrity, Disponibilité, Repartition (CIDRE) inra team CentraleSupelec for supporting this study.

Conflict of interest: The authors declare no conflict of interest.

References

Abdelsalam, M., Idrees, A. M., Shokry, M. (2023). A Proposed Model for Improving the Reliability of Online Exam Results Using Blockchain. IEEE Access.

Asanga, M. P., Essiet, U. U., Ukhurebor, K. E., et al. (2023). Social Media and Academic Performance: A Survey Research of Senior Secondary School Students in Uyo, Nigeria. International Journal of Learning, Teaching and Educational Research, 22(2), 323–337. https://doi.org/10.26803/ijlter.22.2.18

Buja, A. G. (2021). Cyber Security Featuresfor National E-Learning Policy. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(5), 1729–1735. https://doi.org/10.17762/turcomat.v12i5.2169

Cyoy, R. B. (2022). Framework for Effective Management of Cyber Security on E-learning Platforms in Public Universities in Kenya [PhD Thesis]. University of Nairobi. 2022. Available online: http://erepository.uonbi.ac.ke/handle/11295/161726 (accessed on 10 December 2023).

Elmaghrabi, A. Y., Eljack, S. M. (2019). Enhancement of Moodle learning Management System Regarding Quizzes Security and Stability Problems. 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS). https://doi.org/10.1109/cais.2019.8769530

FERPA. (2023). Family Educational Rights and Privacy Act (FERPA)—KSCI-Arc. Available online: https://www.sciarc.edu/institution/resources/policies-anddisclosures/ferpa?gad_source=1&gclid=CjwKCAjwnOipBhBQEiwACyGLuto2DKpkDoizEay4fCMrCLWNDeKnrt63vV0otSrMtLT1Aeg6vBW2wxoCf0kQAvD_BwE (accessed on 10 December 2023).

Hussaini, A. R., Ibrahim, S., Ukhurebor, K. E., et al. (2023). The Influence of Information and Communication Technology in the Teaching and Learning of Physics. International Journal of Learning, Teaching and Educational Research, 22(6), 98–120. https://doi.org/10.26803/ijlter.22.6.6

Korać, D., Damjanović, B., Simić, D. (2021). A model of digital identity for better information security in e-learning systems. The Journal of Supercomputing, 78(3), 3325–3354. https://doi.org/10.1007/s11227-021-03981-4

Mudiyanselage, A. K., Pan, L. (2017). Security test MOODLE: a penetration testing case study. International Journal of Computers and Applications, 42(4), 372–382. https://doi.org/10.1080/1206212x.2017.1396413

Ndunagu, J. N., Ukhurebor, K. E., Adesina, A. (2023). Virtual Laboratories for STEM in Nigerian Higher Education: The National Open University of Nigeria Learners' Perspective. In: Proceedings of the Technology-Enhanced Learning in Laboratories Workshop (TELL). pp.38–48.

Nneji, C. C., Urenyere, R., Ukhurebor, K. E., et al. (2022). The impacts of COVID-19-induced online lectures on the teaching and learning process: An inquiring study of junior secondary schools in Orlu, Nigeria. Frontiers in Public Health, 10. https://doi.org/10.3389/fpubh.2022.1054536

Pérez, S. O., Díez, C. H., García, J. A. M. (2017). Applying Security to Moodle Grades. In: Proceedings of the 2017 International Conference on Security and Management (SAM), Athens, United States: The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 117–123.

Plyer, L., Marcou, G., Perves, C., et al. (2022). Implementation of a soft grading system for chemistry in a Moodle plugin. Journal of Cheminformatics, 14(1). https://doi.org/10.1186/s13321-022-00645-0

Sahoo, K. K., Mishra, P. C., Reddy, R. V. (2020). Utilization of Moodle in Teaching Undergraduate Students in West Africa. Integration of Education, 24(4), 552–560. Internet Archive. https://doi.org/10.15507/1991-9468.101.024.202004.552-560

Sinan, I. I., Degila, J., Nwaocha, V., Onashoga, S. A. (2022). Data Architectures' Evolution and Protection. 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET). https://doi.org/10.1109/icecet55527.2022.9872597