Verifiable Classroom Voting: Where Cryptography Meets Pedagogy

Feng Hao, Dylan Clarke, Carlton Shepherd*

School of Computing Science
Newcastle University
{feng.hao, dylan.clarke, c.g.shepherd}@ncl.ac.uk

Abstract. In this paper, we propose – and have implemented – the first *verifiable* classroom voting system. The subject of secure classroom voting has so far received almost no attention from the security community. Though several commercial classroom voting systems have been available, none of them is verifiable. State-of-the-art verifiable voting protocols all rely on finding a set of trustworthy tallying authorities (who are essentially cryptographers and computer experts) in the first place, and hence are completely unsuitable for classroom voting. Our system design is based on "self-enforcing e-voting" – a new paradigm that was first presented at SPW'12 (Hao, Randell and Clarke). A self-enforcing e-voting scheme provides the same End-to-End (E2E) verifiability as other e-voting schemes but without involving any tallying authorities. The removal of tallying authorities brings several compelling advantages in real-world voting scenarios – here, classroom voting is just one example. We have piloted the use of the developed verifiable classroom voting system in real classroom teaching. Based on our preliminary trial experience, we believe the system is not only scientifically valuable, but also pedagogically useful.

1 Introduction

Classroom voting is a powerful new pedagogy, which was first developed for the physics classroom by Harvard University's Eric Mazur in his influential book: "*Peer Instruction: A User's Manual*" [1], and subsequently extended by other academics to teaching mathematics and other subjects [2].

In this teaching technique, the teacher first poses a set of multiple-choice questions to a class of students, gives them a few minutes to discuss in small groups and asks them to vote for the best answers. Typically, a student submits the vote using a special hand-held device (known as the "clicker" [2, 3]) that sends radio frequency signals to a special receiver installed in the classroom. The receiver tallies votes in real time and displays the results over a projector, providing instant feedback to the students and the teacher alike. Several studies have reported success of using this technique to retain the students' attention, to

* The work was supported by Newcastle University Innovation Funds and partly by the ERC Starting Grant (SEEV), No. 306994.

increase the classroom interactions and to improve the student learning outcome [2, 3].

There have been several commercial classroom voting systems available, e.g., iclicker[1], TurningPoint[2] and eInstruction[3]. In particular, the TurningPoint voting system has been adopted and trialed by a number of universities in the UK, including Newcastle University. (The first author had an opportunity to participate in a demo of the TurningPoint system at Newcastle University. This research work was motivated by that experience.)

However, a notable limitation with TurningPoint – and in fact all existing classroom voting systems – is that the voting results are not verifiable. There is no means for students to check if their votes have been recorded and tallied correctly. The integrity of the results may be affected by many factors: e.g., hardware malfunction of the voting device, lost signal in the radio frequency transmission, software bugs, malicious attacks where an adversary tampers with the back-end software to arbitrarily modify the results.

One might question why we should care about the verifiability at all – if the tallying results turn out to be wrong, it is probably not too big a deal. After all, the classroom voting result is not as sensitive as that in political elections. However, we believe verifiability is still important. First of all, it provides confidence on the accuracy of the tallying results. If any hardware failure or a software bug causes the tallying procedure to go astray, the error in the result will be caught publicly if the system is verifiable. Second, though classroom voting questions are usually not sensitive, there are exceptions: for example, when the system is used as a module assessment tool to rate a lecturer's teaching performance. By taking security into consideration in all conditions, we can make classroom voting more widely useful. Finally, by making the system verifiable, students will have an opportunity to learn and practise the fundamental "trust-but-verify" principle in routine classroom voting. This can prove relevant when they later participate in more serious national elections.

Besides a lack of *verifiability*, there are other limitations with the existing classroom voting systems. They generally use proprietary devices as voting clients. This however not only imposes vendor lock-in but also creates serious logistical issues – simply transporting the physical devices into and out of the classroom can be a laborious task. In addition, they require installing a proprietary receiver in the classroom. This seriously limits the portability of the system, as voting is confined to only designated classrooms.

2 System design

In this section, we will propose a *verifiable* classroom voting system and show a concrete implementation. Our system addresses all the problems we explained above.

[1] `http://www.iclicker.com`
[2] `http://www.turningtechnologies.co.uk`
[3] `http://www.einstruction.com/`

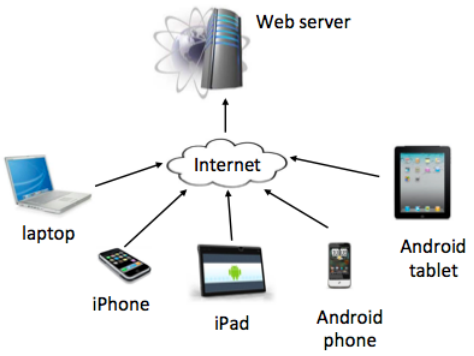


Fig. 1: Verifiable classroom voting system using mobile devices as voting clients

2.1 Overall architecture

Figure 1 shows the overall architecture of our system. At the client side, students use their own computing devices to vote. We have developed two voting clients – an Android app [5] and an iOS app [4] – to support voting from iPhone, iPad, Android phone and Android tablet. In addition we provide a generic web voting interface, so people with any other types of smart phones (e.g., windows 8, blackberry etc) or a laptop can still vote, as long as the device has a web browser and is connected to the Internet.

2.2 System configuration

There are three roles involved in the use of the system: administrator, coordinator and voter. The administrator is responsible to maintain the availability of the web server. A coordinator – usually a teacher – is someone who coordinates voting in a classroom. The system can accommodate many coordinators at the same time. Finally, voters are usually students in a class.

Classroom voting is arranged according to *voting sessions*. A voting session consists of a list of voting questions. We support four types of questions in the system:

1. Single-answer question: students can only choose one answer. (e.g., *which is the largest country in the world? A: Russia; B: China; C: America; D: India*)
2. Multiple-answer question: students can choose multiple answers. (e.g., *which of the following countries are members of Commonwealth? A: Singapore; B: India; C: Austria; D: Canada*)
3. Free numeric input question: there are no given answers and students are free to enter any numeric value (e.g, "*Enter the value of π to the two decimals*")
4. Free text input question: similar as above, except that the entered answer can be any text (e.g., "*Enter the name of the largest ocean on earth.*")

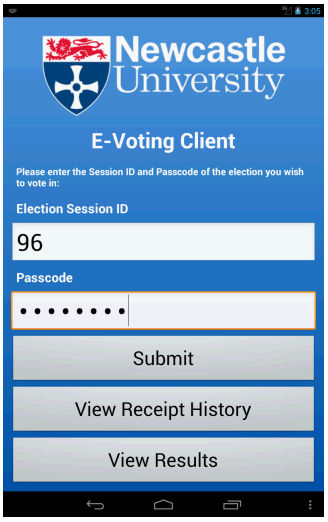


Fig. 2: Login screen for the voting client

The system is only verifiable when the voting questions are of the first two types (our verifiable voting protocol requires knowing the names of the candidates before the election). However, we still support the latter two types of questions, as we consider them useful features, even though the voting results cannot be verifiable in the cryptographic sense.

When a coordinator creates a voting session, there are a few options he needs to configure.

- Whether using a group passcode or individual passcodes.
- The maximum number of students in the class, denoted as N.
- The auditing factor F, which allows each student to audit a vote up to F times (by default $F = 5$)
- The security level L bits (by default $L = 128$)
- The length of the receipt R characters (by default $R = 5$)

A group passcode is a single passcode available to all students in the class. In this setting, the teacher informs students of a session ID and the group passcode, which are needed to log into the particular voting session (see Figure 2). However, one drawback with this authentication mechanism is that one student can vote multiple times by re-using the same passcode. In many circumstances, this is not an issue as there is no incentive for students to double-vote. However, in some cases when voting involves sensitive questions such as rating a lecturer's performance, a group passcode would be inadequate. Individual passcodes should be used instead.

In the individual passcodes scheme, each student is assigned a unique passcode. The web server first generates N random passcodes (recall that N is the maximum number of students in the class). The coordinator then prints out all

N passcodes, each on a paper slip. The paper slips are physically mixed up in front of the students before being distributed to the class. One student can only take one passcode. After voting is finished, the public bulletin board will show how many passcodes have been used. This number should be matched to the actual number of students in the class (e.g., based on a signed class attendance sheet). Any significant discrepancy would suggest something wrong (e.g., ballot stuffing), which demands further investigation.

2.3 Voting protocol

To implement the system, we adopt the Direct Recording Electronic with Integrity (DRE-i) protocol [7], which is under the category of "self-enforcing e-voting" protocols [6]. The DRE-i protocol provides the same end-to-end verifiability as other verifiable voting protocols but without requiring any tallying authorities.

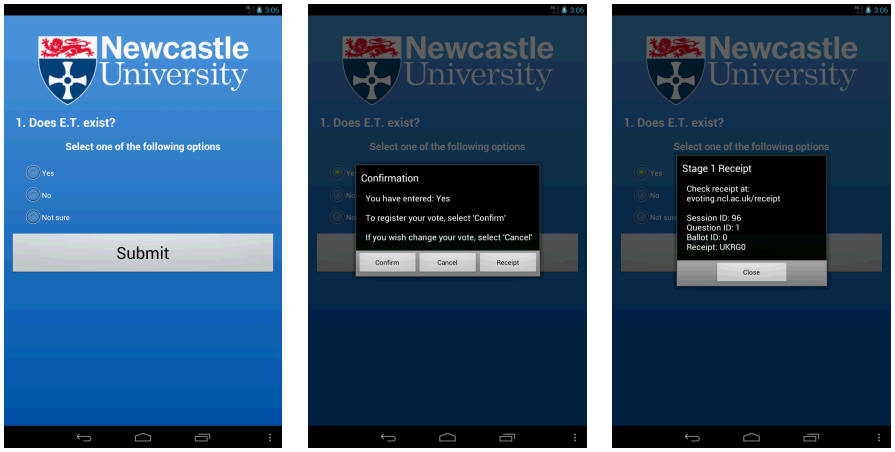
The protocol has three phases: setup, voting and tallying. The setup phase involves pre-computing cryptograms for all electronic ballots, as specified in [7]. Depending on the size of the class and the number of questions, this phase usually takes several minutes to complete.

The second phase is voting. Figure 3a shows the initial voting interface of the Android app for a single-answer question. To cast a vote, a student follows two stages: 1) selecting an answer; 2) confirming or canceling the previous selection.

In the first stage, the student makes a selection: let us assume he selects "Yes". In the next interface, the app shows that "Yes" had been selected and asks the student to "Confirm" or "Cancel" (Figure 3b). There is also a third button "Receipt", which leads to the display of a stage-1 receipt (Figure 3c). The student can verify the receipt by checking that the same content on the receipt has been published on the public bulletin board (a publicly accessible website).

The second stage handles the student's choice of "confirm" or "cancel". Suppose the student chooses to "cancel (essentially, this is to perform voter-initiated auditing [7]). The voting interface will show that the previous selection has been canceled (Figure 4a); the student can proceed to the next question (if any) or re-try the same question. There is also a button "Receipt", which leads to the display of the stage-2 receipt for the cancellation case (Figure 4b). A student can repeat the same cancellation operation up to F times (recall that the value F is configurable). On the other hand, if the student chooses to "confirm", the interface will show the vote has been casted (Figure 4c), together with a stage-2 receipt for the confirmation case (Figure 4d). Same as before, to verify the stage-2 receipt, the student simply needs to check that the content on the receipt matches that published on the public bulletin board. This requires no knowledge of cryptography. As long as all receipts are available on the public bulletin board, anyone with cryptographic knowledge and computing skills will be able to verify all receipts in a batch.

The third - and last - phase is the tallying process. When all students have casted their votes, the coordinator would end that particular voting session through a web interface. The voting results are immediately available. Figure



(a) Initial voting interface (b) After choosing "Yes" (c) Stage 1 receipt

Fig. 3: Stage 1 voting interface and receipt

5 shows an example of the tallying results in a bar chart. The same results are also available on the voting website, together with all receipts (i.e., audit data). We provide an open-source Java program on the voting website to facilitate any interested party to cryptographically verify the integrity of the results based on the audit data.

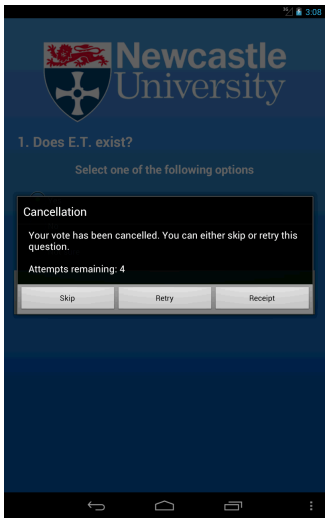
3 Trials

3.1 Usability trial

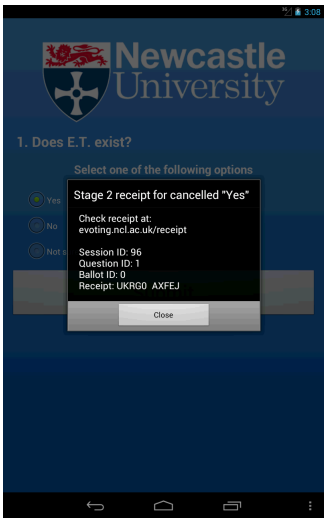
We conducted a voting trial workshop at the School of Computing Science, Newcastle University, on 3 September, 2012. The participants were mainly MSc students who had just submitted their dissertations. We provided pizza catering for all participants. With this workshop, we aimed at "three birds with one stone": to trial our newly developed verifiable classroom voting system; to serve as a farewell party for MSc students as many of them would leave shortly; and finally to give some MSc students a chance to present interesting results in their dissertation projects and let all participants vote for their favorite presentation using the classroom voting system.

There were in total around 40 participants who were mainly MSc students. Five students presented their dissertation projects, and afterwards we asked all participants to vote for the most "entertaining" presentation. In this case, the integrity of the voting result must be ensured, so we used the individual passcodes scheme as described in Section 2.

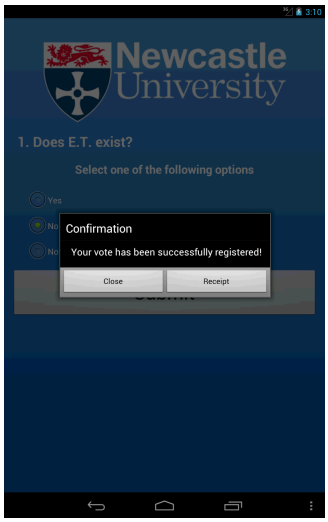
During the trial, participants had two ways to vote: 1) using an Android app [5] (version 1.0.0); 2) using a web interface at `http://evoting.ncl.ac.uk`. (At the time of the trial, the iPhone app was still under development, so iPhone



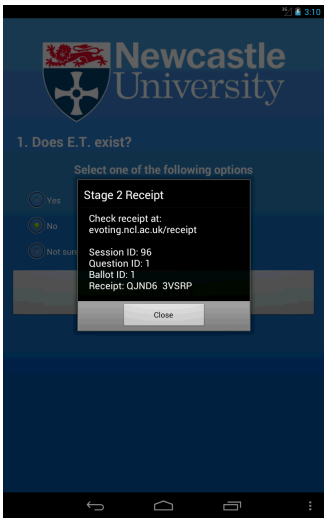
(a) Case A: user chose cancellation



(b) Case A: receipt for cancellation



(c) Case B: user chose confirmation



(d) Case B: receipt for confirmation

Fig. 4: Stage 2 voting interface and receipt

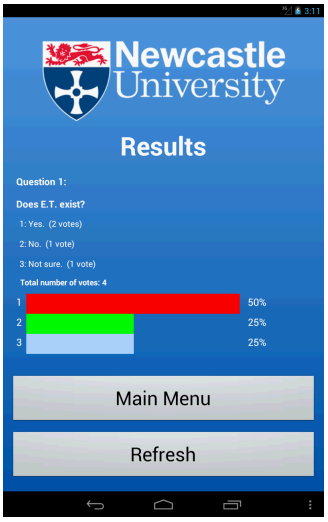


Fig. 5: Display of tallying results

users had to use a web interface to vote.) At the end of the workshop, we received 26 completed questionnaires, among which 17 participants voted through the Android app, and 9 through the web interface.

The feedback questionnaire consisted of 10 statements and respondents were asked to indicate their agreement or disagreement on a Likert scale from 1 to 5 (i.e., "strongly agree", "agree", "neutral", "disagree" and "strongly disagree"). The statements were as follows:

1. Joining a new session was easy.
2. I understood how to join a new session.
3. I understood how to answer questions.
4. Answering questions was easy.
5. I understood how to check the receipt.
6. I understood why I might want to check the receipt.
7. I felt confident that my answers had been recorded correctly.
8. I understood how to view the results.
9. Viewing the results was easy.
10. I felt my answer was sent anonymously.

We summarize the received 26 questionnaire answers in Table 1. There was no obvious difference in the answers between those voting through the Android app and those through the web interface, so we combine all answers in one table.

In general, the feedback was encouragingly positive. Participants generally found our verifiable voting system easy to use (see Table 1). However, some people expressed "neutral" opinions about the security of the system. Despite that we designed the system to be *verifiable* and we physically shuffled the passcodes to ensure *anonymity,* roughly half of the participants indicated they

were not sure whether the vote had indeed been correctly recorded and whether the voting was anonymous. These are useful lessons, which teach us that e-voting is not only a security problem, but also a subject of of psychology and voters' perception of security. We do not believe anyone should immediately accept a new voting system just because it is verifiable or has security proofs. But we do believe that, given a verifiable voting system with all important security elements accounted at the outset of the design, the public confidence in the new system – and their acceptance – will gradually grow.

Question	Strongly Agree (1)	Agree (2)	Neutral (3)	Disagree (4)	Strongly Disagree (5)	Average score *(nearest option)*
1	17	7	2	0	0	1.42 (***Strongly agree***)
2	15	9	2	0	0	1.5 (***Strongly agree***)
3	18	8	1	0	0	1.42 (***Strongly agree***)
4	21	4	1	0	0	1.23 (***Strongly agree***)
5	7	11	5	3	0	2.15 (***Agree***)
6	4	14	5	3	0	2.27 (***Agree***)
7	4	10	8	2	2	2.75 (***Neutral***)
8	10	10	5	1	0	1.88 (***Agree***)
9	11	10	3	1	1	1.96 (***Agree***)
10	3	10	10	3	0	2.5 (***Neutral***)

Table 1: Summary of the 26 received questionnaire answers

3.2 Pedagogical trial

Following the success of the usability trial in September 2012, we made several improvements to the Android app to make it more user-friendly. Also, we provided an iOS app [4] for those using iPhones and iPads to vote. In October and November, 2012, we first trialed the system in real classroom teaching on the "Cryptography" (BSc final year) and "System Security" (MSc first year) modules, in which the first author is the module leader. On 10 January 2013, at the last (revision) lecture of "Cryptography", the first author prepared ten revision questions for the class, gave students 15 minutes to discuss among themselves and asked them to vote for the best answers. At the end of the lecture, a student survey was conducted using the same voting system to collect the feedback. The survey questions and the tallied answers (within brackets) are summarized below:

Question-1 Does the voting make the lecture more fun?

Answers: Yes (26), No (2)

Question-2 Does the voting help you learn?

Answers: Yes (26), No (2)

Question-3 Do you find it useful to have a small group discussion before voting?
Answers: Yes (25), No (3)

Question-4 How do you think that the amount of voting used in this lecture should change?
Answers: More (10), Less (1), Remain the same (16)

Question-5 Do you recommend classroom voting for teaching the same module next year?
Answers: Yes (26), No (1)

The survey results clearly indicate the pedagogical value of the developed classroom voting system. The vast majority of the students in the class found the classroom voting system quite "fun" to use. We believe "fun" is a critical factor in learning – by making learning a fun process, we are able to better retain the students' attention in the class and improve their learning outcome. It is also worth noting that we used about a quarter of the time in a lecture (1 hour) for classroom voting. We were initially concerned if that was too much. But based on the feedback, 16 out of 27 expressed that was an adequae percentage; another 10 (nearly one third) students actually wanted more voting in the class. This is further evidence to show that students generally liked the system.

4 Conclusion

In the paper, we have presented a pioneering classroom voting system that is verifiable. This system serves as a good example to demonstrate the power of the underlying "self-enforcing e-voting" paradigm. Through putting the system into the real classroom teaching and collecting the student feedback, we show that the system has also demonstrated great pedagogical potential to enhance the students' learning experience in a traditonal classroom environment.

References

1. E. Mazur, *Peer Instruction: A User's Manual*, Prentice Hall Series in Educational Innovation, NJ, 1997.
2. K. Cline and H. Zullo, *Teaching Mathematics with Classroom Voting - With and Without Clickers*, Mathematical Association of America, 2011.
3. D. Bruff, *Teaching with Classroom Response Systems - Creating Active Learning Environments*, Jossey-Bass, 2009.
4. Link to the iOS app for the verifiable classroom voting application: `https://itunes.apple.com/us/app/id565080670`.
5. Link to the Android app for the verifiable classroom voting application: `https://play.google.com/store/apps/details?id=uk.ac.ncl.evoting`.
6. F. Hao, B. Randell, D. Clarke, "Self-Enforcing Electronic Voting," Proceedings of the 20th Security Protocols Workshop (SPW'12), Cambridge, UK, 2012.
7. F. Hao, M.N. Kreeger, "Every Vote Counts: Ensuring Integrity in DRE-based Voting System," School of Computing Science, Newcastle University, Technical report No. 1268, 2012.