Formal Analysis of Electronic Exams

Jannik Dreier, Rosario Giustolisi, Ali Kassem, Pascal Lafourcade, Gabriele Lenzini, Peter Y. A. Ryan

Formal Analysis of Electronic Exams*

Jannik Dreier[1], Rosario Giustolisi[2], Ali Kassem[3]
Pascal Lafourcade[4], Gabriele Lenzini[2] and Peter Y. A. Ryan[2]
[1]*Institute of Information Security, ETH Zurich, Switzerland*
[2]*SnT/University of Luxembourg, Luxembourg*
[3] *Université Grenoble Alpes, CNRS, VERIMAG, Grenoble, France*
[4]*University d'Auvergne, LIMOS, France*

Keywords: Electronic Exams, Formal Verification, Authentication, Privacy, Applied π-Calculus, ProVerif

Abstract: Universities and other educational organizations are adopting computer and Internet-based assessment tools (herein called *e-exams*) to reach widespread audiences. While this makes examination tests more accessible, it exposes them to new threats. At present, there are very few strategies to check such systems for security, also there is a lack of formal security definitions in this domain. This paper fills this gap: in the formal framework of the applied π-calculus, we define several fundamental authentication and privacy properties and establish the first theoretical framework for the security analysis of e-exam protocols. As proof of concept we analyze two of such protocols with ProVerif. The first "secure electronic exam system" proposed in the literature turns out to have several severe problems. The second protocol, called *Remark!*, is proved to satisfy all the security properties assuming access control on the bulletin board. We propose a simple protocol modification that removes the need of such assumption though guaranteeing all the security properties.

1 INTRODUCTION

Electronic exams (in short, *e-exams*) are computer-based systems employed to assess the skills, the capabilities or the knowledge of students and professionals. Their importance has raised considerably since several educational and testing institutions began to offer e-exams as a service open to a worldwide-spread audience. For instance, universities like MIT, Stanford and Berkeley have set them up in Massive Open Online Course (MOOC). Students can be marked online, although not yet granted with a legally valid diploma. Other institutions employ e-exams to grant students with certificates which have an officially recognized validity. This happens, for instance, with testing organizations like ETS[2], the world leader in the business of assessing content-knowledge and abilities in various subjects. Other examples are CISCO and Microsoft's career certification programs, and ECDL, the pioneer in assessing people's computer office skills and in releasing "European Computer Driving Licence".

For all these and similar institutes, e-exam systems are a promise for a better and a cheaper organization and management of tests: e-exams are flexible in where and when exams can be set (Hjeltnes and Hansson, 2005), their test sessions can be easily open to a very large public of candidates and, if the implementation allows automatic marking, their results are immediately available.

So far, the main concern about e-exam security has been about student cheating and impersonation (Weippl, 2005). Since such threats mainly come from students, most institutions have arranged invigilated testing. The invigilator can be a person, like a lecturer attending at the test location, or a software, like ProctorU[3] running on the candidate's computer. A proctor is meant to supervise the test and to detect and report, perhaps also discourage, any attempt of fraud. However, e-exams are threatened by more serious problems than student cheating. As evidenced by recent scandals, dishonest acts can come also from other parties than candidates, such as bribed examiners or misbehaving exam authorities. The consequences are usually worse in these cases than in those due to student cheating. In the Atlanta scandal, school authorities colluded in changing student marks to im-

*This research was conducted with the support of the Digital trust Chair from the University of Auvergne Foundation.

[2]https://www.ets.org/

[3]http://www.proctoru.com/

prove their institution's rankings and get more public funds (Copeland, 2013). In a BBC investigation, ETS was shown vulnerable to a fraud perpetrated by official invigilators: in collusion with all the candidates who were there to get their visas, the invigilators dictated the correct answers during the test (Watson, 2014).

We keep a neutral position in the debate about whether e-exams are actually a beneficial and promising choice in promoting and supporting education, but it is a fact that the adoption of computer-based assessment is increasing. It is also a fact that such a growth has not been followed nor was it preceded by a rigorous understanding and analysis of security. There is then a need for a formal framework to define and analyse the security of e-exam protocols.

Contributions: We present the first formalization for e-exams, and define several fundamental security properties for e-exams. We categorize them in two main classes: (a) authentication properties, including *Answer Origin Authentication*, *Form Authorship*, *Form Authenticity*, and *Mark Authenticity*, and (b) privacy properties, containing *Question Indistinguishability*, *Anonymous Marking*, *Anonymous Examiner*, *Mark Privacy*, and *Mark Anonymity*. We develop our formal framework in the applied π-calculus (Abadi and Fournet, 2001), wherein we propose a model for the typical e-exam's processes and phases and define all our properties.

We validate our approach by analyzing two e-exam protocols. The first is an internet exam protocol proposed by Huszti *et al.* (Huszti and Pethő, 2010). The second is a recent protocol proposed by Giustolisi *et al.* (Giustolisi et al., 2014). We model both protocols in the applied π-calculus, and check them against our properties using ProVerif (Blanchet, 2001; Blanchet et al., 2008). Our security analysis reveals several weaknesses in the first protocol, even without considering dishonest parties. We show that the second protocol is secure if all parties are honest, and we also consider the situation where some parties are dishonest, *i.e.,* collaborate with the attacker. In this case we discover a weakness on *Form Authenticity*, and we propose a simple fix to overcome this weakness.

Beyond these results, this work is the first theoretical framework for the security analysis of e-exams protocols. It can be generalized in a straightforward way to study traditional exam protocols.

Related Work: Only a few papers propose e-exam protocols that guarantee some security, mainly under the assumption that some authority is trusted (Huszti and Pethő, 2010; Castellà-Roca et al., 2006; Herrera-Joancomartí et al., 2004; Bella et al., 2011). Few other works (Giustolisi et al., 2013; Furnell et al., 1998; Weippl, 2005) list some relevant properties for e-exams, yet only informally.

Arapinis et al. (Arapinis et al., 2013) propose a cloud-based protocol for conference management system that supports applications, evaluations, and decisions. They identify and analyze a few privacy properties (secrecy and unlinkability) that should hold despite a *malicious-but-cautious* cloud, and they prove, using ProVerif, that their protocol satisfy them.

To the best of our knowledge, no formal definitions have been given for the security properties of e-exam systems. There are instead papers presenting the formalization and verification of properties in domains that seem related to e-exams, namely e-voting (Dreier et al., 2011; Dreier et al., 2012b; Dreier et al., 2012a; Backes et al., 2008a; Delaune et al., 2009; Delaune et al., 2006a) and e-auction systems (Dong et al., 2010; Dreier et al., 2013b; Dreier et al., 2013a).

Some of the security properties therein studied remind those we are presenting for e-exams. For instance, *Answer Origin Authentication* is analogous to voter and bidder authentication. *Mark Privacy* reminds ballot privacy and losing bids privacy. Yet, there are fundamental differences. In e-exams, *Answer Authorship* should be preserved even in the presence of colluding candidates. Conversely, vote (bid) authorship is not a problem for e-voting (e-auction), in fact unlinkability between a voter (bidder) and her vote (bid) is a desired property. An other important property for e-exams is to keep exam questions secret until the exam ends. We do not find such a property in e-voting where the candidates are previously known to the voters, and in e-auction where the goods to bid for are previously known to the bidders. Moreover, properties such as *Anonymous Marking*, meaning that the examiners do not know whose copy they are grading, evaluates to a sort of fixed-term anonymity. This property is meant to hold during the marking, but is trivially falsified when the marks are assigned to the candidates.

Outline: In Section 2, we model e-exam protocols in the applied π-calculus. Then, we specify security properties in Section 3. We validate our framework by analysing the security of two e-exam protocols (Huszti and Pethő, 2010) and (Giustolisi et al., 2014) in Section 4 and 5. Finally, in Section 6, we discuss our results and outline the future work.

2 MODELLING

We model e-exam protocols in the applied π-calculus, a process calculus designed for the verification of cryptographic protocols. To perform the automatic protocol verification, we use ProVerif. This tool uses a process description based on the applied π-calculus, but has syntactical extensions and is enriched by events to check reachability and correspondence properties. Besides, it can check equivalence properties. We use events to define various authentication properties, and we model privacy properties as equivalence properties. Precisely, honest parties are modeled as processes in the applied π-calculus. These processes can exchange messages on public or private channels, create keys or fresh random values and perform tests and cryptographic operations, which are modeled as functions on terms with respect to an equational theory describing their properties.

The attacker has complete control of the network, except the private channels: he can eavesdrop, remove, substitute, duplicate and delay messages that the parties are sending one another, and insert messages of his choice on the public channels (like the Dolev-Yao attacker (Dolev and Yao, 1983)). To capture threats due to collusions and coercions, we assume dishonest parties. They cooperate with the attacker, revealing their secret data (*e.g.,* secret keys) to him, or taking orders from him (*e.g.,* how to answer a question). We model such dishonest parties as in Definition 8 from (Delaune et al., 2006b): if the process P is an honest party, then the process P^{c_1,c_2} is its dishonest version. This is a variant of P which shares with the attacker channels c_1 and c_2. Through c_1, P^{c_1,c_2} sends all its inputs and freshly generated names (but not other channel names). From c_2, P^{c_1,c_2} receives messages that can influence its behaviour. For more details about the applied π-calculus, its standard results and all the definitions used in this paper, we remind to the papers (Abadi and Fournet, 2001; Delaune et al., 2006b). An e-exam system involves different parties, among which are the *candidates* who sit for the exam; the *examiners* who mark the answers submitted by the candidates; the *question committee*, which prepare the exam questions; the *exam authorities*, which conducts the exam, that is registrars, invigilators, exam collectors, and a notification committee. In some protocols, an authority can be responsible of two or more roles.

Definition 1. (E-exam protocol). *An* e-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 commitee,* A_i*'s are the processes executed by the authorities, and* $\tilde{n}_p$ *is the set of private channel names.*

Note that all candidates and all examiners execute the same process, but with different variable values, *e.g.,* keys, identities, and answers.

Definition 2. (E-exam instance). *Given an e-exam protocol an* e-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 set of the protocol's 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 possible 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. Without loss of generality, we assume that* A_1 *is in charge of distributing the copies to the examiners.*

Definition 2 does not specify whether the examiners are machines or humans. For the purpose of our model this distinction is not necessary; it is sufficient that an examiner attributes a mark to a given answer. Note that Q and A_1 could coincide if for instance there is only one authority A, in that case we can write simply $A\sigma_q\sigma_{dist}$ instead of $Q\sigma_q|A_1\sigma_{dist}$. We organize an e-exam's steps in four phases. (1) *Registration:* the exam authority (the registrar) creates a new examination and checks the eligibility of candidates who attempts to register for it. (2) *Examination:* the exam authority authenticates the candidates, and sends to each of them an *exam form* that contains the exam questions. Each candidate fills the form with his answer, and submits it to the exam collector. (3) *Marking:* the authority distributes the forms submitted by the candidates to the examiners, who in their turn evaluate and mark them; (4) *Notification:* once the forms have been evaluated, the marks are notified to the candidates.

3 SECURITY PROPERTIES

We propose a formalization for authentication and privacy properties. They best represent exam security requirements as corroborated by other works (Furnell et al., 1998; Giustolisi et al., 2013; Weippl, 2005).

We introduce four authentication properties meant to ensure the associations between the candidate's identity, the answer, and the mark being preserved through all phases. When authentication holds there is no loss, no injection, and in general no manipulation of the exam forms from examination to notification. We also introduce five privacy properties that ensure the anonymity of critical parties in order to prevent bribing, favouritisms, and to guarantee fairness among candidates. In the context of e-exams, there are other classes of properties that might be of interest, such as verifiability, reliability, or accountability, but we do not study them here. We leave this task as future work.

3.1 Authentication properties

We model our authentication properties as correspondence properties, a well-known approach (Ryan et al., 2000; Ryan and Smyth, 2011). Specific events, whose parameters refer to the pieces of information in the exam form, flag important steps in the execution of the exam. Events are annotations that do not change a process behavior, but are inserted at precise locations to allow reasoning about the exam's execution. In the following id_c is the candidate identity, $ques$ the question(s), ans the answer(s), $mark$ the mark(s), id_form is an identifier of the exam form used during marking, and id_e is the examiner's identity.

- $reg(id\_c)$: is the event inserted into the registrar process at the location where candidate id_c has successfully registered for the exam.
- $submitted(id\_c, ques, ans)$: is the event inserted into the process of candidate id_c in the examination phase, at the location where he sends his answer ans corresponding to the question $ques$.
- $collected(id\_c, ques, ans)$: is the event inserted into the exam collector's process in the examination phase, just after it received and accepted the exam form (id_c, $ques$, ans) from candidate id_c.
- $distrib(id\_c, ques, ans, id\_form, id\_e)$: is the event inserted into the authority process in the marking phase, when it assigns the exam form (id_c, $ques$, ans) from candidate id_c to the examiner id_e using the identifier id_form.
- $marked(ques, ans, mark, id\_form, id\_e)$: is the event inserted into the examiner id_e's process in the marking phase, at the location where he marked the question/answer pair ($ques$, ans) identified by id_form with the mark $mark$.
- $notified(id\_c, mark)$: is the event inserted into the process of candidate id_c in the notification phase, just after he received and accepted the mark $mark$ from the responsible authority.

Note that the id_form is only used to identify an exam form during marking. This could be a pseudonym to allow anonymous marking, or simply the candidate identity if the marking is not anonymous. In exam with only one examiner the event $distrib$ might appear to be unnecessary. Yet, it is needed. It links exam forms to id_form and, for instance, helps revealing when identical answers are graded multiple times (and not necessarily with the same mark).

Our events allow us to express authentication properties as correspondence properties, all having the following structure: "on every trace the event e_1 is preceded by the event e_2". The first authentication property is *Answer Origin Authentication* and concerns both the registration and examination phases. It ensures that only one exam form from each candidate and only the forms submitted by eligible candidates (registered) are actually collected.

Definition 3 (Answer Origin Authentication). *An e-exam protocol ensures* Answer Origin Authentication *if, for every e-exam process EP, each occurrence of the event* $collected(id\_c, ques, ans)$ *is preceded by a distinct occurrence of the event* $reg(id\_c)$ *on every execution trace.*

At examination phase, each candidate submits his exam form with an answer, and the collector collects the forms. *Form Authorship* ensures that the contents of each collected exam form (id_c, $ques$, and ans) are not modified after submission.

Definition 4 (Form Authorship). *An e-exam protocol ensures* Form Authorship *if, for every e-exam process EP, each occurrence of the event* $collected(id\_c, ques, ans)$ *is preceded by a distinct occurrence of the event* $submitted(id\_c, ques, ans)$ *on every execution trace.*

Similarly, *Form Authenticity* ensures that the content of each exam form is not modified after the collection and until after the form is marked by an examiner.

Definition 5 (Form Authenticity). *An e-exam protocol ensures* Form Authenticity *if, for every e-exam process EP, each occurrence of the event* $marked(ques, ans, mark, id\_form, id\_e)$ *is preceded by a distinct occurrence of the events* $distrib(id\_c, ques, ans, id\_form, id\_e)$ *and* $collected(id\_c, ques, ans)$ *on every execution trace.*

At notification phase, the candidate should receive the mark which was assigned by the examiner to his answer. We call this property *Mark Authenticity*.

Definition 6 (**Mark Authenticity**). *An e-exam protocol ensures* Mark Authenticity *if, for every e-exam process EP, each occurrence of the event $notified(id\_c, mark)$ is preceded by a distinct occurrence of the events $marked(ques, ans, mark, id\_form, id\_e)$ and $distrib(id\_c, ques, ans, id\_form, id\_e)$ on every execution trace.*

Note that *Mark Authenticity* ensures that the candidate is notified with the mark delivered by the examiner on the answer assigned to him by the authority. This answer may be different from that submitted by the candidate. Only if also *Form Authorship* and *Form Authenticity* hold then the candidate can be sure that the assigned and submitted answers are identical. *Mark Authenticity* does not guarantee that the mark is computed correctly.

3.2 Privacy properties

We model our privacy properties as observational equivalence, a standard choice for such kind of properties (Ryan and Schneider, 2001; Ryan and Smyth, 2011). We use the *labeled bisimilarity* ($\approx_l$) to express the equivalence between two processes (Abadi and Fournet, 2001). Informally, two processes are equivalent if an observer has no way to tell them apart.

As a notation, we use what in applied π-calculus is called "context". The context $EP_I[\_]$ is the process EP without the identities in the set I; they are replaced by "holes". We use it when we need to specify exactly the processes for candidates id_1 and id_2 without repeating the entire e-exam instance. This is done by rewriting EP as $EP_{\{id_1,id_2\}}[C\sigma_{id_1}\sigma_{a_1}|C\sigma_{id_2}\sigma_{a_2}]$. Notation $EP|_e$ denotes the process EP without the code that follows the event e. The first privacy property says questions are kept secret until the exam starts.

Definition 7 (**Question Indistinguishability**). *An e-exam protocol ensures* Question Indistinguishability *if for any e-exam process EP that ends with the registration phase, any questions q_1 and q_2, we have that:* $EP_{\{id_Q\}}[Q\sigma_{q_1}]|_{reg} \approx_l EP_{\{id_Q\}}[Q\sigma_{q_2}]|_{reg}$.

Question Indistinguishability states that two processes with different questions have to be observationally equivalent until the end of the registration phase. This prevents the attacker from obtaining information about the exam questions before the examination phase starts. This property requires the question committee to be honest; otherwise the property is trivially violated since the committee reveals the questions to the attacker. However, it is particularly interesting to consider dishonest candidates, as they might be interested in obtaining the questions in advance. We can do this by replacing honest candidates with dishonest ones. For example, if we assume that candidate id_1 is dishonest, we obtain $EP_{\{id_1,id_Q\}}[(C\sigma_{id_1}\sigma_{a_1})^{c_1,c_2}|Q\sigma_{q_1}]|_{reg} \approx_l EP_{\{id_1,id_Q\}}[(C\sigma_{id_1}\sigma_{a_1})^{c_1,c_2}|Q\sigma_{q_2}]|_{reg}$.

The next property ensures that the marking process is done anonymously, *i.e.,* that two instances where candidates swap their answers cannot be distinguished until after the end of the marking phase. This may be desirable to ensure fairness of the grading, and is a requirement in some exam settings (at some universities or for competitive examinations).

Definition 8 (**Anonymous Marking**). *An e-exam protocol ensures* Anonymous Marking *if for any 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, we have that:* $EP_{\{id_1,id_2\}}[C\sigma_{id_1}\sigma_{a_1}|C\sigma_{id_2}\sigma_{a_2}]|_{mark} \approx_l EP_{\{id_1,id_2\}}[C\sigma_{id_1}\sigma_{a_2}|C\sigma_{id_2}\sigma_{a_1}]|_{mark}$.

Anonymous Marking ensures that 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. This prevents the attacker to obtain the identity of the candidate who submitted a certain answer before the marking phase ends. For this property, it is interesting to consider dishonest examiners. It can be done using the same technique employed for dishonest candidates outlined above. We can also have some dishonest candidates, however the candidates id_1 and id_2 who are assigned the two different answers have to be honest – otherwise the property can be trivially violated by one of them revealing his answer to the attacker.

To prevent bribing or coercion of the examiners, it might be interesting to ensure their anonymity, so that no candidate knows which examiner marked his copy.

Definition 9 (**Anonymous Examiner**). *An e-exam protocol ensures* Anonymous Examiner *if for any e-exam process EP, any two candidates id_1, id_2, any two examiners id'_1, id'_2, and any two marks m_1, m_2, we have that:* $EP_{\{id_1,id_2,id'_1,id'_2,id_{A_1}\}}[C\sigma_{id_1}\sigma_{a_1}|C\sigma_{id_2}\sigma_{a_2}|E\sigma_{id'_1}\sigma_{m_1}|E\sigma_{id'_2}\sigma_{m_2}|A_1\sigma_{dist_1}] \approx_l EP_{\{id_1,id_2,id'_1,id'_2,id_{A_1}\}}[C\sigma_{id_1}\sigma_{a_1}|C\sigma_{id_2}\sigma_{a_2}|E\sigma_{id'_1}\sigma_{m_2}|E\sigma_{id'_2}\sigma_{m_1}|A_1\sigma_{dist_2}]$ *where σ_{dist_1} attributes the exam form of candidate id_1 to examiner id'_1 and the exam form of candidate id_2 to examiner id'_2, and σ_{dist_2} attributes the exam form of candidate id_1 to examiner id'_2 and the exam form of candidate id_2 to examiner id'_1.*

Anonymous Examiner ensures that a process in which examiner id'_1 grades the exam form of candi-

date 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 id_2 and id'_2 grades that of id_1. Note that to ensure that in both cases the candidates receive the same mark, we also have to swap σ_{m_1} and σ_{m_2} between the examiners. Similar to *Anonymous Marking*, this property prevents the attacker to obtain or guess the identity of the examiner who marked a certain answer. *Anonymous Examiner* requires that the examiners id'_1 and id'_2 are honest, otherwise it will trivially violated by one of them revealing the mark he gave. We can again include dishonest candidates as they might be interested in finding out which examiner marked their copies.

In some exams settings the marks have to remain private. This is formalized in the next property.

Definition 10 (Mark Privacy). *An e-exam protocol ensures* Mark Privacy *if for any e-exam process EP, any marks m_1, m_2, we have that: $EP_{\{id'\}}[E\sigma_{id'}\sigma_{m_1}] \approx_l EP_{\{id'\}}[E\sigma_{id'}\sigma_{m_2}]$.*

Mark Privacy guarantees that two processes where the examiner id'_1 assigns for the same answer, entailed by the same context EP, two different marks m_1, m_2, cannot be distinguished from each other. Depending on the exam policy this can be an optional property since some exams system may publicly disclose the marks of the candidates. However, the intuition here is that candidate's performance should not be known to any other candidate. Again, we can assume that some candidates are dishonest and try to find out the marks of their colleagues, or that an examiner tries to find out the mark achieved by a candidate. The candidate who is assigned the two different marks has to be honest – otherwise the property is violated by him revealing his mark to the attacker. Similarly the examiner assigning the marks has to be honest, otherwise he can reveal the mark himself.

The previous definition of *Mark Privacy* ensures that the attacker cannot know the mark of a candidate. A weaker variant of *Mark Privacy* is *Mark Anonymity*, *i.e.,* the attacker might know the list of all marks, but is unable to associate a mark to its corresponding candidate. This is often the case in practice, where a list of pseudonyms (*e.g.,* student numbers) and marks is published.

Definition 11 (Mark Anonymity). *An e-exam protocol ensures* Mark Anonymity *if for any e-exam process EP, any candidates id_1, id_2, any examiner id'_1, any answers a_1, a_2 and a distribution σ_{dist} that assigns the answers of both candidates to the examiner, and two substitutions σ_{m_a} and σ_{m_b} which are identical, except that σ_{m_a} attributes the mark m_1 to the answer a_1 and m_2 to a_2, whereas σ_{m_b} attributes m_2 to the answer a_1 and m_1 to a_2, we have that: $EP_{\{id_1,id_2,id'_1,id_{A_1}\}}[C\sigma_{id_1}\sigma_{a_1}|C\sigma_{id_2}\sigma_{a_2}|E\sigma_{id'_1}\sigma_{m_a}|A_1\sigma_{dist}] \approx_l EP_{\{id_1,id_2,id'_1,id_{A_1}\}}[C\sigma_{id_1}\sigma_{a_1}|C\sigma_{id_2}\sigma_{a_2}|E\sigma_{id'_1}\sigma_{m_b}|A_1\sigma_{dist}]$*

The definition states that if an examiner id'_1, who is assigned the same answers a_1 and a_2 as σ_{dist} is unchanged, swaps the marks between these answers, the two situations cannot be distinguished by the attacker. This means that a list of marks can be public, but the attacker must be unable to link the marks to the candidates. Again, we can consider dishonest parties, but this definition requires the two concerned candidates and the two concerned examiners to be honest. Otherwise they can simply reveal the answer and the associated mark, which allows to distinguish both cases.

It is also easy to see that a protocol ensuring *Mark Privacy* also ensures *Mark Anonymity*. In fact, σ_{m_a} and σ_{m_b} are special cases of σ_{m_1} and σ_{m_2}.

4 Huszti & Pethő PROTOCOL

We first analyze the Huszti & Pethő protocol (Huszti and Pethő, 2010), which we call concisely H&P protocol. It aims to ensure authentication and privacy for e-exams in presence of dishonest candidates, examiners and exam authorities. The original paper also presents an informal security analysis based on conjectures. Such conjectures contribute to our motivation on the need of a framework for the formal analysis of e-exams. Notably, all the messages within the H&P protocol are sent via a *reusable anonymous return channel* (RARC) (Golle and Jakobsson, 2003) to achieve privacy properties.

A RARC implements anonymous[4] two-way conversations. It allows the party that initiates the protocol to send an anonymous message to a recipient. The recipient can reply without learning the sender's identity, but knowing that his reply will be dispatched to the actual sender. The entire conversation remains untraceable to an external attacker. A RARC is implemented by a re-encryption mixnet. The mix servers jointly generate and share an ElGamal key pair (PK_{MIX}, SK_{MIX}) and a pair of public/private signing keys (SPK_{MIX}, SSK_{MIX}). The sender A and the receiver B also hold ElGamal public/private key pairs, (PK_A, SK_A) and (PK_B, SK_B) respectively. A and B are represented by ID_A and ID_B, identity tags which can be for example A's and B's email addresses.

[4]Note that although the original security definition requires anonymity of the messages, it does *not* require secrecy of the messages.

To send the message m to B, the agent A submits to the mixnet the tuple $Mix(m, A, B)$ that denotes $(\{ID_A, PK_A\}_{PK_{MIX}}, \{m\}_{PK_{MIX}}, \{ID_B, PK_B\}_{PK_{MIX}})$ and proves knowledge of $\{ID_A, PK_A\}$ and of $\{ID_B, PK_B\}$. The proofs are meant to avoid that the attacker decrypts the triplet content by using the mixnet as a decryption oracle (in Section 4.2 we prove this claim to be false considering a Dolev-Yao threat model). The mixnet waits to collect more triplets and then shuffles them. Then, it adds a checksum to the triplets in order to guarantee their integrity while they are shuffled, yet this provides no end-to-end integrity protection. The message m is then re-encrypted with the public key of B using a switching encryption keys technique. The mixnet signs the encrypted public key of A. Thus B receives the pair $(sign(\{ID_A, PK_A\}_{PK_{MIX}}, SK_{MIX}), \{m\}_{PK_B})$ where $sign(x, sk)$ is message x plus the signature with the secret key sk. Then B replies to A with a new message m' by sending to the mixnet $(Mix(m', B, A), sign(\{ID_A, PK_A\}_{PK_{MIX}}, SK_{MIX}))$ and proving only knowledge of $\{ID_B, PK_B\}$. The mixnet checks the proof and the signature, and then processes the tuples like a normal message.

4.1 Protocol Description

We use ProVerif to verify the protocol. The ProVerif model is based on the description presented in original paper (Huszti and Pethő, 2010). Here we only give an overview of the protocol.

The H&P protocol relies upon different cryptographic building blocks. The ElGamal cryptosystem (Elgamal, 1985) is used to provide parties with public/private key pairs. A RARC implements anonymous two-way communication. A network of servers provides a timed-release service (NET), which contributes to create and revoke the candidate's pseudonym. More precisely, their contribution to the pseudonym is shared among the servers using the threshold Shamir secret sharing system (Shamir, 1979). At notification, a subset of the NET servers can use their shares to recover the secret and de-anonymize the pseudonym of the candidate. Then, the exam authority can associate the answer with the corresponding candidate. To avoid plagiarism, the protocol assumes that no candidate reveals its private key to another candidate and that invigilators supervise candidates during the examination. We describe the protocol in five phases, distinguishing the examiner and the candidate registration. However, according to our model, the two registration (sub-)phases are merged into a single phase.

Examiner Registration: The exam authority publishes the public parameters to identify a new examination. The question committee then signs and sends the questions and the starting time of the phases encrypted with the public key of the RARC mixnet. The mixnet forwards the message only when the examination begins. The examiner is then provided with a pseudonym, which is jointly generated by the exam authority and the examiner. The examiner verifies the correctness of the pseudonym by using a zero-knowledge proof (ZKP). Then, the examiner sends his pseudonym to the exam authority, and proves the knowledge of his secret key.

Candidate Registration: The registration of a candidate slightly differs from the registration of an examiner. The candidate pseudonym is jointly calculated by the exam authority, the candidate, and also the NET to provide anonymity for the candidates. The NET stores the secret values used for the pseudonym generation, which can be used to de-anonymize the candidate after the examination has finished. Again, the candidate finally verifies the correctness of his pseudonym using a ZKP.

Examination: The candidate sends his pseudonym via the RARC to the exam authority and proves the knowledge of his private key. Then, the exam authority checks whether the candidate is registered for the examination, and sends him the questions signed by the question committee. The candidate sends his answer, again via the RARC. The exam authority replies with a receipt which consists of the hash of all parameters seen by the exam authority during the examination, the transcription of the ZKPs, and the time when the answer was submitted.

Marking: The exam authority chooses an examiner who is eligible for the examination, and forwards him the answer via the RARC. Then the examiner assigns a mark to the answer, and authenticates them using a ZKP.

Notification: When all the answers are marked, the NET de-anonymizes the pseudonyms linked to the answers, and the exam authority stores the marks.

4.2 Formal Analysis

The equational theory depicted in Table 1 models the cryptographic primitives used within the H&P protocol. The equational theory includes well-known models for probabilistic encryption and digital signatures. Inspired by Backes *et al.* (Backes et al., 2008b), we model the ZKP of knowledge of a secret exponent as two functions, proof and verification. The proof function $zkp\_proof(public, secret)$ takes as arguments a secret and public parameters (i.e. the exponent and the generator to the power of the expo-

nent). It can be constructed only by the prover who knows the secret parameter. The verification function $zkpsec(zkp\_proof(public, secret), verinfo)$ takes as arguments the proof function and the verification parameter $verinfo$. The verifier only accepts the proof if the relation between $verinfo$ and $secret$ is satisfied. However, we support the model for the ZKP of the equality of discrete logarithms zkp_proof with tables in ProVerif. This is due to the difficulties of ProVerif when dealing with associativity of multiple exponents, which is used in the H&P protocol. We also assume the same generator is used for generating the pseudonyms of candidates and examiners, in order to avoid non-termination in ProVerif. This is sound because we distinguish the roles, and each principal is identified by its public key. We replace the candidate identity with his corresponding pseudonym inside the events to check authentication properties. We note that the replacement is also sound because the equational theory preserves the bijective mapping between the keys that identify the candidate and his pseudonym.

$$
\begin{aligned}
decrypt(encrypt(m, pk(k), r), k) &= m \\
getmess(sign(m, k)) &= m \\
checksign(sign(m, k), pk(k)) &= m \\
exp(exp(g, x), y) &= exp(exp(g, y), x) \\
checkproof(xproof(p, p1, g, exp(g, e), e), & \\
p, p1, g, exp(g, e)) &= true \\
zkpsec(zkp\_proof(exp(b, e), e), exp(b, e)) &= true
\end{aligned}
$$

Table 1: Equational theory to model H&P protocol

First we analyzed the RARC alone and found an attack on anonymity and privacy, which is detailed in the next paragraph. We then replaced the RARC with an implementation of a secure RARC using honest parties to check if the protocol ensures some properties given a working anonymous channel. In this case, ProVerif terminates for all properties on H&P[5].

The result of the verification together with the time required for ProVerif to conclude on a standard PC (Intel i7, 8GB RAM), are summed up in Table 2.

Property	Result	Time
Answer Origin Authentication	×	<1 s
Form Authorship	×	<1 s
Form Authenticity	×	<1 s
Mark Authenticity	×	<1 s
Question Indistinguishability	×	<1 s
Anonymous Marking	×	8 m 46 s
Anonymous Examiner	×	9 m 8 s
Mark Privacy	×	39 m 8 s
Mark Anonymity	×	1h 15 m 58 s

Table 2: Summary of our analysis on the formal model of the H&P protocol.

Attack on RARC: ProVerif shows that the RARC fails to guarantee both secrecy of messages and anonymity of sender and receiver identities, which is its main purpose inside the H&P protocol. We refer the triplet $\langle c_1, c_2, c_3 \rangle$ as the encrypted messages that A submits to the mixnet when she wants to send a message to B. From the description of RARC given at the beginning of this section, we recall that c_1 encrypts the A's public key, c_2 encrypts the message to B, and c_3 encrypts the B's public key. All ciphertexts are encrypted with the mixnet's public key.

The attacker in control of the network can use the RARC as a decryption oracle, letting the RARC reveal any of the plaintexts. The attack works as follows. The attacker chooses one of the three ciphertexts (depending on whether he wants to target the contents of the message, or the identities of the sender and receiver) and submits this as a new message. For example, if the attacker targets $c_1 = \{ID_A, PK_A\}_{PK_{MIX}}$, he resubmits c_1 as a new encrypted message, which means that $c_2' = c_1$ in the new triplet. He can leave the encryption of the senders key and the proof concerning the key unchanged, but replaces the encryption of the receiver's key with a public key PK_I for which he knows the corresponding secret key SK_I. In our example this means $c_3' = \{ID_I, PK_I\}_{PK_{MIX}}$. The attacker can also provide the necessary proof of knowledge of plaintext, since he knows this plaintext.

The RARC then mixes the input messages, and sends the encryption of the message under the receiver's public key to the receiver. In our example the attacker receives $\{ID_A, PK_A\}_{PK_I}$. Since the attacker knows the secret key SK_I he can obtain the original message. In our example he gets ID_A, the identity of the sender which should have remained anonymous. Since the attacker can substitute any of the items in the triplet as the new message, the RARC does neither ensure secrecy of the messages nor the anonymity of the sender or the receiver. Note that the checksum meant to guarantee the integrity of the triplet is only added after the submission of the message and is only used inside the mixnet. Hence, the checksum does not prevent the attacker from submitting a modified triplet. Even if it were added before, it would not prevent the attack as the knowledge of the ciphertexts is sufficient to compute the checksum.

[5]All ProVerif codes are available on line `http://apsia.uni.lu/stast/codes/exams/proverif_code_secrypt.tar.gz`

Note that the RARC was originally designed to withstand a passive attacker, which however is allowed to statically corrupt parties. We argue that this is not realistic in the e-exam setting where dishonest parties could try to cheat. Moreover, even static corruption is sufficient to carry out the attack outlined above: one dishonest party that can send and receive messages via the RARC is sufficient. Also, he has to intercept the message before it enters the RARC, but this is difficult to prevent in a normal unsecured network such as the Internet.

Authentication properties: We verified the authentication properties modelling the RARC as an ideal anonymous (yet not secret, according to the original security definition) channel. Note that the following attacks also remain valid if the protocol adopts the RARC.

ProVerif finds an attack on *Answer Origin Authentication* where the attacker can create a fake pseudonym that allows him to take part in an exam for which he did not registered. This is possible because the exam authority does not check whether the pseudonym has been actually created using the partial information provided by the time-release service. The attacker generates his own secret key SK_A, and calculates an associate pseudonym, which sends to the exam authority. The exam authority successfully verifies the received data and that the attacker knows SK_A, thus the exam authority accepts the answer. *Form Authorship* fails due to the same attack: in fact, the exam authority may collects an exam form which is modified by changing the pseudonym to a one chosen by the attacker.

ProVerif also shows that the H&P protocol does not ensure *Form Authenticity*, because there is no mechanism that allows the examiner to check whether the answers have been forwarded by the exam authority. Even if the original RARC is used and the answer is encrypted with the public key of the mixnet, this does not guarantee that the exam authority actually sent the message.

ProVerif provides a similar attack for *Mark Authenticity*. In fact, the attacker can forward any answer to any examiner, even if the answer was not collected by the exam authority. Morevoer, the attacker can notify the candidate by himself with a mark of his choice.

Privacy properties: ProVerif finds an attack on *Question Indistinguishability*. This is because the attack on the RARC exposes the message and the identities of the sender and receiver. As the questions are sent through the RARC, the attacker can obtain them. Moreover, as the candidate's answer is also sent through the RARC, the protocol does not ensure *Anonymous Marking*: the answer can be linked to its corresponding sender. The protocol ensures neither *Mark Privacy* nor *Anonymous Examiner*, as the marks are also sent through the RARC. Hence, they can be decrypted and the examiner can be identified.

We checked the H&P protocol in ProVerif assuming correct RARC (i.e. ensuring anonymity, but no secrecy). Also in this case ProVerif shows an attack for each property. *Anonymous Examiner* can be violated because the attacker can track which examiner accepts the ZKP when receiving the partial pseudonym, and then associate to the examiner the answer that the latter grades. Moreover, a similar attack on *Anonymous Marking* remains: the attacker can check whether a candidate accepts the ZKP to associate him with a pseudonym, and then identify his answer. Finally, neither *Mark Privacy* nor *Mark Anonymity* are ensured because the examiner sends the mark to the exam authority in clear.

To sum up, the H&P protocol ensures no properties at all, and ProVerif discovers attacks. While the authentication properties fail due to a weak protocol design, the privacy properties fail because of an inappropriate use of the RARC, which was neither designed to ensure secrecy nor to withstand active attackers. Moreover, we identified flaws in the RARC even in a static corruption setting, and in the H&P protocol assuming a correct anonymous channel.

5 Remark! PROTOCOL

We first give the protocol presented in (Giustolisi et al., 2014) and then the results of our analysis.

5.1 Protocol Description

The Remark! protocol has the same set of parties of the H&P protocol, but relies on a different approach. The NET is indeed several servers that implement an *exponentiation mixnet* (Haenni and Spycher, 2011). The speciality of exponentiation mixnets is that each server blinds its entries by a common exponent value. On entry X, the mixnet outputs X^r where r is the product of the secret exponent values of the servers. At registration, the NET creates the pseudonyms for the candidates and examiners without involving any of them. The pseudonyms are eventually used as public-key encryption and signature verification keys in such a way to allow parties to communicate anonymously. A bulletin board[6] is used to

[6] A public append-only memory.

publish the pseudonyms, the test questions and the receipts of test submissions. The combination of the exponentiation mixnet and a bulletin board allows the protocol to not rely on a RARC as anonymous channel.

Remark! only assumes that each party is given a pair of public/private key with a common generator g, i.e. the private key x and the public key $y = g^x$. Below, we present the protocol within the four e-exam phases.

Registration: The list of eligible candidates' and examiners' public keys is sent as a batch to the NET. The NET calculates the pseudonyms by raising the initial public keys to a common value $r = \prod_i r_i$. More specifically, each mix server raises the input message to a secret value r_i, and forwards it to another mix server. At the same time the NET blindly permutes the batch of public keys. The so obtained keys eventually become the pseudonyms for candidates and examiners. Along with the pseudonyms $y' = y^r = (g^x)^r$, the NET publishes a new generator h, which is the output of g raised to the product of each mix server secret value, i.e. $h = g^r$. Both the candidates and the examiners can identify their own pseudonyms by raising h to their secret key x, i.e. $h^x = (g^r)^x$. The pseudonyms from now on serve as public encryption and signature verification keys. Two different batches are used for candidates and examiners because only the identities of candidates are revealed at notification.

Examination: The exam authority signs and encrypts the test questions with the candidate's pseudonym and publishes them on the bulletin board. Each candidate submits his answer, which is signed with the candidate's private key (but using the generator h instead of g) and encrypted with the public key of the exam authority. The exam authority collects the test answer, checks its signature using the candidate's pseudonym, re-signs it, and finally publishes its encryption with the corresponding candidate's pseudonym as receipt.

Marking: The exam authority encrypts the signed test answer with an eligible examiner pseudonym and publishes the encryption on the bulletin board. The corresponding examiner marks the test answer, and signs it with his private key (again using the generator h instead of g). The examiner then encrypts it with the exam authority public key, and submits its marks to the exam authority.

Notification: When the exam authority receives all the candidate evaluations, it publishes the signed marks, each encrypted with the corresponding candidate's pseudonym. Then, the NET servers deanonymize the candidate's pseudonyms by revealing their secret exponents. Hence the candidate anonymity is revoked, and the mark can finally be registered. Note that the examiner's secret exponent is not revealed to ensure his anonymity even after the exam concludes.

$$checkpseudo(pseudo\_pub(pk(k), rce), pseudo\_priv(k, exp(rce))) = true$$

$$decrypt(encrypt(m, pk(k), r), k) = m$$

$$decrypt(encrypt(m, pseudo\_pub(pk(k), rce), r), pseudo\_priv(k, exp(rce))) = m$$

$$getmess(sign(m, k)) = m$$

$$checksign(sign(m, k), pk(k)) = m$$

$$checksign(sign(m, pseudo\_priv(k, exp(rce))), pseudo\_pub(pk(k), rce)) = m$$

Table 3: Equational theory to model Remark! protocol

5.2 Formal Analysis

We analyze Remark! with ProVerif, following similar techniques as the one used in the analysis of the H&P protocol. Table 4 sums up the results together with the time required for ProVerif to conclude on the same PC used for H&P. We model the bulletin board as a public channel, and use the equational theory depicted in Table 3. The equations for encryption and signatures are standard, but we also added the possibility of using the pseudonym keys to encrypt or sign. The public pseudonym, which also serves as exam form identifier, is obtained using the function $pseudo\_pub$ on the public key and the random exponent. The function $pseudo\_priv$ can be used to decrypt or sign messages, using the private key and the new generator g^r (modelled using the function exp) as parameters. The function $checkpseudo$ allows us to check if a pseudonym corresponds to a given secret key (or its pseudonym variant).

Authentication properties: Supposing an attacker in control of the network and all parties to be honest, we can successfully verify all authentication properties in ProVerif. For this, we replace the candidate identity with the candidate's pseudonym inside the events. This is sound as each candidate is uniquely identified by his keys, and there is a bijective mapping between keys and pseudonyms by construction of the equational theory (for a given random exponent).

We also verified the authentication properties considering dishonest parties. In this case, all properties are guaranteed except *Form Authenticity*. The attack trace shows that a dishonest candidate can pick the

examiner of his choice by re-encrypting the signed receipt received from the exam authority. It means that the candidate can influence the choice of the examiner who will correct his exam. As the protocol description envisages an access control for publishing into the bulletin board, a feature that we could not code in ProVerif, we cannot claim this to be an attack as the candidate may not be allowed to post on the bulletin board. However, we demonstrate that with a simple fix there is no need of access control policies for publishing into the bulletin board. The fix consists in making the intended pseudonym of an examiner explicit within the signature that designates the examiner as evaluator of an exam. In doing so, the exam authority's signature within the receipt cannot be used by a candidate to designate any examiner because the receipt includes no examiner's pseudonym. The exam authority will only accept exam evaluations that contain its signature on examiner's pseudonym. Considering the fix, ProVerif confirms that Remark! guarantees all the security properties including *Form Authenticity*, even in presence of dishonest parties.

Privacy properties: All the privacy properties are satisfied. For *Question Indistinguishability*, we only assume the exam authority to be honest, and then conclude that the property holds. For *Mark Privacy*, we assume only the concerned candidate and examiner, as well as the exam authority, to be honest. All other candidates and examiners are dishonest, and ProVerif still concludes successfully. Note that this subsumes a case with multiple honest candidates and examiners, since a dishonest party can behave like an honest party. This also implies that the protocol ensures *Mark Anonymity* as noted above. For *Anonymous Examiner*, we assume only the examiners and the NET to be honest. If the NET publishes the pseudonyms in random order, ProVerif concludes successfully. Similarly for *Anonymous Marking*, we assume only the candidates and the NET to be honest. Again, if the NET publishes the pseudonyms in random order, ProVerif concludes successfully.

6 CONCLUSION

We define the first formal framework for the analysis of secure e-exam protocols. We show how to model e-exam protocols in the applied π-calculus, and define nine relevant security properties: four authentication properties and five privacy properties.

Using ProVerif, we analyze the security of two e-exam protocols. The first protocol has only been argued to be secure. Our analysis shows that it indeed satisfies none of the nine properties. Authentication is compromised because of inaccuracies in the protocol design, whereas most of attacks invalidating privacy exploit attacks on the RARC. These attacks compromise secrecy and anonymity of the messages, and exploit the absence of a proof of knowledge of the submitted message to the RARC, which allows its use as a decryption oracle. Such a proof is not explicitly required in the original specification of the RARC, and is certainly missing in the H&P protocol: the "exam authority" is required to forward questions and answers without knowing them, and thus cannot prove knowledge of them when submitting them to the RARC. Even when assuming a perfect RARC ensuring anonymity, we still have attacks on all properties. Thus, we think that fixing the RARC is not sufficient – the protocol requires fundamental changes.

Property	Result	Time
Answer Origin Authentication	✓	< 1 s
Form Authorship	✓	< 1 s
Form Authenticity	✓*	< 1 s
Mark Authenticity	✓	< 1 s
Question Indistinguishability	✓	< 1 s
Anonymous Marking	✓	2 s
Anonymous Examiner	✓	1 s
Mark Privacy	✓	3 m 32 s

Table 4: Summary of our analysis on the formal model of the Remark! protocol. (*) *Form Authenticity* fails with dishonest candidate. It holds after applying our fix.

Also Remark!, the second protocol analyzed, has been only informally argued to be secure in the original paper. It presents a weakness concerning *Form Authenticity*. We propose a fix and formally verify that the (fixed) protocol satisfies all the properties herein considered.

Generally speaking, our framework and our analysis bring e-exams into the attention of the security community. E-exams and in general computer-based assessment tools are becoming widespread, some of them supported by e-learning platforms such as the massive open online courses (MOOC). Nevertheless, they call for being formally proved secure, since most of them have not been submitted to any rigorous security analysis. Such applications are complex and exposed to unprecedented cheating attacks very subtle to be discovered. We set the first research step on the formal understanding of such systems and establishes a framework for the automatic analysis of their security properties.

As a future work we intend to analyze more protocols designed for computer-based tests although obtaining protocol's specifications from the providers is not an easy task. Other interesting research works in-

clude the study of the relation between our security properties as well as the definition of novel properties such as verifiability, reliability, and accountability.

REFERENCES

Abadi, M. and Fournet, C. (2001). Mobile values, new names, and secure communication. In *POPL*. ACM.

Arapinis, M., Bursuc, S., and Ryan, M. (2013). Privacy-supporting cloud computing by in-browser key translation. *Journal of Computer Security*, 21(6):847–880.

Backes, M., Hritcu, C., and Maffei, M. (2008a). Automated verification of remote electronic voting protocols in the applied pi-calculus. In *CSF*. IEEE.

Backes, M., Maffei, M., and Unruh, D. (2008b). Zero-knowledge in the applied pi-calculus and automated verification of the direct anonymous attestation protocol. In *IEEE S & P'08*.

Bella, G., Costantino, G., Coles-Kemp, L., and Riccobene, S. (2011). Remote management of face-to-face written authenticated though anonymous exams. In *CSEDU (2)*, pages 431–437. SciTePress.

Blanchet, B. (2001). An efficient cryptographic protocol verifier based on prolog rules. In *CSFW. IEEE*.

Blanchet, B., Abadi, M., and Fournet, C. (2008). Automated verification of selected equivalences for security protocols. *J. Log. Algebr. Program.*, 75(1):3–51.

Castellà-Roca, J., Herrera-Joancomartí, J., and Dorca-Josa, A. (2006). A secure e-exam management system. In *ARES*, pages 864–871. IEEE Computer Society.

Copeland, L. (2013). School cheating scandal shakes up atlanta. `http://www.usatoday.com/story/news/nation/2013/04/13/atlanta-school-cheatring-race/2079327/`.

Delaune, S., Kremer, S., and Ryan, M. (2006a). Verifying properties of electronic voting protocols. In *Proceedings of WOTE'06*.

Delaune, S., Kremer, S., and Ryan, M. (2009). Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487.

Delaune, S., Kremer, S., and Ryan, M. D. (2006b). Coercion-resistance and receipt-freeness in electronic voting. In *CSFW'06*. IEEE.

Dolev, D. and Yao, A. C. (1983). On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208.

Dong, N., Jonker, H. L., and Pang, J. (2010). Analysis of a receipt-free auction protocol in the applied pi calculus. In *FAST'10*, volume 6561 of *LNCS*. Springer.

Dreier, J., Jonker, H., and Lafourcade, P. (2013a). Defining verifiability in e-auction protocols. In *ASIACCS*, pages 547–552. ACM.

Dreier, J., Lafourcade, P., and Lakhnech, Y. (2011). Vote-independence: A powerful privacy notion for voting protocols. In *FPS*, volume 6888 of *LNCS*. Springer.

Dreier, J., Lafourcade, P., and Lakhnech, Y. (2012a). Defining privacy for weighted votes, single and multi-voter coercion. In *ESORICS*, LNCS. Springer.

Dreier, J., Lafourcade, P., and Lakhnech, Y. (2012b). A formal taxonomy of privacy in voting protocols. In *ICC*, pages 6710–6715. IEEE.

Dreier, J., Lafourcade, P., and Lakhnech, Y. (2013b). Formal verification of e-auction protocols. In *POST*, volume 7796 of *LNCS*, pages 247–266. Springer.

Elgamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *Information Theory, IEEE Transactions on*, 31(4):469–472.

Furnell, S., Onions, P., Knahl, M., Sanders, P., Bleimann, U., Gojny, U., and Röder, H. (1998). A security framework for online distance learning and training. *Internet Research*, 8(3):236–242.

Giustolisi, R., Lenzini, G., and Bella, G. (2013). What security for electronic exams? *8th Int. Conf. on Risk and Security of Internet and Systems (CRiSIS)*.

Giustolisi, R., Lenzini, G., and Ryan, P. (2014). Remark!: A secure protocol for remote exams. In *Security Protocols XXII*, LNCS. Springer. to appear. Draft `http://apsia.uni.lu/stast/codes/exams/preSPW14.pdf`.

Golle, P. and Jakobsson, M. (2003). Reusable anonymous return channels. In *Proc. of the 2003 ACM workshop on Privacy in the electronic society*, WPES '03. ACM.

Haenni, R. and Spycher, O. (2011). Secure internet voting on limited devices with anonymized dsa public keys. In *WOTE'11*. USENIX.

Herrera-Joancomartí, J., Prieto-Blázquez, J., and Castellà-Roca, J. (2004). A secure electronic examination protocol using wireless networks. In *ITCC*. IEEE.

Hjeltnes, T. and Hansson, B. (2005). *Cost Effectiveness and Cost Efficiency in E-learning*. QUIS - Quality, Interoperability and Standards in e-learning, Norway.

Huszti, A. and Pethő, A. (2010). A secure electronic exam system. *Publicationes Mathematicae Debrecen*, 77:299–312.

Ryan, M. and Smyth, B. (2011). Applied pi calculus. In *Formal Models and Techniques for Analyzing Security Protocols*, chapter 6. IOS Press.

Ryan, P. Y. A. and Schneider, S. A. (2001). Process algebra and non-interference. *J. Comput. Secur.*, 9(1-2).

Ryan, P. Y. A., Schneider, S. A., Goldsmith, M., Lowe, G., and Roscoe:, A. W. (2000). *The Modelling and Analysis of Security Protocols: The CSP Approach*. Addison-Wesley Professional, first edition.

Shamir, A. (1979). How to share a secret. *Commun. ACM*, 22(11):612–613.

Watson, R. (2014). Student visa system fraud exposed in BBC investigation. `http://www.bbc.com/news/uk-26024375`.

Weippl, E. (2005). *Security in E-learning*, volume 6 of *Advances in Information Security*. Springer Science + Business Media.