Remark!: A Secure Protocol for Remote Exams

Rosario Giustolisi, Gabriele Lenzini, and Peter Y.A. Ryan(✉)

Interdisciplinary Centre for Reliability, Security and Trust (SnT),
University of Luxembourg, Luxembourg, Luxembourg
{rosario.giustolisi,gabriele.lenzini,peter.ryan}@uni.lu

Abstract. This manuscript presents *Remark!*, an electronic exam protocol which achieves several authentication, (conditional) anonymity, privacy, and verifiability properties without trusted third parties. *Remark!* is primarily designed for invigilated Internet-based exams but it also fits computer-based exams with candidates taking their exam in classrooms.

Keywords: Electronic exams · Security analysis · Security protocol design · Authentication · Anonymity · Privacy · Verifiability

1 Introduction

There is a growing demand for tools able to evaluate remotely the skills and knowledge of people. Remote assessment promises in fact to be cheaper than traditional examination, which usually requires a considerable organizational effort. In its resorting to information and communication technology, computer-based assessment is also able to reach easily a worldwide audience of candidates. However, the use of computers exposes remote examinations to new threats, while it may require to change seasoned procedures successfully used against known threats. Hence, there is an interest in understanding how to design secure remote electronic exams (in short *e-exams*). To date, there has been very little investigation into the design and analysis of e-exams. In this paper we fill this gap: we discuss a number of security properties appropriate to e-exams, and we present and comment a novel scheme designed to achieve them.

1.1 Anatomy of an Exam

Traditional exams consist at least of four phases: *registration*, *testing*, *marking*, and *notification*. During the registration phase, an exam is arranged, usually by a manager, and candidates enrol for it. During the testing phase, the candidates receive and take a test and submit their answers. During the marking phase, examiners assess the answers and assign a mark. Finally, during the notification phase, candidates learn their marks.

E-exams are organized similarly and with the same principals involved in traditional exams: candidates, one or more examiners, and a manager. The roles of

B. Christianson et al. (Eds.): Security Protocols 2014, LNCS 8809, pp. 38–48, 2014.
DOI: 10.1007/978-3-319-12400-1_5

candidate and examiner are straightforward. Essentially, the candidate is responsible to provide answers to the test question, while the examiner is in charge of evaluating the provided answers. The role of the manager includes all remaining tasks necessary to fulfil the examination. Such tasks include: to register eligible candidates and examiners for an examination, to assign the test question to the candidates, to distribute the answered test to examiners, to gather the marks and notify them to the corresponding candidates. According to the specific implementation, the manager's tasks can be increased or split among other principals, such as question committee, invigilator, collector, and notifier.

1.2 Related Work

To our knowledge only a few works propose exam protocols with security in mind. Bella *et al.* propose WATA IV [GRG14], the latest version of computer-assisted exam systems that are paper-based, hence without support for remote examination. Castella-Roca *et al.* [JJA06] propose a computer-based e-exam system with a fully trusted manager. Huszti & Petho [AA10] advance a remote e-exam scheme with fewer trust requirements on principals, but their scheme turns out to have several problem of security. We have identified a number of security flaws but, at time of writing, the work that analyses this protocol and discusses its security is under review [DGK+14].

Contribution and Outline. This paper first identifies threats and requirements for e-exams, in particular the ones that *Remark!* has been designed to counter (Sect. 2). Then, it proposes the details of *Remark!*, an e-exam protocol primarily designed for invigilated internet-based exams (Sect. 3). As we shall see, the protocol also suits computer-based testing where candidates take the exam at examination venue, such as a classroom or a test centre. According to the security analysis provided, *Remark!* achieves authentication, verifiability, and conditional anonymity with minimal reliance on trusted parties (Sect. 4). The paper concludes discussing some future work (Sect. 5).

2 Threats, Security Requirements, and Assumptions

2.1 Threats

E-exams are threatened by outsiders as well as by insiders: each role has in fact incentives to misbehave. In traditional exams, most common threats are due to candidates, who may try to cheat in different ways during the testing phase [che]. In remote exams, candidates may also attempt to exploit the security flaws of the underlying e-exam protocol to abuse any exam phase. For instance, a dishonest candidate may want to tamper with her and other's marks, or find the identity of the examiner who evaluates her test in order to bribe him.

On the other hand, even exam authorities may misbehave as exposed in recent exam scandals [L.13,R.14]. Both managers and examiners might be strongly

motivated to tamper with candidates' marks, especially when they collude with some candidates. This can lead to embarrassing situations as recently occurred in the U.S. Navy nuclear exam cheating scandal [usn].

As we shall see, *Remark!* cannot withstand all possible threats. It is a cryptographic protocol and for this reason designed to withstand network security threats. Plagiarism can still happen: to avoid it there is need of appropriate invigilation. Principals can still collude and communicate via subliminal channels, as it happens when a candidate reveals her identity to the examiner by using steganography. Although it is hard to rule out completely such a threat, steganalysis techniques can be of some help here. Other counter-measures may be needed against collusion attacks that exploit covert channels.

In particular, *Remark!* is designed to resist the following threats:

1. An intruder impersonating a candidate during the testing.
2. An intruder tampering with a candidate's test answer or mark.
3. A candidate seeking to get an higher mark than she deserved (overmarked).
4. A candidate seeking to coerce the examiner who evaluates her test.
5. The manager tampering with the marks.
6. An examiner seeking to assign a biased mark to a specific candidate's test.

2.2 Security Requirements

We have identified several fundamental security requirements that a secure e-exam should fulfil. The list outlined above takes inspiration and extends the requirements described in our previous work [GRG13]:

p1: ***Test Answer Authentication:*** the manager only accepts test answers submitted by registered candidates. This means that a candidate, the test assigned to her, and their association should be authenticated and preserved, for instance against collusion among candidates.

p2: ***Examiner Authentication:*** the manager only accepts evaluations provided by a registered examiner. This rather obvious requirement means that the mark assigned to a test answer is authentic.

p3: ***Anonymous Marking:*** no one learns the author of a test answer before the test is marked. This requirement states that only the candidate who wrote the answers knows the association between her identity and the test. Notably, this should be resistant to collusion between examiner and manager.

p4: ***Anonymous Examiner:*** no candidate learns the identity of the examiner who evaluates their test answers. This requirement ensures that no candidate can coerce an examiner before and even after he evaluates her test answer.

p5: ***Question Secrecy:*** no candidate learns the test question before the testing phase begins. This ensures a desirable degree of fairness among candidates as no one knows the questions in advance, provided that no one is illegitimately allowed to know the answers beforehand.

p6: ***Question Privacy:*** the manager does not learn which test question is assigned to a specific candidate. This requirement ensures that the manager cannot identify a candidate by looking at the test question once it has been submitted for evaluation. In contexts where all students for a given exam receive the same questions this requirement becomes superfluous.

p7: ***Mark Privacy:*** the candidate learns only her mark and not those of other candidates. This rather natural requirement, not always applied, means that the mark of a test is known only by the author of test, and possibly by the manager, who may need it for registering the mark.

p8: ***Test Verifiability:*** the candidate can verify that her test is considered for evaluation. This requirement states that the candidate has a way to check whether her submitted test has been accepted by the manager.

p9: ***Mark Verifiability:*** the candidate can verify that the manager registers the mark she was assigned to by the examiner. This ensures that the candidate can notice when the mark assigned to her test differs from the one registered by the manager.

2.3 Assumptions

Our design and analysis rely on the following assumptions:

1. Each principal holds a long-term public/private pair of keys.
2. The candidate holds a smart card. This carries the candidate's identity visibly engraved and stores her private key securely (*i.e.*, it cannot be extracted).
3. Candidates are invigilated during the testing. Invigilation mitigates cheating, and it can be done remotely by using software such as ProctorU [Pro].
4. The model answers are kept secret from the candidates until after the exam has completed. The examiners may be provided with the model answers after testing has finished.
5. An authenticated append-only bulletin board is available. Everyone is guaranteed to see the same data, though write access might be restricted to appropriate entities (*e.g.*, see [JPV13]). An implementation of a bulletin board together with a detailed description of its security requirements is given in Culnane *et al.* [CS14].
6. Channels ensuring integrity and confidentiality, *e.g.*, SSL/TLS, are available.
7. At least one mixnet server (see next) is honest.

3 *Remark!*: Protocol Description

Remark! relies on several servers that implement an *exponentiation mixnet* [RO11]. The peculiarity of this kind of mixnet is that each mix server re-encrypts the terms by a common exponent value in contrast to a conventional re-encryption mixnet in which each term is independently re-encrypted. As usual, we assume that at least one server among the ones in a mixnet behaves honestly.

Thus, if the mixnet is made of m servers and r_i is the exponent value chosen by the ith server, then the mixnet given the input X, outputs $X^{\overline{r}_m}$, where $\overline{r}_m = \prod_{i=1}^{m} r_i$.

Remark! makes use of exponentiation mixnets at registration, to create the pseudonyms for the candidates and examiners. The mixnet servers may also required at notification to revoke the candidates pseudonyms and retrieve the candidates' identities by revealing $\overline{r}_m$. In particular, the generation of pseudonyms for candidates is separated from that for examiners because, at notification, only the identities of candidates should be revealed.

A bulletin board is used to publish the pseudonyms, the questions, the tests, and the marks. As we shall see, the bulletin board is also used by the mixnet's servers to publish their intermediate shuffling. In so doing anyone can check the authenticity of each mix step.

The following paragraphs detail how to use exponentiation mixnets to generate pseudonyms, and describe all the phases of *Remark!*. The protocol's steps are also illustrated in Appendix in form of a message sequence chart. In the reminder, $\langle X_i \rangle$ is a shorthand for the list $\langle X_1, \ldots X_n \rangle$, and $\overline{r}_k$ is a shorthand for $\prod_{i=1}^{k} r_i$ (so, $\overline{r}_1 = r_1$, and $\overline{r}_2 = r_1 r_2$, *etc.*) and $\overline{\pi}_k$ for $\pi_k \circ \cdots \circ \pi_1$, (so, $\overline{\pi}_1 = \pi_1$, and $\overline{\pi}_2 = \pi_2 \circ \pi_1$, *etc.*).

Registration. The registration uses an exponentiation mixnet to generate pseudonyms for the candidates and examiners, in two independent runs.

In particular, let us assume n eligible candidates with identities $C_1, \ldots, C_n$. Let g denote a generator of a multiplicative subgroup $\mathbb{G}$ of order q. Each C_i has a pair of public/private keys (PK_i, SK_i), each $PK_i = g^{SK_i}$. The identities of the candidates as well as their public keys are public.

The first mix server mix_1 takes $\langle PK_i \rangle$ —the list of the public keys of the candidates— generates a fresh random $r_1 \in \{1, q-1\}$, and computes $\langle PK_i^{r_1} \rangle$ —the list of the public keys to the r_1. Then mix_1 signs and sends this list in a secret shuffled order (*i.e.*, it posts $\langle PK^{r_1}_{\pi_1(i)} \rangle$, where π_1 is the permutation of indexes applied by mix_1) to the bulletin board. It also sends g^{r_1} to the next mix server over a secure channel. Further mix servers repeat these steps as required. Each mix server signs and publishes the shuffled list on the bulletin board, as shown in Fig. 1. The last mixserver, mix_m, publishes also $g^{\overline{r}_m}$. We assume that the bulletin board has appropriate write access control mechanisms, (*i.e.*, only authorities can publish data therein). If the access control can be relied on, the signatures might not be necessary.

While the intermediate steps should be posted to a bulletin board, we do not post intermediate $g^{\overline{r}_1}, \ldots, g^{\overline{r}_{m-1}}$ terms. This is to avoid each candidate tracking their intermediate pseudonyms through the mixnet: although such eventuality is not an attack, it is an undesired feature. The last mix server mix_m publishes the final $h_C := g^{\overline{r}_m}$, and the list of pseudonyms $\langle \overline{PK}_i \rangle = \langle PK^{\overline{r}_m}_{\overline{\pi}_m(i)} \rangle$. Note that zero-knowledge proofs could be used to demonstrate that the mix servers behave correctly. Once the shuffled pseudonyms and the corresponding signatures have been posted along with h_C, each candidate, say C_k, can recognize her pseudonym

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

Fig. 1. Using exponentiation mixnet to generate pseudonyms. All the terms within the box are published on the bulletin board.

among those in the shuffled list $\langle \overline{PK}_i \rangle$ by computing $h_C^{SK_k}$ and finding the match. The pseudonym from now on serves as the pseudo identity for C_k.

After the pseudonyms of candidates have been published, the mixnet generates the pseudonyms for examiners in a similar way. Since a different random value is used by the mix servers to generate the examiner pseudonyms, a different h_E is published at the end of the mix.

Testing. Before starting the testing phase, the manager generates the test questions, signs them with its private key SK_M, and encrypts each test question under a candidate pseudonym. We do not specify how the manager generates the test questions in order to include different forms of assessment (e.g., multiple choice, open-ended, etc.) and assignments (e.g., single question, different questions for candidate, random question permutations, etc.). In general, we observe that a test question denotes a list of questions.

As soon as the testing starts, the manager authenticates the candidate via remote invigilation software. In particular, the manager checks whether the candidate details printed on the top of the smart card matches the candidate identity. When all candidates have been authenticated, the manager publishes the encrypted test questions in the bulletin board. Once all the candidates have received their test questions, they are allowed to work on their test answers.

When the candidate concludes to answer the test, she can submit the test as follows: the candidate appends her pseudonym and the test answer to the test question, so the filled test is $T_C = \langle ques, ans, \overline{PK} \rangle$. Then, she signs T_C with her private key SK_C using the generator h_C instead of g. Thus, the signature can be verified using the candidate's pseudonym $\overline{PK}_C$ with respect to h_C. The candidate then encrypts the signed test with the public key of the manager PK_M, and submits it to the manager. The manager collects and decrypts the test, and then signs a hash of T_C using his private key SK_M. The manager then encrypts the signed hash(T_C) under the corresponding candidate's pseudonym, and publishes the encryption as receipt.

Marking. The manager encrypts the signed test under an eligible examiner pseudonym $\overline{PK}_E$ previously published on the bulletin board by the mixnet.

The designated examiner marks the test. The mark is appended to the signed test, thus generating the evaluation $M_C = \langle Sig\{T_C\}_{SK_M}, mark\rangle$. The examiner then signs M_C with his private key SK_E and the generator h_E. Finally, the examiner encrypts M_C with PK_M and submits his evaluation to the manager.

Note that it is possible to introduce a universally verifiable deterministic assignment of answers to examiners. Thus, for example, the encrypted answers and the examiners pseudonyms could be posted in two lexically ordered lists and the assignment performed cyclically. Such an assignment algorithm should remove any suspicion that the authority might try to perform the assignment in some unfavourable way to the candidates.

Notification. The manager receives the encrypted evaluation from the examiner, which are decrypted and re-encrypted under the corresponding candidate pseudonym $\overline{PK}_C$. The manager publishes all the test evaluations together. Then, the manager asks the mixnet to reveal the random values r used to generate the candidates pseudonyms. In so doing, the candidate anonymity is revoked, and the mark can finally be registered. Note that each candidate (and only him) can see his mark before $\overline{r}_m$ is revealed.

Notification (alternative). Some universities allow candidates decide whether to know the mark or to withdraw their test entirely without knowing he mark. In this case, the mark is forgotten and it is not notified nor registered. This particular way to run a final exam is adopted, for instance, by those universities where candidates are conceded with a limited amount of failures during the exam season, mainly to discourage them from taking the exam without adequate preparation. Other universities, again to discourage candidates to sit at the exam just 'to try it out to get marked', have a rule saying that if a candidate chooses to know her mark and this turns out to be a fail, then she has to skip the next exam session. By giving a candidate the possibility to withdraw a test without knowing the mark, those universities soften the severity of such rules. A candidate can spare wasting one of her attempt token when she realizes, on her own, to have performed insufficiently.

Remark! can include such requirement via an alternative notification phase. In this case, the manager publishes a public commit of the mark instead of the mark. Then, if a candidate wants to know her mark, she proves the knowledge of her private key. If so, the manager reveals the commitment parameter to the candidate, and the candidate can check the commitment. Notably, the notification does not involve the mixnet.

4 Security Analysis

We discuss informally the security of *Remark!* and give arguments supporting the claim that it achieves our security requirements. We organize our argumentation in four sections. The first section discusses authenticity properties, the second anonymity properties, the third privacy properties, and the last verifiability properties.

4.1 Authentication

Test Answer Authentication (p1) is achieved because the manager accepts only the test whose signature can be verified with a pseudonym published by the mixnet. No one but the candidate who holds the corresponding private key can generate a correct signature. Colluding candidates who switch their smart cards are detected by invigilation.

Examiner Authentication (p2) holds because the manager encrypts the test with the examiner's pseudonym. Only the examiner who holds the corresponding private key obtains the test. The manager accepts the evaluation only if it can check the signature using the corresponding examiner's pseudonym.

4.2 Anonymity

The pseudonym guarantees the anonymity of the test submitted by the candidate. The mix servers cannot associate a pseudonym to a candidate's identity, unless all of them collude. Even if a malicious examiner colludes with the manager, Anonymous Marking (p3) holds unless all the mix servers reveal their secret exponents.

Remark! ensures Anonymous Examiner (p4) because the manager encrypts the test with the examiner's pseudonym. The examiner can fairly evaluate the anonymous test answers without fear of being coerced by any candidate, because the pseudonyms of the examiners are not revoked by the mixnet. Even in case of collusion between a candidate and an examiner, if the examination board consists of different examiners, the candidate has no guarantee that the colluding examiner will be assigned to evaluate her test answers.

4.3 Privacy

Question Secrecy (p5) is achieved because the manager publishes the test question once the candidate is under invigilation. The manager cannot learn which test question is assigned to a specific candidate because the test question are encrypted with the anonymous candidate's pseudonym. Thus, *Remark!* ensures Question Privacy (p6).

The protocol also ensures Mark Privacy (p7) because the mark is encrypted with the candidate's pseudonym and then published on the bulletin board. Thus, each candidate only learns her corresponding mark. Only the manager learns the mark after the mixnet reveals the secret exponents.

4.4 Verifiability

Each mix server publishes its generated list of signed pseudonyms (the intermediated and the last), and a zero-knowledge proof of correctness (*e.g.*, all pseudonyms are generated by using the same exponential value). This allows any observer to verify the authenticity and the correctness of the pseudonyms. Once the final pseudonyms are published on the bulletin board, each eligible candidate and examiner can only find their corresponding pseudonym.

Test Verifiability (p8) is guaranteed because the manager publishes the receipt after it receives a valid signature (i.e. the manager can verify a signature using a pseudonym as verification key). Thus the candidate can verify that her test is considered for evaluation. Moreover, she can also prove that her test has been accepted because the manager signs the receipt.

Remark! ensures Mark Verifiability (p9). In fact, the marks are published before the mixnet reveals their secret exponents. Thus, the candidate can verify that the manager registers the correct mark once the mixnet revokes her anonymity. Note that both the manager and the examiner sign the test to which the mark is assigned. Since the mark is signed by the examiner, if the manager registers an incorrect mark, the candidate can prove to an authority the correct mark the examiner assigned to her test.

Note that the manager may post the candidate pseudonyms alongside the marks, which allows everyone to verify the set of posted marks is correctly derived. In this way each candidate can still identify her own mark, while everyone can check, still anonymously, that the marks are distributed among all the candidates. However, such a modification is incompatible with assuring strong mark privacy (p7).

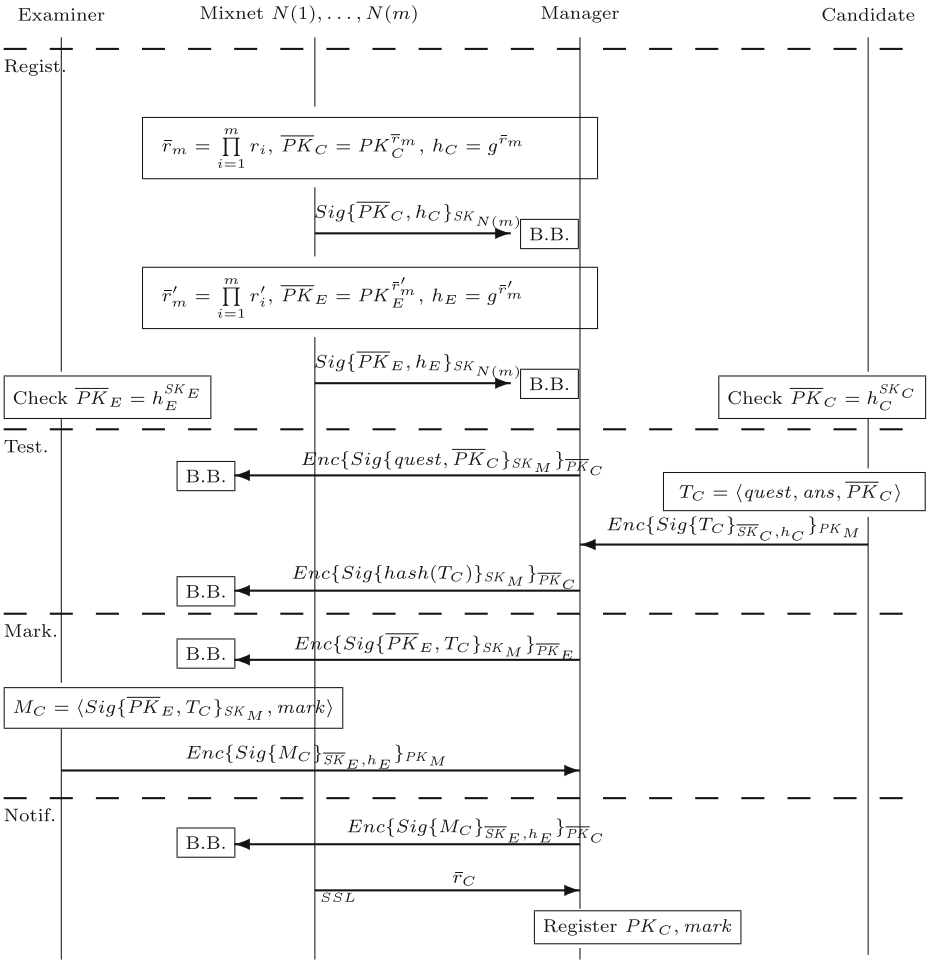
5 Conclusion

This paper proposes *Remark!*, an e-exam protocol that achieves heterogeneous security properties (authentication, privacy, anonymity, and verifiability) in a realistic threat model with minimal trust assumptions. Notably, it requires no trusted parties but that only one mix server behave honestly. *Remark!* can resist collusion of candidates, examiner and manager, or examiner and candidate. Although the paper presents an informal analysis of the protocol, a preliminary formal analysis of *Remark!* in the symbolic model confirms that it ensures all the nine security requirements. As future work, we plan to build a prototype of *Remark!*, and to engineer it as an extension of Moodle. Other future work includes extending our protocol with techniques to detect plagiarism and candidate cheating during the testing phase. We envisage that misbehaviour detection strategies such as data mining used to derive patterns described by Pieczul *et al.* in [OS14] can be useful for our purposes. Another interesting research direction includes the support for collaborative marking, in which the questions are categorised by subject, and examiners evaluate the test answers pertaining to their subject area. Finally, we observe the need of an efficient way to resolve disputes perhaps following the principles outlined in [SR14].

Acknowledgement. We thank Jannik Dreier, Ali Kassem and Pascal Lafourcade for helpful discussions on the security of our protocol.

A *Remark!*: Message Sequence Chart

Notation. A test question is denoted by *quest*, and a test answer by *ans*. SK_X and PK_X denotes the ElGamal private and public keys of the principal X. We assume a common public generator g for the keys of all principals. $\overline{PK}_X$ denotes the pseudonym of the principal X, and r_{X_i} is the secret value used by the mix server i when processing the batch of the role X. The terms *Enc* and *Sig* denote respectively the encryption and signature functions of a message. In particular, the notation $Sig\{msg\}_{\overline{SK}_X, h_X}$ denotes the message *msg* and its signature using the private key SK_X and the parameter h_X rather than g.



References

[AA10] Huszti, A., Pethõ, A.: A secure electronic exam system. Publicationes Mathematicae Debrecen **77**, 299–312 (2010)

[che] How to Cheat on Test Using School Supplies. http://www.wikihow.com/Cheat-on-a-Test-Using-School-Supplies

[CS14] Culnane, C., Schneider, S.: A peered bulletin board for robust use in verifiable voting systems. CoRR, abs/1401.4151 (2014)

[DGK+14] Dreier, J., Giustolisi, R., Kassem, A., Lafourcade, P., Lenzini, G., Ryan, P.Y.A.: Formal analysis of electronic exams (work under review) (2014)

[GRG13] Bella, G., Giustolisi, R., Lenzini, G.: What security for electronic exams? In: 2013 International Conference on Risks and Security of Internet and Systems (CRiSIS), pp. 1–5 (2013)

[GRG14] Bella, G., Giustolisi R., Lenzini, G.: Secure exams despite malicious management. In: Proceedings of 12th Conference on Privacy, Security and Trust (PST) (to appear, 2014)

[JJA06] Castella-Roca, J., Herrera-Joancomarti, J., Dorca-Josa, A.: A secure e-exam management system. In: Proceedings of the First International Conference on Availability, Reliability and Security, ARES '06, pp. 864–871. IEEE Computer Society, Washington, DC (2006)

[JPV13] Benaloh, J., Ryan, P.Y.A., Teague, V.: Verifiable postal voting. In: Christianson, B., Malcolm, J., Stajano, F., Anderson, J., Bonneau, J. (eds.) Security Protocols 2013. LNCS, vol. 8263, pp. 54–65. Springer, Heidelberg (2013)

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

[OS14] Pieczul, O., Foley, S.N.: Collaborating as normal: detecting systemic anomalies in your partner. In: Proceedings of 22nd International Security Protocols Workshop, Cambridge (to appear, 2014)

[Pro] ProctorU. http://www.proctoru.com/

[R.14] Watson, R.: Student visa system fraud exposed in BBC investigation, February 2014. http://www.bbc.com/news/uk-26024375

[RO11] Haenni, R., Spycher, O.: Secure internet voting on limited devices with anonymized DSA public keys. In: Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections, EVT/WOTE'11. USENIX Association (2011)

[SR14] Murdoch, S., Anderson, R.: Security protocols and evidence: where many payment systems fail. In: Financial Cryptography (to appear, 2014)

[usn] U.S. Navy discloses nuclear exam cheating. http://edition.cnn.com/2014/02/04/us/navy-cheating-investigation/