Anonymous Balloting System for Evaluation of Students' Comprehension of Lecture

Marián Novotný, Peter Kempec

Institute of Computer Science
P.J. Šafárik University, Faculty of Science
Košice, Slovakia

PrimeLife / IFIP Summer School 2009 – Privacy and Identity Management for Life

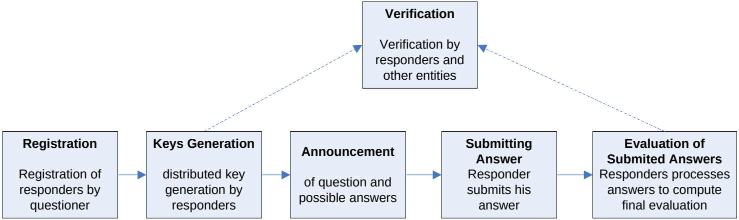
Motivation

- we present design, analysis and implementation of a tool for *education*
- for feedback about students' comprehension of topics of lecture
- teacher makes breaks for questions during the lecture, he prepares questions with answers, where exactly one is correct
- Students have certain time for choosing and submitting their answers
- the teacher obtains results, can repeat explanation of the topic

Security Requirements for Scheme

- Eligibility – only responders who are attending the lecture are eligible to submit their answers
- Privacy – in submission of an answer, the answer must not identify a responder
- Verifiability – responder should be able to verify whether his answer was correctly recorded and the final evaluation was correctly computed
- Accuracy – the scheme must be error-free

Phases of Protocol



Verification

Verification by responders and other entities

Registration

Registration of responders by questioner

Keys Generation

distributed key generation by responders

Announcement

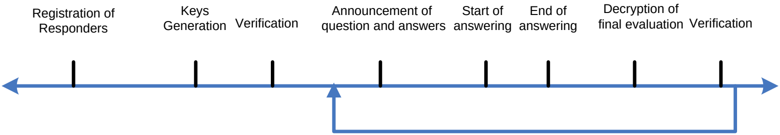
of question and possible answers

Submitting Answer

Responder submits his answer

Evaluation of Submited Answers

Responders processes answers to compute final evaluation



Registration of Responders

Keys Generation

Verification

Announcement of question and answers

Start of answering

End of answering

Decryption of final evaluation

Verification

Building Blocks

- robust threshold (t, n) ElGamal cryptosystem
  - for encryption of submission
  - homomorphic property $E(m_1) \cdot E(m_2) = E(m_1 \cdot m_2)$
  - decryption by cooperation of $t + 1$ shareholders – universally verifiable
- secure distributed key generation
  - without trusted dealer
  - generates shares of secret key for threshold (t, n) ElGamal cryptosystem
- ZK proofs of validity of answers
- proofs of equality of the discrete logarithms

Registration Phase

- authorization of responders is based on knowledge of a password *pass*
- the questioner
  - controls uniqueness of nicknames and network addresses of responders
  - can define the list of network address, nicknames which can be allowed
- responders registers public keys for encryption and signature

Registration Phase - Division into disjoint groups

- after registration responders $R_1, \ldots, R_N$ are randomly uniformly distributed into disjoint groups $G_1, \ldots, G_m$ with similar size n
- we uniformly distribute malicious responders into disjunctive groups
- we assume that the set of malicious responders is static
- the key of one group is used for encryption of answers
  - are rotated
  - members of other groups participate on verification

Secure Distributed Key Generation

- scheme DKG
- distributed generation of a pair of ElGamal public and shared private key for each group of responders
- to implement private channels between responders we use public keys, which are published in the registration phase
- runs in parallel in groups
- groups have the similar size n, it should finish in the same time
- the public key Pk_{G_i} of the group G_i is output in the clear, the private key is shared via threshold scheme, the shareholders publish their public shares

Submitting an Answer

- the public key Pk_{G_i} of the group G_i is used for encryption of responders answers (are rotated)
- the questioner publishes the question q_i and corresponding possible answers $a_1, \dots, a_{l-1}$, where exactly one is correct
- responder sends to the questioner a signed message
  - identification of the question q_i
  - encryption of the representation of the answer a_k
  - non-interactive version of the ZK proof that the encrypted answer is valid – one from l possible answers
- the questioner checks the signature and sends a signed receipt of the submission to the responder

Computing the Final Evaluation

- the questioner
  - 1. checks signatures and ZK proofs all submitted answers
  - 2. publishes the list of correct submissions with ZK proofs and signatures
  - 3. counts and publishes encrypted result $E_{Pk_{G_i}}(g^{result})$
  - 7. checks ZK proofs of decryption parts of the first $t+1$ shareholders and reconstructs g^{result}
  - 8. interpret and publishes *result*
- responders of the group G_i (the public key is used)
  - 4. checks whether his answer is published on the list 2
  - 5. checks signatures of all submissions and correctness of $E_{Pk_{G_i}}(g^{result})$
  - 6. cooperate on decryption of $E_{Pk_{G_i}}(g^{result})$ – publishes his part with ZK proof
- responders of other groups
  - 4. checks whether his answer is published on the list 2
  - 5. checks signatures and ZK proofs of all submitted answers
  - 9. verifies the decryption process

Informal Analysis of the Proposed Scheme

- Eligibility
    - registration of responder is based on the knowledge of the password and the controlling of uniqueness of network addresses and nicknames of responders
    - later responders use their private keys with corresponding published public keys for participating
    - only registered responders are eligible to submit their answers for the question no more than once
- Verifiability
    - the validity of answers is first verified by the questioner
    - later is verified by members of groups which do not cooperate on decryption of the final evaluation.
    - decryption process is verified first by the questioner who recovers the final evaluation and later by all responders

Privacy property (1)

- the used ElGamal system is semantically secure
- the traceability between the responder and his answer should be removed during the multiplication of submitted answers
- by cooperation of $t+1$ dishonest responders from the same group it is possible to decrypt an encrypted answer
- the protocol ensures the privacy of responders when the number of dishonest responders is at most t

Privacy property (2)

- the static set of malicious responders with b members
- If we have N responders, the size of the group is n, the expected number of malicious responders in one group is $exp = n \cdot b/N$

Example

Let $N = 120$, $b = 30$. We can divide responders into

- two groups, $n = 60$, $t = 29$, $exp = 15$, and the probability that 30 malicious responders are in the group is $\binom{90}{30}/\binom{120}{60} < 1/2^{36}$;
- three groups, $n = 40$, $t = 19$, $exp = 10$, and the probability that at least 20 malicious responders are in the group is $\sum_{i=20}^{30}\binom{30}{i}\binom{90}{40-i}/\binom{120}{40} < 1/2^{15}$;
- four groups $n = 30$, $t = 14$, $exp = 7.5$, and , and the probability that at least 15 malicious responders are in the group is $\sum_{i=15}^{30}\binom{30}{i}\binom{90}{30-i}/\binom{120}{30} < 1/2^{10}$.

Implementation

- we count the complexity of operations in protocol
- the most consumed operation from crypto-primitives in the scheme is modular exponentiation
- we built the prototype in Java
- We tested[1] two implementation of JVM – HotSpot and JRockit according to efficiency of computing of modular exponentiation
- one modular exponentiation in HotSpot takes 2.8 ms and in Jrockit 1.4 ms
- he slowest part of the protocol is DKG

---

[1] it was tested on Intel Core 2 Duo E6850 3.0GHz

Conclusions

- we designed scheme for anonymous balloting system for education
- we informally analyzed proposed scheme
- we computed complexity of parts of the scheme
- we built prototype in Java for testing on various computers in order to find appropriate value of the size of the group

Thank you for your attention