Reliable and secure on-premise e-assessment with psi-exam

Andreas Kirsch\textsuperscript{1}, Dominik Herrmann\textsuperscript{2}

Abstract:
Interest in electronic assessment (e-assessment) for higher education is growing, but challenges related to security, deployability, and resource demands persist. Recognizing the shortcomings of current electronic on-premise exam systems, this paper presents psi-exam, an e-assessment solution that combines a simple technological design with low-effort organizational measures to achieve high reliability without compromising security. In psi-exam, students take exams on university-provided laptops, which operate independently during the examination. Developed iteratively through stakeholder engagement, psi-exam has been tested in three real-world exams, each with over 140 participants. Post-exam surveys showed generally positive ratings by examinees. Our contribution shows that an alternative approach to e-assessment can result in a simple, reliable, secure, and flexible system for on-premise examination.

Keywords: e-assessment; electronic exams;

1 Introduction
Despite extensive research into e-assessment methods [Ba21], there are still high barriers [BH23] to their adoption in higher education. Bring-your-own-device (BYOD) approaches (e.g., [FSJ17]) suffer from various hardware challenges, while solutions that rely on computer pools (e.g., [BF21]), scale poorly to the needs of entry-level courses with hundreds of examinees. In addition, many solutions rely on familiar Learning Management Systems (LMS) such as ILIAS (e.g., [Reg23]) or Moodle (e.g., [GLP21]), often resulting in inflexible yet complex and support-intensive e-assessment environments. What is missing is an e-assessment system that can handle arbitrary exam tasks and allowed aids, is easy to deploy and maintain, robust against failures, and secure against attacks.

Our research addresses this gap. We propose psi-exam [KH24] whose design is rooted in the practical stakeholder requirements of a mid-sized university. Specifically, we ask: How can we design an e-assessment system that overcomes the limitations of existing solutions? Drawing on the approaches of Design Science Research [Pe07] and Action Research [Gi17] and recognizing the shortcomings of current electronic on-premise exam systems, we created a system that aims for technical simplicity.

\textsuperscript{1}Universität Bamberg, Lehrstuhl Privatsphäre und Sicherheit in Informationssystemen, 96045 Bamberg, Deutschland, andreas.kirsch@uni-bamberg.de
\textsuperscript{2}Universität Bamberg, Lehrstuhl Privatsphäre und Sicherheit in Informationssystemen, 96045 Bamberg, Deutschland, dominik.herrmann@uni-bamberg.de

doi: 10.18420/sicherheit2024_012
Besides striving for simplicity, we mimic the familiar processes and look-and-feel of paper-based exams, where appropriate. This improves the acceptability by a university’s administration. Unlike other e-assessment approaches (Sect. 2), we minimize infrastructure and third-party dependencies. Students provide responses to exam tasks in a web browser that runs locally on hardened exam laptops provided by the university. Besides the browser, auxiliary applications can be preinstalled as allowed aids. During the exam, the laptops are isolated from the network and do not depend on external services, resulting in high reliability. For convenience, the design includes a supervisor laptop that can be used to distribute exam tasks to the examinee laptops and collect examinees’ responses, which are stored as plaintext files. Plaintext files improve traceability, can be batch-processed during grading, and simplify post-exam reviews and archival, topics which are out of the scope of this paper.

In this paper, we describe the requirements and design of psi-exam (Sect. 3 and 4, respectively), which have been iteratively refined through stakeholder dialog, a pilot exam, and evaluation through student feedback (Sect. 5). Based on our experience with three exams with 142 to 242 participants from an introductory computer science course, we are confident that psi-exam meets the necessary security, flexibility, and deployability requirements to make it a good fit for real-world applications.

Our contribution is twofold. First, this paper informs the field of applied information security by concisely presenting an intentionally simple solution consisting of organizational and technical measures. Second, our efforts contribute to the discourse on e-assessment systems by constructing a solution that responds to the complexities of security, flexibility, and minimal resource dependency. Our work also opens up new opportunities for research in higher education, such as providing real-time feedback to students on selected tasks during the exam, a feature we are currently experimenting with.

2 Related Work

Two common approaches for on-premise e-assessment deployment are (1) utilizing existing computer pools and (2) adopting a bring-your-own-device (BYOD) model. The BYOD approach varies in system control, from installing specialized browsers like the Safe Exam Browser [SEB23] to booting dedicated OS images on personal laptops. Many approaches rely on Learning Management Systems (LMS) like Moodle (e.g., [GLP21]) or ILIAS (e.g., [Reg23]), demanding the availability and secure configuration of these LMS. Comprehensive e-assessment solutions like Dynexite [Dyn23] entail high complexity and substantial infrastructure requirements, including power, network, and server availability. Research in this field is, for instance, concerned with sophisticated high-availability architectures [GPS22] and identity verification measures, e.g., via a Public-Key Infrastructure [KPS17] and multi-factor authentication [Ma21]. BYOD implementations, such as those developed within PePP [Pep23] and at the Alpen-Adria-Universität Klagenfurt [FNS19; FSJ17], face hardware challenges, e.g., related to UEFI boot complexities; other solutions
depend on complex multi-tier software stacks [BF21; GPS22] and external services like GitHub [KPS17], requiring extensive expertise for support and maintenance.

Especially with the popular BYOD approaches, maintainability and reliability issues are not the only concerns. Vulnerabilities and fundamental design flaws in tools like the Safe Exam Browser, as identified by Cronqvist and Kortesaari [CK23] and Dawson [Da16] demonstrate the difficulties of securing these complex environments. The alternative approach, i.e., relying on existing computer pools, may reduce complexity, but comes at the cost of poorer scalability. But even comparatively lightweight systems like YAPS [BF21] and Examuntu [GLP21], which focus on building boot images for university computer pools, typically assume the availability of a Moodle server, DHCP servers, and Domain Controllers.

While the mentioned approaches have been practically deployed, they do, in our opinion, not sufficiently address the constraints of many universities. When asked about barriers to the adoption of e-exams, examiners at a recent symposium cited, among others, insufficient examination spaces for large classes and a lack of technical and pedagogical support personell [BH23]. This is why we focus on building a solution that can handle hundreds of examinees in one room, has low network and server infrastructure demands, and does not involve sophisticated technologies.

3 Requirements

We built psi-exam iteratively, alternating between gathering requirements and refining the design. Throughout its development, we talked to all the relevant institutional stakeholders at our university, namely the examination office, legal department, IT services, and facilities management, as well as the university’s board of directors and registrar. We also spoke with examiners, students, and supervisors from several universities to learn from their experiences with e-assessment solutions.

To concisely describe psi-exam and explain its design tradeoffs (see Sect. 4), we present only the most important (non-)functional requirements. For ease of understanding, the requirements are organized according to the FURPS+ [Gr92] categories. Since this paper focuses on the components used during an exam, we will not describe our tools for authoring exams, grading student responses, conducting post-exam review, and archiving exams.

The FURPS+ requirements classification was proposed by Grady at Hewlett Packard [Gr92]. FURPS+ considers the following classes of requirements: functionality, usability, reliability, performance, supportability. Moreover, it includes constraints (the “+”) regarding the overall system architecture (e.g., examinees’ responses must be stored in plain text files), its implementation (e.g., use Linux on laptops), interfaces to other systems (e.g., exams must be archived on paper), and physical environment (e.g., rooms where exams take place have neither reliable Wi-Fi nor power). Requirements are in the present tense because they are properties of the current psi-exam design.
In FURPS+, **functional requirements** capture the main features desired by the stakeholders. Like paper-based exams and as shown in Fig. 3 in the appendix, *psi-exam* supports arbitrarily designed tasks consisting of text and images (R1) and free-text responses typed into text input fields (R2). Support for drawing figures and mathematical notation could be implemented in future work. Supervisors have full control over the start and end times of an exam (R3), so examinees can arrive late (unless another examinee has left) or leave early, and people with disabilities can get individual time extensions. Most importantly, *psi-exam* ensures fairness and equality of opportunity by guaranteeing confidentiality, integrity, authenticity, and non-repudiation: First, it is impossible to submit responses to exam tasks in the name of other examinees (R4); second, it is impossible to impersonate the supervisor’s laptop (R5); third, the exam tasks cannot be obtained before the exam starts and cannot be tampered with (R6); fourth, the responses of other examinees cannot be obtained and cannot be tampered with (R7); fifth, the responses submitted to the supervisor laptop cannot be tampered with (R8). In addition, examinees cannot use the provided laptop as a means of communication (R9).

**Usability requirements** address effectiveness and efficiency of use, as well as user satisfaction. We consider three main usability requirements. First, responses are saved by pressing a button to prevent unintentional changes (R10). Second, as shown in Fig. 4 in the appendix, examinees explicitly hand in their exams, giving both supervisors and examinees clarity about the final version of their responses (R11). Third, the task layout is responsive, allowing examinees to change the font size or window width (R12).

**Reliability requirements** have a high priority in *psi-exam*. Once the exam has started, malfunctions or attacks must have minimal impact. Network and power failures during the exam are tolerable because the examinee laptops run autonomously (R13). If a laptop fails, the system allows for low-effort failover, i.e., the most recent version of an examinee’s responses can be transferred to another laptop at any time (R14). Examinees’ responses are backed up to multiple locations during and after the exam (R15).

**Performance requirements** are relevant not only to the examinee during the exam, but also to those who set up the exam infrastructure (quick setup improves acceptance). The laptops used for the exam are prepared and charged on the days before the exam. On the day of the exam, preparations require minimal effort (R16): The laptops can be placed on the tables in any order, which significantly speeds up room setup. For convenience, the laptops retrieve the exam tasks from a supervisor laptop before the exam starts, and the solutions are collected on the supervisor laptop after the exam (R17). During the exam, the exam application runs completely locally, with response times in the millisecond range (R18). Switching to a spare laptop in case of failure takes a few minutes (R19).

**Supportability requirements** ensure long-term use and high acceptance of the solution. To this end, system architecture, concepts, and processes are intentionally kept simple; software dependencies are kept to a minimum (R20). Supervisors do not need to know the
technical intricacies of the system (R21). Since the exam does not require a central exam server, psi-exam can handle hundreds of examinees simultaneously (R22).

psi-exam is subject to various system constraints ("*"), The most important physical constraint relates to facilities: To reduce the number of supervisors required, electronic exams take place in large lecture halls, which have unreliable network uplinks and very few electrical outlets (hence the need to use laptops). An important design constraint is the use of dedicated exam laptops with a custom system image, rather than relying on the bring-your-own-device concept (R23), to reduce complexity while maintaining security. To maximize transparency and traceability, response handling is intentionally lightweight. Each response entered by an examinee is stored in a separate plaintext file on disk "as is" and is not modified by the software (R24). The most notable implementation constraint is the choice of established open source software to ensure continued support. There is also a desire to make psi-exam acceptable for all involved divisions of the university administration (interface constraints). To this end, for now, several administrative processes mimic paper-based exams (R25): For instance, at the start of the exam, examinees sign a physical exam cover sheet, acknowledging that they have been informed of the exam conditions. The signed sheet is collected at the end of the exam and archived with a hard copy of the responses.

4 Design

With psi-exam, we aim to achieve a level of reliability and security comparable to that of paper-based exams, i.e., the e-assessment environment should not introduce additional attack vectors. To this end, we follow a defense-in-depth approach that combines several simple technological and organizational security measures to avoid single points of failure. As a result, our security assumptions are very conservative: psi-exam does not require a permanent network connection during the exam and its security does not rely on the confidentiality of the exam laptop’s disk image. While the design includes a Wi-Fi to conveniently manage exam laptops during an exam, the security does not depend on the secrecy of the Wi-Fi password (pre-shared key). In general, attacks can increase the effort for supervisors to conduct an exam, but they cannot compromise fairness and equality of opportunity.

Before explaining the components and processes in detail, we provide a high-level summary of the exam procedure (see Fig. 1), which consists of three phases: first, a pre-exam preparation phase in which supervisors put the laptops on tables, initiate the exam download, and scan the device tokens shown immediately after boot. No examinees are allowed in the exam room during this phase. Second, the exam phase, during which examinees enter the room, the exam is decrypted, examinees sign the physical exam cover sheet (cf. R25), and examinees work on the exam tasks. This phase ends when the exam is finished. Then, the supervisors verify short submission hashes that are shown on screen when an exam is handed in and have been written onto the physical exam cover sheet by the examinees (see Fig. 5 in the appendix). After that, the examinees leave the room. Third, in the post-exam...
Fig. 1: Procedure of an e-assessment using psi-exam, showing the steps carried out by the supervisors (upper part) and by the examinees (lower part). Horizontal ordering expresses dependency.

Fig. 2: Components of psi-exam after the exam was downloaded from the supervisor laptop, decrypted by entering the exam key, and the examinees (Alice and Bob) began answering the exam questions.

phase, supervisors check that all submissions have been uploaded to the supervisor laptop, and, finally, reset, shutdown, and collect the exam laptops.

In the following, we provide details about the components of the system (cf. Fig. 2), which consists of university-provided examinee laptops and a supervisor laptop (cf. R23). All laptops are connected to an isolated, password-protected exam Wi-Fi network with enabled client isolation, which prevents communication between examinee laptops (cf. R9).

4.1 Examinee Laptop

The examinee laptops are placed on each examinee’s desk in the exam room. To prevent examinees to read the content of their neighbors, we restricted the viewing angle of the
screens by applying privacy filters (cf. R7, R9). The laptops are randomly distributed among the desks. Since examinees are required to take a particular desk identified by its number and correct seating is checked by the supervisors, targeted manipulation of a particular laptop becomes infeasible (cf. R16). In addition, spare laptops are kept available (cf. R14, R19). To prevent examinees from working on the exam tasks after the end of the exam time, the laptops must be closed when announced by the supervisors (cf. R3).

All examinee laptops run an identical system image, which is updated prior to the exam using mass deployment techniques (e.g., with partclone and udpcast). The system image contains the software required for the exam but not the exam tasks. The exam tasks are downloaded in the exam room shortly before the exam, which reduces the risk of unauthorized access (cf. R6).

**Operating System.** The operating system (OS) on the examinee laptops is a hardened Debian Linux with Gnome. The Debian distribution was chosen for its reliability and long-term maintainability (cf. R20). The examinees have only access to a user account with limited capabilities (cf. R6). The OS prevents access to its network configuration, i.e., examinees cannot connect to networks under their control, e.g., mobile hotspots (cf. R9). Furthermore, all laptops use a sufficiently strong BIOS password to prevent adversaries from booting from external devices.

Every one to two minutes, the OS attempts to connect to the exam Wi-Fi to obtain a job from the supervisor laptop and to upload the examinee’s responses to the supervisor laptop (cf. R17). The OS then disconnects from the Wi-Fi, which minimizes load on the network (cf. R18, R22). Every uploaded submission is encrypted using age [Va20], which implements a hybrid authenticated encryption scheme. Encryption uses the public key that belongs to the supervisor laptop (pub_supervisor), which prevents eavesdroppers on the network from accessing or manipulating the submissions (cf. R7, R8). Besides the examinee’s responses, every submission contains a random nonce to prevent adversaries from replaying previous submissions (cf. R4, R8) as well as the laptop’s device token that is randomly generated during its first boot and is used to identify the examinee.

During the exam, the uploaded submissions serve as a backup of the examinee’s work on the exam. Another backup is stored on an always-connected USB flash drive (cf. R15) that is updated whenever a response is entered or changed. In the event of a laptop failure, a supervisor removes the flash drive, plugs it into a spare laptop, and executes a single command to restore the most recent version of the responses. Thus, the examinee can continue with minimal recovery time (cf. R19, R21).

**Exam Web Application.** The exam web application is a locally-running Flask [Ro10] web service (cf. R18, R20) that fulfills four tasks. Firstly, it generates and displays the device token. The device token is a non-guessable random value that identifies a laptop, is not known to examinees, and is added to every submission (cf. Fig. 2). The device token is only shown after the initial boot of a laptop. Before the examinees enter the room, supervisors
quickly record the mapping between examinees and laptops by scanning a corresponding on-screen QR code and a QR code on the physical exam cover sheet, which encodes the seat number (cf. Fig. 1). Like in paper exams, supervisors check the identity of the person at every seat and whether every examinee has taken their preallocated seat by checking an official proof of identity (e.g., national ID card). When combined, these measures detect attempts of identity spoofing (cf. R4). Furthermore, the obtained device tokens serve as a whitelist of valid tokens. This whitelist defends against adversaries who have obtained the laptop image and, thus, can access the exam Wi-Fi. Such adversaries cannot upload valid submissions since they cannot present a valid device token.

Secondly, the application decrypts the encrypted exam tasks once they have been downloaded from the supervisor laptop. The exam tasks are encrypted to prevent eavesdroppers on the Wi-Fi from learning the exam tasks prematurely (cf. R6). Decryption of the exam is supposed to take place when the exam begins. This is achieved as follows. The exam tasks are encrypted using age on the supervisor laptop with a separate key pair (exam key). For decryption, the examinees’ laptops need the corresponding private key. This private key is split into two shares. The first share is downloaded from the supervisor laptop along with the encrypted exam tasks. The second share has to be entered by the examinees into the web application. This share is announced or distributed on paper once all examinees are seated, no further communication is tolerated, and the exam is about to begin (cf. R3).

Thirdly, the web application allows the exam tasks to be viewed using a browser (cf. R12). The browser displays the task description, represented in HTML (cf. R1), and input fields for the examinees to enter their responses (cf. R2). The responses are stored as plaintext files in the file system (cf. R20, R24). To save the response for a given task, the examinee explicitly presses a button next to the corresponding input field (cf. R10).

Fourthly, the web application calculates and displays the submission hash when an examinee hands in the exam. The submission hash is a truncated SHA256 hash of that examinee’s response files (including the device token). Once the hash is displayed, no further changes can be made to the responses. By writing the hash on the physical exam cover sheet, an examinee commits on the submitted responses. This measure prevents examinees from claiming that the responses that were graded by the examiner differ from the ones that they handed in (cf. R11).

4.2 Supervisor Laptop

The supervisor laptop serves as the controller of the exam and, therefore, contains sensitive information. Confidentiality and integrity are maintained by storing it in a secure location and full-disk encryption (cf. R6). Also, an identical laptop is available as a spare (cf. R14).

During the exam, the supervisor laptop fulfils three tasks. Firstly, and most critically, it allows the examinee laptops to download the encrypted exam tasks.
Secondly, the supervisor laptop can be used to conveniently orchestrate the exam by instructing the examinee laptops to, e.g., download the exam, upload submissions periodically (cf. R17), triggering the final submission hash verification procedure at exam end, and resetting and shutting down the laptops. Exam tasks and job announcements are signed with \textit{minisign} [De22] using a private signing key (\texttt{priv\_supervisor}) and additionally protected against replay attacks. Examinee laptops verify the signatures using the corresponding public key \texttt{pub\_supervisor} and reject messages with invalid signatures. This prevents adversaries from impersonating the supervisor laptop and distributing bogus jobs or exam tasks (cf. R5).

Thirdly, the supervisor laptop stores the uploaded submissions. At the end of the exam, it allows each examinee laptop to check whether the most recent version of its uploaded submission matches the most recent version stored locally on the laptop (cf. R11). The result of the verification is displayed on screen, which signals to supervisors that this laptop can be reset and collected. Note that this is merely a convenience feature; network failures during or after the examination time have no critical impact (cf. R13). If uploads and checks cannot be conducted automatically, for instance, because an adversary jams the Wi-Fi network, submissions can still be collected network-independently from the USB flash drives.

Finally, at the end of the exam, all submissions are copied from the supervisor laptop to two encrypted USB drives that are carried by different supervisors to prevent data loss (cf. R15).

5 Evaluation

\textit{psi-exam} has been evaluated in three exams so far, a pilot exam \((n = 155\) registered examinees), and two final exams in August and October \((n = 242\) and \(n = 142\)). The aim of the evaluation was to understand obstacles and design flaws. Based on feedback from the pilot exam, we streamlined the procedures and improved the user interface of the exam, e.g., by alerting students to unsaved responses.

While the supervisors were collecting the exams, the examinees were filling out feedback forms. In the following, we report only the results for the August exam \((127\) forms with at least one question answered), since the October evaluation shows similar results. Very few examinees reported technical \((5\%)\) or organizational \((4\%)\) problems. For example, some reported difficulties using Gnome or the keyboard because they were Mac or tablet users. In response, we created a tutorial video that included a preview of all the procedures and instructions on how to use the laptops (e.g., changing font size and typing special characters). When asked for advice for future examinees, several examinees recommended bringing a mouse.

81\% of the 84 examinees who answered this question said that the “real-time feedback” provided for some of the input fields was helpful \((19\%\) were confused by it). In terms of allowed tools, 86\% of the examinees reported using a calculator, 70\% the Linux shell, 27\% the provided web pages, and 22\% the provided text editor.
6 Discussion and Future Work

Like other e-assessment systems, psi-exam has limitations. In this paper, we focus on three key issues relevant to cheating and reliability that also affect comparable systems. First, the availability of the supervisor’s laptop during the exam download phase could be compromised by adversarial actions such as Wi-Fi jamming. To mitigate this, exams could be preloaded onto USB drives for manual distribution to laptops, although this may result in a delayed start of the exam. Second, allowing students to bring in personal USB mice or keyboards introduces significant cheating risks. Devices such as Rubber Ducky and O.MG Cable [Fa22a; Fa22b], which can inject pre-programmed keystrokes and exfiltrate data to external collaborators, are difficult to detect due to their ability to mimic regular USB devices. One possible countermeasure is for universities to ban personal peripherals and provide standardized equipment. Third, psi-exam faces potential vulnerabilities from its hardware and software components. We reduce this risk by minimizing system complexity, limiting reliance on third-party components, using a minimal Linux distribution, and applying software updates prior to exams.

We consider three activities for future work. First, a comparative analysis of the different e-assessment solutions could be performed. Security aspects and reliability could be assessed by extending the threat modeling approach of Hietanen [Hi21]. In addition, organizational (e.g., complexity of setup, flexibility of use, long-term maintainability) and legal (e.g., privacy compliance) aspects should be considered. The results of such an analysis would provide useful insights into the trade-offs inherent in different approaches to e-assessment in higher education. Second, because psi-exam is deliberately simple in design, it can be easily extended, e.g., with tools for creating exams, toolchains for automated grading and exam archiving, or even the derivation of bootable device images to support BYOD scenarios and remotely proctored online exams. Third, once user-friendly tools are available, we could evaluate the acceptance of psi-exam by examiners.

7 Conclusion

This paper presents psi-exam, an e-assessment solution for higher education that is designed with simplicity in mind. Using a defense-in-depth approach and mimicking paper exams as recommended by stakeholders, psi-exam effectively addresses the challenges of scalability, support, flexibility, and security in e-assessment. Unlike traditional BYOD and computer pool-based systems, it runs independently on university-provided laptops, reducing infrastructure and third-party dependencies to improve reliability and maintainability. We have validated psi-exam in three real-world exams and received positive feedback.

We hope that psi-exam resonates with the security community. By starting with a basic system and only adding necessary measures, psi-exam reminds us that this straightforward approach can lead to a system that is robust yet intuitive to understand. This approach, reminiscent of Gall’s Law, should be considered more often when systems are designed.
References


A Screenshots

This section contains three screenshots of the user interface of the exam web app of psi-exam.

Fig. 3: Screenshot of a task displayed on an examinee laptop. There are unsaved changes.
Fig. 4: Screenshot of the exam hand-in workflow shown to the examinee.

Fig. 5: Screenshot of the hand-in workflow shown to the examinee after the exam has been handed in.