Publ. Math. Debrecen
77/3-4 (2010), 299–312

A secure electronic exam system

By ANDREA HUSZTI (Debrecen) and ATTILA PETHŐ (Debrecen)

Dedicated to Professor Pál Dömösi on the occasion of his 65*th birthday*

Abstract. Electronic exam is a difficult part of e-learning security. This paper describes a cryptographic scheme that possesses security requirements, such that authenticity, anonymity, secrecy, robustness, correctness without the existence of a Trusted Third Party. The proposed protocol also provides students a receipt, a proof of a successful submission, and it is based on existence of anonymous return channels.

1. Introduction

Student learning through computerized systems has become a hot topic in recent years. E-exam management is one of the most important building blocks of an e-learning environment. An electronic assessment system raises more security issues than the other parts of an e-learning software. However an e-exam scheme should achieve all features that traditional paper-based exams provide, we require that the electronic solution will decrease our duty, save costs and time (cf. [16]). All security obligations should be completely fulfilled, hence its design should take a special care of security.

Face to face exams not only provide an opportunity to check student's identity, but also ensure examinees to comply with the rules (students must not talk to each other etc.). In case of distance or virtual exams achieving a legitimate operation require special security concerns. One of the major challenges is to

Mathematics Subject Classification: 94A60, 68P25.
Key words and phrases: ElGamal encryption, secret sharing.

identify exam takers, to make sure far from the examinee that the person who answers the questions is the one who is supposed to take the exam. Besides the traditional password based solution, several times fingerprint biometrics authentication (cf. [13]) is applied. In the present paper we use public key cryptography for authentication. This is secure provided that the owner of the private key does not give it to another person. The only way to prevent our proposed system from such a cheating is that the secret key is used not only for the exam.

The other challenge is to monitor the examinee to assure he does not use illegal tools. The proposed electronic exam system assumes the existence of an examination center controlled by a supervisor, where all exam stages are performed electronically.

This protocol is designed not only for multiple-choice tests, but for write in tests, too. Hence submitted exams are sent to teachers to correct them. We also carry out a feature that in case of face to face exams is difficult to be achieved, our scheme guarantees anonymity of teachers and students, as well. Teachers do not know the identity of the student that prevents partiality and students do not know who will correct their exams, either, hence they cannot bribe or threaten a teacher in order to receive a better grade. A closer look shows that the requirements for the anonymity of students and teachers are different. At the end of the exam students will get their marks, thus they are interested in recovering their real identity. This is not necessary for teachers, their real identity should remain secret.

The authors in [12] introduced an e-exam protocol for wireless networks, in 2006 J. Castella-Roca, J. Herrera-Joancomarti and A. Dorca-Josa in [2] designed a secure e-exam management system, containing three kinds of participants: students, teachers and a manager. Manager is an authority that is responsible for the whole process, manages questions, answers, grades. Manager is assumed to be honest, so the scheme is based on a Trusted Third Party. Managers authenticate students and teachers by their key pairs, hence their real identity are revealed. Property of anonymity is achieved by applying the honest manager. Our aim is to achieve anonymity without relying on an honest manager.

There are several commercial solutions, but they do not describe their security measures ([5], [9], [15]), and papers dealing with implementation issues ([1]).

1.1. Our solution. All the security requirements are accomplished by applying cryptographic primitives. In order to identify students pseudonyms are employed. A student at the beginning of his studies should possess a secret key, and for each exam a new pseudonym is generated deriving from this original master key, hence

it should be kept secret. Since at the end of the exam grades should be inserted into the on-line database, thus general identity of the student should be able to retrieved. This is managed with a timed-release solution, meaning before a certain time no one can connect the pseudonym to the student, but after the deadline students' identification information is revealed that gives a connection between a pseudonym and the corresponding student. We do not assume that the Exam Authority is honest. During the process neither the teacher, nor the Exam Authority knows the real identity of the student and neither the authority, nor the student knows who corrects the student's paper. The proposed scheme possesses all the necessary requirements without applying a Trusted Third Party. Only Registry, which is responsible for generating key pairs and system parameters during the setup stage, is honest. There are n servers (NET) provide the timed-release service, in order to achieve anonymity, these servers compose a Mixnet, too. Since there are complete conversations between the participants an anonymous return channel [8] is applied.

2. Exam scheme

2.1. Security requirements. Security requirements for an exam scheme are as follows:

Authenticity: Only eligible students' tests should be considered, hence the authority has to verify whether the sender is allowed to take the exam or not. A student after registration can reveal his pseudonym to another student asking her to take the test instead of him. Authenticity avoids this attack. Students must be sure they have received valid questions, i.e. generated by teachers of the university. It should be verified whether the exam grade is proposed by a teacher, who qualified for, i.e. only eligible teachers are allowed to correct papers.

Anonymity: Students may try to bribe or threaten teachers to get a better grade. The system provides anonymity for students and teachers, as well. Teachers do not know which paper belonging to which student he is correcting and students do not know who corrects their papers. Students' and teachers' authentication is managed without revealing their real identity.

Secrecy: Exam questions and answers are kept secret. During the examination process neither the questions nor the generated answers are revealed. At the end of the exam the grades should be published in a way, that only the corresponding student should know his mark.

Robustness: Exam questions and answers can not be altered and after submission no one is allowed to modify them.

Correctness: Students are not allowed to take the same exam more than once and an already submitted exam cannot be denied.

Receipt: After sending the solution students are able to make sure of the successful submission.

2.2. Participants. The protocol participants are the Registry, students, teachers and the Exam Authority.

Registry (R): Registry provides secret and public keys for the participants, generates the necessary system parameters during the setup stage. R is trusted, does not collude with other participants.

Students (S): Students wish to take the exam, we suppose they might be malicious.

Teachers (T): Teachers check the tests and give a grade.

Exam authority(EA): Exam authority issues pseudonyms for eligible participants, manages the exam process, checks authenticity, chooses a teacher for a student anonymously and at the end of the exam updates the database with the student's grade.

2.3. Functions. In the exam scheme we will use primitives of public key cryptography, mainly ElGamal encryption, anonymous return channel and time-release service, which will be defined in the next section.

An exam starts with a registration process, when students and teachers get a pseudonym. This pseudonym is unique for each participant, but cannot be connected to the real user before the grading stage. Each participant gets at most one pseudonym. A pseudonym is constructed in a way, that Exam Authority can verify the identity of the user, and his authenticity for taking the exam or correcting the paper. Before sending the questions to a student or an answer sheet to a teacher EA verifies whether he/she possesses an authorized pseudonym. Eligible students receive exam questions and send the corresponding answers back with the duration time. EA checks whether the student has taken this test before, if not, sends the answer sheet to an eligible teacher, who corrects it and sends a grade back. At the end EA gets the real identity of students back from the pseudonym and inserts the corresponding grade. We define an exam scheme as

$$ExS = \{register, ifeligible, takeexam, getidentity\},$$

where

(1) Function
$$register(g_U, PK_U, SK_U, \bar{s}, random\ value) \to pseudonym$$
takes g_U, PK_U, the participant's (student or teacher) public keying material and authenticates it with $\bar{s}$ private key of the exam and SK_U participant's private key. Applying random values ([6], [7], [10], [11]) pseudonyms cannot be linked to the participants. For each exam a new pseudonym is generated.

(2) Function
$$ifeligible(pseudonym, subject) \to \{0, trans\}$$
checks eligibility of the participant, *i.e.* whether the pseudonym is authorized for the subject given as input. The owner of the pseudonym is verified by running an interactive zero knowledge proof. It outputs transcript *trans* if it is correct and a 0 otherwise.

(3) Function
$$takeexam(pseudonym, quest, answ, time) \to grade$$
takes a pseudonym, questions, the answers and the duration time of the exam and outputs a grade.

(4) Function
$$getidentity(pseudonym) \to identity$$
takes a pseudonym and determine the corresponding real identity of the student, in order to give his/her grade.

3. Building blocks

3.1. ElGamal encryption. ElGamal [4] is a probabilistic public-key cryptosystem. Let P and Q be large primes so that $Q|(P-1)$. G_Q denotes a group of prime order Q, g a generator of G_Q, $SK \in \mathbb{Z}_Q$ a private key and $PK = g^{SK} \pmod P$ the corresponding public key. Plaintexts are in G_Q and ciphertexts in $G_Q \times G_Q$. ElGamal encryption of $m \in G_Q$ is
$$Enc(m) = (g^k \pmod P, m \cdot PK^k \pmod P),$$
where $k \in \mathbb{Z}_Q$ is randomly chosen. ElGamal is semantically secure under the assumption that the Decisional Diffie Hellman (DDH) problem is hard in the group G_Q.

3.2. Reusable anonymous return channel. Reusable anonymous return channel [8] enables a complete anonymous conversation. Any recipient can deliver anonymous messages and even send back one or more anonymous replies to the sender. The representation of this channel is a re-encryption mix network emp-

loying ElGamal cryptosystem, based on the fact that ElGamal encryption allows for re-encryption of ciphertexts.

Setup: Servers jointly generate the public and private parameters of an ElGamal cryptosystem. Public information, G_Q, g, PK are published and SK private key is shared among the mix servers in a threshold manner. The mix servers also establish a shared signing key.

Submission of messages: We assume that A wants to send anonymously a message M to B. A has an identifying tag ID_A and a public key PK_A and submits

$$(Enc_{Mix}(ID_A \| PK_A), Enc_{Mix}(M), Enc_{Mix}(ID_B \| PK_B))$$

to the Mixnet and proves knowledge of $ID_A \| PK_A$[1].

Mixing: When there are sufficiently many messages in the batch, each mix server mixes and re-encrypts the triplets in a way that the elements of a triplet are not separated or reordered. A checksum is appended to each triplet guaranteeing the integrity of them.

Delivery of messages: Mixnet converts $Enc_{Mix}(M)$ to $Enc_{PK_B}(M)$ and generates a signature on $Enc_{Mix}(ID_A \| PK_A)$. Recipient B receive

$$(Enc_{Mix}(ID_A \| PK_A), Sig, Enc_{PK_B}(M)).$$

Submitting a reply: Let suppose B wants to reply N to the sender, then submits to the Mixnet

$$(Enc_{Mix}(ID_B, \| PK_B), Enc_{Mix}(N), Enc_{Mix}(ID_A \| PK_A), Sig))$$

with the proof of knowledge $ID_B \| PK_B$. If the signature is valid and the proofs are correct, then the replies are mixed and delivered in the same way as the original messages.

For security properties, we refer to [8].

3.3. Timed-release service. Timed-release service is applied to get students' identity back in the stage of grading. During registration a student's pseudonym is randomized by NET, a network containing n servers. We suppose there are at least $n - t + 1$ trusted servers. In order to synchronize the servers a Time Server Authority (TSA) providing absolute time reference is needed. Registry plays the role of TSA. Let P and Q be large primes so that $Q|(P-1)$. G_Q denotes a group of prime order Q. The NET receives message $g_U \pmod P$ as input and calculates $g_U^{\Gamma} \pmod P$ output, where Γ is secret, shared among the n servers, employing (t, n) threshold Shamir's secret sharing system (cf. [14]). Registry chooses n distinct, non-zero elements of $\mathbb{Z}_Q$, denoted by x_i, $1 \leq i \leq n$ and gives

[1]Here and in the sequel $x \| y$ denotes the concatenation of the words x and y.

the value x_i to server i, where values x_i, $1 \le i \le n$ are public. In order to share the value Γ Registry secretly chooses $a_1, a_2, \ldots, a_{t-1}$ elements of $\mathbb{Z}_Q$, computes and sends $y_i = a(x_i)$ $(1 \le i \le n)$ to server i in a secure way, where

$$a(x) \equiv \Gamma + \sum_{j=1}^{t-1} a_j x^j \pmod{Q}.$$

For the request of Registry t servers calculate the output message applying Lagrange interpolation:

$$g_U^{\Gamma} \equiv \prod_{j=1}^{t} g_U^{b_j y_{i_j}} \pmod{P},$$

where

$$b_j = \prod_{1 \le k \le t, k \ne j} \frac{x_{i_k}}{x_{i_k} - x_{i_j}} \pmod{Q}.$$

Each server stores $(time, y_i, g_U)$, where *time* declares when the server is allowed to publish value g_U. When *time* arrives t servers calculate g_U^{Γ} from their shares and publish g_U.

4. The proposed scheme

4.1. Assumptions. Our model is constructed in a way, that EA is able to manage several exams at the same time. We assume there are more eligible teachers for a given subject, and a teacher might correct papers in more topics. All participants have digital certificates, they are responsible for their secret key, revealing a secret key causes crucial advantage for an other user. Security of key-generation procedure depends on computational security assumptions. The adversary's computational resources are bounded by a polynomial in the security parameter.

4.2. Our protocol. During the **setup phase** R generates all system parameters, participants' public and secret keys. Let P and Q be large primes so that $Q|(P-1)$. G_Q denotes $\mathbb{Z}_P^*$'s multiplicative subgroup of order Q. Each participant U chooses randomly a secret key $SK_U \in \mathbb{Z}_Q$ and an arbitrary element $g_U \in G_Q$, calculates $PK_U \equiv g_U^{SK_U} \pmod{P}$ public information. These public and secret keys are used during several exams. We suppose the existence of an on-line database containing the student's public key (g_S, PK_S) and information about whether they are allowed to take an exam. Before each exam EA chooses

$\bar{s} \in \mathbb{Z}_Q, \bar{g} \in G_Q$ and publishes exam verifier information: $(\bar{g}, \bar{h})$, where $\bar{h} \equiv \bar{g}^{\bar{s}} \pmod{P}$, and keeps $\bar{s}$ secret. Each server gets ElGamal key pair and jointly setup the Mixnet.

In our scheme exam questions are generated by a committee. Let PK_C, SK_C denote committee's ElGamal public and secret keys. Test questions are sent encrypted by Mixnet's public key, authenticity of them is assured by the committee's signature. EA receives

$$Enc_{Mix}(quest \| Sig_C(quest \| time_1)),$$

where $time_1$ denotes the time when the exam starts. This case EA does not know the questions and cannot change them.

There are three distinctive stages of the protocol.

Registration. Students' and teachers' pseudonym is calculated with $register(g_U, PK_U, SK_U, \bar{s}, random\ value)$. All messages are sent through an anonymous return channel.

(1) EA: Checks eligibility of the user in the database and calculates $\tilde{p} \equiv PK_U^{\bar{s}} \pmod{P}$ and stores $\tilde{p}$.

(2) **if** U is a student **do**

  (a) $EA \to NET$: $(\tilde{p}, g_U)$

  (b) NET: Calculates $p' \equiv \tilde{p}^{\Gamma} \pmod{P}$ and $r \equiv g_U^{\Gamma} \pmod{P}$, each server securely stores $(time, y_i, \tilde{p}, g_U)$, where $time$ denotes the time when g_U can be published and y_i is its share applied for secret sharing.

  (c) $NET \to U$: (r, p')

  else

  (a) $EA \to U$: $(\tilde{p}, g_U)$

  (b) U : Calculates $p' \equiv \tilde{p}^{\alpha} \pmod{P}$ and $r \equiv g_U^{\alpha} \pmod{P}$, where α is randomly chosen.

(3) **endif**

(4) U : Calculates $p \equiv r^{SK_U} \pmod{P}$.

(5) $U \leftrightarrow EA$: EA as a prover and U as a verifier run an interactive zero knowledge proof of the equality of discrete logarithms of $(p, p'), (\bar{g}, \bar{h})$ (cf. [3]).

At the end of registration a user possesses: (r, p, p'). Let denote eligible student's pseudonym: (a, b, b') and eligible teacher's pseudonym: (e, f, f'). The only difference is that there is no need timed-release service for the teachers. We do not need to connect their pseudonyms to their real identity. As we pointed out in the Introduction students will have their marks at the end of the exam, thus they

are interested in recovering their real identity. The teachers do not have similar demand.

Exam. EA checks students' and teachers' eligibility with $ifeligible((a,b,b'), subject)$ and $ifeligible((e,f,f'), subject)$, where (a,b,b') and (e,f,f') are pseudonyms of students (S) and teachers (T) respectively. They run function $takeexam((a,b,b'), quest, answ, time)$ together.

(1) S: Calculates message $M = (a\|b\|b'\|subject)$.

(2) $S \to EA$: S submits

$$(Enc_{Mix}(ID_S\|PK_S), Enc_{Mix}(M), Enc_{Mix}(ID_{EA}\|PK_{EA}))$$

to Mixnet, where ID_S a randomly chosen identification number for the student S, PK_S is the public key of S. Participants may use different identifying tags and public keys for each message. Mixnet collects a batch of messages over a certain period of time. Each server mixes and re-encrypts each set of messages in the batch using ElGamal encryption and provides a proof of correct mixing. EA receives

$$(Enc_{Mix}(ID_S\|PK_S), Sig_{Mix}(ID_S\|PK_S), Enc_{PK_{EA}}(M)),$$

where $Sig_{Mix}(ID_S\|PK_S)$ is jointly generated by Mixnet.

(3) $T \to EA$: T calculates message $M = (e\|f\|f'\|subject)$ and sends to EA via Mixnet in a simmilar way like in step (2).

(4) EA: After decryption EA verifies whether the received pseudonym is authorized for taking or correcting the exam in $subject$ by checking the congruence $b^{\bar{s}} \equiv b' \pmod P$ or $f^{\bar{s}} \equiv f' \pmod P$, checks whether S submitted a test before. EA runs an interactive zero knowledge (ZK) proof of knowledge of the secret key (cf. [3]) with students and teachers, and securely (*i.e.* encrypted with Mixnet's public key) stores

$$((a,b,b'), trans_S, (e,f,f'), trans_T,\ Enc_{Mix}(ID_T\|PK_T), subject),$$

where $trans_U$ the transcript of U's ($U \in \{S,T\}$) ZK proof.

(5) $EA \to S$: EA sends questions back on the anonymous return channel to S with the actual time. EA submits to Mixnet

$$(Enc_{Mix}(ID_{EA}\|PK_{EA}),$$
$$Enc_{Mix}(M), Enc_{Mix}(ID_S\|PK_S), Sig_{Mix}(ID_S\|PK_S)),$$

where

$$M = quest_S\|Sig_C(quest_S)\|time_1$$

and $Sig_C(quest_S)$ denotes the exam questions signed by the committee.

(6) $S \to EA$: S verifies authenticity of the questions and creates the answer sheet $answ_S$ and sends the message
$$M = a \| b \| Enc_{Mix}(answ_S) \| time_2$$
back through the anonymous return channel, where $time_2$ denotes the exact time of submitting the exam answers.

(7) $EA \to S$: EA securely stores $(quest_S, time_1, time_2, Enc_{Mix}(answ_S))$ to the corresponding student's section in the database and sends hash value of all data:
$$Hash(a \| b \| b' \| subject \| trans_S \| quest_S \| time_1 \| time_2 \| Enc_{Mix}(answ_S))$$
to S as a receipt.

(8) EA: Chooses for each submitted exam a teacher and submits
$$(Enc_{Mix}(ID_{EAP} \| PK_{EA}), Enc_{Mix}(answ_S), Enc_{Mix}(ID_T \| PK_T))$$
to the Mixnet, where ID_{EAP} is a specially generated identification tag, it is different for each exam. EA stores $Enc_{Mix}(ID_T, \| PK_T)$ for the corresponding exam information.

(9) $T \to EA$: T corrects the exam, gives a grade and sends the following message through the anonymous return channel:
$M = (grade \| Hash(grade \| answ_S) \| [Hash(grade \| answ_S)]^{SK_T} \| noninttrans)$, where $noninttrans$ is the transcript of a non-interactive ZK proof of equality of discrete logarithm of
$$(Hash(grade \| answ_S), [Hash(grade \| answ_S)]^{SK_T}, e, f).$$

At the end of the exam stage, EA possesses students' pseudonyms and the corresponding grades.

Grading. After a certain time defined at the beginning of the exam timed-release service generates necessary information for each pseudonym in order to retrieve the real identity of the students, EA and NET run $getidentity(b', time)$ function. All messages are sent through an anonymous return channel.

(1) $EA \to NET$: Sends b', where b' is the part of students (a, b, b') pseudonym, remember $a \equiv g_S^{\Gamma} \pmod P, b \equiv PK_S^{\Gamma} \pmod P, b' \equiv PK_S^{\Gamma \bar{s}} \pmod P$.

(2) $NET \to EA$: If the time is after $time$, then t servers of the Mixnet calculates $b' \equiv \tilde{b}^{\Gamma} \pmod P$ and together with values $\tilde{b}$ sends encrypted back to EA.

(3) EA: Decrypts messages, gets student's real identity according to $(b', \tilde{b})$ and inserts
$Enc_{Mix}(ID_T \| PK_T)$ with
$$Hash(grade \| answ_S) \| [Hash(grade \| answ_S)]^{SK_T} \| noninttrans$$
and the grades into the database to the corresponding student.

4.3. Security analysis.

Theorem 4.1. *The proposed scheme possesses authenticity, anonymity, secrecy, robustness, correctness property and students receive a receipt after submission.*

PROOF. **Authenticity:** During the exam stage $ifeligible((a, b, b')\|subject)$ and $ifeligible((e, f, f'), subject)$ functions check eligibility of students and teachers. Exam authority checks whether the pseudonym is authorized for the corresponding subject, *i.e.* congruences $b^{\bar{s}} \equiv b' \pmod{P}$ and $f^{\bar{s}} \equiv f' \pmod{P}$ and, where $\bar{s}$ is the secret exponent of the given subject. If congruences hold EA verifies whether the sender is the owner of the pseudonym by running an interactive ZK-proof of knowledge of secret key for values a and b in case of students and e and f in case of teachers.

Another student, different from the owner of the pseudonym, cannot take the test without knowledge of the secret key. If a student (S_1) gives values a, b' to another student (S_2) asking S_2 to answer exam questions, it is detected during the interactive ZK-proof, since S_2 does not know S_1's secret key or if S_2 employs his secret key to get value b, the pseudonym will not be authorized for the corresponding exam anymore. The only way to get a correct pseudonym if S_1 reveals his secret key for S_2, we suppose secret keys are used not only for educational purposes.

Students receive test questions in the Exam stage in step 5, with the actual time and the signature of the committee, proving authenticity of them. Grades authenticated by the corresponding teacher are inserted into the database, with the transcript of the non-interactive ZK proof.

Anonymity: Anonymity of students and teachers is achieved by applying pseudonyms and an anonymous return channel. Students' pseudonyms are generated interactively by EA and NET. NET randomizes $g_S \pmod{P}$ and $\tilde{b} \equiv PK_S^{\bar{s}} \pmod{P}$ in case of students, hence EA cannot connect $a \equiv g_S^{\Gamma} \pmod{P}$ and $b' \equiv \tilde{b}^{\Gamma} \pmod{P}$ to g_S and $\tilde{b}$. Teachers' pseudonyms are also randomized by the teachers themselves.

Randomization is processed by Shamir's secret sharing scheme, hence only an authorized set of NET servers is able to generate the randomized pseudonym. We suppose there are at least $n - t + 1$ trusted servers, hence the randomized pseudonym cannot be connected to the real identity. Real identity of students are not revealed till the stage of grading. At the end students' identity should be retrieved. It is assured by a timed-release solution. We do not consider the case, when students deliberately reveal their

real identity by inserting their name or any special information in the answer of one of the write in questions.

Secrecy, Robustness: Before exam starts EA receives the encrypted, authenticated questions from a committee

$$(Enc_{Mix}(quest \| Sig_C(quest) \| time_1)),$$

hence neither EA not other participants know the questions and cannot change them. During the exam stage in step 6. students verify authenticity of them. Students send encrypted answers $(Enc_{Mix}(answ_S))$ to EA. EA forwards it to a teacher.

Teacher after correcting them sends

$$grade \| Hash(grade \| answ_S) \| [Hash(grade \| answ_S)]^{SK_T} \| noninttrans$$

back, where the grade and the corresponding answers are hashed and authenticated by the teacher. Authentication is proved with a transcript of a non-interactive ZK proof that is inserted into the database.

Students are able to verify that their answers are not modified by the receipt and the information inserted with the grade into the database. Questions and answers are transferred encrypted with the public key of Mixnet, hence it is not known by other participants. At the end of grading a student knows only his own grade.

Correctness: During the exam stage in step 4. EA checks whether the student has submitted a test before, if he has, EA does not consider the second one. If a student has submitted a test, it is corrected by a teacher and the student's identity is revealed, he cannot deny it.

Receipt: EA in step 7. in the Exam stage calculates and publishes hash value of

$$Hash(a, b, b', subject, trans_S, quest_S, time_1, time_2, Enc_{Mix}(answ_S))$$

as a receipt for the student. □

4.4. Comments. If a student has questions about his test after grading, he has an ability to ask the teacher, who has corrected his paper. With his grade he has received $Enc_{Mix}(ID_T \| PK_T)$, as well. Student S submits

$$(Enc_{Mix}(ID_S \| PK_S), Enc_{Mix}(questions), Enc_{Mix}(ID_T \| PK_T))$$

to the Mixnet, the corresponding teacher is able to answer the questions anonymously through the anonymous return channel. We assume, there is a certain time for submitting questions and answers, hence there should be enough messages for the Mixnet.

The proposed system is planned to be implemented in order to evaluate its usability in near future in frames of a project.

Acknowledgement. The work is supported by TÁMOP 4.2.1/B-09/1 /KONV-2010-0007 project. The project is implemented through the New Hungary Development Plan co-financed by the European Social Fund, and the European Regional Development Fund. The first author is partially supported by the project GOP-1.1.2-07/1-2008-0001 and also by the Hungarian National Foundation for Scientific Research Grant No. K75566, the second one is by No. T67580.

References

[1] T. S. M. Barhoom and Shen-Sheng Zhang, Trusted exam marks system at IUG using XML-signature, Proceeding of the Fourth International Conference on Computer and Information Technology (CIT'04), 2004, 288–294.

[2] J. Castella-Roca, J. Herrera-Joancomarti and A. Dorca-Josa, A secure e-exam management system, Proceeding of the First International Conference on Availability, Reliability and Security (ARES'06), 2006, 864–871.

[3] D. Chaum and T. Pryds Pedersen, Wallet databases with observers (extended abstract), Advances in Cryptology CRYPTO '92, *Springer-Verlag*, 1992, 89–105.

[4] T. El Gamal, A public key cryptosystem and a signature scheme based on discrete logarithms, Proceedings of CRYPTO 84 on Advances in cryptology, 1985, 10–18.

[5] ExamSoft Worldwide, `http://www.examsoft.com`.

[6] János Folláth, Construction of pseudorandom binary sequences using additive characters over GF(2^k), *Period. Math. Hungar.* **57**(1) (2008), 73–81.

[7] János Folláth, Construction of pseudorandom binary sequences using additive characters over GF(2^k) II., *Period. Math. Hungar.* **60**(2) (2010), 127–135. .

[8] P. Golle and M. Jakobsson, Reusable anonymous return channels, Proceeding of the 2003 ACM workshop on Privacy in the electronic society, 2003, 94–100.

[9] GurukulOnline Learning Solutions, `http://www.gurukulonline.co.in/index.htm`.

[10] Tamás Herendi, Uniform distribution of linear recurring sequences modulo prime powers, *Finite Fields Appl.* **10**(1) (2004), 1–23.

[11] Tamás Herendi, Construction of uniformly distributed linear recurring sequences modulo powers of 2 (*to appear*).

[12] J. Herrera-Joancomarti, Josep Prieto-Blazquez and J. Castella-Roca, A secure electronic examination protocol using wireless networks, Vol. 2, International Conference on Information Technology: Coding and Computing (ITCC'04), 2004, 263–267.

[13] Y. Levy and M. M. Ramim, A theoretical approach for biometrics authentication of e-exams, Chais Conference on Instructional Technologies Research, 2007.

[14] A. Shamir, How to share a secret, *Comm. ACM* **22** (1979), 612–613.

[15] Software Secure, Securexam, `http://www.softwaresecure.com`.

[16] Wasan S. Awad and Resala Al Adraj, On the effectiveness of e-exam, http://www.econf.uob.edu.bh/admin\Paper\285.swf.

ANDREA HUSZTI
FACULTY OF INFORMATICS
UNIVERSITY OF DEBRECEN
P.O. BOX 12, H-4010 DEBRECEN
HUNGARIAN ACADEMY OF SCIENCES
AND UNIVERSITY OF DEBRECEN
HUNGARY

E-mail: Huszti.Andrea@inf.unideb.hu

ATTILA PETHŐ
FACULTY OF INFORMATICS
UNIVERSITY OF DEBRECEN
P.O. BOX 12, H-4010 DEBRECEN
HUNGARIAN ACADEMY OF SCIENCES
AND UNIVERSITY OF DEBRECEN
HUNGARY

E-mail: Petho.Attila@inf.unideb.hu

(Received July 30, 2009, revised July 19, 2010)