Securing Online Assessments : A Cryptographic Approach to Combat Academic Dishonesty

AARAV ASHISH MEHRISHI, University of Twente, The Netherlands

In the light of the rapid expansion and digitization of education, adoption of popular learning platforms such as Canvas LMS (Learning Management System), Moodle and Google forms for students at universities has become pervasive. This research focuses on these platforms due to their widespread adoption in universities and significant role in modern online education. Despite their prevalence, there exist concerns regarding the integrity of data particularly in the realm of online assessments. During and post COVID-19 pandemic, there was a surge in the number of online cheating cases of various types such as unauthorized access, content sharing, plagiarism and use of external resources by students during the assessment which indicate the current vulnerabilities of these educational platforms. This research aims to address gaps that promote online cheating in Canvas LMS, Moodle and Google forms. Additionally, after addressing the gaps, this study proposes solutions for each targeted issue and integrate them into a prototype. The prototype incorporates cryptographic protocols for each phase of the examination using encryption and digital signatures to maintain the integrity of the exam data. Moreover it includes features like Two-Factor-Authentication(2FA), Safe Exam Browser (SEB) and no copy-paste functionality which help in mitigating academic dishonesty among students. The developed prototype is designed to help secure online assessments and uphold the integrity of exam data while safeguarding student's privacy.

Additional Key Words and Phrases: Safe Exam Browser (SEB), Canvas LMS, Moodle, Google Forms, Encryption, Digital Signature

1 INTRODUCTION

With the onset of the COVID-19 pandemic, online education has been widely integrated and promoted across educational institutions which has in turn led to the adoption of remote electronic exams (E-exams) as a predominant assessment method [9]. While online assessments play an important role in enhancing academic performance in higher education, this sudden shift to widely used remote assessment platforms like Canvas LMS, Moodle and Google forms may have introduced additional repercussions. According to Dayananda et al. (2021), repercussions such as academic dishonesty performed by students during the exam, raise questions on the integrity of the platform. These concerns encompass more than just gaining unauthorized access to exam materials beforehand. They also involve the alteration of questions, answers and grades. Moreover, various forms of cheating have been observed [17] such as, plagiarism, browsing the internet for solutions during the examination or content sharing with other students. [6],[24]. This research explores the most commonly used educational and assessment platforms at a university level such as Canvas LMS, Moodle and Google forms in depth to understand the current vulnerabilities and gaps of these platforms which promote various online cheating methods commonly used by students during an assessment such as plagiarism, use of external resources and unauthorized access. Even though these platforms have taken steps to maintain their privacy and security, each of them have their own shortcomings. For example, according to Manuel et al. (2023), Canvas LMS allows users to maintain concurrent logins which will be addressed in this research. Additionally, this research proposes a combination of cryptographic protocols inspired by [6], for each stage of an online assessment from the teacher creating an exam to the student attempting it. These proposed solutions are integrated to a prototype of an online assessment platform and tested for their efficiency. Moreover, to further secure the integrity of the online assessment, features like browser lockdown and prevention of copy-paste functionality are implemented to the prototype. This prevents the student from opening a new tab on their browser to access external resources during the assessment and not being able to duplicate their answers.

By developing a prototype of a secure online assessment platform, this research aims to contribute to the field of online assessments by recognizing the current vulnerabilities and propose solutions for privacy concerns in educational applications, especially focusing on the prevalent issue of online cheating and academic dishonesty. The next section of this paper, Section 2, reviews related work and literature. Section 3 defines the problem statement and mentions the research questions addressed in this paper. This is followed by the proposed solutions for this research discussed in Section 4 mentioning the current vulnerabilities identified and the features of the developed prototype. Section 5 presents the tests conducted on the prototype and an analysis of the results. Section 6 discusses the limitations of this research and the scope for future improvements. This paper concludes in Section 7 which mentions the concluding remarks and summary of the findings.

2 RELATED WORK

Manuel et al. (2023) discusses how some learning management systems such as, Canvas LMS, permit users to have and maintain concurrent logins. This functionality creates a vulnerability where a student, while attempting a quiz on Canvas LMS in a supervised environment could potentially share their login details to an external party. This poses questions on the integrity of the assessment platform. It enables the third party in this case to provide unauthorized assistance remotely while bypassing the system's security measures. The paper by Noorbehbahani et al. (2022), mentions the importance of knowing the popular types of cheating methods in order to mitigate them. Popular individually used cheating methods include use of forbidden resources or accessing solutions before the assessment. On the other hand, those used in a group include, impersonation and collaboration. In the context of proposing solutions to maintain the integrity of online examinations, Jung and Yeom (2009), discuss a solution that enhances the security and integrity

TScIT 41, July 5, 2024, Enschede, The Netherlands

of online assessments with the use of group cryptography. This technique involves dividing important information and sensitive data into cryptographic keys distributed among the users. The paper suggests the SeCOnE System Software which implements group cryptography. This technique involves multiple users to collaborate in order to decrypt data. Each user is provided with a unique key and during the exam, users need to communicate and collaborate by combining their cryptographic keys using the SeCOnE System Software to access the encrypted content. Abdelsalam et al. (2024) propose a model using blockchain technology in order to secure the exam contents and improve the overall transparency in the assessment process. Another paper by Venukumar and Pathari (2016), proposes a secure, efficient and flexible Multi-Factor-Authentication (MFA) system using a technique known as Threshold Cryptography. This technique is used to divide a cryptographic key into multiple parts. When a minimum number of these parts are combined, they key can be used for authentication or decrypting data.

The above mentioned studies, showcase some of the current vulnerabilities that still exist in the realm of online examinations that allow scope for academic dishonesty. Moreover, the existing and proposed solutions mentioned have their own limitations and loopholes. Students could potentially use external resources such as their notes or online resources on their browser, students may also replicate or duplicate answers from these external resources. Additionally, some students might attempt to exploit the system using techniques to gain unauthorized access to the assessment contents or have some other user take the test on their device.
The motivation of this research is to identify the current gaps or issues in Canvas LMS, Moodle and Google Forms that promote academic dishonesty. Furthermore, this research proposes solutions to the identified vulnerabilities along with cryptographic protocols for different phases of the exam such as creating, approving and requesting the exam.

3 PROBLEM STATEMENT

This section serves as the foundation of this research project. The problem statement of the project revolves specifically around the issue with online cheating and academic dishonesty due to the widespread adoption of various online assessment platforms such as Canvas LMS, Moodle and Google forms. The use of unauthorized collaboration, external resources and various other technological aids pose a threat to the integrity and credibility of online assessments. Moreover, the surge in online cheating cases during and post COVID-19 pandemic [22] period highlight the importance to address the issue and propose effective solutions for the same. This research aims to address the following research questions:

(1) **Research Question 1 (RQ1):** What are the current vulnerabilities of popular online assessment platforms used at a university level such as Canvas LMS, Moodle and Google forms which contribute to online cheating and academic dishonesty?
(2) **Research Question 2 (RQ2):** How can a combination of existing or proposed cryptographic protocols and solutions to the identified vulnerabilities be leveraged to enhance the security and integrity of an online assessment platform for each phase of an assessment while also prioritizing student privacy?
(a) **Sub Research Question 1:** How can the prototype of an online assessment platform mitigate common web application threats like SQL Injection and Cross-Site Scripting(XSS) to ensure both security and student privacy?

These research questions will help serve as a guiding path for investigating the current gaps of existing online assessment platforms which contribute to academic dishonesty and propose tested alternative solutions which would combine identified solutions discussed in the next section with the help of a prototype to enhance the security and integrity of these platforms.

4 PROPOSED SOLUTION

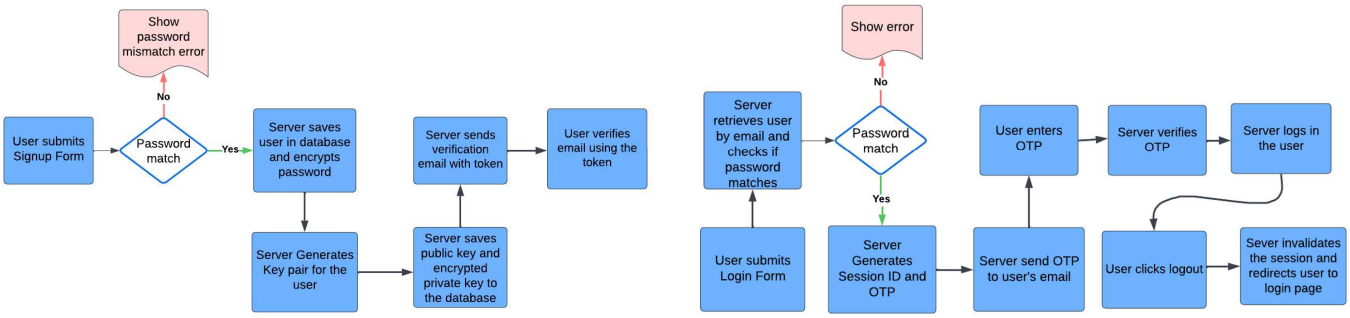
This section discusses some findings of this research for the defined research questions as well as propose solutions to current online educational and assessment platforms in order to tackle academic dishonesty. This section first discusses the current vulnerabilities of popular online assessment platforms used at a university level such as Canvas LMS, Moodle and Google forms which contribute towards academic dishonesty. Moreover, a prototype of a secure online assessment platform will be presented which would propose solutions for the mentioned vulnerabilities and design cryptographic protocols inspired by Castella-Roca et al. (2006), in order to secure the contents of the examination from unauthorized access or tampering while in transit over the network.

Table 1. Current Vulnerabilities of Popular Platforms and Solutions

Platform	Current Vulnerabilities	Solution
Canvas LMS	Concurrent logins	Track Session ID
Moodle	Cross-Site Scripting (XSS) attacks	Input Sanitization, Parameterized queries
	Weak authentication	2-Factor-Authentication (2FA)
	Insecure data handling	Encryption, Digital Signature
Google forms	Access to external resources during assessment	Browser lockdown, visibility change detection, Full screen mode
	Allowing copy-paste functionality	Restrict copy-paste to and from the assessment

4.1 Current Vulnerabilities in Online Assessment Platforms (RQ1)

In order to address the first research question (RQ1) regarding the current vulnerabilities inherent in commonly utilized online assessment platforms at a university level such as Canvas LMS, Moodle and Google forms which contribute to online cheating and academic dishonesty, a comprehensive study of these platforms as well as a review of literature was conducted to understand the common gaps among them and cheating methods used individually or in a group. The



(a) Sign up flow (b) Login flow

Fig. 1. Sign up and Login flow

findings of this research question is crucial in guiding the development of the proposed solutions and the prototype. The common issues currently faced during an assessment on these platforms are mentioned in Table 1. To further explain the vulnerabilities and their corresponding solutions, a brief explanation is provided below:

4.1.1 Concurrent logins: Allowing concurrent logins enable users to have multiple sessions using the same credentials. This can cause unauthorized access to the sensitive exam content when considering it for an online assessment platform. Students could share their login credentials with an external party and cheat during the exam. The paper [19] mentions how allowing concurrent login sessions presents a loophole where while attempting an assessment on Canvas LMS in a supervised environment, students might potentially share their credentials with an external party in order to solve the assessment. Therefore, in order to mitigate this issue, the session id of the user should be tracked and the user should be logged out of their previous session if they attempt to log in again on another tab, browser or even on another device with the same credentials.

4.1.2 Cross-Site Scripting Attacks (XSS):. The Cross-Site Scripting (XSS) Attacks, allow an attacker to inject malicious into the web pages. For an online assessment platform, this could pose a huge threat as it could lead to unauthorized access and manipulation of student or exam data causing disruptions. Such a vulnerability can pose questions on the fairness and integrity of the online assessment. Popular e-learning platforms such as Moodle, has been found to have a lot of vulnerabilities and a weak authentication mechanism which have resulted in its database being compromised in the past. The paper [20], discusses how even after containing the more serious vulnerabilities, the newer versions of the software still have some issues and attacks which include Cross-Site Scripting Attacks and SQL Injection. In order to contain this issue, input sanitization and parameterized queries should be integrated to the system in order to avoid malicious scripts being used by an attacker [7].

4.1.3 Weak Authentication: A system with a weak authentication mechanism could lead to an unauthorized user gaining access to the platform. In the context of online examination platforms, this could cause an identity fraud where a non-registered individual could take the exam or gain access to the contents thereby leading to academic dishonesty. As discussed earlier, Moodle platform has been found to have a weak authentication system despite being a popular learning and assessment platform across universities. In order to improve and strengthen the authentication mechanism, Two-Factor-Authentication should be implemented which involves two separate forms of identifications before logging in the user to system.

4.1.4 Access to external resources: Allowing students to access external resources during an online exam would enable students to gain access and refer to unauthorized websites or materials which would defeat the purpose of an examination [5],[21]. Google forms does not restrict or notify the teacher when a student opens a new tab during a quiz [12]. Hence, in order to mitigate this issue, students should be restricted to access different tabs or applications by using a Safe Exam Browser (SEB) while attempting their exam.

4.1.5 Insecure data handling: Insecure data handling in the context of an online assessment platform can lead to the sensitive exam data including the questions and answers to be compromised. In order to maintain the integrity of the assessment platform and the exam contents, strong encryption techniques and digital signatures should be used to secure the important data from access or manipulation [16].

4.1.6 Copy-paste Functionality: Allowing copy-paste functionality during the exam allows a student to share their answers with other students and perform plagiarism [13] thus undermining the originality aspect in their answers and assess their knowledge. An online assessment platform should be able to restrict copy-paste capabilities while a student is taking the exam. This could help reduce cheating and plagiarizing answers during the exam.

Each of the vulnerabilities mentioned above pose a significant threat to online assessment platforms. It is essential to address these issues to insure the security of the platform and the integrity of the exam data.

4.2 Prototype Development (RQ2)

The second research question (RQ2) involves leveraging a combination of solutions for the vulnerabilities identified in Table 1 into a prototype of a secure online platform. The prototype is implemented to enhance the security and integrity of online assessments while also protecting the privacy of students. The prototype developed in this research, consists of the solutions mentioned in Table 1 including Two-Factor-Authentication (2FA), tracking user session, input sanitization, encryption, digital signature, browser lockdown and restricting copy-paste functionality while the student attempts the exam. In addition to this, a combination of cryptographic protocols inspired by the work of Castella-Roca et al. (2006), for each stage of the assessment have been implemented. This research focuses on the following three stages of an exam for the protocols:

- Protocol 1 : Create Exam
- Protocol 2 : Approve Exam
- Protocol 3 : Request Exam

In Protocol 1, the teacher creates an exam for the students. This is followed by Protocol 2, carried out by the manager who approves the exam for the students. Finally, in Protocol 3, the student requests the exam from the manager and proceeds to answer it.

These protocols require that each student, teacher and the manager have a key pair of a public key cryptosystem. P_{entity} is the public key and S_{entity} is the private key. In this context, "entity" refers to any of the roles (Student, Teacher or Manager) involved in the assessment process.

- (P_T, S_T) teacher's key pair.
- (P_S, S_S) student's key pair.
- (P_M, S_M) manager's key pair.

4.3 Cryptographic Algorithms Used

This research makes use of a combination of advanced cryptographic algorithms for the developed prototype which will be discussed in this section. In order to generate key pairs (private and public) for each user, RSA (Rivest–Shamir–Adleman) algorithm is used to generate unique 2048-bit key pair. RSA is the most widely used algorithm for generating key pairs and can not be breached easily [14].

Moreover, the prototype mainly uses the Advanced Encryption Standard (AES) along with 256-bit key for encrypting the exam data with the public key of the user. This is because the AES algorithm is proven to be both a fast and an effective encryption algorithm [10, 18]. Furthermore, this setup uses Cipher Feedback Mode (CFB) which allows to encrypt exam data of varying lengths. This is followed by the encryption getting a unique random Initialisation Vector (IV) which enhances the security of the exam data.

4.4 Authentication and User Management

This section discusses how the authentication mechanism and user management have been implemented in the developed prototype of this research in order to address the current issues discussed in previous sections related to weak authentication and concurrent logins.

The sign up and login process of the Flask web application is designed to securely store and authenticate the user. The Figure 1a and Figure 1b describe the sign up and login flow for the application. After filling the sign up form, the server creates a user object of the particular role (Student, Teacher or Manager) and a key pair is generated for the user using RSA. The user's password is hashed using bycrypt algorithm before being stored to our SQLite database using SQLAlchemy, which is a powerful Object-Relational Mapping tool. Moreover, the public key is stored in plain text but the private key of the user is encrypted using AES encryption before being stored. Once the account has been created, the user receives a verification link to verify their email address. When the user attempts to login, they provide their registered email address and password. After verifying the credentials using bcrypt's hash comparison and before proceeding with authentication, the system checks whether the user has verified their email address through a token-based verification system. This step ensures that only validated users can access the application which reinforces security measures. Upon successful authentication, a unique session ID is generated and saved associated with the user in the database. This session ID is crucial for maintaining the user's authenticated state during their browsing session and helps prevent unauthorized access and concurrent logins. Additionally, a six-digit One-Time Password is generated and sent to the user's registered email address as an added layer of security. The OTP, generated using PyOTP library, serves for the Two-Factor Authentication mechanism. Finally, after the user enters the OTP and it is verified, the user is directed to their respective dashboard.

4.5 Exam Management

This section explains the protocols implemented for the developed prototype of this research in order to address the issue of insecure data handling. There are different protocols designed for each phase of the examination which involve the Teacher, Manager and the Student. The protocols, initially inspired by the paper [6], have been refined and are enhanced to suit the requirements of the developed prototype in this research.

4.5.1 Protocol 1 (Create Exam): This step of creating an exam can only be done by the Teacher. After the teacher logs in to their dashboard, they are provided with a form to create an exam. The form asks for the details of the exam as well as questions for the exam. The teacher can add two types of questions (Multiple Choice Questions or Subjective questions). If the teacher chooses to add a multiple choice question, they would have to provide the question, options and the correct

answer to that question. After the form has been filled and submitted, the following protocol takes place:

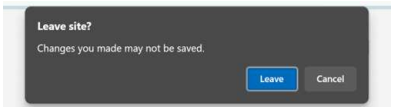
1. Form details are validated and parsed on the server side.
2. Exam questions, options, and correct answers are structured into a JSON format.
3. Unique exam identifier is computed using subject name, code, semester, exam date, fixed time, and a serial number.
4. Exam details and the generated exam ID are concatenated into a string.
5. The concatenated string is digitally signed using S_T with SHA-256 hashing and PSS padding.
6. A random AES key is generated for symmetric encryption.
7. Exam data is encrypted using AES in CFB mode with an initialization vector (IV) for added security.
8. The AES key is encrypted using RSA with P_M.
9. The digital signature, encrypted exam data, and the encrypted AES key are stored to the database.

4.5.2 Protocol 2 (Approve Exam): This step of approving the exam can only be done by the manager. After the manager logs in to their dashboard, they are presented with a list of pending exams for approval. When the manager clicks on the approve button next to the exam that was created by the teacher, the following protocol takes place:

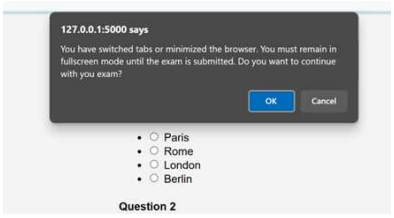
1. Fetch the encrypted exam data from the database using the exam's ID.
2. Convert the hexadecimal strings of encrypted exam data, encrypted AES key, and digital signature into byte arrays.
3. Decrypt the AES key using S_M with RSA-OAEP decryption with SHA-256 hashing and PSS padding.
4. Separate the initialization vector (IV) and encrypted content from the decrypted exam data.
5. Decrypt the exam content using AES in CFB mode with the decrypted AES key and IV.
6. Split the decrypted content to get the exam ID, exam JSON, and digital signature
7. Convert the received digital signature from hexadecimal to bytes for verification.
8. Verify the received digital signature against the stored signature
9. If signature is verified, parse the decrypted exam JSON content for manager's display.

4.5.3 Protocol 3 (Request Exam): After approving the exam, the student requests the manager for the exam. When the student logs in to their dashboard, they can view the exam that the manager had approved and when they click on the Take exam button next to it, the following steps take place:

1. Fetch exam details from the database using exam ID.
2. Convert encrypted exam data, encrypted AES key, and digital signature from hexadecimal string representation to byte arrays.
3. Use S_S to decrypt the AES key via RSA-OAEP decryption with SHA-256 hashing and PSS padding.
4. Separate the initialization vector (IV) from exam data.
5. Decrypt exam content using the decrypted AES key in CFB mode.
6. Use a delimiter to split the string into exam ID, exam JSON content, and the received digital signature.
7. Verify the received digital signature against the stored signature
8. Extract questions, options, and other exam details
9. Iterate through the exam content to remove the correct answers from the data
10. Render the HTML page with the exam questions and pass the exam content and exam ID to the template for student to take the exam.



(a) Notification when user tries to open a new tab



(b) Notification when user exits current screen

Fig. 2. Notifications when attempting to use external resources during the exam

4.6 Additional Functionalities

The cryptographic protocols are implemented to ensure confidentiality of exam data using encryption. The integrity of the data is ensured using digital signatures signed by a user's private key. Unauthorized access attempts can be detected and denied which ensures only authorized users of a particular role can create, approve and request the exam. In addition to these protocols, some more features have been implemented to further reduce the risks of a student to cheat while taking an exam. While in the exam environment, features such as disabling copy-paste to and from the exam, detecting and preventing opening of new tabs, detecting and notifying when the screen is minimized or another application is opened have been incorporated to the web application. This is done using JavaScript to disable restricting the use of shortcuts for copying (CTRL+C) and pasting (CTRL+V). When the system detects a student trying to open a new tab, a notification prevents them to open a new tab as seen in Figure 2a. If the student wants to continue with their exam, they can click cancel on the notification but if they approve that they want to leave the current tab, they would be logged out of the exam and their progress will be saved. On the other hand, if the user tries to minimize their screen or open a new application, a

similar notification is sent as seen in Figure 2b and if the user fails to respond to the notification within a time limit that would be set, the user would be logged out of the system and the progress would be saved. Lastly, in order to prevent the system from SQL injection and Cross-Site Scripting (XSS) attacks, the prototype uses SQLAlchemy, an Object-Relational Mapping (ORM) which uses parameterized queries to prevent SQL injection and Bleach which is a python library used to sanitize user inputs in order to avoid XSS attacks.

Type	Measurement	Data Size (Bytes)	Encryption Time (s)	Decryption Time (s)
Small	1	331	0.001350	0.014006
	2	322	0.000383	0.007573
	3	319	0.000238	0.004291
Medium	1	730	0.000223	0.011090
	2	730	0.000558	0.007126
	3	742	0.000363	0.007493
Large	1	1463	0.001967	0.013000
	2	1490	0.000421	0.006957
	3	1506	0.000797	0.006449

Table 2. Measured Data Sizes and Times for Different Data Types

Test	F-Value	P-Value
Encryption Times	1.0097	0.4188
Decryption Times	0.003133	0.9969

Table 3. ANOVA Test Results for Encryption and Decryption Times

5 TESTING AND RESULTS

This section includes the different types of tests performed on the implemented prototype of the web application as well as on the cryptographic protocols incorporated and analyze the results.

5.1 Performance Benchmark Test

In testing the performance of the cryptographic protocols implemented in the prototype web application, a performance benchmark test was conducted to measure and analyze the time taken for encryption and decryption of the exam data. For a data size of 319 Bytes in Table 2, the results of this test revealed that the encryption of the data was completed in a remarkably fast time of 0.000238 seconds. On the other hand, the decryption of the data took 0.004291 seconds which is slightly more than the encryption but it is acceptable since decryption is more complex and involves more steps in the defined protocols. In order to validate these findings, the paper by Andriani et al. (2018), compares various file sizes and their encryption time using AES 256-bit keys. For a 5.296 KB file, the encryption time recorded was 1.623 seconds. While, the paper does not mention the decryption times recorded, they can be estimated using a relative comparison.
Comparing the above findings with the test results, validate the efficiency of the encryption and decryption algorithms used for this research. For the data size of 319 bytes (0.319 KB) used in this test, the encryption and decryption times of 0.000238 seconds and 0.004291 seconds demonstrate the speed and efficiency of the implementation done in this research.

5.2 ANOVA Test on the Effect of Data Size on Encryption and Decryption Times

In order to determine if there exists any significant difference in the encryption and decryption times across different data sizes, an ANOVA test was conducted on the recorded values. Table 2 presents the measured data sizes (in bytes) and their corresponding encryption and decryption times (in seconds) recorded for three different data size categories (Small, Medium and Large). Moreover, each category has three measurements that were recorded to ensure the reliability and consistency for the ANOVA test to be conducted. The description of the exam data for the mentioned categories is defined below:

- **Small:** The exam data contained 2 Multiple Choice Questions (MCQ) and 2 Subjective questions.
- **Medium:** The exam data contained 5 Multiple Choice Questions (MCQ) and 5 Subjective questions.
- **Large:** The exam data contained 10 Multiple Choice Questions (MCQ) and 10 Subjective questions.

After recording the data sizes and times (encryption and decryption) for the above mentioned data sizes, the ANOVA test was conducted.

For the ANOVA test on encryption time, the following hypotheses were defined:

- **Null Hypothesis (H_0)**: There is no significant difference in encryption times across different data sizes.
- **Alternative Hypothesis (H_a)**: There is significant difference in encryption times across different data sizes.

For the ANOVA test on decryption time, following hypotheses were defined:

- **Null Hypothesis (H_0)**: There is no significant difference in decryption times across different data sizes.
- **Alternative Hypothesis (H_a)**: There is significant difference in decryption times across different data sizes.

As seen in Table 3, the ANOVA test observed an F-value of 1.0097 and P-value of 0.4188 for the encryption times. Since the P-value is greater than the significance level of 0.05, the test failed to reject the null hypothesis. This suggests that there is no significant difference in the encryption times across different data sizes (small,medium and large). Similarly an F-value of 0.003133 and a P-value of 0.9969 were observed for decryption times. Since the resulted P-value is greater than the significance level of 0.05, the test failed to reject the null hypothesis and concluded that there is no significant difference in the decryption times across different data sizes (small,medium and large).

5.3 Automated Security Testing using OWASP ZAP

In this section, the results of an automated security testing using OWASP ZAP (Zed Attack Proxy) tool are analyzed. OWASP ZAP is a penetration testing tool that assists in detecting and finding vulnerabilities in a web application. The

automated scan conducted for the developed prototype web application of this research identified potential security vulnerabilities and categorized according to risk levels (High, Medium, Low and Informational) and confidence (User Confirmed, High, Medium, Low) as displayed in a tabular form in Figure 3. High risk vulnerabilities which pose a significant threat to damage or allow unauthorized access for the web application were not detected. Medium risk issues comprising 50% of the issues, are important but not severe. Low risk issues comprising 20% of the alerts, do not pose a significant threat to the developed prototype. 30% of the alerts comprise of Informational alerts which help provide insights for security improvements.

For the confidence levels, there were no User Confirmed or Low Confidence issues. Moreover, most of the medium risk issues had a High Confidence which indicates true positives where as Medium Confidence issues had a less chance of being true positives.

The key observations of this tests indicate no critical vulnerabilities present since there were no high-alerts found. Furthermore, the absence of high-risk authentication alerts suggest that the login mechanism developed is reasonably secure. The solution to most of the alerts found as presented in the Table 4 during the test were out of the scope for this research. However, medium risk alerts were mostly related to Content Security Policy (CSP) issues which could potentially allow XSS attacks and highlight areas for improvement.

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Medium	0 (0.0%)	4 (40.0%)	1 (10.0%)	0 (0.0%)	5 (50.0%)
	Low	0 (0.0%)	1 (10.0%)	1 (10.0%)	0 (0.0%)	2 (20.0%)
	Informational	0 (0.0%)	1 (10.0%)	2 (20.0%)	0 (0.0%)	3 (30.0%)
	Total	0 (0.0%)	6 (60.0%)	4 (40.0%)	0 (0.0%)	10 (100%)

Fig. 3. Alerts categorized by risk level and confidence

Alert Type	Risk	Count
CSP: Wildcard Directive	Medium	4 (40.0%)
CSP: script-src unsafe-inline	Medium	4 (40.0%)
CSP: style-src unsafe-inline	Medium	4 (40.0%)
Content Security Policy (CSP) Header Not Set	Medium	4 (40.0%)
Missing Anti-clickjacking Header	Medium	2 (20.0%)
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	7 (70.0%)
X-Content-Type-Options Header Missing	Low	2 (20.0%)
Authentication Request Identified	Informational	1 (10.0%)
Session Management Response Identified	Informational	4 (40.0%)
User Agent Fuzzer	Informational	12 (120.0%)
Total		10

Table 4. Alert counts by alert type and risk level from OWASP ZAP scan

6 LIMITATIONS AND FUTURE WORK

This thesis presented an approach to secure online assessments by proposing solutions to the identified vulnerabilities in Canvas LMS, Moodle and Google Forms. However, this research has its limitations. Firstly, in the current implementation, a key pair (Public and Private) is generated for each user after they sign up on the platform and remain consistent throughout. However they should be regularly changed by automating key rotation while securely managing key life cycles which could improve the security of the protocols using the private keys of users to decrypt the exam data. Secondly, this implementation assumes the online exam for students would be carried out under some supervision. In order to hold exams in a less supervised environment, an automatic proctoring system as described in the papers by Atoum et al. (2017) and Gadkar et al. (2023) should be implemented in the future. This would utilize the webcam and microphone to monitor the exam after complying with the student's privacy as well as GDPR regulations to ensure ethical use.

Additionally, the current developed system does not include all phases of an examination. Including grading and reviewing of the exam would create a complete usable assessment platform. Features like automated grading for multiple-choice questions and an opportunity for students to review the grading would improve the usability of the application. Lastly, improving the user interface and making it more intuitive could improve the usability and adoption of the system.

7 CONCLUSION

Amidst the COVID-19 pandemic, the realm of online education witnessed a surge in cases of academic dishonesty by students. This called for the urgent need of robust solutions to address and solve the vulnerabilities inherent in popular educational platforms such as Canvas LMS, Moodle and Google forms. This research identified the current gaps present in these platforms that help to contribute cheating during online assessments. Moreover, this research proposed solutions to the identified vulnerabilities and incorporated them in a prototype of a secure online assessment web application. The prototype includes Two-Factor-Authentication (2FA), tracking user session, input sanitization, encryption, digital signature, browser lockdown and restricting copy-paste functionality while the student attempts the exam. In

addition to this, the developed protocol incorporates cryptographic protocols for different stages of the exam which include creating, approving and requesting the exam. The algorithms used in the protocols were tested for their efficiency and security. After conducting tests on the developed cryptographic protocols as well as the security of the web application, it was concluded that the system uses fast and effective algorithms to encrypt and decrypt the exam data in the protocols and does not have any high risk vulnerability making it fairly secure. The system successfully manages to propose effective solutions to each of the identified vulnerabilities present in Canvas LMS, Moodle and Google Forms for educational or assessment purposes.

This research aims to contribute to the field of online education by focusing on recommending solutions to mitigate academic dishonesty and maintaining the integrity of an online assessment platform. Through continued improvements to the provided solution, this research seeks to create a more secure environment for online assessments.

8 ACKNOWLEDGEMENT

The author would like to extend gratitude to Dr. Dipti Kapoor Sarmah for her supervision and support throughout this research.

REFERENCES

[1] Aarav. 2024. Secure Online Assessment. https://github.com/Aarav2402/SecureOnlineAssessment. Accessed: 2024-06-30.

[2] Mohamed Abdelsalam, Marwan Shokry, and Amira M. Idrees. 2024. A Proposed Model for Improving the Reliability of Online Exam Results Using Blockchain. *IEEE Access* 12 (2024), 7719–7733. https://doi.org/10.1109/ACCESS.2023.3304995

[3] Ria Andriani, Stevi Ema Wijayanti, and Ferry Wahyu Wibowo. 2018. Comparision Of AES 128, 192 And 256 Bit Algorithm For Encryption And Description File. In *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*. 120–124. https://doi.org/10.1109/ICITISEE.2018.8720983

[4] Yousef Atoum, Liping Chen, Alex X. Liu, Stephen D. H. Hsu, and Xiaoming Liu. 2017. Automated Online Exam Proctoring. *IEEE Transactions on Multimedia* 19, 7 (2017), 1609–1624. https://doi.org/10.1109/TMM.2017.2656064

[5] G. Bubaš and A. Čižmešija. 2023. A Critical Analysis of Students' Cheating in Online Assessment in Higher Education: Post-COVID-19 Issues and Challenges Related to Conversational Artificial Intelligence. In *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*. 905–910. https://doi.org/10.23919/MIPRO57284.2023.10159826

[6] J. Castella-Roca, J. Herrera-Joancomarti, and A. Dorca-Josa. 2006. A secure e-exam management system. In *First International Conference on Availability, Reliability and Security (ARES'06)*. 8 pp.–871. https://doi.org/10.1109/ARES.2006.14

[7] Yanpeng Cui, Junjie Cui, and Jianwei Hu. 2020. A Survey on XSS Attack Detection and Prevention in Web Applications. In *Proceedings of the 2020 12th International Conference on Machine Learning and Computing* (Shenzhen, China) *(ICMLC '20)*. Association for Computing Machinery, New York, NY, USA, 443–449. https://doi.org/10.1145/3383972.3384027

[8] DPD Dayananda, KGL Chathumini, and S Vasanthapriyan. 2021. A Novel Framework for Online Exams during the Pandemic of COVID-19: Evaluation Methods, Students' Priorities and Academic Dishonesty in Online Exams. In *2021 IEEE 1st International Conference on Advanced Learning Technologies on Education Research (ICALTER)*. 1–4. https://doi.org/10.1109/ICALTER54105.2021.9675092

[9] Lina Elsalem, Nosayba Al-Azzam, Ahmad A. Jum'ah, and Nail Obeidat. 2021. Remote E-exams during Covid-19 pandemic: A cross-sectional study of students' preferences and academic dishonesty in faculties of medical sciences. *Annals of Medicine and Surgery* 62 (2021), 326–333. https://doi.org/10.1016/j.amsu.2021.01.054

[10] Umar Fayyaz, Shahab Ahmad Niazi, Abdul Aziz, and Asjad Amin. 2023. A Secured Care Service System Using AES for Internet of Medical Things. In *2023 IEEE International Conference on Emerging Trends in Engineering, Sciences and Technology (ICEST)*. 1–4. https://doi.org/10.1109/ICEST56843.2023.10138856

[11] Siddhesh Gadkar, Bansari Vora, Krupa Chotai, Sparsh Lakhani, and Priyal Katudia. 2023. Online Examination Auto-Proctoring System. In *2023 International Conference on Advanced Computing Technologies and Applications (ICACTA)*. 1–7. https://doi.org/10.1109/ICACTA58201.2023.10392679

[12] Google Classroom Community. 2024. Google Classroom Community. https://support.google.com/edu/classroom/thread/73840136/will-the-admin-teacher-notified-if-a-student-opens-a-new-window-or-tab-during-a-quiz?hl=en. Accessed: 2024-06-21.

[13] Arto Hellas, Juho Leinonen, and Petri Ihantola. 2017. Plagiarism in Take-home Exams: Help-seeking, Collaboration, and Systematic Cheating. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education* (Bologna, Italy) *(ITiCSE '17)*. Association for Computing Machinery, New York, NY, USA, 238–243. https://doi.org/10.1145/3059009.3059065

[14] Raza Imam, Qazi Mohammad Areeb, Abdulrahman Alturki, and Faisal Anwer. 2021. Systematic and Critical Review of RSA Based Public Key Cryptographic Schemes: Past and Present Status. *IEEE Access* 9 (2021), 155949–155976. https://doi.org/10.1109/ACCESS.2021.3129224

[15] Im Y. Jung and Heon Y. Yeom. 2009. Enhanced Security for Online Exams Using Group Cryptography. *IEEE Transactions on Education* 52, 3 (2009), 340–349. https://doi.org/10.1109/TE.2008.928909

[16] Ravneet Kaur and Amandeep Kaur. 2012. Digital Signature. In *2012 International Conference on Computing Sciences*. 295–301. https://doi.org/10.1109/ICCS.2012.25

[17] C.C. Ko and C.D. Cheng. 2004. Secure Internet Examination System Based on Video Monitoring. *Internet Research* 14, 1 (2004), 48–61. https://doi.org/10.1108/10662240410516318

[18] Dr. Prerna Mahajan and Abhishek Sachdeva. 2013. A Study of Encryption Algorithms AES, DES and RSA for Security. *Global Journal of Computer Science and Technology* 13, 15 (2013), 16–20. Issue 1.0. https://globaljournals.org/GJCST_Volume13/4-A-Study-of-Encryption-Algorithms.pdf

[19] Castro Manuel, Sathiamoorthy Manoharan, Ulrich Speidel, Xinfeng Ye, and Jiayi Zu. 2023. Observations of Cheating Behaviours in Online Examinations and Tools for Mitigation. In *2023 IEEE Frontiers in Education Conference (FIE)*. 1–7. https://doi.org/10.1109/FIE58773.2023.10343363

[20] Đorđe Milošević, Kristijan Kuk, Brankica Popović, and Petar Čisar. 2022. Endangered data in Moodle platform with malicious plugins. In *2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH)*. 1–5. https://doi.org/10.1109/INFOTEH53737.2022.9751251

[21] Alvin Natawiguna and M.M. Inggriani Liem. 2016. Virtualization methods for securing online exam. In *2016 International Conference on Data and Software Engineering (ICoDSE)*. 1–7. https://doi.org/10.1109/ICODSE.2016.7936145

[22] Philip M. Newton and Keioni Essex. 2023. How Common is Cheating in Online Exams and did it Increase During the COVID-19 Pandemic? A Systematic Review. *Journal of Academic Ethics* (2023). https://doi.org/10.1007/s10805-023-09485-5

[23] F. Noorbehbahani, A. Mohammadi, and M. Aminazadeh. 2022. A systematic review of research on cheating in online exams from 2010 to 2021. *Education and Information Technologies* 27, 6 (2022), 8413–8460. https://doi.org/10.1007/s10639-022-10927-7

[24] Camille F. Rogers. 2006. Faculty Perceptions About E-Cheating During Online Testing. *J. Comput. Sci. Colleges* 22, 2 (2006), 206–212. https://dl.acm.org/doi/abs/10.5555/1181901.1181936

[25] V. Venukumar and V. Pathari. 2016. Multi-Factor Authentication Using Threshold Cryptography. *6th Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI)* (2016). https://doi.org/10.1109/icacci.2016.7732291

A USE OF AI

During the preparation of this work the author(s) used ChatGPT in order to assist in debugging code, improving the readability of this research paper as well as organizing figures and tables. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the work.