Watching the watchers: bias and vulnerability in remote proctoring software

Ben Burgess
Princeton University

Avi Ginsberg
Georgetown Law

Edward W. Felten
Princeton University

Shaanan Cohney
University of Melbourne
shaanan@cohney.info

***Abstract*—Educators are rapidly switching to remote proctoring and examination software for their testing needs, both due to the COVID-19 pandemic and the increasing virtualization of the education sector. These software are used not only for regular school and university exams, but for high stakes testing such as legal and medical licensing exams. Three key concerns arise with the use of these complex software: exam integrity, exam procedural fairness, and exam-taker security and privacy.**

We conduct the first systematic, technical analysis of each of these concerns through a case study of all the major platforms used in U.S. law school and state attorney licensing exams. Through a large scale survey, we identify the four exam software suites utilized by 92 percent of the ranked U.S. law schools and 100 percent of remote state bar exam administrators.

We reverse engineer these exam proctoring suites and evaluate both their protective measures against a dishonest student and the security risks they pose to all students. We find that despite the promises of high-security, all platforms' anti-cheating measures can be trivially bypassed and can pose significant security risks to the user.

A number of anti-cheating platforms use facial recognition models in an attempt to identify non-technical forms of cheating. We test for skin-tone and racial biases in the legal exam proctoring suite with the largest market share. We evaluate their model against the best off-the-shelf classifiers and find it significantly under-performs, noting that faces from various racial groupings are substantially more likely to trigger false positives or false negatives than would be expected from a state-of-the art model. This has the potential to exacerbate already significant disadvantages faced by minorities in the legal profession. Finally, we offer recommendations to improve the integrity and fairness of the remotely proctored exam experience.

I. Introduction

Educators and professional accreditation bodies are rapidly adopting remote proctoring suites—software for monitoring students while they take exams. This trend has accelerated due to the COVID-19 pandemic [1]. While such suites are attractive to administrators and educators, they come with substantial risks to (i) the integrity of the exam (ii) the fairness of exam procedures and (iii) the security and privacy of exam-takers. These three risks, having received recent media coverage, are increasingly salient and are the focus of this work [2], [3].

While the mass adoption of remotely proctored exams is a phenomenon that can be attributed to the pandemic, computerized exams in an exam hall with proctoring similar to traditional paper based exams have seen steady adoption over the past few years [4]. Traditional computerized exam proctoring in exam halls operates on a significantly different threat model than remote proctoring solutions.

Examinations taken in an exam hall setting benefit from a more restricted threat model due to the watchful eyes of the in-person proctor. Computerized exams taken remotely on student controlled hardware require a threat model that takes into account the relative benefits and costs of cheating, along with the complexity of doing so.

Remote exam proctoring suite operating environments attempt to mitigate the increased risks of cheating by installing pervasive, highly privileged services on students' computers. Such software attempts to mitigate forms of academic misconduct such as accessing online resources during exams and pasting-in or accessing pre-written materials. This introduces considerable privacy concerns since, unlike an institutional computer in an exam hall, a student's laptop is not generally a single use device that only contains class related material. A student using their own hardware faces the risk of their sensitive and/or personal information being damaged or leaked by the exam proctoring software. As the software is highly privileged, this may compromise information owned by other users of the system as well.

These remotely proctored exams are not only used for general examination in tertiary education but also for high stakes professional licensing exams in regulated professions such as medicine and law. There are significant societal costs to illegitimately passing students, magnified in the legal profession where an inept lawyer can put an individual's liberty at stake or in medicine where an incompetent physician can cause significant injury and trauma to patients. The time and monetary burden of professional education and licensing places extreme pressure on students and this, along with the benefits of passing, mitigates the risk associated with cheating for unethical students. Maintaining public confidence that degrees earned and licenses obtained ensure a minimum degree of competency and knowledge is imperative. Equally important is the confidence that no individual who merits entrance into a profession has been blocked due to false cheating allegations. Research into whether remote exam proctoring puts either of these in jeopardy is lacking in current literature and merits attention from the security and privacy community.

We conduct the first systematic analysis of the remote proctoring ecosystem, scoping our study to those used by law

school and state bar exam boards. We aggregate public data from (and where necessary survey) all law schools and state bar exam boards to determine which proctoring suites they use. We find four exam suites—Examplify, ILG Exam360, Exam4, and Electronic Blue Book—implemented by 93 percent of U.S. law schools and 100 percent of remote state bar exam administrators. By limiting our scope to a particular regulated profession, our case studies are able to adequately represent the entire set of platforms in use within their examination settings, increasing our confidence in the results. However, we do not have reason to suspect that applying our methodology to other examination settings would yield substantially different results.

We then reverse engineer the four exam suites and find vulnerabilities of varying complexity that would allow a user to compromise the purportedly secure testing environments. We evaluate the suites in the context of three potential adversaries: a law student; a law student with computer science experience; and an experienced reverse engineer, and discuss vulnerabilities we find in the context of these adversaries. We analyze the impact exam security guarantees have on a student's privacy and find the majority of the exam proctoring suites we analyzed install a highly privileged system service that has full access to a user's activities. We find private information being transmitted to the exam proctoring suite vendor's servers during the exam that contains logs created before the exam began, highlighting the trade off in providing these guarantees.

Through this analysis, we determine that Examplify implements a facial recognition classifier to authenticate a student against pre-existing photographs prior to starting an exam. The facial recognition classifier is then run during the exam, depending on the settings selected by the educator, to determine whether the student who opened the exam and was authenticated is the student who is taking the exam. We are able to extract the name of the facial recognition system Examplify is using, face-api.js, and note that they are employing the pretrained models that are publicly available on the face-api's GitHub. We test these models against the current state-of-the-art (SOTA) classifiers across several different racial groups and demonstrate significant variance between subjects compared to the SOTA algorithms. We discuss execution time and licensing concerns of the SOTA algorithms to help understand why the current model was adopted. Finally, we discuss the terms of service and user interface of the exam proctoring suite to determine whether a user is giving informed consent for all of the monitoring.

We conclude with recommendations for steps that educators and students can take to limit the privacy impact of remote exam proctoring and provide vendors with suggestions to improve exam integrity while lessening the student privacy impact.

Contributions.

- We conduct a survey of the top 180 law schools and all state bar associations in the United States to determine their remote proctoring practices and will release the dataset as part of publication. (Section III-C).
- We reverse engineer four exam proctoring suites and identify the security the proctoring suite provides the institution and then the impact providing this security has on a student's privacy (Section IV).
- We build (and on publication, will release) a tool for automatically and rapidly extracting key security and privacy properties of exam suite software.
- We perform a detailed evaluation of racial biases in the facial recognition model used in the software with the dominant market-share for remote proctoring in law schools (Section V).

We evaluate the privacy and bias concerns of software that powerful organizations require students to use, with the express goal of aiding dis-empowered individuals and marginalized groups. Current heightened tensions regarding racial equality and identity bias, along with the COVID-19 related need for new methods of examination, motivate the need for this and similar contributions.

Research Ethics & Limitations.

While our survey involved contacting law schools and state bar associations to determine what platforms they use and how they use them, our work was exempt from IRB review as we did not collect data pertaining to individuals.

Our analysis of facial recognition systems used a data set containing images of incarcerated individuals who were not given an opportunity by the original researcher to consent to its use for research. We consulted with an applied ethicist independent from the project and determined criteria for appropriate use of these images, most importantly the principle of beneficence. While we are unable to rectify the consent issue, after consideration we believe that our work fulfills the principle of beneficence as:

- Our work aims to aid marginalized groups and dis-empowered individuals by evaluating software that powerful organizations require students to use.
- Our work does not cause additional harm to the individuals in our dataset, beyond perpetuating its use in academic literature.

Thus, while we are unable to uphold the principle of autonomy to the greatest extent, we believe our research is nonetheless appropriate.

Our analysis of facial recognition systems focuses on racial biases in algorithms, despite evidence that system performance is more closely tied to skin-tone with race as a proxy. However, as our very large reference data sets are racially coded rather than coded by skin tone, a more sophisticated distinguishing analysis was outside our scope.

We necessarily restrict ourselves to a sub-sector within education (law and the regulated professions), as well as centering our work on a limited set of products. This allows us to more comprehensively cover our chosen area and produce timely research in light of the current social need.

We intentionally refrained from evaluating server side components and functionality of the software packages to avoid the accompanying legal and ethical concerns.

Our automated tool is an incremental advance over prior tooling and thus our evaluation of it is limited—however we believe discussing and releasing the tool will simplify replication of our results and facilitate similar analyses of other products. We therefore include a brief discussion of its design

and evaluation as a minor contribution and part of our larger work.

II. Related Work

Several previous studies [5], [6] have discussed the different threat model remote proctoring solutions face and provided recommendations for security features that could mitigate these new vulnerabilities such as improved authentication measures or 360 degree cameras. Slusky [7] extended this by conducting a study of the security features 20 different exam proctoring suites claim to provide and discussing their strengths and weaknesses against various threat models. Teclehaimanot, ET AL. [8] conducted a study of eight John Madison University professors and determined that a significant number of their students appeared to have gained an advantage on remotely proctored exams despite the use of an exam proctoring suite. Cohney, ET AL. [9] performed a multidisciplinary analysis of the risks faced by institutions as they rapidly adopt EdTech in light of the COVID-19 pandemic.

A few studies have been conducted that attempt to quantify how a student's perceived stress and performance varies between remote and in person exam environments. Teclehaimanot, ET AL. [8] performed a meta analysis of test scores recorded in proctored and unproctored environments, finding a 0.20 STD variation between the two sets. Karim, ET AL. [10] performed a study on 582 subjects taking a cognitive test in an exam hall versus in a remote setting with exam proctoring software. They found the scores were similar between the two groups but that subjects in the remote setting indicated a higher perceived stress level. Teclehaimanot, ET AL. [8] conducted a survey of eight experts from different universities who had previous experience with remote exam proctoring solutions to determine the primary factors that influenced their adoption of the solutions. They found the perceived trust of the vendor and the security of the offering to be the primary factors. The recent work of Balash, ET AL. [11] presented an in depth user-study of student responses to remote proctoring software in light of the pandemic.

Several previous studies have demonstrated biases in the different components of facial recognition systems [12], [13]. Singh, ET AL. [14] extended this by creating targeted attacks to cause false positives by the classifier. We do not investigate these attacks in the context of Examplify's classifier but anticipate no reason they would not be applicable. Nagpal, ET AL. [15] evaluated several different machine learning classifiers using the Multi-PIE and Morph-II datasets and found significant racial biases. We closely structure our methodology for detecting biases in Examplify's classifiers to follow this work. The National Institute of Standards and Technology (NIST) conducted the Face Recognition Vendor Test (FRVT) to quantify face recognition accuracy at a very large scale [16] and providing methodological expertise that guides our work.

Most importantly, both teachers and students have documented their concerns—in increasing number since the start of the pandemic. Students have performed their own small scale experiments testing various proctoring suites [17], [18] while teachers have voiced misgivings [19] about use of facial recognition.

III. Preliminaries

Cheating on exams is a well-known problem faced by educational and license-granting institutions prompting, increased reliance by these institutions on exam proctoring software for cheating detection and prevention. While preventing cheating is important, competing values exist which are equally important such as preserving privacy and security, guarding against racial bias, and ensuring fairness. Ensuring exam integrity while respecting these competing values becomes increasingly difficult with the introduction of remote exam proctoring software that is downloaded and run on a student's personal computer.

A. Exam Software

Exam software generally consists of a user interface that allows a student to take a computerized exam by displaying multiple choice questions with answers and/or providing text boxes for students to enter answers into. This is coupled with a series of cheating detection and deterrence tools tailored to the threat model assumed by the exam software vendor. The general assumption is that students will try to cheat by searching the internet, opening documents or programs on their computers, or consulting a device, person, or printed material during the exam. To this end, exam software generally block internet access and access to non-approved applications on the user's machine, perform audio recording to detect verbal communication with another person, and run facial recognition to ensure the appropriate individual takes the entire exam and does so without looking away to consult another source of information.

Facial recognition is a particularly problematic aspect of exam software because (as we discuss in Section V) the facial recognition models used by leading exam software vendors do not exhibit equal accuracy between racial groups creating the possibility for certain groups to be flagged more frequently for potential cheating. The importance of exam software treating all users fairly, regardless of race, gender, or other appearance attributes cannot be overstated.

As a highly regulated industry, law pays particular attention to ethical and professional standards such that when a student cheats within law school or during a licensing exam, there is a high likelihood that the accrediting organization will bar them (potentially permanently) from practice. We infer that these high standards within the profession affect the choice of remote proctoring software platform utilized.

B. Legal Education

The threat model for law school and bar exams needs to take into account the legal profession testing structure and the nature of the exams themselves. Law school exams are typically graded on tight curves and course grades depend primarily on a single final exam worth at least 85% of the course grade. Law school grades, and thus exams, are closely tied to job prospects (more-so than many other fields) [20] and bar exam scores are a determining factor in lawyer licensing. These factors combined with the high-debt burden associated with legal education place extreme pressure on students, often providing strong motivation for unscrupulous students to cheat, even by means of paying outside individuals for materials,

Exam Proctoring Suite	Schools	Percentage
Examplify	99	55
Exam4	52	28.89
Electronic Blue Book	13	7.22
Canvas	4	2.22
MyLaw	3	1.67
ILG Exam360	1	0.56
Other	5	2.78
Unknown	3	1.67

Table I: Proctoring Suite Market Share in Law School. We tabulate the adoption of remote exam proctoring suites by the top 180 law schools in the United States. Examplify is the leading proctoring suite followed by Exam4 and Electronic Blue Book. A few smaller schools use less well known software offerings such as Canvas or MyLaw. We were unable to determine software used for three schools as two did not respond to inquiries and one closed down operations. Top 180 status was determined in accordance with the US News and World Report 2021 ranking.

technical bypasses, or cheating tools. Law school exams often take the format of a story riddled with legal issues that a student must identify and analyze. Exam answers often consist of several pages of written text so that cheating by simply copying another student's answer would be painfully obvious. Student cheating by attaining or predicting exam contents in advance in order to pre-write answers or using messenger apps during the exam to consult friends comprise the likely threat model and represent a more difficult to detect cheating scheme. COVID-19 has caused bar associations and schools to rapidly adopt a remote (at home) testing model forcing administrators to look for additional assurances that cheating attempts will be detected.

C. Software Usage Survey

We identify the proctoring software used by the top 180 law schools by scraping public facing websites and making private inquiries to law school administrations. We repeat this process for every state bar association's licensing exam. Examplify, ILG Exam360, Exam 4, and Electronic Blue Book are the four primary exam software suites in use by over 91% of law schools and 100% of bar exam associations. We select these for analysis in the remainder of this work.

Table I depicts the results of our survey of remote proctoring software adoption across ABA-accredited United States law schools while Figure 1 illustrates our compiled parallel list for every state bar association's licensing exam. Our data set with the individual school and state bar remote proctoring suite adoption choices is available at https://github.com/proctoringsecurityndss2022/ProctoringSuiteAdoption.

IV. Cheating

In order for remote exam proctoring software to provide a significant benefit to exam integrity, it is critical that it effectively prevents attempts to cheat. This is a non trivial task as there is a very broad threat model due to the relatively uncontrolled setting and hardware they must account for. To evaluate the security of the currently used offerings, we present a reverse engineering methodology that allows us to accurately analyze the security provided by these exam proctoring suites. We propose three theoretical adversaries that would be realistic adversaries to a remotely proctored law exam. We then discuss potential ways each of these adversaries could compromise the security features we found and potential attacks this would allow them to perform.

A. Methodology

The four exam suites we analyzed are not open source, so we reverse engineer the binaries using existing static and dynamic analysis tools and target three primary questions: (1) Do the exam suites provide the security guarantees they promise? (2) What privacy is the user required to give up to achieve these security guarantees? and (3) Are the exam integrity checks performed fairly across all examinees?

We isolate suspected critical functions in the exam proctoring suites using common reverse engineering methods such as system log inspection and recording user interface dialogues. We also use traditional disassembly and reverse engineering workflows to manually inspect binary areas of interest. We first target the critical functions we identified earlier to achieve a reference position then develop this to other functions by analyzing function references and dependencies. This approach only allows us to view data statically stored in the binary which is only a sub-portion of the entire application's operating state. We employ dynamic analysis to extend this analysis and allow us to view the state of the exam proctoring suite as it is running. A few exam proctoring suites we analyzed implement a scheme to detect when a debugger is attached and run an alternative code flow. This can be removed, however, through patching of the binary which allows the standard control flow to be followed.

To evaluate exam transit integrity, we use standard network interposition techniques to evaluate whether the connection to retrieve the exam is over an encryption TLS connection or in plaintext. If the connection is over TLS, we evaluate the program's response to being served: (1) a valid certificate that has an incorrect common name; and (2) a self signed certificate that is not recognized by any certificate authority (CA) but that bears the correct common name. We also attempt to force a downgrade by blocking access to the port before the handshake can occur and then watching for further plaintext retries on a separate port.

We defer the evaluation of exam fairness to Section V, to allow for its increased depth.

B. Threat Model for Exam Proctoring

To ensure that our work fairly represents the context, we informally model three adversaries likely to interfere with the fair and secure operation of a remotely proctored law

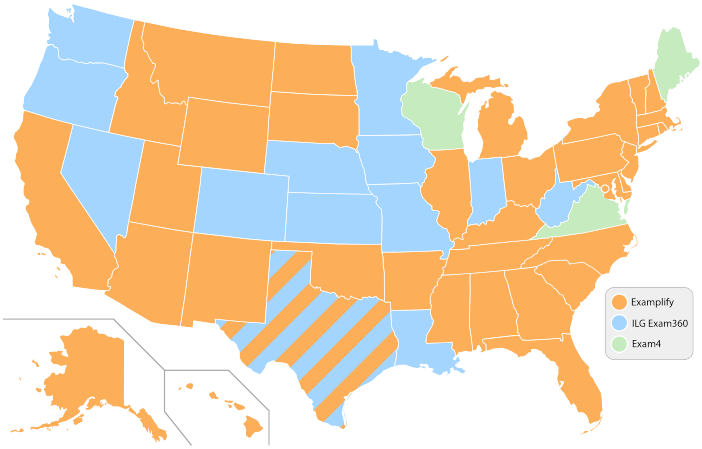

Figure 1: Proctoring Suite Market Share Across Bar Exams. The figure depicts adoption of remote exam proctoring suites by state bar exam associations is mapped out across the United States. Examplify has a dominant, majority market share, followed by ILG Exam360 and then Exam4.

exam. We note that any student could easily become a more sophisticated adversary than their background would suggest by either colluding with other students or hiring additional help.

Law Student. We consider law students to be individuals without experience in reverse engineering software or programming. They can adjust basic system and file settings, configure simple hardware devices, and use known passwords. They cannot modify binary settings or extract encryption keys. Attacks with a budget ranging from a few to several thousands of dollars are feasible, given the hundreds-of-thousands of dollars spent on law school.

Law Student with CS Background. We consider law students with computer science background as individuals who do not have extensive experience reverse engineering software but who have significant programming experience, familiarity with basic system administration, and the capability to extract keys from the binary but no ability to modify any portion of the binary.

Experienced Reverse Engineer. We consider experienced reverse engineers as individuals with all the prior capabilities with additional experience analyzing binaries, disassembling software, applying patches, and rebuilding binaries. They are familiar with advanced system administration and are able to adjust any setting, configure hardware devices, modify drivers, and use custom encryption tools. They are also able to extract encryption keys from the binary and modify the control flow of the binary. While we expect the number of students with this background is low, such individuals may sell their services or cracked versions of software.

C. Results

Exam proctoring suites use a few distinct components to insure exam integrity is maintained: **exam monitoring** which prevents or detects the student accessing unauthorized content during the exam; **exam content protection** which attempts to prevent the student from accessing the questions or answers to the exam outside of the proctoring environment; and **identity verification and authentication** which ensures the student is actually taking the exam versus a third party. We categorize our findings based on these components for the following four proctoring suites: Examplify, Exam 4, Electronic Blue Book, and ILG Exam360.

1) Exam Monitoring: The exam monitoring components in an exam suite aim to prevent a student from accessing unauthorized resources during the exam and may even restrict the student from beginning the exam if certain parameters are not met. Table II provides an overview of the features used by each suite. The features we identified as comprising the exam monitoring component of exam integrity are detailed with each exam suite's feature implementation outlined:

Virtual Machine Protection. A student running the exam proctoring software inside a virtual machine environment can easily exit the environment with a hotkey and completely evade any monitoring the exam proctoring suite hoped to provide. To prevent this, most suites feature virtual machine detection to detect and prevent attempts to run the software inside a virtualized container or hypervisor. The most common implementation of this is a simple check of the computer's CPU manufacturer to see if the string matches known virtual machine software vendors. An additional check of CPU temperature for constant values or known presets can be run since a virtual CPU does not often pass through the real value from the actual CPU.

All of the exam suites we examined implement a virtual machine check by comparing the CPU vendor field against a list of known virtual machine vendors. Examplify extends this by retrieving the CPU temperature and flagging a device if it reports a CPU temperature of 100C as this is the typical default value virtual machine vendors use. Electronic Blue Book also checks the computer, hard drive, network adapter, and bios vendor information to see if the field contains the string 'virtual'. If a virtual machine is detected, they log the attempt and prompt the user to run the software outside of a virtual machine. Table III provides a list of popular virtual machine hypervisors that are blocked by each proctoring suite.

Virtual Webcam/Microphone Detection. Virtual webcam and microphone programs exist that allow a user to generate a virtual device which either returns data from a remote endpoint or from a file. This can allow a student to bypass identity verification performed by the exam proctoring suite by either returning video of themselves in another location while someone takes the exam for them or by returning a prerecorded video of them taking the exam. Exam proctoring suites attempt to mitigate this threat by checking the device vendor and bus location against a list of flagged vendors. If one of these blocked devices is detected, the software will either flag the exam for further review or prevent the student from beginning to take it.

Examplify and ILG Exam360 detect virtual webcams and microphones by retrieving the operating system's device list and comparing it to known virtual device vendors. Exam4 and Electronic Blue Book do not use the computer's webcam or microphone so they don't implement a check. A list of popular

		Examplify	Exam4	EBB	ILG Exam 360
Security	**Encryption at Rest**	AES-256	AES-256	3DES	AES-256
	Encryption in Transit	HTTPS	HTTP*	HTTP*	HTTPS
	Virtual Machine Protection	Block List	Block List	Block List	Block List
	Virtual Device Detection	Block List	N/I	N/I	Block List
	Clipboard Management	Integrated	Cleared	Cleared	Integrated
	Screenshot Capture	N/I	N/I	N/I	App Window
	Process Restrictions	Allow List	Block List	Block List	Allow List
	Network Access Restriction	Route Table	Adapter Disable	N/I	Null DNS
Privacy	**Initial Identity Verification**	Automated	N/I	N/I	Human
	Continuous Identity Check	Automated	N/I	N/I	Human
	System Service	Always Running	App Running	App Running	App Running
	Device Identifiers	App List OS Hardware	N/I	N/I	OS Hardware

Table II: Security and Privacy Features of Proctoring Software. The security and privacy features of Examplify, Exam4, Electronic Blue Book (EBB), and ILG Exam 360 are summarized above. Entries marked N/I indicate that a given feature was not implemented by a specific product.

vendors that are on the list of blocked hypervisors can be seen in Table III.

Clipboard Management. Students copying pre-written text into an exam is a major concern, especially in the field of law where exam essay answers may require lengthy summaries of law and/or analysis that can be prepared before the exam. Exam proctoring suites attempt to prevent this by either clearing the clipboard before the exam or logging its contents for later analysis. During the exam the clipboard generally can be used inside the proctoring suite but copying from outside apps is prohibited or logged using a similar method.

The exam proctoring suites implement clipboard protection by calling the system clear function. The content is not captured before the clear operation by any of the suites. Exam4 and ILG Exam360 implement a custom restricted clipboard for use inside the test environment which limits what can be copied to only plaintext items.

Screenshot Capture. Exam proctoring suites often offer screenshots of the student's screen during the exam to allow a proctor to retroactively review the exam session to determine if unauthorized resources were accessed on the computer. These screenshots are normally captured using a highly privileged system level service which leads to potential privacy issues when an exam is not in progress.

ILG Exam360 is the only exam suite that provides screenshot captures of the student's computer during an exam. The screenshot is captured by calling their highly privileged system service using a Javascript call which uploads the screenshot to an Exam360 controlled server.

Process/Application Restrictions. Process restrictions are normally used to limit what applications a student can access during an exam. These are generally implemented using a process allow list that contains processes specifically allowed by the exam in addition to critical processes the operating system needs to maintain the computer's function. A weaker implementation that may also be used involves process block lists that prevent certain processes such as web browser activation from being started. Both of these approaches are implemented using the exam proctoring suite's highly privileged system service which starts a watchdog service that forcefully kills unauthorized processes.

Examplify and ILG Exam360 compare the processes currently running on the system to a list of processes they allow along with any processes allowed by the exam. They call their service helper to forcibly kill and continuously monitor for any processes that are running but not included on the list. Exam4 and Electronic Blue Book have a list of services that they disallow which are killed upon an exam being started.

Network Interception. Closed book exams require limitation on a student's ability to easily search for information on the internet. The different approaches suites can implement to block internet access include: dropping internet traffic, inserting an active man in the middle to capture traffic, or redirecting the traffic to the vendor's servers. The simplest approach is dropping the traffic using a routing rule.

Examplify restricts network traffic by inserting a null route into the default operating system routing table. Exam4 disables the network adapter during the examination. ILG Exam360 inserts a null DNS entry into the network configuration to cause domain name resolution to fail. Electronic Blue Book does not implement any network restrictions other than blocking access to common websites through their process block list. None of the implementations we inspected captured browser traffic or redirected it to a site controlled by the exam proctoring suite vendor.

2) Exam Content Protection: Exams are often downloaded to student computers before the actual exam begins. The security of an exam in transit from the servers of the exam proctoring suite vendor to the client is paramount since traffic can be easily intercepted using off the shelf solutions. The exams need to be protected both during the download and while they sit on the student's computer to prevent early or unauthorized access. In transit and at rest exam suite encryption implementation is detailed below.

Encryption In Transit. Examplify and ILG Exam360 use transport layer security (TLS) for all of their connections. The certificate chain, expiration date, and common name are correctly verified which mitigates active man in the middle attacks. The connection is never downgraded to plaintext HTTP even if the software is unable to successfully complete the handshake. Examplify includes their own certificate store inside the software to prevent potentially using a modified system certificate store. Exam4 and Electronic Blue Book allow the individual institution to select their transport layer security settings. We note several institutions in the Electronic Blue Book binary are not configured to use HTTPS. Additionally, the school which we obtained the Exam4 binary from did not have TLS configured.

Encryption at Rest. Examplify, ILG Exam360, and Exam4 implement AES-256 for encrypting the exams at rest on the student's computer. ILG Exam360 and Exam4 use SHA1 to derive the AES key from a password. Examplify uses 10,000 iterations of the PBKDF2 function to derive the exam password. The exam manifest which contains the main exam password salt, exam information, allowed resources, and monitoring settings is encrypted separately using a static key stored in the Examplify binary. Electronic Blue Book allows the institution to choose between Triple DES and AES for encrypting their exams and 1,000 iterations of SHA1 is used by default for the password derivation but the institution can configure the iteration count to use. The password salt is statically stored in the Electronic Blue Book binary and is set to 'Kosher'.

3) Identity Verification and Authentication: Exam suites all implement some form of user authentication to ensure that the test taker matches the individual to be assessed.

Logins and Photographic ID. Exam4, ILG Exam360, and Electronic Blue Book implement standard single factor logins. OAuth is not supported by any of these solutions so institutions cannot easily add more extensive identity verification measures such as two factor verification. Examplify implements a similar single factor login, however, institutions can enable an automated photographic identity verification before a student is allowed to take an exam. ILG Exam360 also offers photographic verification of a student's identity but they implement a remote webcam approach which connects a student with a human proctor to conduct the verification before the exam begins.

General Interaction Fingerprinting. Several exam suites implement general interaction fingerprinting which analyzes the pattern of a student's key strokes and mouse movements against the class average to determine if there are any anomalies. If an anomaly is detected the exam is flagged for a human proctor to review. This poses the risk of potentially unfairly flagging students with disabilities or who otherwise deviate from the class average's pattern for legitimate reasons.

Facial Recognition. Exam proctoring suites employ facial recognition to perform identity verification of the student taking the exam. This is implemented to serve as an analog to students showing their ID upon entering an exam hall and acts as a countermeasure against another person taking the exam for the student. The student's image is normally compared against a trusted reference image of the student and the distance of the facial vectors is compared and verified if the distance is below a certain threshold.

ILG Exam360 performs facial recognition in some cases but also employs a remote human verification method so that the final verification resembles that of an exam hall session. Examplify's verification implementation relies on an automated facial recognition classifier. Our research identified bias introduced by Examplify's process and we detail the results of our case study into Examplify's process in Section V-C.

D. Impact Analysis

To evaluate the impact of remote exam proctoring on exam integrity we must determine whether the security features are effective against various adversaries; whether privacy concerns are generated by the use of these features; and whether these features introduce bias into the exam taking process. We start by discussing common attacks that adversaries could use to bypass the security features the proctoring suites offer and relate this to the knowledge level required to perform the attack to demonstrate whether it's a realistic threat to an exam.

1) Security Feature Vulnerabilities: The proctoring suites include various features to prevent students from bypassing the protections the suites offer.

Virtual Machines. Virtual machine software allows a guest operating system to be run inside the primary environment,

		Examplify	Exam4	EBB	ILG Exam 360
Virtual Machine Software Successfully Detected	**VirtualBox**	✓	✓	✓	✓
	VMWare Workstation	✓	✓	✓	✓
	VMWare Fusion	✓	✗	✗	✗
	Parallels	✓	✓	✗	✓
	Hyper-V	✓	✗	✗	✗
	QEMU	✓	✗	✗	✗
Virtual Webcam/Microphone Successfully Detected	**ManyCam**	✓	N/I	N/I	✓
	YouCam	✓	N/I	N/I	✓
	MyCam	✓	N/I	N/I	✓
	Logitech Capture	✗	N/I	N/I	✗
	OBS Studio	✗	N/I	N/I	✓

Table III: Detected Virtual Devices. We tabulate here the set of virtual machines and virtual webcams/microphones detected and blocked by the exam proctoring suites we analyzed. N/I indicates that the exam proctoring suite did not implement any features to detect a virtual device.

bypassing any monitoring an exam proctoring suite could hope to achieve. The CPU vendor check implemented by all of the exam suites can be easily bypassed using common virtual machine software since this field is generally configurable. The additional CPU temperature check Examplify conducts is also easily defeated by configuring the virtual machine to pass through the CPU temperature. The student could either set it to their actual computer's CPU vendor or they could generate a random string not on the block list the exam proctoring suites are using. A law student with a CS background, capable of installing a virtual machine and configuring the CPU vendor, could readily complete this attack.

Virtual Webcams/Microphones. Virtual webcam and microphone devices can be used to take exams without being watched by replaying a prerecorded file or piping a connection to a separate device. The only mitigation employed to prevent virtual devices is a driver list search which looks for known virtual device vendors along with the running process restrictions. A student can easily install a virtual driver not on the known vendor list to evade detection. The signing key of the driver is not checked by the exam proctoring suite allowing a student to rename the driver of a blocked virtual device. Compared to finding an unblocked virtual device, this attack would require significantly more skill but all these attacks would be well within the capabilities of a student with a CS background. A student who was able to create a custom virtual device that masqueraded as a legitimate driver would be able to easily redistribute this to other students for use.

Clipboard Protection. Exam proctoring suites protect against clipboard content being carried into the exam by calling the operating system's clear function before the exam begins. This does not preclude the use of an external hardware connected clipboard such as a KVM or a built in keyboard macro which allows note storage. The majority of these hardware connected devices simply present as standard hardware interface devices which don't require any additional drivers. Exam proctoring suites could attempt to protect against this by fingerprinting the input rate of a student's keystrokes. Purchasing a hardware device capable of maintaining an external clipboard is an attack any student could perform. We do not investigate the Mac version of the exam proctoring suites in this paper but colloquial evidence suggests that students may be able to use the iCloud clipboard sharing to bypass these protections by loading information from their phone's clipboard into the Mac keyboard through the service during the exam.

Process Restrictions. Process restrictions are implemented either with a list of allowed processes or with a list of disallowed processes that are killed upon starting the exam. To subvert either restriction, a student with CS background could recompile a piece of open source software to report a different process name. As an example, Chromium could be recompiled to report as 'explorer.exe' which is allowed by every testing suite we looked at since it is a critical user interface component for Windows based systems. For suites using block lists, any student would be capable of finding a process not present on the disallowed list through trial and error.

2) Student Privacy Concerns: Two major privacy questions arise with remote proctoring software use: (1) Is the user appropriately informed of the information being collected upon engaging with the remote exam software? and (2) Does the potential for pervasive monitoring after the student is no longer actively taking an exam exist? To this end, we develop an analysis tool to assist other researchers in identifying remote exam software privacy issues.

Informed Consent.

The way in which students interact with exam proctoring software raises significant issues as to the ability of the students to meaningfully consent. An examinee cannot provide meaningful consent to the activities performed by the software if they are unable to learn what information the software collects or what actions it performs. Examinees are not informed as to what specific data these exam proctoring software are actually collecting or the mechanisms used to surveil for cheating and are prohibited from discovering such information by reverse engineering the software.

Attempts to glean information about specific data collected or surveillance mechanisms by reading privacy policies, end user license agreements, or similar documents will not be fruitful. For example, ExamSoft's privacy policy notes "in order to secure the exam takers device, ExamSoft must access and, in some instances, modify device system files." This broad statement is devoid of meaningful information and essentially informs the reader that ExamSoft's Examplify software may do virtually anything to their computer. Other privacy policies contain conflicting statements about the software's activities. For example, the Extegrity Exam4 privacy policy states "Exam4 does not access any data on the laptop other than the data created by the act of taking an exam" and "Exam4 monitors activity the student engages in while taking an exam and creates an encrypted log report for evidentiary purposes." It is not possible for Exam4 to monitor activity the student engages in if it does not access any data other than that created by the act of taking an exam. These types of statements thwart a student's attempt at gaining information on data collection or exam software actions and inhibit their ability to meaningfully consent.

Even if an examinee was able to truly understand and consent to all information being collected and the actions software was performing, such consent would not be meaningful given the examinee's position. In most instances, the examinee does not have bargaining power. They are faced with a simple and daunting choice: accept and use the software as-is or refrain from taking the exam and accept the associated consequences, which in the field of law would mean inability to become a licensed lawyer. Such a "choice" is not a choice at all. While it provides a degree of legal cover for the exam software companies, it fails to meet conventional ethical standards for consent [21].

Post Exam Monitoring. Examplify installs a highly privileged system service that is constantly running on the computer even if Examplify isn't open. The currently running applications are logged to a debugging file that is uploaded periodically once the application is open. The service also regularly reaches out to the Examplify server to check for and install updates for the service or the binary. Exam4 and ILG Exam360 also implement a system service for exam monitoring but stop them when the exam is terminated gracefully. Electronic Blue Book directly hooks into the Windows system service with their binary to provide the exam monitoring features. This guarantees that no additional background monitoring is being performed once the binary is closed.

E. Automated Analysis of Privacy Impact

We created an analysis tool based on the RADARE2 and Ghirda frameworks [22] to simplify the reverse engineering process for researchers who want to quickly analyze the privacy impact of other exam proctoring solutions and will release the tool on publication. The approach we use in this paper using traditional tools like IDA work well for in depth studies, but they are not well suited for providing a quick summary of an exam proctoring suite's privacy impact. Our tool requires a user to simply run the Python script on the binary they want to analyze and a high level overview of the application will be provided.

1) Design: The analysis tool first loads all of the shared object files the binary uses then performs auto analysis using RADARE2's built in analysis suite. This attempts to locate the segments and functions in order to generate a control flow graph. From this control flow graph, we are able to extract cross references to lines in any part of the code which allows us to more easily establish where certain data elements are being used.

We are able to detect privacy sensitive calls such as calls to a microphone, webcam, or video driver by fingerprinting common vendor and system library names. We also attempt to extract information about the security features the exam proctoring suite implements including whether it detects virtual machines, uses a secure connection to reach the back-end server, and whether it encrypts on disk content. If on disk encryption if found, we display the cipher suite being used then attempt to extract the encryption key and initialization vector by searching for keys of the correct bit length in a user definable window around any data references found in the encryption function. For a more complete analysis, the analysis tool can be run with the live memory option which initializes the binary, attaches GDB, and runs to a user defined breakpoint then performs the analysis. This allows a more complete analysis of libraries and code segments which are stored encrypted at rest or loaded from a remote endpoint.

The user can opt to view a summary of the binaries security and privacy properties or a more in depth analysis which features control flow graphs and decompilations with Ghirda of code segments the tool believes are relevant.

2) Evaluation: We evaluate the automated analysis tool of the four exam suites with analyzed in this paper to determine whether it accurately identifies relevant security and privacy information about the proctoring suites. We evaluate the tool without using the live memory analysis option since we believe this would be the most commonly run configuration of the tool due to the relatively large perfomance and memory overhead of searching the entire live memory space multiple times on consumer hardware.

We find the tool is able to correctly identify camera and microphone usage in the exam suites besides a false positive which is triggered when analyzing Electronic Bluebook. We determined this false positive is due to the inclusion of a large English dictionary in the binary which incorrectly triggers one of the vendor searches we run. The relevant function control flow graph and decompilation is presented to the user which would allow the user to trivially identify it as irrelevant and therefore disregard.

The tool performs similarly well when finding virtual machine detection, insecure connections, and on disk encryption with results mirroring the results we obtained in our manual

analysis. The encryption key Examplify uses is successfully extracted and presented to the user along with a few false positives. We write a simple Python script to test the encryption keys and initialization vectors the tool outputs and are able to successfully decrypt the on disk libraries for Examplify in under one minute.

V. Identity Verification & Fairness

In this section, we evaluate the accuracy and then the fairness of the facial recognition system used in Examplify, the only one of the products within scope that uses such a feature. However, as this functionality is increasingly penetrating the broader market our section begins with a more generalized analysis (Sections V-A to V-B) informed by the product decisions in Examplify.

When forms of identity verification are used as part of examination procedures, it is critical to the fairness of the examination that these procedures minimize bias when determining if the exam taker is the same individual to whom a grade or license will be awarded. This motivates our analysis of identity verification systems in the context of remote proctoring, prompting us to assess whether sufficient attempts have been made to increase accuracy and minimize bias.

Traditionally, human verification both through simple recognition of a student and identification card checks have been used to ensure exam integrity. While these checks could still be conducted in a remote setting using a human proctor who monitors the testing session, there is a large incentive for exam proctoring vendors to move to a fully automated model because it reduces staffing costs and increases the number of students who can be verified within the same time frame. The adoption of an automated approach introduces the possibility of algorithmic bias and unintended situations where a student may be unable to take their exam or would be incorrectly flagged for cheating during the exam.

We separate the facial recognition steps an exam proctoring suite would need to perform into two steps: the initial verification against a student's identification photo to bootstrap their identity; and the continuous verification that insures the student who verified their identity initially continues to take the entire exam. We select a set of open source state of the art algorithms along with 'face-api.js' which is deployed in the real world identity verification system used by Examplify. Given that the expertise of remote proctoring firms is outside AI/ML and that in the current business environment such firms are unlikely to develop and train their own models (an assumption borne out by our analysis of the leading market product), it is reasonable to select pre-trained, off-the-shelf models for comparison.

We select a collection of datasets that provide close analogs to real world conditions that would be experienced by a real world implement to allow us analyze the general accuracy of the facial recognition implementation Examplify uses. We select Morph-II which provides a set of prisoner mugshots with a significant time gap between captures. This allows us to replicate the conditions that would experienced when trying to verify a person against a photograph on their driver's license or identification card which would generally be a relatively old picture of the person. We select the Multi-PIE dataset for verifying the continuous verification mode Examplify uses the ensure the student who completed the verification is the student who completes the rest of the exam. This dataset features images of subjects taken at different head rotations which we believe would be a close analog to the head rotations a student would make while filling out an exam. Both of these datasets provide at a controlled background and lighting condition which allows us to control for these factors which may otherwise increase the variance in our results.

We select an additional collection of datasets for our analysis of potential racial biases in the facial recognition implementation. We select FairFace and UTKFace which both provide a racially balanced selection of faces from in the wild datasets. They provide closely cropped images of each face which allow us to minimize the affect of different backgrounds in subsequent captures of the same subject. We select the 'White', 'Black', and 'Asian' ethnicities for our analysis.

We note that the LFW dataset provides only a limited representation of the faces an exam proctoring facial recognition system would process as the LFW images are captured with uncontrolled backgrounds, facial poses, age differences, and lighting.

It is reasonable to assume that all of these factors could be more precisely controlled during an exam setting. Therefore, while we select our four algorithms using the benchmark LFW dataset, we simulate the identify verification steps utilized in remote proctoring using two additional datasets, Morph-II [23] for initial verification and Multi-PIE [24] for continuous verification, and run them through the algorithms. These two datasets feature controlled backgrounds, facial poses, and lighting that we believe more accurately reflect images a facial recognition system used in an exam proctoring suite would be expected to process.

The algorithms we select are based on their accuracy using Labeled Faces in the Wild (LFW) [25], the current dataset used most prominently in the literature for benchmarking the fairness of facial recognition algorithms [26]. This results in a selection containing VGG-Face [27], ArcFace [28]–[31], FaceNet [32], and OpenFace [33]. We note that the LFW dataset provides only a limited representation of the faces an exam proctoring facial recognition system would process as the LFW images are captured with uncontrolled backgrounds, facial poses, age differences, and lighting. It is reasonable to assume that all of these factors could be more precisely controlled during an exam setting. Therefore, while we select our four algorithms using the benchmark LFW dataset, we simulate the identify verification steps utilized in remote proctoring using two additional datasets, Morph-II [23] for initial verification and Multi-PIE [24] for continuous verification, and run them through the algorithms. These two datasets feature controlled backgrounds, facial poses, and lighting that we believe more accurately reflect images a facial recognition system used in an exam proctoring suite would be expected to process.

We first detail false match, false non-match rates, and image quality in relation to a subject's reported race for both Morph-II and Multi-PIE in a generally applicable way. We then discuss the performance of the algorithm and parameters Examplify implemented over the Morph-II and Multi-PIE datasets. Finally, we discuss the impact Examplify's verification system could

Algorithm	African American	European
ArcFace	0.483	0.520
FaceNet	0.424	0.492
VGG-Face	0.163	0.132
OpenFace	0.254	0.393
face-api.js	0.356	0.481

Table IV: Average false non match rate. We tabulate the false non match rate by racial categorization for each of our evaluated models. An increased false non match rate is seen for the "European" group across all algorithms except VGG-Face.

have on a student attempting to complete an exam with the software.

A. Initial Identity Verification

The Morph-II dataset contains images of 13,658 prisoners taken by the Bureau of Prisons throughout their prison sentences with racial classifications available for each subject. These images are analogous to the driver's license images a facial recognition system would use to establish the root of trust for a student's identity during the initial identity verification step. The extended time range, 4.2 years, between subsequent captures of subjects in Morph-II is advantageous in this case as the average driver's license photo refresh rate is 5.7 years in the United States [34]. This allows us to simulate the aging that would likely occur between the image on a student's identification and the image the testing suite captured of them. We conduct our analysis on subjects in the two primary ethnicities provided in the Morph-II dataset: "White" and "Black"—noting the complexities that come from assigning racial classifications based on facial skin tone, one of the primary drivers of model unfairness. We randomly select an equal number of subjects from each racial group and from these groups further select subjects so that the average time between captures is the same for both groups. The result is an average time between captures of 4.4 years with a total of 2,684 subjects in each group.

1) False Non Match Rate (FNMR): We evaluate the false non match rate of the various algorithms across subjects in the African American and European racial groups to quantify the number of students in each group who would be flagged by the system—indicating that the system believes the test taker is 'cheating'. We find the average false non match rate for the African American group to be lower than the European group for all of the algorithms except VGG-Face as can be seen in Table IV. We calculate the variance in the results for each racial group to determine whether a certain group would have an increased number of outliers above the 0.60 distance threshold used by Examplify (we provide a more complete product specific analysis in Section V-C). We find the number of subjects above the cheating threshold track the false non match rate results with more subjects in the European group above the threshold for cheating for all of the algorithms except VGG-Face.

2) False Match Rate (FMR): We measure the false match rate by comparing every subject in our selection of subjects to each subject in their racial group subset. A false match would occur where the test taker is not the indented individual (ie, a hired test-taker), but the system falsely concludes the two individuals are the same. The average false match rate for the African American group is higher than the average false match rate for the European group for all of the algorithms we tested. The false match rate average across both racial groups is 1 in 150 which is a relatively high false match rate when compared to the false match rate the LFW benchmark would suggest for the algorithms.

3) Image Quality: We use FaceQNet [35] which is based on the International Civil Aviation Organization (ICAO) image standard used for passports on the images of subjects in both racial groups to quantify the quality of a subject's image. We calculate the number of non-compliant images in each group and find 5% more of the images in the African American group were compliant with the standard. Previous work cited a bias against African American subjects in the ICAO standard which may suggest an even larger bias in favor of the African American group in our sample set once the bias is corrected.

4) Analysis: Facial recognition algorithms verifying African American subjects with a lower false non match rate than European subjects is a novel result that is rarely seen in facial recognition studies. Conversely, we find the opposite result in our false match rate analysis which shows that African Americans are more likely to be incorrectly verified compared to Europeans. We verify the image quality of subjects in each racial group to determine whether this is influencing the final result and find a slight bias in favor of the African American group. We also calculate the number of subjects in each group who were noted to have facial hair or eyewear differences between their captures and find similar numbers on subjects in each group. We conclude that this result may be due to feature extraction being calibrated for European subjects resulting in feature vectors that do not accurately describe the facial features they intend to map for the African American group. By more randomly placing these feature points, the average distance may be reduced compared to correctly placed feature points but this also reduces the accuracy of the algorithm resulting in the higher false match rate we see in the African American group.

B. Continuous Identity Verification

The Multi-PIE dataset provides images of 337 subjects in 15 different head positions under 19 different illumination conditions. This allows us to accurately simulate the varying facial tilts encountered by a facial recognition system taking random student captures during an exam. We analyze 15, 30, 45, 60, and 90 degree head angles on subjects classified as "Asian American" and "European" which results in 200 and 120 subjects respectively. We do not include the African American subjects in this analysis as there was not a statistically sufficient number of subjects in the dataset.

1) False Non Match Rate (FNMR): We observe similar false non match rates for subjects in either racial group for facial tilts up to +/-15 degrees with almost every subject being successfully recognized by the systems. We begin to see divergence in the recognition accuracy for the different classifiers after +/- 30 degrees facial tilt. Subjects in the

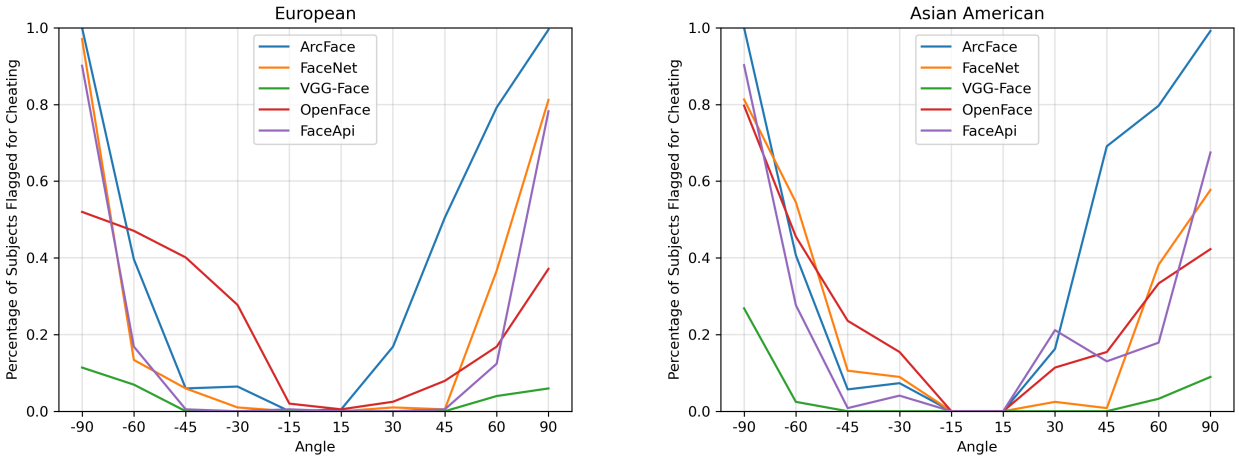

Figure 2: False Non Match Rates above 'cheating' threshold. These graphs depict the percentage of subjects in the Asian American and European groups respectively who have a false non match rate above Examplify's cheating threshold of 0.60 across a variety of angles.

"Asian American" group experience a higher false non match rate for all of the algorithms once +/-30 degrees facial tilt is achieved. We use the 0.60 distance threshold used by a real world exam proctoring system to calculate the number of subjects who would be flagged for cheating based on the facial tilt they exhibited during the capture. We see a close mirror of our raw false non match rate results with a significantly higher percentage of "Asian American" subjects flagged for cheating than "European" subjects at facial tilts above +/-30 degrees. The VGG-Face classifier outperforms all of the other algorithms for the false non match rate across subjects in either racial group but a bias against subjects in the "Asian American" group is still seen at facial tilts exceeding +/-45 degrees.

2) False Match Rate (FMR): We evaluate the false match rate for subjects in either racial groups by comparing each subject to all of the other subjects in the same racial group. We find a higher average false match rate for subjects in the "Asian American" group versus the "European" group for all facial tilts. We note a significant increase in the bias between racial groups once a +/- 30 degree facial tilt is reached. VGG-Face outperforms again with the lowest false match rate across all facial tilts and racial groups. A bias is still exhibited with VGG-Face once the facial tilt exceeds +/- 60 degrees, but this is a significantly more extreme facial tilt than the tilt at which the other algorithms diverged.

3) Image Quality: We evaluate the image quality of the images in each group using FaceQNet to determine if any of the variability we saw in the false match and false non match rate performance can be attributed to varying image quality. We find no non-compliant images in any of the images used for subjects in either racial group for facial tilts under +/-30 degrees. This is to be expected as Multi-PIE uses a highly controlled posing system and illumination setup. We find increasing non-compliance rates for facial tilts above +/-30 degrees which is also expected given that the ICAO standard FaceQNet is based on expects straight on passport style captures.

4) Analysis: We see the performance of all of the systems degrade as the facial tilt increases. This can be reasonably attributed to the difference in image quality as measured by the ICAO standard as more of the face is occluded. When we evaluate the performance in the context of the 0.60 false non match rate threshold used by Examplify to detect cheating, the results suggest a high rate of false positives would be recorded. The majority of algorithms we tested would flag a student for cheating if they turned their head more than +/-30 degrees when the randomized capture took place as can be seen in Figure 2. We argue that this is a reasonable facial tilt for a student to exhibit during an exam and would not be considered cheating in an exam hall environment. VGG-Face out performed all of the other algorithms by beginning to significantly flag students once they achieved a +/-45 degree facial tilt.

We see a higher false match and false non match rate for subjects in the "Asian American" group once a +/-30 degree facial tilt is reached. We theorize that this is due to the classifier being less optimized to extract facial feature vectors from subjects in the "Asian American" group. Previous studies have also raised the possibility that capture conditions are optimized for European faces causing other racial groups to either be under or overexposed [36]. The ICAO standard would not necessarily flag the image quality for an exposure issue that affected the classifier performance as it defines a relatively large acceptance window in relation to image exposure.

C. Evaluating Real World Implementation

We reverse engineer the facial recognition module Examplify uses to provide identity verification for exams. We find 'face-api.js' was used as the classifier with the publicly available models found on the project's Github. Examplify uses the two distinct verification modes we described above with a 0.60 distance used as the threshold for when to flag a student for cheating. We note that when a student is flagged, the images are sent to a remote server run by Examplify which may employ a

separate secondary automated verification step. We are unable to test this further as we were not able to obtain a test exam in which we had administrative privileges to see what images were finally marked as potential violations.

We find the face-api.js classifier that Examplify uses underperforms the open source state of the art algorithms in both false non match rate and false match rate on the Morph-II dataset which simulates the initial verification step. The high false non match rate and significant variance suggests a student may run into a situation where they are unable to begin their exam due to the automated verification failing. Examplify does not currently have a manual override that would allow a student to bypass this automated verification step in favor of a human verification step.

Similar results can be seen for the 'face-api.js' classifier on the Multi-PIE dataset. Students would begin being flagged once they achieved a facial tilt greater than 30 degrees. This does not result in a denial of service concern like the initial identity verification step but rather flags the student's exam for an instructor to review while grading. This raises concerns over prejudicing the instructor to believe the student cheated based on the presentation of the alert or causing them to simply take disciplinary action on the presence of an alert. The variability is further increased by Examplify using a random silent capture window versus a prompted verification step where a student is instructed to look straight at the camera.

Additionally, we see significant variability in both verification steps based on race. We see higher incidents of flagging for Europeans in the initial verification step and a significantly higher incident of flagging for Asian Americans in the continuous verification step. Given the variability and bias in the facial recognition steps, we believe a human based verification model is a significantly fairer approach to insuring exam integrity.

If an automated classifier is to be used, we recommend training models on a dataset that contains a balanced sampling of subjects from different races versus using pretrained default models. We also recommend evaluating the performance of the algorithms on datasets that realistically represent the use case of the system. The datasets that are commonly used for classifier performance evaluation, such as LFW, cannot be accurately used as an overall benchmark when the images the production system will be processing greatly differ from the dataset used for evaluation. Based on our analysis of our two datasets, chosen to closely resemble images a remote proctoring facial recognition system would process, VGG-Face outperforms 'face-api.js'. Therefore, we believe that adopting VGG-Face would significantly improve the accuracy of the system and slightly reduce the bias. We evaluate the average comparison time of VGG-Face versus 'face-api.js' and note similar comparison times making the cost of adoption minimal.

VI. Discussion

Impact on Marginalized Groups. Minorities and other marginalized groups are traditionally underrepresented in the legal profession. Using exam software with built-in biases, such as incorrectly flagging certain races for cheating at higher rates, creates an invisible barrier which will likely harm the chances of minorities successfully entering the legal profession and will perpetuate the systemic racism issues the US is grappling with today. Such invisible discrimination, which shows up in the US legal system in the form of unequal enforcement of laws by police and courts, must not be allowed to pervade academia unchecked. Further, while law schools, medical schools, and other tertiary educational institutions have put emphasis in recent years on diversity and addressing the underrepresentation of minorities on their campuses, such efforts are in vain if exam administration is plagued with invisible racial biases. Additional research should be performed by law schools, bar associations, and other educational and licensing institutions on the invisible biases in the exam software they utilize so that marginalized groups are not discriminated against by exam administration processes.

Fundamental Challenges of Remote Examination. Remote exam proctoring suites suffer from a few fundamental limitations in the threat model they can protect against since they are running on untrustworthy hardware unlike traditional exam hall based computerized exams. Security features and monitoring schemes can be created but since the student has full administrative access they will always be able to bypass these protections given enough time. Exam proctoring suite vendors can attempt to increase the time and skill level necessary to compromise the exam by adding complexity to the process through techniques such as obfuscation and active anti-debugging measures.

In order to create a truly secure remote exam proctoring suite, a vendor needs to establish a trusted environment on the device that restricts the student's ability to extract or modify part of the exam suite. Intel SGX and other similar trusted execution environments could potentially be employed for this, however, it may introduce significant usability and availability concerns due to a much more complex bootstrapping procedure. An application providing a rich user experience while providing sufficient security guarantees through Intel SGX is currently an unsolved problem due to significant components of the operating system having to run outside of the trusted environment in the current operating system model such as the video card driver.

Risks of Privileged Software. The software we evaluated all request privileged system access. Operating systems increasingly restrict and safeguard such access—a common source of malfeasance. Buggy but well-intentioned code that is given such access substantially broadens the attack surface of the OS and serves as a glaring target. This is so critical that experts increasingly recommend against granting such access even to third-party antivirus software [37]!

Compounding the problem, students are likely to be unaware that privileged system services from the proctoring packages persist well after the exam is over and do not uninstall automatically [11], putting them at long term risk.

Privacy, Surveillance, and Ethical Concerns. In an attempt to meet their design goals, platforms engage in sweeping surveillance, with many taking a 'kitchen sink' approach. This includes everything from keylogging, screen captures, drive access, process monitoring, and A/V feeds, through which the software has access to personal data stored on the device. Outside of this context, these features appear only in malware, highlighting the unusual capabilities of these software. Some

of the binaries also included anti-debugging mechanisms in their code, further limiting the ability of student advocates to assess the security and safety of the software.

The context in which students are required to install proctoring software mitigates their ability to meaningfully consent to the substantial impositions on their privacy and security. This is compounded by the veneer of legitimacy of the platform conveyed by the institutional backing of the school or testing company. Students thus are not in a position to meaningfully object to the use of these platforms and even if they did, are not provided with a reasonable alternative. This dynamic is partially captured in the results of Balash, ET AL. [11] who find that trust in institutions substantially reduces the extent to which students are willing to object to remote proctoring.

Recommendations. The strongest recommendation we offer is that where allowable, educators should design assessments that minimize the possible advantage to be gained by cheating. Project work, open book essays, or other similar modes with unrestricted access to resources feature fewer opportunities to gain unfair advantage.

Where re-imagining of assessment is not possible, students should be offered a *meaningful* chance to opt-out of digital testing on their own hardware and should be given the choice of using either provided hardware or paper-copy and live proctoring. Furthermore, schools often put substantial efforts into helping students install proctoring software and should match this with equal efforts to help them uninstall the software while advising them of the risks of retaining it.

Given the substantial fairness concerns with facial recognition systems, our primary recommendation is to avoid using these systems. Where infeasible, it is vital that human review remain a central component and that such review be conducted by multiple diverse individuals so as to reduce human biases as well. Lastly, until such time as more advanced facial recognition technology is developed, if these current algorithms are going to see continued use, candid conversation regarding generalized differences in facial features between racial groups needs to be addressed by programmers through dialogue with racially diverse focus groups and accounted for in calibration settings in an attempt to reduce the incidence of false identification.

ACKNOWLEDGEMENTS

We thank Paul Ohm for early discussions on the project.

REFERENCES

[1] Raman, R., Sairam, B., Veena, G., Vachharajani, H., and Nedungadi, P., Adoption of online proctored examinations by university students during covid-19: Innovation diffusion study, *Education and Information Technologies*, 1–20, 2021 (cit. on p. 1).

[2] Reed, A. *Online Bar Exams Come With Face Scans, Bias Concerns*, Bloomberg Law, https://news.bloomberglaw.com/health-law-and-business/online-bar-exams-come-with-face-scans-discrimination-concerns, 2020 (cit. on p. 1).

[3] Singer, N. *Online Cheating Charges Upend Dartmouth Medical School*, NYTimes, 2021 (cit. on p. 1).

[4] Thurlow, M., Lazarus, S. S., Albus, D., and Hodgson, J., Computer-based testing: Practices and considerations. synthesis report 78. *National Center on Educational Outcomes, University of Minnesota*, 2010 (cit. on p. 1).

[5] Langenfeld, T., Internet-based proctored assessment: Security and fairness issues, *Educational Measurement: Issues and Practice*, vol. 39, no. 3, 24–27, 2020 (cit. on p. 3).

[6] Turani, A. A., Alkhateeb, J. H., and Alsewari, A. A. Students online exam proctoring: A case study using 360 degree security cameras, in *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, IEEE, 2020, 1–5 (cit. on p. 3).

[7] Slusky, L., Cybersecurity of online proctoring systems, *Journal of International Technology and Information Management*, vol. 29, no. 1, 56–83, 2020 (cit. on p. 3).

[8] Teclehaimanot, B., You, J., Franz, D. R., Xiao, M., and Hochberg, S. A., Ensuring academic integrity in online courses: A case analysis in three testing environments, *The Quarterly Review of Distance Education*, vol. 12, no. 1, 47–52, 2018 (cit. on p. 3).

[9] Cohney, S., Teixeira, R., Kohlbrenner, A., ET AL., Virtual classrooms and real harms, *USENIX Symposium on Usable Privacy and Security*, 2021 (cit. on p. 3).

[10] Karim, M. N., Kaminsky, S. E., and Behrend, T. S., Cheating, reactions, and performance in remotely proctored testing: An exploratory experimental study, *Journal of Business and Psychology*, vol. 29, no. 4, 555–572, 2014 (cit. on p. 3).

[11] Balash, D. G., Kim, D., Shaibekova, D., Fainchtein, R. A., Sherr, M., and Aviv, A. J., Examining the Examiners: Students Privacy and Security Perceptions of Online Proctoring Services, *USENIX Symposium on Usable Privacy and Security*, 2021 (cit. on pp. 3, 13, 14).

[12] Leslie, D., Understanding bias in facial recognition technologies, *arXiv preprint arXiv:2010.07023*, 2020 (cit. on p. 3).

[13] Wu, W., Protopapas, P., Yang, Z., and Michalatos, P. Gender classification and bias mitigation in facial images, in *12th ACM Conference on Web Science*, 2020, 106–114 (cit. on p. 3).

[14] Singh, R., Agarwal, A., Singh, M., Nagpal, S., and Vatsa, M. On the robustness of face recognition algorithms against attacks and bias, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, 13583–13589 (cit. on p. 3).

[15] Nagpal, S., Singh, M., Singh, R., and Vatsa, M., Deep learning for face recognition: Pride or prejudiced? *arXiv preprint arXiv:1904.01219*, 2019 (cit. on p. 3).

[16] Grother, P. J., Grother, P. J., and Ngan, M. *Face recognition vendor test (frvt)*. US Department of Commerce, National Institute of Standards AND Technology, 2014 (cit. on p. 3).

[17] Feathers, T., Proctorio is using racist algorithms to detect faces, *VICE*, 2021, https://www.vice.com/en/article/g5gxg3/proctorio-is-using-racist-algorithms-to-detect-faces (cit. on p. 3).

[18] Johnson, E., Hey @proctorio @artfulhacker how do you explain this? *Twitter*, 2020, http://web.archive.org/web/20210715164139/https://twitter.com/ejohnson99/status/1303121786637373443 (cit. on p. 3).

[19] Swauger, S., *MIT Technology Review*, 2020, https://www.technologyreview.com/2020/08/07/1006132/software-algorithms-proctoring-online-tests-ai-ethics/ (cit. on p. 3).
[20] Be prepared: Law school doesnt even resemble your college experience, (cit. on p. 3).
[21] Appelbaum, P. S., Lidz, C. W., and Meisel, A., Informed consent: Legal theory and clinical practice, 1987 (cit. on p. 9).
[22] Cheng, E. Binary analysis and symbolic execution with angr, Ph.D. dissertation, PhD thesis, The MITRE Corporation, 2016 (cit. on p. 9).
[23] Ricanek, K., and Tesafaye, T. Morph: A longitudinal image database of normal adult age-progression, vol. 2006, May 2006, 341–345 (cit. on p. 10).
[24] Gross, R., Matthews, I., Cohn, J., Kanade, T., and Baker, S., Multi-pie, *Image and vision computing*, vol. 28, no. 5, 807–813, 2010 (cit. on p. 10).
[25] Serengil, S. I., and Ozpinar, A. Lightface: A hybrid deep face recognition framework, in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, 2020, 23–27 (cit. on p. 10).
[26] Kemelmacher-Shlizerman, I., Seitz, S. M., Miller, D., and Brossard, E. The megaface benchmark: 1 million faces for recognition at scale, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 4873–4882 (cit. on p. 10).
[27] Parkhi, O., Vedaldi, A., and Zisserman, A. *Deep face recognition. university of oxford*, 2015 (cit. on p. 10).
[28] Deng, J., Guo, J., Yuxiang, Z., Yu, J., Kotsia, I., and Zafeiriou, S. Retinaface: Single-stage dense face localisation in the wild, in *arxiv*, 2019 (cit. on p. 10).
[29] Guo, J., Deng, J., Xue, N., and Zafeiriou, S. Stacked dense u-nets with dual transformers for robust face alignment, in *BMVC*, 2018 (cit. on p. 10).
[30] Deng, J., Roussos, A., Chrysos, G., ET AL., The menpo benchmark for multi-pose 2d and 3d facial landmark localisation and tracking, *IJCV*, 2018 (cit. on p. 10).
[31] Deng, J., Guo, J., Niannan, X., and Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition, in *CVPR*, 2019 (cit. on p. 10).
[32] Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, 815–823 (cit. on p. 10).
[33] Amos, B., Bartosz, L., and Satyanarayanan, M. "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016 (cit. on p. 10).
[34] Tefft, B. C., Driver license renewal policies and fatal crash involvement rates of older drivers, united states, 1986–2011, *Injury epidemiology*, vol. 1, no. 1, 1–11, 2014 (cit. on p. 11).
[35] Hernandez-Ortega, J., Galbally, J., Fierrez, J., Haraksim, R., and Beslay, L. Faceqnet: Quality assessment for face recognition based on deep learning, in *2019 International Conference on Biometrics (ICB)*, IEEE, 2019, 1–8 (cit. on p. 11).
[36] Vangara, K., King, M. C., Albiero, V., Bowyer, K., ET AL. Characterizing the variability in face recognition accuracy relative to race, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019 (cit. on p. 12).
[37] Anthony, S. *It might be time to stop using antivirus*, Ars Technica, https://arstechnica.com/information-technology/2017/01/antivirus-is-bad/, 2017 (cit. on p. 13).