




EA and BYOD: Threat Model and Comparison to Paper-based Examinations

Bastian Küppers^{1,2} ^a, Thomas Eifert² ^b, Richard Zameitat² and Ulrik Schroeder¹ ^c

¹Learning Technologies Research Group, RWTH Aachen University, Ahornstraße 55, Aachen, Germany

²IT Center, RWTH Aachen University, Seffenter Weg 23, Aachen, Germany

Keywords: Electronic Examinations, Electronic Assessment, Computer based Assessment, Bring Your Own Device.

Abstract: e-Assessment provides a wide range of opportunities to improve students' experience and apply innovative examination methods, therefore gaining growing interest from institutes of higher education. Despite the advantages of EA, considering practical use raises concerns, one of which are security risks, even more when students' devices should be used instead of a centrally managed hardware pool. Approaching these concerns requires to evaluate and break down potential security implications and appropriate mitigations. This paper focusses in particular on relative security in comparison with paper-based examinations, presuming that EA does not have to be perfect, instead it is sufficient to be on par with traditional approaches. With that assumption in mind and based on work by Sindre and Vegendla, a threat model for electronic examinations is developed, taking own research into account. For each threat included in the threat model, a counter measure is proposed. Afterwards, the level of security for EA and the level of security for paper-based examinations are compared. The results of this comparison are quite promising regarding the level of security that EA can offer.


1 INTRODUCTION


E-Assessment (EA) is a topic of growing interest for institutes of higher education (IHE), since EA offers a wide range of advantages over paper-based examinations (PBE) (Conole and Warburton, 2005; Csapó et al., 2011; Küppers and Schroeder, 2017). However, security risks are a concern that is raised when considering EA, especially when it comes to using the students' devices to carry out the EA (*bring your own device*, BYOD) (Dawson, 2015; Søggaard, 2016; Heintz, 2017). To be able to develop decent countermeasures to potential threats, these threats have been determined. Therefore, a threat model for EA in a BYOD scenario has to be developed in order to identify potential threats. However, EA is not equal to EA and one BYOD setting does not equal another setting, as there are many possible ways to conduct EA with students' devices (Küppers and Schroeder, 2016).


In our project **FLEX** (*Framework For FLExible Electronic EXaminations*) we develop an EA framework that utilizes the students' devices for on-campus

examinations in order to provide the students with a known working environment, which turns out to be important for the students (Hillier, 2015). Additionally, by making use of a BYOD approach, the threshold to introduce EA at IHEs can be kept as low as possible, since no expensive computer labs have to be installed and administered. Based on the software architecture of **FLEX**, we will consider a setting for EA where students have to execute an application on their device and a central component, a server, is responsible for handing out the assignments and also for collecting and saving the answers. To have a solid basis for the derivation of the threat model, the software architecture the EA framework that the threat model targets will be discussed in detail.

This paper reviews related work and discusses the software architecture of **FLEX**. Based on these information, an updated threat model is derived. This model is then used to update countermeasures to existing threats and to develop countermeasures to new threats in order to increase security and reliability of EA. Afterwards, the resulting framework for EA is compared to the security and reliability of PBEs. The paper closes with a summary and research outlook.

^a  <https://orcid.org/0000-0002-5882-2125>

^b  <https://orcid.org/0000-0003-1996-0944>

^c  <https://orcid.org/0000-0002-5178-8497>

2 SOFTWARE ARCHITECTURE OF FLEX

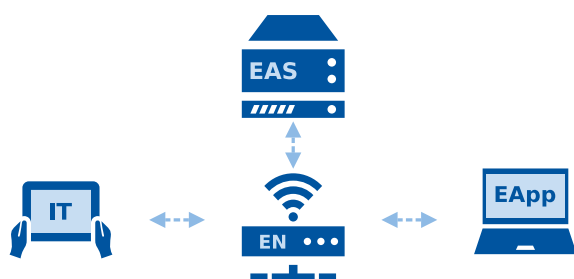


Figure 1: Basic Architecture.

FLEX utilizes a client-server architecture, which consists of four components, as shown in Fig. 1. The particular components and their relation, as well as related processes will be described shortly in the following sections. For more details we refer to our paper *Practical Security for Electronic Examinations on Students’ Devices* (Küppers et al., 2019). **FLEX** is generally designed in a modular fashion to support arbitrary types of assignments, for example programming assignments.

EA Application (EApp). The EA application is executed on the students’ devices. It provides the interface, which is used to solve the assignments of the EA. Additionally, it provides a detection mechanism to identify cheating related actions on these devices (Küppers et al., 2019).

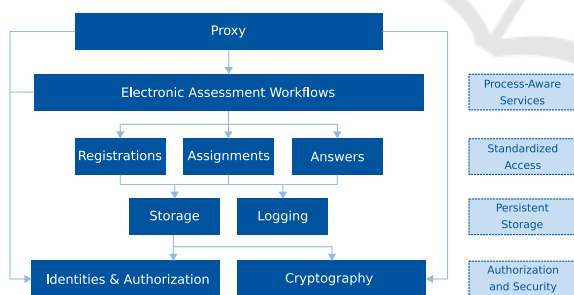


Figure 2: EAS Architecture.

EA Server (EAS) A microservice pattern (Namiot and Sneps-Snepp, 2014) allows reducing dependencies between the different modules of the EAS. Modeling clearly separated functional units that can be maintained and exchanged easily is crucial to not jeopardize the security of the server infrastructure. The EAS therefore separates into four functional tiers and a proxy tier to shield the server from unauthorized access. The process aware tier offers various workflows to support the examination processes. It defines the primary interface for the EA application.

The standardized access tier then allows accessing resources on an equal semantic level and homogeneous nomenclature, that also orients towards the supported workflows. The persistent storage tier then translates generic storage requirements towards storage implementations and concerns file systems, databases and protocols. Last but not least, all tiers base on the authorization and security tier offering information about identities and their roles within the processes as well as strong cryptographic and signing functionalities to secure workflows. To achieve clear separation-of-concerns and allow re-usability of the different modules in the tiers, each one clearly defines interfaces and dependencies. The tiers are then designed in a way that higher tiers may only depend on lower tier modules but not the other way around to prevent circular dependencies.

Invigilator Tablet (IT). The Invigilator Tablet (IT) serves mainly as the replacement for paper-based registration lists. It can download the current registration list from the EAS and invigilators can use this digital copy of the list in a registration procedure similar to the one used in current paper examinations. The students can sign their attendance to the examination on the tablet. Additionally, the IT has the possibility to verify the connection of a client in order to avoid cheating (Küppers et al., 2019).

Examination Network (EN). The connection between EA application and EAS is established over a special examination network (EN), which has to be the only way to access the EAS. During an EA, the user accounts for registered students are transferred from the regular network, for example *eduroam*, to the EN. Additionally, each student can use the credentials for the EN only to establish one connection and connection between students are prohibited.

3 RELATED WORK

Sindre and Vegendla (called *SV* hereafter) did a similar analysis in their paper *E-exams versus paper exams: A comparative analysis of cheating-related security threats and countermeasures* (Sindre and Vegendla, 2015), which will serve as a basis for this paper. In their paper they model threats to EA with Attack-Defense-Trees (ADT) (Kordy et al., 2011) and derive a reasonable model. However, research in our project **FLEX** suggests that the threat model has to be updated in order to keep track with technological development. Additionally, some of the proposed countermeasures are not allowed by law in Germany or

are not suited for **FLEX** for other reasons, so that we have to come up with different solutions. Particularly important are the statements that “paper-based exams do not have perfect security either [...]. Hence, if e-exams have advantages in other respects they need not have better security than traditional paper-based exams, only a similar level of security.” and that “it is not at all obvious that BYOD e-exams will generally be less secure than paper exams. Rather, this will depend on the exact implementation of the paper exam, and of the e-exam”. Following these insights, EA is exempted from the need of providing perfect security, but could be widely accepted if the same threshold for security as in PBEs is within reach.

Beyond the work of *SV*, other papers that deal with the security of summative examinations. Apampa et.al. analyzed impersonation as a particular threat to EA in their paper *User Security Issues in Summative EA Security* (Apampa et al., 2010). This paper targets a biometric authentication of the students by pointing out that any other sort of authentication can be circumvented by sharing credentials or similar measures. Even for invigilated exams, it is pointed out that the invigilator could be “collude with fraudulent students to allow the fraudulent act”. We are not able to rely on biometric authentication, because in a BYOD scenario we cannot guarantee that every student has appropriate hardware available. However, as in our scenario the examiner is also the invigilator of an examination, we consider the possibility of a fraudulent collusion negligible. However, humans as the weak link of an information security system is something that has to be considered in general (Met-alidou et al., 2014).

Kiennert et. al. describe in their paper *Security Challenges in EA and Technical Solutions* (Kiennert et al., 2017), which was written in the context of the TeSLA project of the European Union’s *Horizon 2020* project, a technical infrastructure which is secured by using a public key infrastructure (PKI). This approach will also be taken to resolve some of the threats described later on. However, the TeSLA Project has a different aim, as it deals with remote examinations, whereas the focus of **FLEX** are on-campus examinations.

4 THREATS TO EA

SV identified seven potential threats to EA and stated #1, #2 and #4 as the most important threats: Impersonation (1), Assistance / Collaboration (2), Plagiarism (3), Use of Unauthorized Aids (4), Timing Violations (5), Lying to Proctors (6) and Smuggling Out

Exam Questions (7). As for *SV*, #7 is not a severe threat in our scenario and the same holds for #6. Additionally, both threats can also occur in PBEs, which makes them not special to EA. Plagiarism may be a threat to EA if students hand in other students’ solutions for simple and schematic assignments, where the plagiarism may not be obvious. This could be, for example, the case for simple programming assignments. However, this may also be considered to be some sort of *assistance* or *collaboration*, which is why we will not discuss it in detail. Timing violations may not be a severe threat, but there are effective countermeasures to timing violations in EA.

In order to get a reasonably complete threat model for EA, at least one threat has to be added to the list derived by *SV*: *Manipulation of the Exam Results*. However, even then the list may not be complete, because there could exist threats yet unknown.

Assistance / Collaboration & Use of Unauthorized Aids. The ADT derived by *SV* for *Assistance / Collaboration* is depicted in Fig. 3. According to the ADT, every possible threat is resolved, since there is no threat (oval shapes) that is not answered by a counter measure (rectangular shapes). For the left subtree (*Traditional in-room Communication*) we agree to *SV* that this is, when implementing the proposed countermeasures, not a threat. However, for the right subtree (*Distance Communication*) we argue that there are unresolved threats. First, banning laptops in an EA is obviously not working, especially when utilizing a BYOD approach. Hence, the threat is unresolved. Additionally, mitigating wireless communication as proposed in the ADT does not work in Germany. We can not conduct body searches and jamming is not an option either, because both are not within the law. Scanning for ad-hoc wireless network and bluetooth connections might work, however, if the students use the cellular network there is nothing suspicious that could be detected. In our paper *Practical Security for Electronic Examinations on Students’ Devices* (Küppers et al., 2019) we discussed security measures for these threats under the previously stated principle that EA has not to be bullet proof, but has to offer reasonable security in comparison to PBEs. Based on our paper we propose the update to the right subtree of the ADT which can be seen in Fig. 4.

It will never be possible to hinder students from going to the toilet during an examination. Hence, it will not be possible to prevent students from cheating on the toilet, especially if body searches are not allowed. Therefore, the only possibility to at least mitigate cheating on the toilet is to reduce the portability of the questions, as proposed by *SV*. As banning

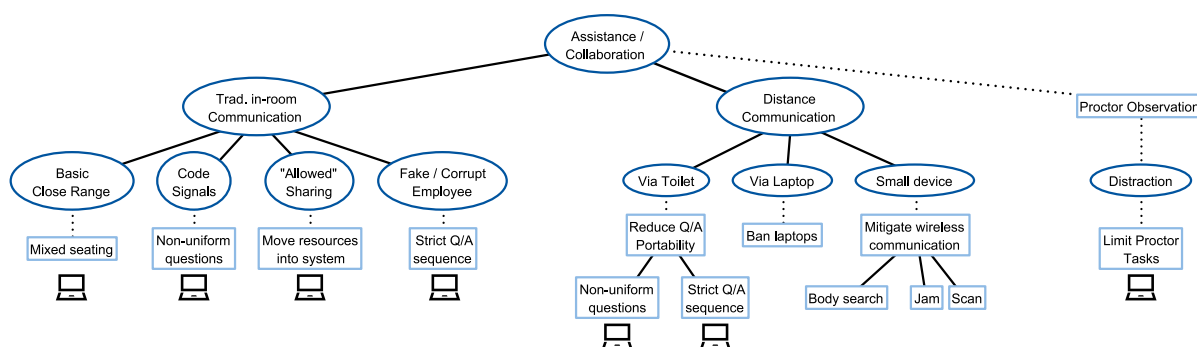


Figure 3: ADT for Assistance / Collaboration (Sindre and Vegdla, 2015).

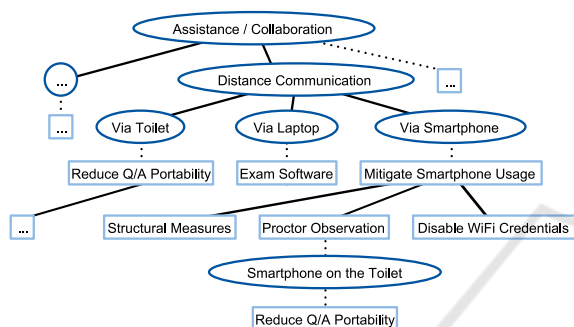


Figure 4: Updated ADT for Assistance / Collaboration.

laptops from EA is not possible, there has to be an exam software on the laptops that mitigates the possibilities of cheating and communication on the devices during an examination. In our paper *Beyond Lockdown: Towards Reliable e-Assessment* (Küppers et al., 2017) we tackled this threat and proposed an approach for a software, which does not try to lock down the system but relies on monitoring and logging, as lock down approaches seem to have security issues (Søgaard, 2016; Heintz, 2017). Additionally, our approach does not require admin rights on the students' devices, since students may not be willing to grant these, and does not integrate itself deep into the operating system, as this could be problematic in terms of data privacy. That exam software can also be used to resolve the remaining threat for *Use of Unauthorized Aids* (see Fig. 5). In the same way as for *Collaboration / Assistance*, it can monitor accesses to the local harddrive and therefore to unallowed aids on the laptop, which leads to an update of the ADT as depicted in Fig. 6.

Last, the use of smartphones for cheating can be suppressed by structural measures, i.e. building an examination room which resembles a Faraday cage. However, even by building such an examination room, that would still allow the students to use their smartphones with the university's WiFi network, which has to be available in the examination room in

order to conduct EA. Hence, the students' WiFi credentials have to be disabled during the examination, so that it is only possible for the students to connect their laptop with provided credentials for the EA, but not to use their smartphones with the regular WiFi network, e.g. *eduroam*. If structural measures are not possible, the proctors in the examination room have to observe the students for illegal use of their smartphones. Again, this is not possible on the toilet. Hence, students can potentially use their smartphones for communication on the toilet. However, this is not a threat which is special for EA, but can also occur in PBEs.

Impersonation. The threat model for *Impersonation* derived by SV, see Fig. 7, does not take advantage of digital encryption mechanisms (Kaur and Kaur, 2012). Hence, we introduce those into the threat model as depicted in Fig. 8. For both main threats, *Spoof Candidate* and *Label Swapping*, we added digital certificates using a *Public Key Infrastructure* (PKI), for example the DFN PKI (Deutsches Forschungsnetz,). A digital certificate, which is validated against the PKI, can be used to identify a student. At first sight, a certificate could be handed to an impostor, just like username and password as indicated by SV. However, if a general certificate is used for a whole student lifecycle, that certificate key is much more valuable to a student than login credentials for an EA account or a one-time private certificate. That certificate in combination with a certificate used by the examiner can also be used to prevent the manipulation of examination results (see Section 4). The first part of that process can be seen in the right subtree in Fig. 8, where a digital signature was added to the ADT. The step *Sign Results via PKI* includes a digital signature by the student, which resembles the manual signature in a PBE, and another digital signature by the examiner, which resembles an examiner checking that label.

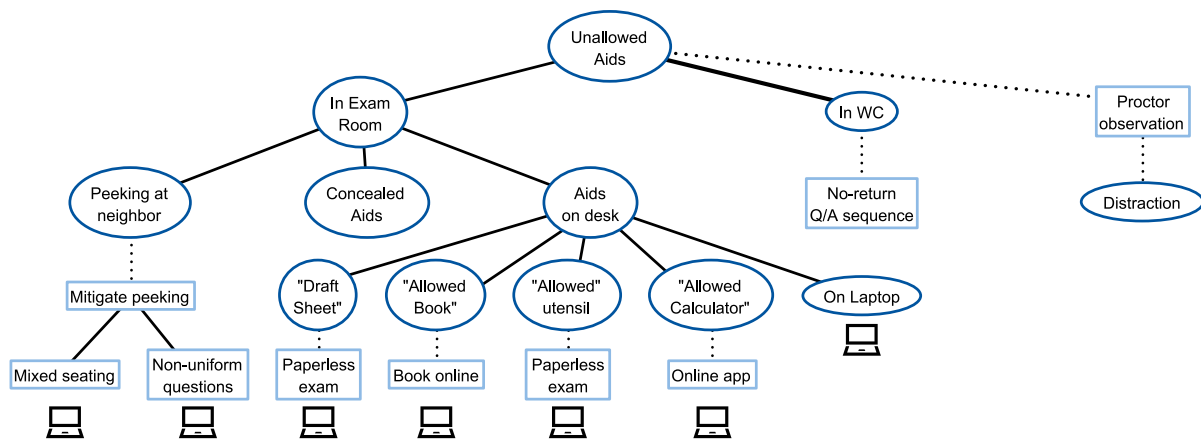


Figure 5: ADT for *Use of Unauthorized Aids* (Sindre and Vegndla, 2015).

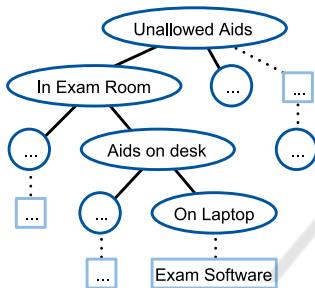


Figure 6: Updated ADT for *Use of Unauthorized Aids*.

Timing Violations. The threat model for *Timing Violation* is rather simple, as students could only start early or hand in late. For PBEs that may indeed be a problem. The exam sheets have to be handed out prior to the exam in order to enable all students to start simultaneously. Hence, it is possible that some students start to work on the assignments before all students have the sheets and are unnoticed by the proctors. The same holds for the end of the exam, where all sheets have to be collected in order to ensure that no students are working on the assignments anymore. Again, it is possible that some students do not stop their work after the end of the exam until their answering sheets are collected. For EA these threats, even if those are not severe treats as stated by *SV*, can easily be countered. The assessment server is the key component to the solution, as it serves the assignments simultaneously to all students and does not accept answers from students anymore after the deadline has passed.

Manipulation of the Exam Results. The manipulation of exam results, i.e. the answers that students handed in or even final grades determined by the examiner, are a severe threat, even if that threat is not very likely to occur. Even if the server of the EA framework does not accept answers anymore after the

exam's deadline has passed, that is no guarantee per se that the answers of the students are not modified afterwards by both, student or examiner. In a PBE that would be comparable to a student breaking into the examiner's office and attaching new sheets with answers or the like or, on the other side, an examiner disposing sheets for different reasons, in the best case (or worst case, depending on the point of view...) by accident if a sheet is simply lost. In EA, there are no hard copies of the students' answers, which means two things. First, there is nothing like a handwriting to identify a student's set of answers. Second, there is no need to break into a physical office anymore, but an attack can be carried out from home - in theory. Hence, as stated earlier (see Section 4), the students' results are signed by two parties: the students themselves and the examiner. Thus, an examiner can not dispose a part of a student's set of answers, because the remaining subset was not signed by the student. On the other hand, the student can not modify the results later on, because these changes are not signed by the examiner. Still, the examiner could dispose the whole set of answers of a student. To prevent this, the students get a receipt signed by the assessment server, not the examiner, at the end of the exam, which proves they handed in. The worst case would be a student who acquires the examiner's private key so that it would be possible to sign a modified dataset after the exam's deadline has passed. In theory that is possible, however, this is a threat that is omnipresent in the cyberspace, as online banking or the like rely on the ability of a server to keep a private key secret. Returning to the principle that EA has not to be bullet proof, it can be assumed that it is good enough for EA if it is good enough for online banking. With the same reasoning, final grades can not be changed after, except by the examiners themselves, the exam was corrected if the list of grades is

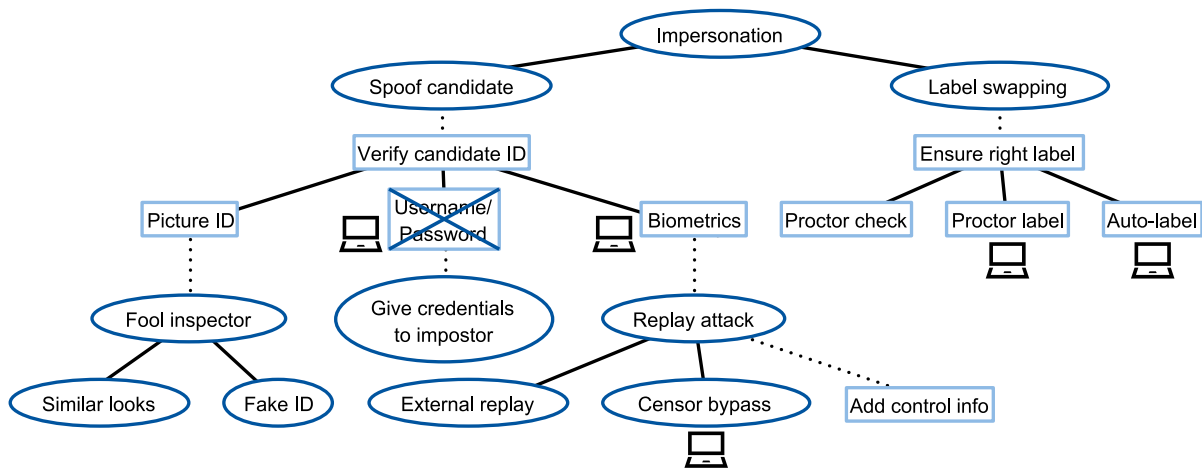


Figure 7: ADT for Impersonation (Sindre and Vegdla, 2015).

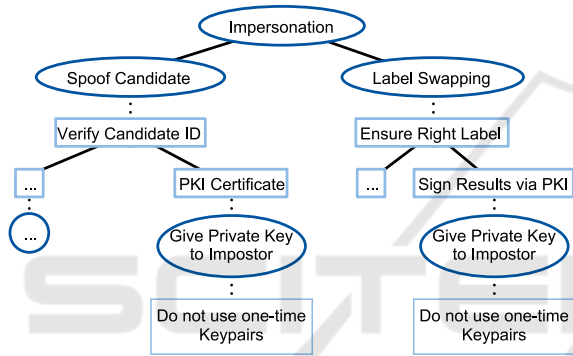
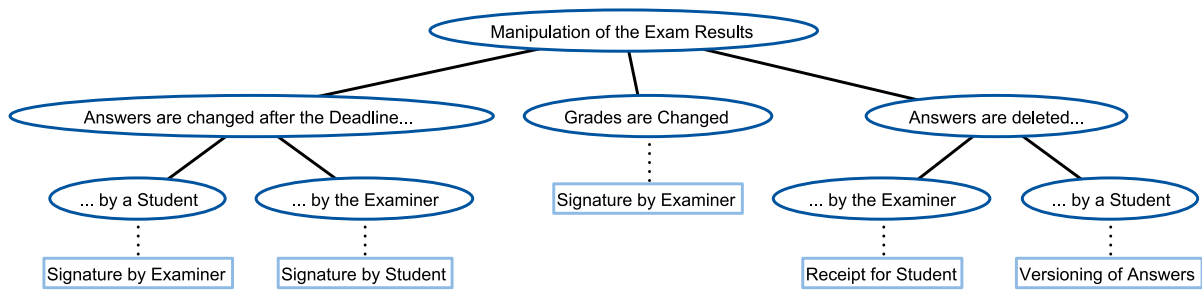
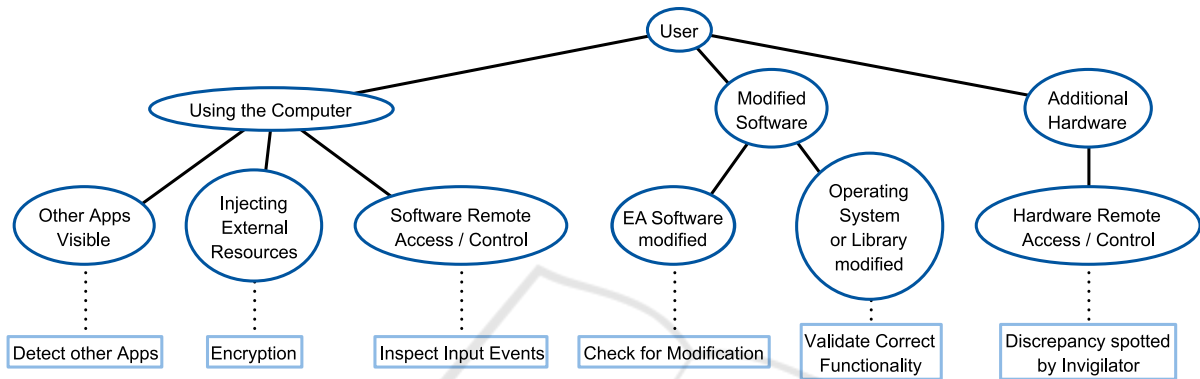


Figure 8: Updated ADT for Impersonation.

signed by the examiner. Additionally, the destruction of the results that were handed in by a student has to be prevented. This scenario could be invoked by a student who rather deletes everything that was handed in instead of getting a bad grade. However, this could only be achieved by hacking into the system, because there is no legit way for a student to delete everything after the exam is over. This can easily be prevented by digitally signing the whole set of the students' answers immediately after a change occurred, i.e. after a student saved answers to the EAS. This can be done repeatedly during the exam, because this way it is ensured that every valid state of the set of answers is digitally signed. Hence, there is no way for a student to delete a single set of answers, because that would change the whole set of answers which is detectable. To prevent the loss of digital data in general, the data has to be replicated across at least two independent storage systems in order to minimize the risk of hardware failure. But again, that is no problem of EA in particular, but a problem of reliable data storage in general, which can be considered solved satisfyingly (Stanek and Eifert, 2012).

Threat Model for FLEX . Since *Exam Software* was proposed as a countermeasure in the threats *Assistance / Collaboration* and *Unallowed Aids*, a dedicated threat model for **FLEX** has to be developed in order to ensure that the EA software can in fact serve as a suitable counter measure. Hence, corresponding requirements have to be identified that **FLEX** needs to fulfill to be able to do that. Most importantly, it has to be ensured that the EA software works as intended. Thus, it is essential that the software is not modified and that it does not run in a modified environment. Hence, a way has to be found to check on-the-fly whether the EA software and the execution environment on the students' devices are unmodified. However, even if the EA software itself is unmodified, there is the risk that additional software is used during the exam, for example remote access / control software, which could be used to get help from students outside the examination room. Hence, it is important to monitor the working environment during the exam, as discrepancies between the expected behavior of the EA software and extraordinary behavior in case of cheating can be detected this way. In order to prevent the injection of external data into the network traffic, the communication between the EA software and the EAS needs to be encrypted. If students take the extra step to utilize additional hardware, e.g. for remote access, it is really difficult for the EA software to detect this. However, the effects of this additional piece of hardware can be noticed by the proctors, as these would influence the behavior of the EA software. For example, if characters appear on the screen while the student in front of the device is not typing. If a proctor has a suspicion that a particular student is cheating, a mechanism has to be available to interact with the EA software on that student's device to ensure that it is not remotely controlled.

Figure 9: ADT for *Manipulation of the Exam Results*.Figure 10: ADT for **FLEX**.

5 DISCUSSION

Based on the threat model developed by SV and updated by us, it can be concluded that EA indeed offers at least a similar level of security compared to PBEs. The easiest way to cheat in an exam seems to be using a smartphone on the toilet, which can only be mitigated but not prevented. However, this way of cheating is not special to EA, but can be applied in a similar fashion to PBEs. In fact, EA offers more ways to mitigate this way of cheating, as discussed by SV, e.g. by randomized ordering of questions or no-return Q/A sequence. Given, that proctors are still present in the room, even if it is called *e-Assessment*, cheating in the examination room seems as likely as for PBEs. With a reliable exam software, this probability can even be reduced because new ways of detecting cheats can be introduced with this software. The reliability of the software can be preserved even in a BYOD scenario, as described in our papers *Practical Security for Electronic Examinations on Students' Devices* and *Beyond Lockdown: Towards Reliable e-Assessment*. Actually, as described in this paper, EA does not only provide a similar level of security than PBEs, but an improved level of security. By introducing digital certificates, students' answers can be made identifiable in a way that offers the same level of security than modern online banking or similar applica-

tions. Hence, it is not longer possible that students' answers are modified, by accident or intentionally, after the students handed in by both parties, students and examiners. In the same manner, EA can provide improved security regarding *Timing Violations*, as the server of the assessment framework treats every student equally at the same time, i.e. assignments are provided for all students at the same time and answers are accepted from all students until the deadline. In addition to the previously discussed issues, EA offers possibilities to detect cheating a-posteriori. Particularly for specialized types of assignments, for example in programming courses, the digital implementation of examinations enables examiners to utilize machine learning techniques to gain indications about the actual author of a set of results (Opgenrein et al., 2018).

6 SUMMARY AND OUTLOOK

In this paper we derived a threat model for EA based on work by Sindre and Vegendla which targets the architecture of our software framework for EA. Due to the rather specific scenario, the threat model by Sindre and Vegendla had to be updated to match our requirements. This updated threat model is, however,

applicable to other scenarios for EA when leaving out our special requirements. Based on this threat model, we described counter measures based on our research for our project **FLEX**. The subsequent discussion of the level of security for EA in comparison to PBEs revealed that EA is not *per se* less secure than PBEs, which is in line with the results of Sindre and Vegendla. Based on the software architecture discussed at the beginning of the paper, we were able to conclude that EA is in part even more secure than PBEs. The results in this paper are quite promising regarding the security of EA, which affects the acceptance of EA by students and examiners. However, the discussed threat model was designed to fit the architecture of our software framework. The next steps have to be deriving a generalized threat model for EA to be able to show the level of security of EA for a more general setting.

REFERENCES

- Apampa, K. M., Wills, G., and Argles, D. (2010). User security issues in summative e-assessment security. *International Journal for Digital Society*, 1(2):135–147.
- Conole, G. and Warburton, B. (2005). A review of computer-assisted assessment. *Research in Learning Technology*, 13(1).
- Csapó, B., Ainley, J., Bennett, R. E., Latour, T., and Law, N. (2011). Technological issues for computer-based assessment. In *Assessment and Teaching of 21st Century Skills*, pages 143–230. Springer Netherlands.
- Dawson, P. (2015). Five ways to hack and cheat with bring-your-own-device electronic examinations. *British Journal of Educational Technology*, 47(4):592–600.
- Deutsches Forschungsnetz. Überblick DFN-PKI. <https://www.pki.dfn.de/ueberblick-dfn-pki/>. Last visited on 2019-06-07.
- Heintz, A. (2017). Cheating at Digital Exams - Vulnerabilities and Countermeasures. Master's thesis, Norwegian University of Science and Technology, Norway.
- Hillier, M. (2015). e-Exams with student owned devices: Student voices. In *Proceedings of the International Mobile Learning Festival 2015*, pages 582–608.
- Kaur, R. and Kaur, A. (2012). Digital signature. In *2012 International Conference on Computing Sciences*. IEEE.
- Kiennert, C., Rocher, P.-O., Ivanova, M., Rozeva, A., Durcheva, M., and Garcia-Alfaro, J. (2017). Security challenges in e-assessment and technical solutions. In *2017 21st International Conference Information Visualisation (IV)*. IEEE.
- Kordy, B., Mauw, S., Radomirović, S., and Schweitzer, P. (2011). Foundations of attack–defense trees. In *Lecture Notes in Computer Science*, pages 80–95. Springer Berlin Heidelberg.
- Küppers, B., Kerber, F., Meyer, U., and Schroeder, U. (2017). Beyond Lockdown. In Igel, C., Ullrich, C., and Wessner, M., editors, *Bildungsräume 2017 - DeLFI 2017*, volume 237 of *Lecture Notes in Informatics (LNI)*, pages 191–196, Bonn. DeLFI 2017 - 15. e-Learning Fachtagung Informatik, Chemnitz (Germany), 5. Sep 2017 - 8. Sep 2017, Deutsche Gesellschaft für Informatik e.V. (GI).
- Küppers, B., Politze, M., Zameitat, R., Kerber, F., and Schroeder, U. (2019). Practical Security for Electronic Examinations on Students' Devices. In Arai, K., Kapoor, S., and Bhatia, R., editors, *Intelligent Computing*, volume 857 of *Advances in Intelligent Systems and Computing*, pages 290–306, Cham. Computing Conference 2018, London (United Kingdom), 10. Jul 2018 - 12. Jul 2018, Springer International Publishing.
- Küppers, B. and Schroeder, U. (2016). Bring Your Own device for e-Assessment. In Gómez Chova, L., López Martínez, A., and Candel Torres, I., editors, *EduLearn 16 : 8th International Conference on Education and New Learning Technologies*, pages 8770–8776, Valencia. EDULEARN 2016 - 8th International Conference on Education and New Learning Technologies, Barcelona (Spain), 4. Jul 2016 - 6. Jul 2016, IATED Academy.
- Küppers, B. and Schroeder, U. (2017). Vergleich von papierklausuren und elektronischen prüfungen. In *INFORMATIK 2017*, pages 307–318. Gesellschaft für Informatik, Bonn.
- Metalidou, E., Marinagi, C., Trivellas, P., Eberhagen, N., Skourlas, C., and Giannakopoulos, G. (2014). The human factor of information security: Unintentional damage perspective. *Procedia - Social and Behavioral Sciences*, 147:424–428.
- Namiot, D. and Sneps-Sneppé, M. (2014). On Microservices Architecture. *International Journal of Open Information Technologies*, 2(9):24–27.
- Opgen-Rhein, J., Küppers, B., and Schroeder, U. (2018). An application to discover cheating in digital exams. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research - Koli Calling 18*. ACM Press.
- Sjøgaard, T. M. (2016). Mitigation of Cheating Threats in Digital BYOD exams. Master's thesis, Norwegian University of Science and Technology, Norway.
- Sindre, G. and Vegendla, A. (2015). E-exams versus paper exams: A comparative analysis of cheating-related security threats and countermeasures. In *Norwegian Information Security Conference (NISK 2015)*.
- Stanek, D. and Eifert, T. (2012). Maßnahmen für verlässliche und schnelle datenwiederherstellung. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 35(3).