Secure and Verifiable Coercion-Resistant Electronic Exam

Mohammadamin Rakeei*[1], Rosario Giustolisi[2], Gabriele Lenzini[1], Dhekra Mahmoud[3], Jannik Dreier[4], and Pascal Lafourcade[3]

[1] *SnT, University of Luxembourg*
[2] *Department of Computer Science, IT University of Copenhagen*
[3] *Université de Clermont Auvergne, LIMOS*
[4] *Université de Lorraine, CNRS, Inria, LORIA*

Total words: 9634

Abstract

Since they enable efficient assessment of large cohorts of students and test-takers, electronic exams (e-exams) have become popular. However, the transition from pencil-and-paper tests to e-exams comes with challenges: researchers needed to ensure a comparable level of security and privacy as that enjoyed before the transition; at the same time, they have to address threats due to the use of information and communication technology. Research has shown that, for the reason of assessment fairness, e-exams should satisfy a list of peculiar security properties, for instance, about authentication, secrecy, integrity, anonymity, and correctness, including their universal and individual verifiability. Recently, e-exams have been scrutinized for their resistance to collusion and coercion. Subsets of participants have an interest in teaming up, or forcing one another, to gain an unfair advantage over the honest others. In this work, we study coercion-resistance for e-exams. We propose a novel strong definition of coercion where all secrets are leaked to the attacker. Under this threat, we prove that a recent coercion-resistant exam protocol is subject to attacks. We improve the protocol by ensuring that all its properties are maintained and that it is coercion-resistant under the new threat model. Our new protocol is also verifiable, which is a must-have property whenever there is the need to prove that fairness is preserved despite anyone attempting to subvert it. All our claims are formally verified using ProVerif. Notably, our formal verification includes proving the security of a recent exponentiation mixnet framework proposed in the literature.

Keywords Coercion-Resistant, E-Exam Protocols, Formal Verification, Mixnet, Verifiability, ProVerif

*Corresponding author: 6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg Luxembourg, `amin.rakeei@uni.lu`

1 Introduction

People's skills and knowledge are generally assessed with tests or exams: universities run tests to select students and follow their progression and proficiency with quizzes; professional organizations release certifications after applicants have passed exams; and companies resort to IQ and attitudinal tests to decide on recruitment and promotion.

Since the 1990s, exams have relied more and more on information and communication technology. Depending on the degree of dependence, exams can be computer-aided or entirely computer-based, but in general, we talk in both cases of *electronic exams* (in short, e-exams). Electronic exams promise to be better than paper-and-pencil-based testing: they ease exam administration, enable faster display of results, and make identification and authentication of candidates stronger.

However, the hasty adoption of information and communication technology and the development of testing applications have been motivated by efficiency reasons and by educational best practices rather than security. In the professional context of computer-based assessment and testing, the few discussions about security have remained informal, addressing standard systems security issues, such as access control, password strength, and network security.

Only recently, the security of exams have become the subject of research and thus of formal analysis. Dreier *et al.* [11] were the first to define a framework in the applied π-calculus [1] to analyze authentication and privacy, taking two existing electronic exam protocols as use cases. The formal approach to security led to the discovery of numerous attacks, calling for new designs and leading to the definition of peculiar security properties. This also includes individual and universal verifiability, whose formal analysis in traditional and electronic exams has been pioneered in [10]. With time, the set of security requirements got richer: for instance, accountability, anonymous marking, and dispute resolutions were formalized and realized in a prototype by Bella *et al.* [3].

Recently, electronic exam protocols have been studied for their resilience to coercion and collusion [23]. Differently from other domains such as e-voting, where coercion has the goal of directing votes towards specific candidates, in the world of e-exams, coercion happens because of an interest to corrupt certain parties (*e.g.,* examiners), to subvert what is probably the most important high-level and transversal principle guiding any exam: a fair assessment for anyone. Coercion threatens that process, especially when one party tries to change the outcome of the assessment. For instance, a candidate could blackmail an examiner to get higher marks; in anonymous marking, an examiner can force a test taker to reveal his/her identity to bias the assessment in his/her disfavour. Collusion also threatens fairness when a subset of players tries to take advantage of other honest parties, including the exam administration, to tamper with the markings in diplomas and certifications, assets of extreme importance since, with them, one can get access to jobs or apply for VISAs.

In pioneering the study of coercion in electronic exams, Rekeei *et al.* [23], proposed a protocol called CREX, which they proved coercion-resistance under a threat model where some, but not all, private information are leaked to the attacker/coercer. Getting technical, the security of CREX also relies on a specific assignment sub-protocol, only informally argued to be secure while formally assumed to be so in the formal analysis. These are known limitations, and as we will see later, by refining the model using a concrete instantiation of the assignment protocol, the protocol suffers from attacks.

CREX is surely an interesting and rich protocol, the first one that addresses coercion in exams, but differently from the other protocols proposed previously, it does not ensure verifiability, a class of properties that is necessary accountability [7] and in support of that overarching guarantee of fairness that must hold for the whole assessment process. We also provide a novel design that instead is fully verifiable.

Contributions Also this work is about coercion resistance. It builds from the state of the art in this topic, which is the mentioned protocol CREX. In particular, it studies the limitations of the protocol, finds and describes a few vulnerabilities that in the original protocol remained hidden under optimistic assumptions, proposes a stronger definition of coercion resistance, and extends CREX with more security against coercion under the stronger threat model. Since we stressed the importance of ensuring verifiability, this work proposes a coercion-resistant protocol that is also fully verifiable. All the security properties, the new and the old, are validated in Proverif with a substantial work of modelling and with several rounds of formal analysis (all the code is available). Listed in items, the contributions of this work are:

- a stronger definition of coercion, where all pieces of information are leaked to the intruder. Under this new definition, we revisit the definition of coercion, and in particular the security of CREX. This new definition allows us to discover a weakness in the Anonymous Submission property. We also refine the assignment sub-protocol from CREX, which allows us to find a second attack on privacy.

- a new protocol that corrects the issues of the CREX protocol and offers better security. We call this protocol *Secure* CREX, in short, SCREX. In SCREX, we conserve the spirit of CREX, but modify some message exchanges to achieve strong coercion resistance and verifiability (missing in the CREX design).

- a concrete instantiation of the assignment protocol using a recent MixNet model proposed in [12] We are also the first to give a full formal model in Proverif and a complete security analysis of this particular Mixnet, whose security has been, so far, only informally argued.

- a full model of SCREX and all its cryptographic components and a rigorous formal analysis in Proverif of all the security claims we have on the protocol. This includes a formal analysis of the verifiability of SCREX. All our code and proofs are available as an artefact in an public repository.

Outline Section 2 discusses the related work on e-exams and recalls the basics of the applied π-calculus and ProVerif, the automatic verification tool that we use to prove the security of our protocol. In Section 3, we present how to model electronic exams in ProVerif and review relevant security properties: authentication, privacy and verifiability. In Section 4, we describe our novel SCREX protocol. Section 5 discusses the model of the protocols that we used for the formal analysis of SCREX, including that of the Mixnet, and the result of the analysis. All our ProVerif files are available online [1].

2 Previous works, Background, and Tools

The discussion about security in e-learning and online exams was first introduced by Foley *et al.* in [13], where they formalized confidentiality in electronic exams. The notions of security and privacy were later informally expanded in [14], [21], and [26]. Giustolisi *et al.* in [16] introduced additional security properties and proposed a standard e-exam system by defining essential security properties for the various players and phases of an e-exam protocol. Dreier *et al.* [11] proposed the first formal framework in applied π-calculus for defining and analyzing authentication and security properties of both traditional and electronic exam protocols. Dreier *et al.* in [10] defined another framework to formally express and prove verifiability properties in e-exams. Rakeei *et al.* in [23] introduced the notion of coercion in e-exams and formalized it by defining two new properties.

In the literature, several e-exam protocols have been proposed with different threat models. Castella-Roca *et al.* [8] designed an e-exam management system with student, teacher, and manager entities, whose security is based on the existence of a trusted-third-party (TTP) in the manager role. Huszti and Petho in [20] proposed a new e-exam protocol that aimed to offer security and privacy guarantees without relying on a trusted party. However, Dreier *et al.* [11] demonstrated an attack on the secrecy and anonymity of the Reusable-anonymous-return-channel [18], the main building block of [20], showing that their scheme could not meet the main security and privacy requirements. Giustolisi *et al.* in [17] designed the *Remark!* e-exam protocol with minimal trust assumptions to ensure authentication, privacy, and verifiability properties.The guarantee for the anonymity of candidates and examiners is mainly provided by exponentiation mixnet servers, assuming that only one of them is trusted. Recently, Rakeei *et al.* [23], through pen-and-paper cryptanalysis, and Dreier *et al.* [12], through a model checker security proof, demonstrated a linkability attack on Haenni-Spycher's exponentiation mixnet [19] used in *Remark!*, breaking all anonymity features of the *Remark!* protocol. Bella *et al.* [2] developed WATA IV, a computer-assisted exam protocol in the presence of corrupted examiners and an honest-but-curious TTP as the anonymizer. In WATA IV, visual cryptography, a kind of secret sharing scheme, plays the main role in anonymizing candidate identities. Although informally discussed, the authors

[1] ProVerif Files

equipped their scheme with a set of verifiability tests and a dispute resolution mechanism. They extended their research on the WATA IV protocol in [4] by adding more security requirements and formally proving its security using the ProVerif model checker [5]. Rakeei *et al.* [23] proposed the CREX protocol as the first e-exam scheme designed to address the coercion threat model. They introduced a fix to prevent the linkability attack on Haenni-Spycher's mixnet and used this secure version of the mixnet to ensure anonymity of identities in their protocol. The most recent protocol in the literature, advanced by Vecsi and Petho [24], focuses on accountability. They adopted a similar shuffled answer technique and pre-assignment protocol used in [23] to mitigate collusion during the marking phase. To establish anonymous communication, the authors used a modified version of the Loopix anonymizer [22], called the Scalix network [25], which features enhanced logging for auditing authorities.

2.1 Applied Π-Calculus

In this work, we adopted the symbolic model to formally represent our protocol using the applied π-calculus [1] and employed the ProVerif verification tool [6] for analysis. The symbolic model allows cryptographic protocols to be analyzed under the Dolev-Yao attacker model [9], where the adversary has complete control over the network—able to read, modify, delete, or inject messages. However, the attacker can only execute cryptographic operations if they possess the correct keys, an assumption known as perfect cryptography. The applied π-calculus, an extension of the π-calculus, serves as a language for modeling security protocols and is designed to describe and analyze the properties of concurrent computations. In the following, we provide an overview of the ProVerif tool used to analyze our protocol and present the formal modeling of e-exams in the applied π-calculus.

2.2 ProVerif

A very large number of tools exists for verifying security protocols in the symbolic model. The protocol verifier ProVerif [5] analyzes protocols using an abstract internal representation via a set of Horn clauses. ProVerif is fully automatic: the user only gives the specification of the protocol and the property to verify. For example, to test the secrecy of a ground term M, i.e., whether the attacker can learn M, the following query is included in the input file before the main process: **query attacker** (M).

The tool can handle many different cryptographic primitives (encryption, signature, hash function, Diffie-Hellman Key agreement) specified as rewrite rules or equations. ProVerif can also handle an unbounded number of sessions and an unbounded message space.

ProVerif is able to prove the following properties:

- Secrecy (as reachability properties): the adversary cannot learn the secret.

- Authentication (as correspondence properties): if an event has been executed, then other events have been executed as well.
- Equivalences between processes (as observational equivalence): the adversary does not see the difference between two processes.

3 Formalizing e-Exams and Security Properties

This section reminds a formal toolkit to model and analyze e-exams in applied π-calculus and ProVerif.

We refer to the e-exam model in applied π-calculus introduced in [11], while for all the properties here in considered, we refer to the formalization proposed in different works [11, 10, 23].

The reason for referring to these works, in addition to relying on models and definitions that work perfectly fine, is that we intend to analyze the security of our novel protocol against all those previous properties before showing that our protocol is also correction-resistant and verifiable. In other words, our protocol must also preserve all the good properties of previous state-of-the-art e-exam design(s). Using the same formal framework of modelling and analysis of previous work is, therefore, a sound experimental choice to ensure consistency with state-of-the-art research and replicability for future research.

3.1 E-Exams Modelling

Electronic exams have been formally defined by Dreier *et al.* in [11]. In a nutshell, an *exam protocol* is a tuple $(C, E, Q, A_1, \ldots, A_l, \tilde{n}_p)$, where C is the process executed by the candidates, E is the process executed by the examiners, Q is the process executed by the question committee, A_i's are the processes executed by the authorities, and $\tilde{n}_p$ is the set of private channel names. Given the tuple $(C, E, A_1, \ldots, A_l, \tilde{n})$ an *exam instance* is a closed process $EP = \nu\tilde{n}.(C\sigma_{id_1}\sigma_{a_1}|\ldots|C\sigma_{id_j}\sigma_{a_j}| E\sigma_{id'_1}\sigma_{m_1}|\ldots|E\sigma_{id'_k}\sigma_{m_k}|Q\sigma_q|A_1\sigma_{dist}|\ldots|A_l)$, where $\tilde{n}$ is the set of all restricted names, which includes the private channels; $C\sigma_{id_i}\sigma_{a_i}$'s are the processes run by the candidates, the substitutions σ_{id_i} and σ_{a_i} specify the identity and the answers of the i^{th} candidate respectively; $E\sigma_{id'_i}\sigma_{m_i}$'s are the processes run by the examiners, the substitution $\sigma_{id'_i}$ specifies the i^{th} examiner's identity,and σ_{m_i} specifies for each question/answer pair the corresponding mark; Q is the process run by the question committee, the substitution σ_q specifies the exam questions; the A_i's are the processes run by the exam authorities, the substitution σ_{dist} determines which answers will be submitted to which examiners for grading.

Given the process EP, $EP_I[\cdot]$ is the corresponding context, without the identities in the set I and $EP|_e$ stands for process EP without the code that follows the event e.

3.2 Authentication Properties

In ProVerif, authentication properties are proven using correspondence assertions based on defined events. In the applied π-calculus, these events are internal messages that, along with arbitrary arguments, act as markers to capture the state of a process at specific points within a protocol trace. They do not affect the protocol's behavior and are inaccessible to attackers. Instead, they help the verifier reason about the reachability of certain states and thereby model and prove authentication properties. To prove authentication properties in our SCREX protocol, we first define all relevant events and the situations in which they are emitted as follows:

- $\texttt{registered}(id_c)$: triggered by the exam authority at the point where the registration of candidate (id_c) is completed.
- $\texttt{submitted}(id_c, ques, ans)$: raised by the candidate id_c when she submits the test $(ques, ans)$ to the exam authority.
- $\texttt{collected}(id_c, ques, ans)$: triggered by the exam authority at the location where it accepts the test $(ques, ans)$ from the candidate id_c.
- $\texttt{marked}(id_e, ques, ans, mark)$: triggered by examiner id_e at the point where he assigns the mark $mark$ to the test $(ques, ans)$.
- $\texttt{recorded}(id_c, id_e, ques, ans, mark)$: triggered by the exam authority at the point where it officially records the mark $mark$, assigned by examiner id_e to the test $(ques, ans)$ of candidate id_c.
- $\texttt{notified}(id_c, id_e, ques, ans, mark)$: triggered by candidate id_c when she accepts the mark $mark$, assigned by examiner id_e to her submitted test $(ques, ans)$.

Based on the events defined earlier, we now define four authentication properties within the SCREX protocol. The first property, called *Test Origin Authentication*, ensures that the exam authority accepts tests only from registered candidates.

Definition 3.1 (Test Origin Authentication [15]) An exam protocol guarantees Test Origin Authentication if for every exam process EP the following assertion holds across all execution traces

$$inj\text{-}\texttt{collected}(id_c, ques, ans) \rightsquigarrow inj\text{-}\texttt{registered}(id_c)$$

Note that the term *inj* refers to an injective correspondence assertion, signifying a one-to-one relationship between events. This means that each occurrence of the event $\texttt{collected}(id_c, ques, ans)$ is preceded by a unique occurrence of the event $\texttt{registered}(id_c)$.

The next property is *Answer Authenticity*, which ensures that the content of the candidate's submitted test is not modified when the ea collects it.

Definition 3.2 (Answer Authenticity [15]) An exam protocol ensures Answer Authenticity if for every exam process EP the following assertion holds across all execution traces

$$inj\text{-}\texttt{collected}(id_c, ques, ans) \rightsquigarrow inj\text{-}\texttt{submitted}(id_c, ques, ans)$$

The *Mark Authenticity* property guarantees that the mark assigned to a test by the examiner is recorded without any modification.

Definition 3.3 (Mark Authenticity [15]) An exam protocol ensures Mark Authenticity if for every exam process EP the following assertion holds across all execution traces

$$inj\text{-}\texttt{recorded}(id_c, id_e, ques, ans, mark) \rightsquigarrow inj\text{-}\texttt{marked}(id_e, ques, ans, mark)$$

The final property, *Mark Authentication*, certifies that the mark the candidate receives for her test at the end of the examination is recorded without any changes.

Definition 3.4 (Mark Authentication [15]) An exam protocol guarantees Mark Authentication if for every exam process EP the following assertion holds across all execution traces

$$inj\text{-}\texttt{notified}(id_c, id_e, ques, ans, mark) \rightsquigarrow inj\text{-}\texttt{recorded}(id_c, id_e, ques, ans, mark)$$

3.3 Privacy Properties

We give formal definitions of privacy properties of exam protocols as defined in [11]. These properties are modeled as observational equivalences, a standard choice for such kind of properties [1]. We use *labeled bisimilarity* ($\approx_l$) to express the equivalence between two processes. In a nutshell, two processes are considered equivalent if an observer cannot differentiate between them.

The first property called *Question Indistinguishability*, requires that two processes with different questions remain observationally equivalent until the end of the registration phase. This prevents the attacker from acquiring information about the exam questions before the examination phase begins. For this property to hold, the question committee must be honest; otherwise, it can be easily compromised if the committee reveals the questions to the attacker.

Definition 3.5 (Question Indistinguishability [11]) *Two processes with different questions have to be observationally equivalent until the registration phase. For every e-exam process EP that ends with the registration phase any questions q_1 and q_2*

$$EP_{\{id_Q\}}[Q\sigma_{q_1}]|_{reg} \approx_l EP_{\{id_Q\}}[Q\sigma_{q_2}]|_{reg}$$

The second property ensures that the evaluation procedure is carried out anonymously, i.e., that two instances in which candidates exchange their responses cannot be distinguished until the evaluation phase has concluded. This anonymity is desirable to guarantee impartiality in the grading process and may be a mandatory requirement in certain exam settings, such as those at specific universities or for competitive examinations.

Definition 3.6 (Anonymous Marking [11]) *A protocol ensures Anonymous Marking if the process where id_1 answers a_1 and id_2 answers a_2 is equivalent to the process where id_1 answers a_2 and id_2 answers a_1. Formally, for every e-exam process EP that ends with the marking phase, any two candidates id_1 and id_2 and any two answers a_1 and a_2:*

$$EP_{\{id_1,id_2\}}[C\sigma_{id_1}\sigma_{a_1}|C\sigma_{id_2}\sigma_{a_2}]|_{mark}\approx_l EP_{\{id_1,id_2\}}[C\sigma_{id_1}\sigma_{a_2}|C\sigma_{id_2}\sigma_{a_1}]|_{mark}$$

To prevent bribing or coercion of the examiners, it may be advantageous to guarantee their anonymity, thereby ensuring that no candidate is aware of the examiner who assigned their copy. The third property is meant to guarantee examiners' anonymity.

Definition 3.7 (Anonymous Examiner [11]) *A protocol ensures Anonymous Examiner if process in which examiner id'_1 grades the exam form of candidate id_1 and examiner id'_2 grades that of candidate id_2 cannot be distinguished from a process in which id'_1 grades the exam form of candidate id_2 and examiner id'_2 grades that of candidate id_1. Formally, for every e-exam process EP, any two candidates id_1 and id_2, any two examiners id'_1 and id'_2 and any two marks m_1 and m_2:*

$$\begin{aligned}EP_{\{id_1,id_2,id'_1,id'_2,id_{A_1}\}}[C\sigma_{id_1}\sigma_{a_1}|C\sigma_{id_2}\sigma_{a_2}|E\sigma_{id'_1}\sigma_{m_1}|E\sigma_{id'_2}\sigma_{m_2}|A_1\sigma_{dist_1}] \approx_l \\ EP_{\{id_1,id_2,id'_1,id'_2,id_{A_1}\}}[C\sigma_{id_1}\sigma_{a_1}|C\sigma_{id_2}\sigma_{a_2}|E\sigma_{id'_1}\sigma_{m_2}|E\sigma_{id'_2}\sigma_{m_1}|A_1\sigma_{dist_2}]\end{aligned}$$

The next property ensures that the mark itself has to remain private. Depending on the exam policy, this can be an optional property since some exam systems may publicly disclose the marks of the candidates.

Definition 3.8 (Mark Privacy [11]) *A protocol ensures Mark Privacy if two processes where the examiner id'_1 assigns for the same answer two different marks m_1 and m_2 cannot be distinguished from each other. Formally, for any e-exam process EP any marks m_1 and m_2*

$$EP_{\{id'\}}[E\sigma_{id'}\sigma_{m_1}] \approx_l EP_{\{id'\}}[E\sigma_{id'}\sigma_{m_2}]$$

3.4 Verifiability

Dreier *et al.* [10] propose a formalization to reason about individual and universal verifiability. We summarize the core ideas of the framework. The reader can refer to the original article for the full details of the definitions.

Definition 3.9 (Exam Abstract Model)[10]) *An exam E is a tuple $(I,Q,A,M,\ \alpha)$ where I of type $\mathcal{I}$ is a set of candidate identities, Q of type $\mathcal{Q}$ is a set of questions, A of type $\mathcal{A}$ is a set of answers, M of type $\mathcal{M}$ is a set of marks, and α is the set of the following relations:*

Definition	Individual Verifiability	Universal Verifiability
Registration		$\texttt{R}_{\texttt{UV}}(e) \Leftrightarrow$ $I_r \supseteq \{i : (i,x) \in \texttt{Accepted}\}$
Question Validity	$\texttt{QV}_{\texttt{I}}(i,q,a,m,p) \Leftrightarrow (q \in Q_g)$	
Marking Correctness	$\texttt{MC}_{\texttt{IV}}(i,q,a,m,p) \Leftrightarrow$ $(\texttt{Correct}(q,a) = m)$	$\texttt{MC}_{\texttt{UV}}(e) \Leftrightarrow$ $(\forall (i,x,m) \in \texttt{Marked},$ $\texttt{Correct}(x) = m$
Test Integrity	$\texttt{TI}_{\texttt{IV}}(i,q,a,m,p) \Leftrightarrow$ $\big((i,(q,a)) \in \texttt{Accepted}$ $\wedge \exists m' : (i,(q,a),m') \in \texttt{Marked}\big)$	$\texttt{TI}_{\texttt{UV}}(e) \Leftrightarrow \texttt{Accepted} =$ $\{(i,x) : (i,x,m) \in \texttt{Marked}\}$
Mark Integrity	$\texttt{MI}_{\texttt{IV}}(i,q,a,m,p) \Leftrightarrow$ $\exists m' : \big((i,(q,a),m') \in \texttt{Marked}$ $\wedge (i,m') \in \texttt{Assigned}\big)$	$\texttt{MI}_{\texttt{UV}}(e) \Leftrightarrow \texttt{Assigned} =$ $\{(i,m) : (i,x,m) \in \texttt{Marked}\}$
Mark Notification Integrity	$\texttt{MNI}_{\texttt{IV}}(i,q,a,m,p) \Leftrightarrow$ $(i,m) \in \texttt{Assigned}$	

Table 1: Individual and universal verifiability.

- $\texttt{Accepted} \subseteq I \times (Q \times A)$: *the candidates' tests accepted by the collector authority;*
- $\texttt{Marked} \subseteq I \times (Q \times A) \times M$: *the marks given to the candidates' tests;*
- $\texttt{Assigned} \subseteq I \times M$: *the marks assigned to the candidates;*
- $\texttt{Correct} : (\mathcal{Q} \times \mathcal{A}) \rightarrow \mathcal{M}$: *the function used to mark a test.*

Definition 3.9 can be extended with two specific subsets:

- $I_r \subseteq I$ as the set of identities of candidates who registered for the exam;
- $Q_g \subseteq Q$ as the set of questions generated by the question committee.

The structure of a verifiability requirement consists of a verifiability test function and a predicate modeling a specific property. The evidence needed for the verifiability test comes from information about the exam execution and from private knowledge of involved roles.

According to [10], an exam is *testable* if it provides an algorithm (i.e., the verifiability test) that checks a specific property on the exam execution. The verifiability test must be both sound and complete for the given property. In Table 1, this is represented by the symbol $\Leftrightarrow$.

Individual verifiability considers the perspective of exam candidates who have access to private knowledge acquired during the exam. Universal verifiability, on the other hand, involves any auditor who does not necessarily have prior knowledge of the exam. The auditor relies on public information to conduct the verifiability tests. The requirements concern the verifiability of candidate registration, the validity of questions, and the integrity of tests, marks, and notification. The requirements are concisely listed in Table 1.

Regarding individual verifiability, the candidate uses public information and private knowledge to verify some aspects of the exam. The candidate knows her identity i, the test she submitted, which consists of question q and answer a, and the notified mark m. The candidate also knows the perspective p from the exam process, which includes the messages sent and received by the candidate during the exam. Note that the candidate's perspective p is not necessary to specify the predicate that models the properties to verify. In fact, the perspective never appears in the predicate c. However, the perspective may be necessary for implementing the verifiability-test. The five individual verifiability requirements are Question Validity, Marking Correctness, Test Integrity, Mark Integrity, and Mark Notification Integrity

The definitions of universal verifiability are not pivoted around any exam role but consider the viewpoint of an external auditor. The auditor runs the verifiability tests on the public information available after an exam protocol run. Hence, the knowledge of the auditor consists of a general variable e that contains evidence. Moreover, the auditor does not necessarily know the questions and the answers given during the exam. This is captured in the definitions with a generic term x.

The four universal verifiability requirements concern the registration of the candidates and the integrity of the batch of tests from the submission until after the marking. Therefore, the requirements are similar to the ones already seen for individual verifiability, plus Registration. The requirements of Question Validity and Mark Notification Integrity, are only relevant for individual verifiability because the external auditor has no knowledge of the questions nor the marks received by the candidates, but only of public data.

3.5 Coercion Properties

Coercion resistance may be regarded as an extension of the basic property of privacy. It is a strong form of privacy that assumes the adversary can interact with examiners and candidates. Specifically, the adversary may instruct targeted examiners to reveal their private keys after registration or direct them to correct exams in a particular manner. If the adversary can determine whether the examiners followed these instructions, it gains the ability to influence the correction process. In [23], Rakeei *et al.* were the first to define and formalize coercion properties in e-exams. Coercion resistance means that an exam protocol should preserve privacy between candidates and examiners, even if one party attempts to force the other to reveal secrets that could lead to deanonymization.

The first coercion resistance property, called *Anonymous Submission*, is an enhanced version of the privacy property *Anonymous Marking* defined in Section 3.6. This property ensures that examiners cannot discover the link between candidates and their submitted tests, even if they coerce candidates into revealing their secrets. It relies on the distinguishability of the answers submitted by two known candidates.

Definition 3.10 (Anonymous Submission [23]) *An e-exam protocol guarantees Anonymous Submission if for every e-exam process that ends with the* *marking phase where a candidate id_1 answers c according to the intruder's wishes is indistinguishable from the process where id_1 answers something else (a_1), i.e. if there exists a closed process C' such that for all answers c and a_1:*

- $C'^{\backslash out(chc,\cdot)} \approx_l C\sigma_{id_1}\sigma_{a_1}$
- $EP_{\{id_1,id_2\}}[(C\sigma_{id_1}\sigma_c)^{chc}|C\sigma_{id_2}\sigma_{a_1}]|_{mark} \approx_l EP_{\{id_1,id_2\}}[C'|C\sigma_{id_2}\sigma_c]|_{mark}$

C' represents a process in which candidate id_1 answers a_1 while pretending to cooperate with the coercer. The coercer cannot distinguish between the scenario where id_1 genuinely cooperates and submits c and the scenario where id_1 feigns cooperation but actually submits a_1."

The second coercion resistance property named *Single Blindness* is a strong version of the privacy property *Anonymous Examiner* defined in Section 3.6. It aims to guarantee that candidates cannot learn the link between examiners and tests they marked even if candidates force examiners to reveal their secrets. It is based on the distinguishability of the marks provided by two examiners.

Definition 3.11 (Single Blindness [23]) *An e-exam protocol guarantees Single Blindness if for every e-exam process that ends with the marking phase where an examiner id'_1 marks an e-exam according to the intruder's wishes e is indistinguishable from the process where id'_1 marks something else m_1, i.e., if there exists a closed process E' such that for all marks e and m_1:*

- $E'^{\backslash out(che,\cdot)} \approx_l E\sigma_{id'_1}\sigma_{m_1}$
- $EP_{\{id'_1,id'_2\}}[(E\sigma_{id'_1}\sigma_e)^{che}|E\sigma_{id'_2}\sigma_{m_1}]|_{mark} \approx_l EP_{\{id'_1,id'_2\}}[E'|E\sigma_{id'_2}\sigma_e]|_{mark}$

E' describes a scenario where examiner id'_1 gives the mark m_1 while feigning cooperation with the coercer. The coercer cannot distinguish between genuine cooperation, where id'_1 assigns the mark e, and a situation where id'_1 is only pretending to cooperate but actually assigns m_1.

4 A New Verifiable Coercion-Resistant Protocol

The protocol, which we call Secure CREX (SCREX), includes four roles: Examiner(E), Mixnet(M), Exam Authority(EA), and Candidate(C). It is organized in five phases: *Registration*, *Assignment*, *Testing*, *Marking*, and *Notification*. We assume the existence of a Public Append-only Bulletin Board (B.B.) used for publishing public protocol parameters. Anyone has access to this B.B., including potential attackers; hence, any message posted on this board must be signed by its respective author. Additionally, we assume the presence of an Untappable channel between EA and C that prevents the attacker from discovering the contents of any communicated message and from being aware of the communication occurrence itself.

4.1 Registration

This phase aims to generate two sets of pseudonyms for examiners and candidates in a way that ensures only the owner of a public key can learn the corresponding pseudonym. To achieve this, the mixnet M takes as inputs the list of public keys for the examiners and candidates. Utilizing the secure exponentiation mixnet introduced in [23], it produces two lists of pseudo-public keys: one for the candidates and another for the examiners. The mixnet then signs these lists and publishes them on the bulletin board (B.B.). Later in Section 5, we demonstrate that this pseudonymization is necessary to achieve *Anonymous Marking*, *Anonymous Examiner*, *Anonymous Submission*, and *Single-Blindness*. The detailed generation of pseudonyms for candidates is illustrated in Figure 1. The same procedure is applied to the examiners. Note that in this setup, along with a public key, the mixnet receives a strong zero-knowledge proof (SZKP) as described in [12], which proves the knowledge of the corresponding private key.

	inputs	mix_1	mix_2		mix_m		outputs
C_1	$(PK_1, SZKP_1)$	$PK^{\overline{r}_1}_{\overline{\pi}_1(1)}$	$PK^{\overline{r}_2}_{\overline{\pi}_2(1)}$	$\cdots$	$PK^{\overline{r}_m}_{\overline{\overline{\pi}}_m(1)}$	$=$	$\overline{PK}_1$
C_2	$(PK_2, SZKP_2)$	$PK^{\overline{r}_1}_{\overline{\pi}_1(2)}$	$PK^{\overline{r}_2}_{\overline{\pi}_2(2)}$	$\cdots$	$PK^{\overline{r}_m}_{\overline{\overline{\pi}}_m(2)}$	$=$	$\overline{PK}_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		
C_n	$(PK_n, SZKP_n)$	$PK^{\overline{r}_1}_{\overline{\pi}_1(n)}$	$PK^{\overline{r}_2}_{\overline{\pi}_2(n)}$	$\cdots$	$PK^{\overline{r}}_{\overline{\overline{\pi}}_m(n)}$	$=$	$\overline{PK}_n$
	g	$g^{\overline{r}_1}$	$g^{\overline{r}_2}$	$\cdots$	$g^{\overline{r}_m}$	$=$	h_C

Figure 1: Pseudonym generation using secure exponentiation mixnet. All the terms within the box are published on the bulletin board.

4.2 Assignment Protocol

One of the responsibilities of the exam authority in an e-exam protocol is to distribute tests to eligible examiners. One possible solution to this problem is the *Assignment Protocol* introduced in [23]. However, this scheme has two main drawbacks: (i) The protocol is only discussed informally, with no formal proof provided to ensure its security. (ii) The authors implicitly assume that in case of examiner coercion, the coercer is not aware of the communications that occur during this protocol. In other words, the coercion threat is only considered during the Marking phase. To address these issues, we propose a secure assignment protocol that allows the exam authority to anonymously allocate a set of numbers, representing test identifiers, to a group of examiners. This ensures each examiner will only learn their assigned numbers and will have no knowledge of the numbers assigned to others. Additionally, the link between the numbers and the examiners remains secret from the exam authority. Later in Section 5, we demonstrate how our SCREX protocol, which utilizes this assignment protocol, maintains security across defined threat models, including examiner coercion. Now, we describe our *Assignment* protocol in two steps:

Step 1 Let's consider an exam with n candidates, d examiners. First, the exam authority EA randomly generates d sets denoted as $\{P_1,\dots,P_d\}$ where $P_i \subset \{1,\dots,n\}$, $\forall i \in \{1,\dots,d\}$. Then, the exam authority publishes message M_1 on the BB.

$$M_1 = \{Enc_1,\dots,Enc_d\} = \{Enc(P_1)_{PK_{EA}},\dots,Enc(P_d)_{PK_{EA}}\}$$

In the above message, Enc is a probabilistic encryption function and PK_{EA} is the public key of the exam authority. The idea of encrypting messages with PK_{EA} is hiding the content of the subsets $\{P_1,\dots,P_d\}$ and preparing a proof by the exam authority for designated examiners in the next step.

Step 2 Following publishing M_1, the exam authority securely transmits the following message

$$\begin{aligned} M_2 =& \{Enc'_1,\dots,Enc'_j,\dots,Enc'_d\} \\ =& \{Enc(P_1)_{\overline{PK}_{E_1}},\dots,Enc(P_j)_{\overline{PK}_{E_j}},\dots,Enc(P_d)_{\overline{PK}_{E_d}}\} \end{aligned}$$

through an untappable channel to the examiners where $\overline{PK}_{E_j}$ represents the pseudo-public key of examiner j. Additionally, the exam authority generates d proofs as:

$$\begin{aligned} (x_1,\omega_1) \in S_1 \Leftrightarrow & \Big(SK_{EA} = f(PK_{EA}) \wedge F\big(Enc_1, Enc'_1\big) = true\Big) \\ & \vee \Big(\overline{SK}_{E_1} = f(\overline{PK}_{E_1})\Big) \\ & \vdots \\ (x_j,\omega_j) \in S_j \Leftrightarrow & \Big(SK_{EA} = f(PK_{EA}) \wedge F\big(Enc_j, Enc'_j\big) = true\Big) \\ & \vee \Big(\overline{SK}_{E_j} = f(\overline{PK}_{E_j})\Big) \\ & \vdots \\ (x_d,\omega_d) \in S_d \Leftrightarrow & \Big(SK_{EA} = f(PK_{EA}) \wedge F\big(Enc_d, Enc'_d\big) = true\Big) \\ & \vee \Big(\overline{SK}_{E_d} = f(\overline{PK}_{E_d})\Big) \end{aligned}$$

where $x_j = (PK_{EA}, Enc_j, Enc'_j)$, $Enc_j = Enc(P_j)_{PK_{EA}}$, $Enc'_j = Enc(P_j)_{\overline{PK}_{Ej}}$, and $\omega_j = (SK_{EA}, P_j) \vee (SK_{Ej})$. The exam authority sends $\pi = \{\pi_1,\dots,\pi_d\}$ to each examiner where π_i, for $i \in \{1,\dots,d\}$, is the ZK proof for the statement S_i. In this following statement, $F(Enc(m_1)_{\overline{PK}_1}, Enc(m_2)_{\overline{PK}_2})$ is a deterministic polynomial-time function that returns true if $m_1 = m_2$. Upon receiving $\{M_2,\pi_i\}$, E_i first extracts his corresponding P_i from M_2 message. Then he verifies $\{\pi_1,\dots,\pi_d\}$ and checks if $P_i \subseteq \{1,\dots,n\}$. As the above proofs imply, at the end of the *Assignment* protocol, each examiner not only learns the assigned subset to him but also can verify that message M_2 is encrypting the same subsets, previously committed by the exam authority through message M_1.

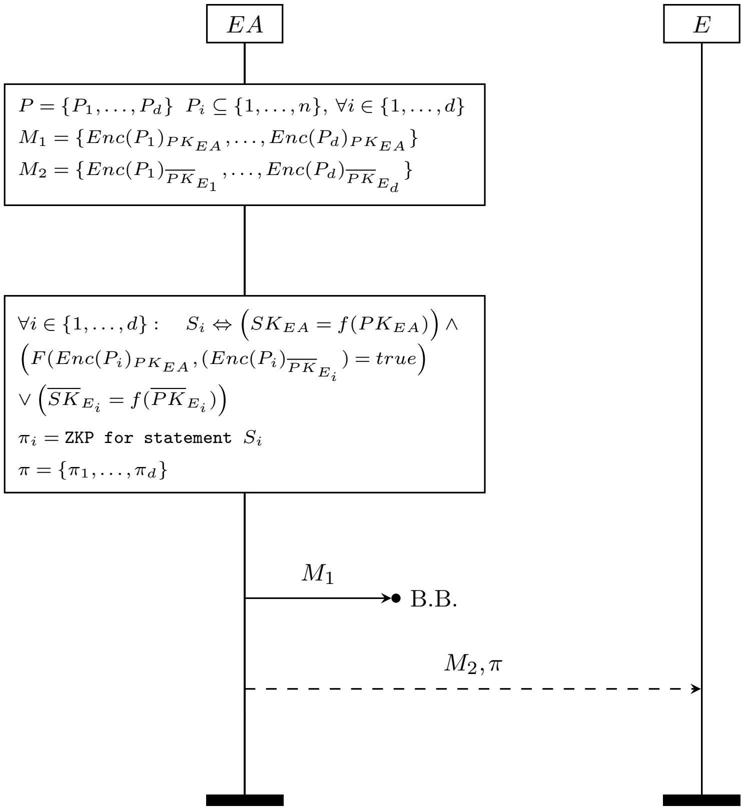
EA
E
$P = \{P_1, \dots, P_d\}$ $P_i \subseteq \{1, \dots, n\}, \forall i \in \{1, \dots, d\}$
$M_1 = \{Enc(P_1)_{PK_{EA}}, \dots, Enc(P_d)_{PK_{EA}}\}$
$M_2 = \{Enc(P_1)_{\overline{PK}_{E_1}}, \dots, Enc(P_d)_{\overline{PK}_{E_d}}\}$
$\forall i \in \{1, \dots, d\}: \quad S_i \Leftrightarrow \Big(SK_{EA} = f(PK_{EA})\Big) \wedge$
$\Big(F(Enc(P_i)_{PK_{EA}}, (Enc(P_i)_{\overline{PK}_{E_i}}) = true\Big)$
$\vee \Big(\overline{SK}_{E_i} = f(\overline{PK}_{E_i})\Big)$
$\pi_i =$ ZKP for statement S_i
$\pi = \{\pi_1, \dots, \pi_d\}$
M_1
B.B.
M_2, π

Figure 2: The message sequence chart of Assignment Protocol
Public transfer: $\longrightarrow$ Untappable channel: $\dashrightarrow$

4.3 Testing

This phase pertains to the process of assigning questions to candidates and gathering their corresponding answers. In the SCREX protocol, we assume that exam authority is responsible for creating the exam questions. For each candidate C, the exam authority signs and encrypts the test with the candidate's pseudonym and then posts it on the B.B. Candidate C decrypts the test, verifies the signature, and proceeds to answer the questions. Then, she sends $[Sign_{SK_C, h_C}(quest, ans, \overline{PK}_C)]_{PK_A}$ to the exam authority over an untappable channel where SK_A, PK_A, $quest$, ans and $\overline{PK}_C$ represent the private key of the exam authority, the public key of the exam authority, the questions, the answers, and the candidate pseudonym, respectively. Additionally, the exam authority generates a receipt for the candidate as $[Sign_{SK_A}H(quest, ans, \overline{PK}_C, \alpha)]_{PK_A}$ where H and α denote a secure hash function and a random value generated by the exam authority, respectively, and publishes it on the BB.

It is worth mentioning that in the *Testing* phase, we implement two countermeasures against candidate coercion threats. First, the use of an untappable channel ensures that a coercer cannot determine whether communication occurs between C and EA. Second, the receipt generated by exam authority is masked by a secure α value that will be revealed at the end of the exam. Consequently, the participation of a specific candidate cannot be detected by a coercer until the notification phase. The security analysis in Section 5 confirms this claim.

4.4 Marking

In this phase, the tests that have been collected in the previous phase, are assigned to the examiners by the exam authority. Upon gathering the tests from the candidates, the EA constructs a matrix named T comprising all candidates' $(question, answer)$ pairs. Afterward, it selects a secure permutation matrix named Π and applies it to T to generate a new matrix T^{π}, denoted as $\Pi(T) = T^{\pi}$. The following mathematical transformation illustrates the shuffling procedure.

$$T = \begin{bmatrix} (q,a)_{1,1} & \cdots & (q,a)_{1,n} \\ \vdots & & \vdots \\ (q,a)_{k,1} & \cdots & (q,a)_{k,n} \end{bmatrix} \xrightarrow{\Pi} \begin{bmatrix} (q,a)_{(1,\pi_1^1)} & \cdots & (q,a)_{(1,\pi_n^1)} \\ \vdots & & \vdots \\ (q,a)_{(k,\pi_1^k)} & \cdots & (q,a)_{(k,\pi_n^k)} \end{bmatrix} = T^{\pi}$$

Here, assuming k questions and n candidates, $(q,a)_{(x,y)}$ represents the answer of candidate y to question x. Let T and T^{π} be sets of n vectors $T = [V_1, \ldots, V_n]$ and $T^{\pi} = [V_1^{\pi}, \ldots, V_n^{\pi}]$, respectively. V_y, $1 \leq y \leq n$, denotes the test of candidate y, while V_y^{π}, $1 \leq y \leq n$, signifies a new test where each question belongs to a random author. The exam authority signs each element of T^{π} with its private key and publishes the signed matrix on the BB. Now E_j marks $(q,a)_{(x,y)}$, $\forall y \in P_j$ and sends $Sign_{SK_E, h_E}(q, a, m, H(q, P_E))_{PK_A}$ to the exam authority over an untappable channel where m is the mark for the (q,a) test and H is a secure hash function.

The shuffling mechanism applied in this phase serves two main purposes. First, by breaking down a complete test and distributing (q,a) pairs to multiple examiners, a candidate's test is marked by several examiners rather than just one. This approach mitigates the threat of examiner-candidate collusion. Second, compared to a non-shuffled solution, a test graded by multiple examiners enhances the candidate's perception of *fairness* in marking.

4.5 Notification

The main objective of this phase is to notify the candidate of her mark. Let T_C be the C^{th} column of T, representing the test of candidate C. Furthermore,

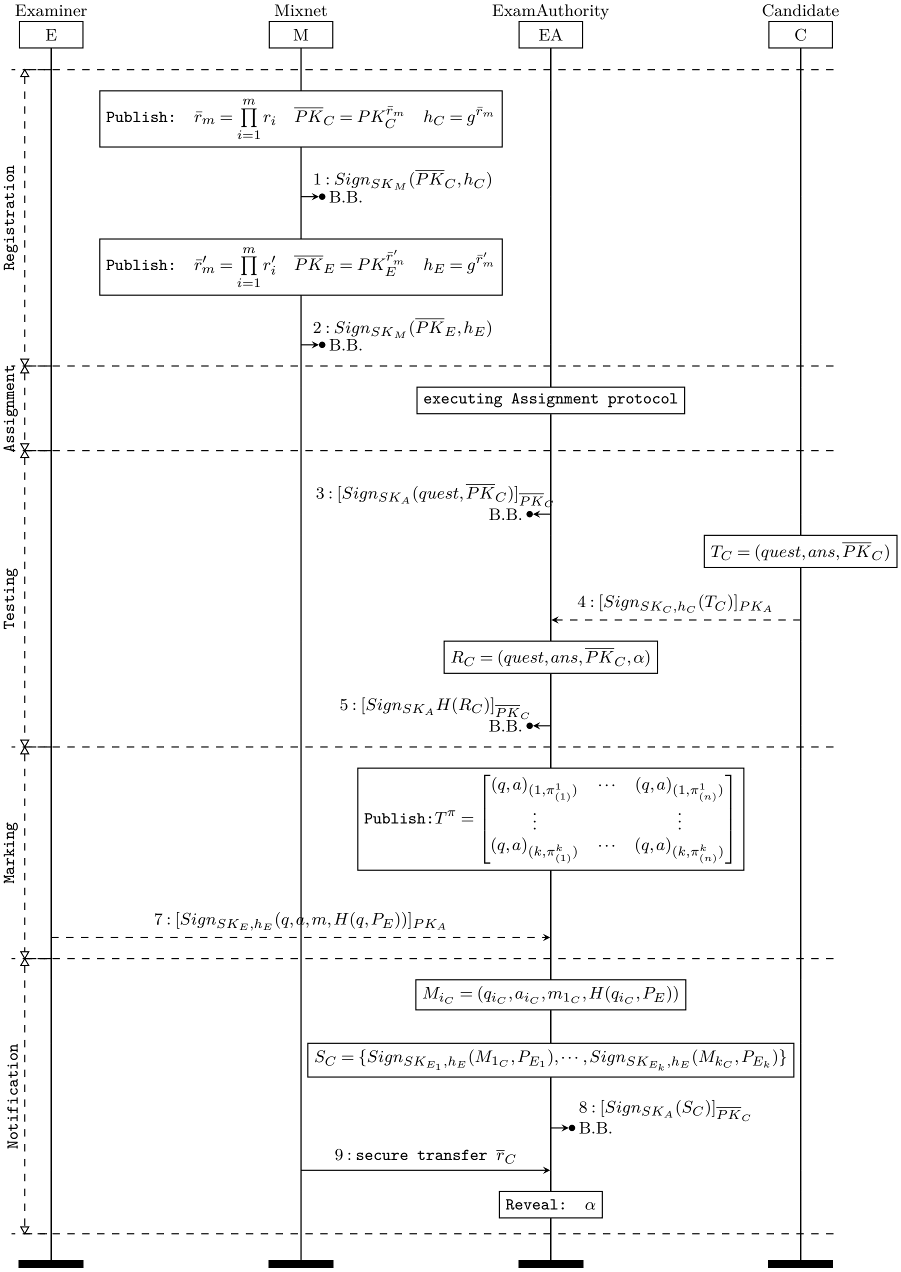
Examiner
E
Mixnet
M
ExamAuthority
EA
Candidate
C
Registration
Publish: $\bar{r}_m = \prod_{i=1}^{m} r_i$ $\overline{PK}_C = PK_C^{\bar{r}_m}$ $h_C = g^{\bar{r}_m}$
$1 : Sign_{SK_M}(\overline{PK}_C, h_C)$
B.B.
Publish: $\bar{r}'_m = \prod_{i=1}^{m} r'_i$ $\overline{PK}_E = PK_E^{\bar{r}'_m}$ $h_E = g^{\bar{r}'_m}$
$2 : Sign_{SK_M}(\overline{PK}_E, h_E)$
B.B.
Assignment
executing Assignment protocol
Testing
$3 : [Sign_{SK_A}(quest, \overline{PK}_C)]_{\overline{PK}_C}$
B.B.
$T_C = (quest, ans, \overline{PK}_C)$
$4 : [Sign_{SK_C, h_C}(T_C)]_{PK_A}$
$R_C = (quest, ans, \overline{PK}_C, \alpha)$
$5 : [Sign_{SK_A} H(R_C)]_{\overline{PK}_C}$
B.B.
Marking
Publish: $T^{\pi} = \begin{bmatrix} (q,a)_{(1,\pi^1_{(1)})} & \cdots & (q,a)_{(1,\pi^1_{(n)})} \\ \vdots & & \vdots \\ (q,a)_{(k,\pi^k_{(1)})} & \cdots & (q,a)_{(k,\pi^k_{(n)})} \end{bmatrix}$
$7 : [Sign_{SK_E, h_E}(q, a, m, H(q, P_E))]_{PK_A}$
Notification
$M_{i_C} = (q_{i_C}, a_{i_C}, m_{1_C}, H(q_{i_C}, P_E))$
$S_C = \{Sign_{SK_{E_1}, h_E}(M_{1_C}, P_{E_1}), \cdots, Sign_{SK_{E_k}, h_E}(M_{k_C}, P_{E_k})\}$
$8 : [Sign_{SK_A}(S_C)]_{\overline{PK}_C}$
B.B.
$9 :$ secure transfer $\bar{r}_C$
Reveal: α

Figure 3: The message sequence chart of SCREX protocol
Public channel: $\longrightarrow$ Untappable channel: $\dashrightarrow$

let T_{iC} denote the i^{th} element of vector T_C. When the exam authority receives the message $S_{q,a} = Sign_{SK_E,h_E}(q, a, m, H(q, P_E))_{PK_A}$ from examiner E, it first decrypts the message with its private key and verifies whether (q, a) exists in T. Then, it regenerates $H(q, P_E)$ to ensure it matches the hash received from E. After collecting all messages from the examiners, the exam authority constructs $S_C = \{S_{C_1}, \cdots, S_{C_k}\}$ where $S_{C_i} = S_{q,a}$ s.t. $(q, a) = T_{iC}$. Finally, it signs S_C, encrypts it with $\overline{PK}_C$ and publishes the result as the notification message for candidate C on the BB. The term S_C contains the mark of candidate C signed by corresponding examiners, which allows C to verify that the mark she received is graded by legitimate examiners and is not modified.

In general, the $P_1, \ldots, P_d$ subsets are not necessarily disjoint, meaning that two or more examiners might be assigned the same (q, a) pair and consequently provide different marks. In such cases, the exam authority selects one of the marks and notifies the candidate through the S_C term. As we will prove in Section 5.4, the *Marking Correctness* verifiability test ensures that the mark assigned to the candidate's test is correct, regardless of which examiner generated it. Therefore, the notification of marks remains robust even if multiple examiners grade the same (q, a).

5 Formal Security Analysis

In applied π-calculus and ProVerif, term manipulation by any participant, including adversaries, is handled through *destructor* functions, which are part of what's called equational theory. The specific equational theory we used is detailed in Table 2. The first four equations, covering *Exponentiation*, *ElGamal Decryption*, *Digital Signatures* and *Non-Interactive Zero-Knowledge Proofs (NIZKP)* were initially proposed by Dreier *et al.* [11, 12]. The final equation in the table, which we are presenting for the first time, describes the designated verifier ZKP used in the *Assignment* protocol. We will now go through each part of this equational theory in more detail.

- Exponentiation: The symbol *exp* here represents the exponentiation operation used by the participants in the protocol. The first part of the equation ensures that the protocol behaves as expected, specifically that $(g^x)^y = (g^y)^x$. The second part addresses an attack highlighted in [23], which targets exponentiation mixnets used in the protocol to conceal the link between public keys and pseudonyms. It is important to note that g in these equations is a constant; otherwise, ProVerif's pre-treatment of the equations will not terminate.

- ElGamal Decryption: The function symbol *enc* takes as argument a plaintext m, a basis h, a public key $exp(h, sk)$ and a random coin r. To decrypt the encrypted message m, one needs to specify the ciphertext, the basis used for the encryption h and the corresponding secret key sk.

- Digital Signature: The function symbol *sign* is used to model a digital signature. It takes as argument a message m to sign, a basis h and a secret key sk. Thus, to verify the signature, we apply $checksign$ with the corresponding public key $exp(h, sk)$.

- NIZKP: Since the participants must provide zero-knowledge proofs to the mixnet, we added an equation to model the non-interactive zero-knowledge proof of the discrete log.
- Designated Verifier ZKP: In this equational theory, the $checkdzkp$ function returns $true$ under two conditions: (i) if a valid proof is correctly generated by the exam authority, or (ii) if any fake proof is generated by an examiner. The function $priv\_zk$ is the proof generator, which takes as inputs an exponentiation base g, secret key sk, subset p and randomness r. The first equation returns $true$ if the two encryption functions have the same arguments and the exam authority generates the proof. The second condition is satisfied if the proof is generated by an examiner holding the private key skE'. It is important to note that the randomness used in the $priv\_zk$ proof generator is crucial, as it prevents a coercer who forces an examiner to reveal all secret keys and proofs from reproducing the proof. Otherwise, the coercer could deduce which subset was sent to the examiner.

Primitive	**Equation**
Exponentiation	$exp(exp(g, x), y) = exp(exp(g, y), x)$ $exp(exp(exp(g, x), y), z) = exp(exp(exp(g, x), z), y)$
ElGamal Decryption	$dec(enc(m, h, exp(h, sk), r), h, sk) = m$
Digital Signature	$checksign(sign(m, h, sk), h, exp(h, sk)) = m$ $getmessage(sign(m, h, sk)) = m$
NIZKP	$zkpcheck(zkp(A, g, x), g, exp(g, x)) = true$
Designated Verifier ZKP	$checkdzkp(exp(g, skEA), exp(exp(g, rce), skE),$ $enc(x, g, exp(g, skEA), r1),$ $enc(x, exp(g, rce), exp(exp(g, rce), skE), r2),$ $priv\_zk(g, skEA, x, r3)) = true$ **otherwise** $checkdzkp(exp(g, skE'), exp(exp(g, rce), skE), c, d,$ $priv\_zk(g, skE', p, r3)) = true.$

Table 2: Equational Theory.

The results of our formal analysis using ProVerif are summarized in Table 5. The timings for the latter were done on macOS with an M1 processor and 16 GB of RAM. In this table, the *Corrupted Parties* column indicates the maximum number of parties that can be compromised while still ensuring the property holds. In other words, the property remains satisfied even if all the specified roles are compromised, but it will break if one additional role is corrupted beyond those listed. Based on the security properties defined earlier in Section 3, we will now discuss the results of our analysis in more detail.

5.1 Authentication and Secrecy in SCREX

Given the model we introduced for authentication properties in Section 3.2, ProVerif demonstrates that all the correspondence assertions hold. Additionally, using reachability queries in ProVerif, we investigate two secrecy properties: *Mark Secrecy* and *Subset Secrecy*. The former ensures that marks generated by examiners remain secret from the attacker, while the latter guarantees the secrecy of the subsets assigned to examiners. ProVerif successfully shows that both these secret values are unreachable to the attacker. As shown in Table 5, for authentication and secrecy properties, even if the mixnet is corrupted, the secrecy and authentication requirements in SCREX are still satisfied. This result supports the fact that the role of the mixnet is merely to pseudonymize identities to achieve privacy and coercion properties, and its trustworthiness is not necessary for maintaining authentication and secrecy properties.

5.2 Privacy in SCREX

Table 5 demonstrates that SCREX satisfies all defined privacy requirements. In the SCREX protocol, since the exam authority generates the questions, it must remain honest to ensure *Question Indistinguishability* is satisfied. For *Anonymous Marking*, at least two honest candidates are necessary; otherwise, an attacker could collude with all corrupted candidates to easily compromise the mark anonymity of the sole honest candidate. Similarly, *Anonymous Examiner* requires at least two honest examiners to uphold the anonymity of the examiners. For *Mark Privacy*, which concerns the privacy of a mark given to a candidate, anyone except the concerned candidate, the examiner and the exam authority can be dishonest. As the result shows, the mixnet should be trusted for maintaining *Anonymous Examiner* and *Anonymous Marking*.

5.3 Coercion in SCREX

In SCREX, the coercer is assumed to have access to all of the victim's secret information, including the randomness used in probabilistic encryption, whereas, in the CREX protocol, it was only assumed that the coercer would obtain the private key of the coerced party. However, the assumption regarding the temporariness of *Examiner Coercion* in CREX remains unchanged in SCREX, meaning that coercion is only considered relevant until the end of the marking phase.

If we apply the new coercion threat model to the CREX protocol, ProVerif demonstrates that both *Anonymous Submission* and *Single-Blindness* are compromised. The primary issue lies in the public communication between the exam authority and examiners, as well as between the exam authority and candidates. In the SCREX protocol, as shown in Figure 3, these two channels are replaced

Requirement	**Soundness**	**Completeness**
Question Validity	✓ (EA)	✓(all)
Test Integrity	✓	✓(all)
Marking Correctness	✓ (EA)	✓(all)
Mark Integrity	✓	✓(all)
Mark Notification Integrity	✓	✓(all)

Table 3: Summary of the analysis of SCREX for I.V. requirements

with *untappable* channels, and ProVerif successfully verifies the aforementioned coercion-resistance properties.

The security analysis indicates that the exam authority must remain honest to achieve *Anonymous Submission* and *Single-Blindness*. If the exam authority is dishonest, it could expose the data transmitted over untappable channels, allowing the coercer to detect non-cooperation by the coerced party. Additionally, the mixnet must also be trusted to ensure anonymity, thereby satisfying *Anonymous Submission* and *Single-Blindness*.

5.4 Verifiability in SCREX

The ProVerif model used for authentication and privacy is also used to analyze SCREX for verifiability.

All relations are built from the posts that appear on the bulletin board. The tuples $(i, (q, a), m)$ are generated from the marked test signed by the examiner, that is, $Sign_{SK_E, h_E}(M_C)$ The tuples (i, m) instead are built from the encryption of the marked test generated by the exam authority, that is, $\{Sign_{SK_E, h_E}(M_C)\}_{\overline{PK_C}}$. The function `Correct`, which is the algorithm used to mark the tests, is modeled as a ProVerif.

Individual verifiability definitions require the existence of verifiability tests that candidates run to check the properties of the protocol. We show that SCREX has the necessary verifiability tests. In ProVerif, the event `OK` signals that the verifiability test succeeds, while the event `KO` that the verifiability test fails.

We check the event `OK` is always preceded by the event emitted in the part of the code where the predicate becomes satisfied to prove soundness. This can be modeled as a reachability property in ProVerif. We assume an honest candidate who plays the role of the verifier. The other principals are usually corrupted, if not stated otherwise. The verifiability test receives the data from the candidate via a private channel, and the remaining data posted on the bulletin board via public channels. This allows an attacker to manipulate the input data. Corrupted principals may collude with the attacker.

We check that the event `KO` cannot be emitted to prove completeness. We consider only honest principals as the test succeeds if its input data is correct.

Table 3 summarises the results of the individual verifiability analysis of SCREX and reports the roles required to be honest.

Requirement	**Soundness**	**Completeness**
Registration	✓	✓(all)
Marking Correctness	✓ (EA)	✓(all)
Test Integrity	✓	✓(all)
Mark Integrity	✓	✓(all)

Table 4: Summary of the analysis of SCREX for U.V. requirements

To prove the soundness of the universal verifiability definitions, we check that the event `KO` cannot be emitted. Every time the verifiability test succeeds, we check that the decryption of the concerned ciphertext gives the expected plaintext. If not, the event `KO` is emitted. This is necessary as the candidate can be corrupted, hence it can be hard to find a ProVerif process that can be annotated with events to check soundness via correspondence assertions.

Table 4 summarises the results of the universal verifiability analysis of SCREX and reports the roles required to be honest. Note that it is not possible to automatically prove the universal verifiability requirements in ProVerif because ProVerif cannot iterate over all candidates (*i.e.,* ProVerif does not support loops). In [10], manual induction proofs that generalize the ProVerif result are provided to fully prove the universal verifiability requirements. The same strategy can be used in our case.

The results of the analysis are depicted in Table 5 and show that all our properties are verified.

6 Conclusion

Exams are often seen as a way to evaluate students' knowledge in schools, but the concept of exams, or assessments in general, covers a much broader range of activities. In fact, assessments can be more significant than typical school or university exams. For example, they might include professional skills evaluations in companies, language proficiency tests like TOEFL or IELTS, or even peer reviews for academic papers. Security is a major concern in all these types of exams, as the results are crucial for measuring a candidate's competence and can have far-reaching consequences, such as influencing hiring decisions or visa approvals.

When exams are conducted electronically, e-exams not only need to meet the functionality of traditional exams but also address new security challenges that arise due to their digital nature. Recently, research revealed that security in e-exams is defined through a set of authentication, integrity, and confidentiality properties that are peculiar and differ in their specification from those in similar domains, like e-voting. They bring new insights in their design of secure systems. One of the latest is about coercion, a threat that has indeed found relevance in the news since a party can be interested in altering the result of an assessment (think of where e-exams are a precondition to applying for a

	Property	Corrupted Parties	Result	Time
Secrecy	Mark Secrecy	N	✓	2 s
	Subset Secrecy	C, N	✓	2 m 52 s
Authentication	Test Origin Authentication	N	✓	4 s
	Answer Authenticity	N	✓	3 s
	Mark Authenticity	N	✓	5 s
	Mark Authentication	N	✓	4 m 1 s
Privacy	Question Indistinguishability	C, E, N	✓	5 s
	Anonymous Marking	C^*, E, EA	✓	7 s
	Anonymous Examiner	C, E^*, EA	✓	14 s
	Mark Privacy	N	✓	19 m 32 s
Coercion	Single Blindness	C	✓	12 m 40 s
	Anonymous Submission	C^*, E	✓	23 m 5 s
Individual Verifiability	Question Validity	C, E, N	✓	11 m 13 s
	Marking Correctness	C, E, N	✓	2 m
	Test Integrity	C, E, EA, N	✓	2 m 16 s
	Mark Integrity	C, E, EA, N	✓	3 m 4 s
	Mark Notification Integrity	C, E, EA, N	✓	1 m 15 s
Universal Verifiability	Registration	C, E, EA	✓	50 s
	Marking Correctness	C, E, N	✓	24 s
	Test Integrity	C, E, EA, N	✓	23 s
	Mark Integrity	C, E, EA, N	✓	25 s

Table 5: Analysis results of our SCREX protocol. The * symbol indicates a scenario where all parties except two are corrupted.

VISA or a promotion). In coercion scenarios, a candidate, examiner, or even an external coercer may force a party to reveal their identity or act in a way that manipulates the outcome of the exam.

The CREX protocol introduced in [23] was the first to address the coercion threat in e-exams, but it had some security flaws and incomplete aspects. In this work, we first analyzed the security of the CREX protocol. To do so, we refined the assignment process, which was not fully specified in the original paper, and considered a stronger coercion model. Through this more precise modeling, we uncovered attacks on privacy properties in CREX.

As a result, we developed SCREX, a new e-exam protocol designed to resist stronger coercion threats. Additionally, verifiability, which ensures fairness in exams, was missing in the original CREX scheme. We designed SCREX to include verifiability as well, using a state-of-the-art framework to prove this property for e-exams. Our protocol was shown to be verifiable, and we created formal models in ProVerif to automatically verify the security properties of SCREX. Our analysis highlights the importance of having a clear, detailed specification for all aspects of a protocol and taking a realistic approach to coercion threats. Moving forward, we intend to investigate stronger threat models in which the exam authority may act dishonestly under coercion. Furthermore, we plan to extend our methodology to a broader class of security protocols that require both privacy and verifiability

Acknowledgment

Rakeei and Lenzini's research is supported by the ANR–FNR international project INTER/AN/20/14926102, ”Secure and Verifiable Electronic Testing and Assessment Systems” (SEVERITAS). Lafourcade and Mahmoud's research is partially supported by the ANR project SEVERITAS (ANR-20-CE39-0009) and the ANR project PRIVA-SIQ (ANR-23-CE39-0008). Dreier's research is funded by the French National Research Agency (ANR) under the France 2030 programme, reference ANR-22-PECY-0006.

References

[1] Martin Abadi, Bruno Blanchet, and Cedric Fournet. “The applied pi calculus: Mobile values, new names, and secure communication”. In: *Journal of the ACM (JACM)* 65.1 (2017), pp. 1–41. DOI: 10.1145/3127586.

[2] Giampaolo Bella, Rosario Giustolisi, and Gabriele Lenzini. “Secure exams despite malicious management”. In: *2014 Twelfth Annual International Conference on Privacy, Security and Trust*. IEEE. 2014, pp. 274–281. DOI: 10.1109/PST.2014.6890949.

[3] Giampaolo Bella, Rosario Giustolisi, Gabriele Lenzini, et al. “A Secure Exam Protocol Without Trusted Parties”. In: *ICT Systems Security and Privacy Protection*. Ed. by Hannes Federrath and Dieter Gollmann. Cham: Springer International Publishing, 2015, pp. 495–509. ISBN: 978-3-319-18467-8. DOI: 10.1007/978-3-319-18467-8_33.

[4] Giampaolo Bella, Rosario Giustolisi, Gabriele Lenzini, et al. “Trustworthy exams without trusted parties”. In: *Computers & Security* 67 (2017), pp. 291–307. DOI: 10.1016/j.cose.2016.12.005.

[5] Bruno Blanchet. “An Efficient Cryptographic Protocol Verifier Based on Prolog Rules”. In: *14th IEEE Computer Security Foundations Workshop (CSFW-14 2001), 11-13 June 2001, Cape Breton, Nova Scotia, Canada*. IEEE Computer Society, 2001, pp. 82–96. DOI: 10.1109/CSFW.2001.930138.

[6] Bruno Blanchet. “The security protocol verifier ProVerif and its horn clause resolution algorithm”. In: *arXiv preprint arXiv:2211.12227* (2022). DOI: 10.48550/arXiv.2211.12227.

[7] Alessandro Bruni, Rosario Giustolisi, and Carsten Schuermann. “Automated analysis of accountability”. In: *Information Security: 20th International Conference, ISC 2017, Ho Chi Minh City, Vietnam, November 22-24, 2017, Proceedings 20*. Springer. 2017, pp. 417–434. DOI: 10.1007/978-3-319-69659-1_23.

[8] Jordi Castella-Roca, Jordi Herrera-Joancomarti, and Aleix Dorca-Josa. “A secure e-exam management system”. In: *First International Conference on Availability, Reliability and Security (ARES’06)*. IEEE. 2006, 8–pp. DOI: 10.1109/ARES.2006.14.

[9] D. Dolev and A. Yao. “On the security of public key protocols”. In: *IEEE Transactions on Information Theory* 29.2 (1983), pp. 198–208. DOI: 10.1109/TIT.1983.1056650.

[10] Jannik Dreier, Rosario Giustolisi, Ali Kassem, et al. “A framework for analyzing verifiability in traditional and electronic exams”. In: *International Conference on Information Security Practice and Experience*. Springer. 2015, pp. 514–529. DOI: 10.1007/978-3-319-17533-1_35.

[11] Jannik Dreier, Rosario Giustolisi, Ali Kassem, et al. “Formal analysis of electronic exams”. In: *2014 11th International Conference on Security and Cryptography (SECRYPT)*. IEEE. 2014, pp. 1–12. DOI: 10.5220/0005050901010112.

[12] Jannik Dreier, Pascal Lafourcade, and Dhekra Mahmoud. “Shaken, not Stirred - Automated Discovery of Subtle Attacks on Protocols using Mix-Nets”. In: *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, 2024, pp. 3135–3150. ISBN: 978-1-939133-44-1.

[13] Simon N Foley and Jeremy L Jacob. “Specifying security for computer supported collaborative working”. In: *Journal of Computer Security* 3.4 (1995), pp. 233–253.

[14] SM Furnell, PD Onions, Martin Knahl, et al. “A security framework for online distance learning and training”. In: *Internet Research* 8.3 (1998), pp. 236–242. DOI: 10.1108/10662249810217821.

[15] Rosario Giustolisi. *Modelling and Verification of Secure Exams*. Information Security and Cryptography. Springer, 2018. ISBN: 978-3-319-67106-2. DOI: 10.1007/978-3-319-67107-9.

[16] Rosario Giustolisi, Gabriele Lenzini, and Giampaolo Bella. “What security for electronic exams?” In: *2013 International Conference on Risks and Security of Internet and Systems (CRiSIS)*. IEEE. 2013, pp. 1–5. DOI: 10.1109/CRiSIS.2013.6766348.

[17] Rosario Giustolisi, Gabriele Lenzini, and Peter YA Ryan. “Remark!: A secure protocol for remote exams”. In: *Security Protocols XXII: 22nd International Workshop, Cambridge, UK, March 19-21, 2014, Revised Selected Papers 22*. Springer. 2014, pp. 38–48. DOI: 10.1007/978-3-319-12400-1_5.

[18] Philippe Golle and Markus Jakobsson. “Reusable anonymous return channels”. In: *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society*. 2003, pp. 94–100. DOI: 10.1145/1005140.1005155.

[19] Rolf Haenni and Oliver Spycher. “Secure internet voting on limited devices with anonymized {DSA} public keys”. In: *2011 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 11)*. 2011.

[20] Andrea Huszti and Attila Petho. “A secure electronic exam system”. In: *Publicationes Mathematicae Debrecen* 77.3-4 (2010), pp. 299–312. DOI: 10.5486/PMD.2010.4682.

[21] Khalil El-Khatib, Larry Korba, Yuefei Xu, et al. “Privacy and security in e-learning”. In: *International Journal of Distance Education Technologies (IJDET)* 1.4 (2003), pp. 1–19. DOI: 10.4018/jdet.2003100101.

[22] Ania M Piotrowska, Jamie Hayes, Tariq Elahi, et al. “The loopix anonymity system”. In: *USENIX Security Symposium*. 2017, pp. 1199–1216. DOI: 10.48550/arXiv.1703.00536.

[23] Mohammadamin Rakeei, Rosario Giustolisi, and Gabriele Lenzini. “Secure internet exams despite coercion”. In: *International Workshop on Data Privacy Management*. Springer. 2022, pp. 85–100. DOI: 10.1007/978-3-031-25734-6_6.

[24] Adam Vecsi and Attila Petho. “Auditable Anonymous Electronic Examination”. In: *Cryptography* 8.2 (2024), p. 19. DOI: 10.3390/cryptography8020019.

[25] Adam Vecsi and Attila Petho. “Scalix mix network”. In: *Acta Cybernetica, to Appear* (2022). DOI: 10.14232/actacyb.307671.

[26] Edgar R Weippl. *Security in e-learning*. Vol. 16. Springer Science & Business Media, 2005. DOI: 10.1145/1070939.1070943.