Punchscan in Practice: An E2E Election Case Study

Aleks Essex
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, K1N 6N5
aesse083@site.uottawa.ca

Jeremy Clark
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, K1N 6N5
jclar037@site.uottawa.ca

Rick Carback
Dept. CS
University of Maryland Baltimore County
Baltimore, Maryland, 11022
rcarback@umbc.edu

Stefan Popoveniuc
Dept. CS
George Washington University
Washington, D.C., 11022
poste@gwu.edu

***Abstract*— This paper presents a case study of the E2E voting system Punchscan and its first use in a binding election. The election was held in March 2007 at the University of Ottawa for several offices within the university's graduate student association. This case study presents a walkthrough of the election and offers discussion as to how the voters and poll workers responded to the Punchscan system, with implications to E2E systems in general.**

I. Introduction

The University of Ottawa's *Graduate Students' Association / Association des étudiant(e)s diplômé(e)s* (GSAÉD) voted unanimously to adopt the Punchscan [6][3] voting system for their 2007 executive election. Among the reasons they cited for their decision was the desire to speed up the tally process, increase the integrity of election results, provide a means to identify double-voting, and, at an academic level, play a leading role in voting systems research.

The GSAÉD's chief returning officer (CRO) in conjunction with the Punchscan team proceeded to conduct one of the first end-to-end cryptographic (E2E) votes in a binding election. This paper presents the details and findings of this event and is organized as follows; Section 2 defines the notion of an E2E election, the criteria that make a voting system E2E, and briefly explains how the Punchscan voting system satisfies these criteria. Sections 3-6 will chronicle the steps taken leading up to, during, and after the election. In Section 7, we present our findings and discuss the reactions of voters and pollworkers to Punchscan. In Section 8, we offer concluding remarks with proposals for future directions in research.

II. E2E: End-to-End Cryptographic Independent Verification Voting Systems

In 2005, the American Election Assistance Commission (EAC) released a set of voluntary voting system guidelines [1] that includes a description of what they refer to as "End to End Cryptographic Independent Verification" (E2E) systems. According to the EAC, typical distinguishing features of an E2E voting system are as follows:

- A paper receipt is issued to the voter that contains information that permits the voter to verify that their choices were recorded correctly. The information does not permit the voter to reveal his or her selections to a third party.
- The voter has the option to check that his or her ballot selections were included in the election count, e.g., by checking a web site of values that should match the information on the voter's paper receipt.
- Such a system may provide an assurance not only that their ballot choices were correctly recorded (cast-as-intended), but that those selections were included in the election count (counted-as-cast).

A. Punchscan in the scope of E2E

The EAC found that the range of proposed E2E systems have points of commonality, and they attempted to summarize these in a list of properties. Here we present some of them and briefly explain how Punchscan does or does not exhibit their properties. Note that this is not a list of requirements for a system to be classified as E2E but rather a preliminary sketch of the typical of properties of these systems.

Property 1. *Voters' ballot selections are encrypted for later counting by designated trustees.*
A Punchscan ballot consists of two pages. The top page contains a list of contests and candidates with set of randomly ordered symbols beside the candidate names. There are holes in the top sheet that display a corresponding (but independently and randomly ordered) set of symbols. To vote on a Punchscan ballot, the voter observes the symbol appearing beside their chosen candidate's name, and locates the matching symbol in the holes. The voter then marks that hole with an implement such as a bingo-style dauber. The implement is sized slightly larger than the hole such that the ink mark will be made on both sheets. One of these sheets is destroyed in a cross-cut paper shredder. The remaining sheet represents the voter's receipt and is now "encrypted." [1] Only the threshold number of trustees (aka the election authority) have the ability to reconstruct the information contained on the destroyed sheet.

Property 2. *Voting will produce a receipt that would enable the voter to verify that their ballot selections were recorded correctly and counted in the election.*

[1] Since both sheets contain random but independent orderings of the symbols, possessing only one of the sheets does not give you information about the corresponding symbol on the other sheet.

Punchscan uses a robust audit procedure, including a process by which a voter can visit the election website and look up their ballot using the serial number contained on their receipt and verify what they hold in their hand matches what was recorded by election authority.

Property 3. *The receipt preserves voter privacy by not containing any information that can be used to show the voters selections.*
Because one of the sheets is shredded, and assuming that the ordering of symbols contained on that page were uniformly random and independent from the page that was retained (aka the receipt), then no information about the destroyed sheet is contained on the retained sheet, and therefore the vote cannot be guessed with any advantage.

Property 4. *No one designated trustee is able to decrypt the records; decryption of the records is performed by a process that involves multiple designated trustees.*
Punchscan employs a threshold based password scheme whereby a pre-designated number of trustees must correctly enter their passwords before the records can be reconstructed.

Property 5. *End to end systems store backup records of voter ballot selections that can be used in contingencies such as damage or loss of its counted records.*
Punchscan in its original form relies on voter receipts to reconstruct an election should the counted records be destroyed. However, as will be discussed in the next section, the implementation of Punchscan used in this case study expanded the originally proposed system to include a paper-based backup of the ballot receipts.

Property 6. *The backup records contain unique identifiers that correspond to unique identifiers in its counted records, and the backup records are digitally signed so that they can be verified for their authenticity and integrity in audits.*
The backup ballot receipts used in this election contained a serial number which matched the serial number of the ballot. While the ballot receipts themselves were digitally signed, the paper backups were not as it was agreed that the backup records were very unlikely to be needed. Should they have been used, they would have been published and thus their integrity would be ensured through voter verification. In future elections, consideration will be given to digitally signing the backups as well as the receipts.

Property 7. *The documentation includes extensive discussion of how cryptographic keys are to be generated, distributed, managed, used, certified, and destroyed.*
The source code for all the software used by Punchscan is open source and can be examined by anyone. Furthermore, the Punchscan team has attempted to document the underlying cryptography of the system through papers, presentations, and other documentation available from the Punchscan website [2].

Property 8. *Vote capture stations used in end to end systems must meet all the security, usability, and accessibility requirements.*
The security of the vote capture station in a Punchscan election is similar to that of a paper ballot voting station. Two additional security measures are taken: one is to lock the ballot to a clipboard and the second is to ensure a high-integrity paper shredder. The purpose of this case study, in part, is to examine the usability of Punchscan and will be discussed further in later sections.

Property 9. *Reliability, usability, and accessibility requirements for printers in other voting systems apply as well to receipt printers used in end to end systems.*
Punchscan can be easily implemented with inexpensive off-the-shelf equipment. As will be examined further in this case study, reliability and usability issues emerged. However Punchscan is largely hardware independent and could be adapted to use proprietary voting-dedicated equipment.

Property 10. *Systems for verifying that voter ballot selections were recorded properly and counted are implemented in a robust secure manner.*
Punchscan allows the voter to verify the proper scanning of their ballot at the polling station before it is cast, in addition to their ability to check the receipt online. The security of the Punchscan tallying process is dependent on well-studied cryptographic primitives [6] and no implementation vulnerabilities have been discovered to date.

III. Election Requirements

In consultation with GSAÉD, several requirements for this election were arrived upon. GSAÉD required there to be five polling stations located on the University of Ottawa campus, which were to be operated during normal business hours across a three day period. Two poll workers were to be present at all times. Being a fully bilingual university, the ballot was required to be worded in both English and French, and it was decided that this would be accomplished through a single bilingual ballot, as opposed to separate ballots for each language. The offices, candidate names, as well as French translations were provided by GSAÉD.

In addition to these basic requirements, it was agreed that several other additions over the basic Punchscan system would be implemented to provide greater security and robustness. They are as follows:

A. Ballot Receipt Digital Signatures

An E2E ballot receipt allows a voter to verify their vote was counted correctly, but also serve as proof when it was not. At it's core it protects the voter from mistakes or malicious activity on the part of the election trustees. However given

[2] http://punchscan.org

that Punchscan ballots are produced using low cost drilling and printing methods, fabricating a counterfeit receipt for the purposes of invalidating the election results is entirely feasible. Therefore we need a means to protect the election trustees (and in turn the election results) from malicious voters. This threat was addressed by the introduction of a barcode-based digital signature printed onto the ballot receipt at the time that it was cast. At the polling place, the ballot receipt is scanned and the Polling Place software detects and encodes the serial and mark positions into an XML file. The ballot receipt is then placed in a printer and green 'O' shaped overlays are printed over top of letters marked by ink daubs to confirm and 'lock-in' the location of the daub mark. Additionally 'X' shaped overlays are printed over unmarked positions. The marked positions are then digitally signed and printed on to the bottom left corner of the ballot receipt in a scanable barcode format, allowing the election trustees to establish the authenticity of a ballot receipt in the event of a future dispute.

B. Paper-based Backup

Punchscan records ballot marks by optically scanning and electronically storing the marks. Until this point however, GSAÉD had only ever employed traditional (i.e. strictly paper) ballot and had procedures for their handling, counting, and eventual disposal. Because Punchscan is an optical scan system, and because the only surviving physical portion of the ballot (i.e. the receipt) is retained by the voter, the concern was raised by GSAÉD over keeping a strictly electronic record of the election. For example, their election procedure stipulates a date (after the election results have been ratified) by which the election ballots are to be shredded. Other concerns expressed by the council included power outages at the polling stations, accidental deletion, hardware/hard-disc failure, and other electronic attacks. The Punchscan team consulted with them at length about a means to provide a paper-based ballot backup. It was agreed that a small space in the lower right-hand corner of the ballot was to be reserved for printing an encoded copy of the marked positions, which was performed at the same time as the overlays/digital signature. Using a pair of scissors, the poll worker cut off and retained this corner before returning the receipt to the voter. These corners were collected and retained by GSAÉD functioning in the stead of actual ballots.

C. Electronic Pollbook

In past GSAÉD elections, voters were authenticated at the polling place using local paper copies of the voters list. Voters' university-issued graduate student cards were used as the primary credential to vote. Although this scheme allows the eventual identification of students who cast more than one ballot (so-called "double voting") at different polling stations, it does not allow a means by which to invalidate the double votes from the tally. GSAÉD had experienced problems with double voting in past elections and had employed solutions such as the use of walkie-talkie radios to communicate the student numbers of voters in a real-time fashion. However the local copies of voter lists were still being updated manually which is inefficient given a list of 4000 eligible voters and 5 polling stations. The Punchscan team offered and implemented an alternative solution: a centralized electronic pollbook. A MySQL database of eligible student numbers was created and maintained on the `Punchscan.org` server. The poll worker interface was through any standard web browser and internet connection. Using the campus wireless network, the poll workers were able to log into a password protected, SSL secured web form and perform voter list queries and updates. Querying a voter's student number would return one of three possibilities: a message indicating that the student number was not found in the database, an option to mark that individual as having voted, or an alert that that student had voted already.

D. Ballot Clipboard and Lock

Although it was agreed to be unlikely to occur in this election environment, the Punchscan team decided to test a countermeasure to the vote buying/intimidation technique known as "chain voting." If a chain voting perpetrator is able to remove one uncast ballot from a polling station without detection, they could, using the cooperation of subverted voters, "require that the subverted voter take the ballot to a polling place, exchange the pre-marked ballot for the blank ballot issued to that voter at the polling place, and return the blank ballot to the perpetrator to enable the next cycle" [5]. To increase the difficulty of someone being able to remove a ballot from the polling station without detection, the Punchscan team developed special clipboards that employ a *Medeco* plunger lock affixed to the clipboard. When the locking mechanism (located in the upper-right corner) was depressed, the lock's bolt would extend down into a hole in the clipboard. Each ballot also had a hole drilled in the top right corner. Before the voter is presented with an unmarked ballot it is placed on the clipboard. When the lock cylinder is depressed, the plunger would seamlessly pass through the ballot and clipboard locking it into place. When the voter returns to cast their vote, the poll worker checks to ensure that the ballot is still locked to the clipboard and that the paper corner is not torn. Additionally the clipboards were fitted with a cover to obscure from view the unique information printed on the ballot.

IV. Preparing for the Election

There were approximately 4000 registered graduate students at the University of Ottawa, all of which were eligible to vote. Voter turnout in past years was quite low (about 5%), and even with optimistic estimates it was not realistically expected to exceed 10% turnout for this election. Nevertheless we decided to err on the side of caution, resolving to print 3000 ballots.

A. Ballot Design

There are two steps for creating a Punchscan ballot. In the first step the layout of a Punchscan ballot is created using what is referred to as *ballot template* software, which can be done using almost any standard desktop publishing application. In

addition to whatever standard text and graphics are placed on the ballot, the only requirement for the ballot template is the ability to mark certain locations with simple colored shapes. For this election, we used Microsoft Publisher to create the ballot template. In the second step each individual ballot is generated containing its own unique information using what is referred to as the *ballot authoring* software. The ballot authoring software searches the ballot layout (created by the ballot template software), and locates the colored shapes, overlaying the appropriate unique information at these locations and exporting the completed ballots in Portable Document Format (PDF). The specification calls for colored disk shapes to represent items that will appear on the top page, and colored ring shapes to represent items that will appear on the bottom page. These positional markers are used to place four categories of items on the ballot:

1) Ballot serial number (top and bottom pages).
2) Letters appearing beside candidate names (top page).
3) Letters appearing through the holes (bottom page).
4) Scanner alignment marks (top and bottom pages).

The ballot authoring software differentiates the respective categories by color. Once the ballot layout has been finalized, a PDF version is loaded into the Punchscan ballot authoring software. This software uses image recognition to determine the X-Y coordinates of all of the positional markers. Later when ballots are being prepared to be printed, the unique information contained on each ballot (serial numbers and permutations) is read from an XML file and placed on the ballot in the appropriate locations.

B. Ballot Manufacture

Ballot manufacture refers to the process of creating holes in the appropriate locations of the physical ballot stock. However drilling holes happens to be a reasonably simple, inexpensive and available process in this application. When ballot authoring software recognizes the ballot template markers (discs and rings), it produces a special XML formated "drill file" which is used by a machinist to accurately drill holes in reams of 500 sheets of paper at a time using a computer-positioned vertical mill. As per their election procedure, the GSAÉD council met one week before the election to approve the ballot wording and layout. That evening the Punchscan team produced the drill file and took it to the machinist who drilled 3000 blank ballots. The ballots were then shipped and arrived in Ottawa the next morning. At 19 holes per ballot and 6 reams of paper, the setup and drilling was completed in under half an hour. Given the machine and operator rate of $75/hour, 3000 Punchscan ballots were produced at a unit cost of $0.01, which is about half the cost of the paper component. Using standard quality white 8.5x11" office paper costing $0.01 per sheet, the overall until cost of an unprinted Punchscan ballot was $0.03.

C. The First Meeting of the Trustees

The security of a Punchscan election rests entirely in a shared secret key, which is distributed among the trustees and not wholly known by anyone (similar to Shamir's secret sharing scheme [9]). Ideally the trustees form a zero-sum relationship (i.e., political adversaries) and have no incentive to collude with each other. The Punchscan team implemented a threshold based secret sharing scheme using standard hash and symmetric cryptographic primitives based on the passphrases chosen by the election officials. This scheme allows a predetermined minimum threshold of election officials to convene and regenerate the master election key at each of the subsequent meetings. However given the stakes in this election were relatively low, the option of distributing the key was not exploited and the CRO acted as the sole trustee.

At the first meeting (the election specification), the CRO supplied her passphrase and the open source software generated all the election data from it. This includes the association of the ballot serial numbers with a pseudo-randomly generated permutation of letters. Additionally, the information needed to reconstitute, or "decrypt" the vote after one half of the ballot is destroyed is also generated. This "decryption table" will be partially revealed during the post-election audit. However to increase the probability of catching tampered ballots during the audit, more (independent) decryption tables can be created, each with a unique keystream. For this election we used 10 such tables. After this secret information is created, each piece is run through a bit commitment function. These commitments are publically posted. Later as some of this secret information is revealed during the audits, it can be run through the same commitment function (by anyone) to ensure it matches the original commitments, establishing that the results have not been modified. These commitments, and all the election data referred to throughout this case study, are available from the Punchscan webpage [3].

D. Pre-Election Audit

After all ballots had been (digitally) generated and committed to, half of the ballots are chosen at random to be audited. Because we ultimately wish to print 3000 ballots, during the election specification we (digitally) generated 6000 ballots. The audit checks that the unique information on the ballot matches the commitment published after the first meeting, and that the decryption information for the ballot in the database is properly formed (i.e., it contains the proper permutation to return the ballot to its canonical form). It is designed as cut-and-choose protocol to catch fraudulent commitments or decryption information. Auditing a ballot involves publically posting that ballot's secret information and allowing any interested party to verify it matches its corresponding commitment. At that point, those ballots are considered "spoiled" and are not used in the election.

The integrity provided by this audit could be compromised if the selection of ballots could be known in advance of committing to the data in the first meeting, or if the outcome of the selection could be rigged. Therefore for the audit to be effective, it must be infeasible for anyone to know during the first meeting of the trustees which ballots will be audited

[3]http://punchscan.org/gsaed/

with any advantage. The Punchscan team followed the random selection procedure presented in [2] which uses volatile stock market data to make the ballot selections. Stock data is well-suited for this task because it is unpredictable *a priori*, yet easily verified *a posteriori*. The stock portfolio comprised of 32 stocks—a subset of the Wired 40 [7] companies traded on NASDAQ. These indeces are known to be sufficiently volatile that we can reasonably expect at least 1 bit of entropy from each index. Using this portfolio to seed a pseudo-random number generator (PRNG), we can produce a selection stream[4] that ensures each ballot will be selected with a 50% probability regardless of how the market will close.

The pre-election audit took place in two stages. First the CRO entered her passphrase to open and publish the information about the selected ballots. After this data had been published, the second stage was to actually audit this information against the published commitments from the first meeting. This comparison was performed using an open-source auditing tool which checks the ballot against its commitment. An open source implementation of a pre-election auditor is provided by the Punchscan team and available from the Punchscan website. The audit report generated by the program determined that all the revealed ballots matched their commitments. It is important to note however that the audits form the core of the "independent verification" and is meant to be as available and performable by anyone interested. This audit does not require the software provided by Punchscan—it follows an open specification and can be independently implemented by any interested party.

E. Printing Ballots

After the completion of the pre-election audit, the CRO met with the Punchscan team to print the remaining 3000 ballots on paper. Under the supervision of the CRO, the ballots were printed in parallel on six inkjet printers. Three printers were loaned by Hewitt Packard to the University for the election (two HP-K5400's and one HP-K550's), and three more were provided by the Punchscan team (HP-K550's). Although strictly speaking the printing is not dependent on a particular model of printer, a few considerations need to be taken. Firstly, the ballots contained colored scanner-alignment marks. Secondly the ballots must be accurately aligned so that the serial number and letters on the bottom page show through the holes. This is usually resolved through trial and error, and using the ballot authoring software to introduce a compensation in the ballot PDF. Thirdly it was discovered for certain models that printing would always halt when the print head reach a drill hole. Finally paper feeding was an issue because the drilling process creates a small cusp in the paper around the drill hole causing the sheets to "stick" together as they were fed into the printer. This often results in numerous pages being scrapped. However most of these issues had been previously addressed and the printing process for the 3000 election ballots took approximately one hour. Upon completion, the ballots were placed into boxes, sealed, and signed along the seal by the CRO.

[4]If the bit in the selection stream at position X is 0, the ballot with serial number X to be audited. If the bit is 1, then the ballot is left unopened and used in the election.

F. Poll Worker Training

On the night prior to the election, the Punchscan team hosted a two-hour poll worker training program. For the first hour, an introduction to Punchscan was given explaining the theory behind the system, outlining the polling place procedures (which was also provided in written form), and answering questions. The second hour provided hands-on experience with the polling place equipment. The poll workers were provided with mock ballots to vote on and they practiced the procedure of scanning and casting the ballots.

V. Conducting the Election

A. Contests

The GSAÉD election consisted of six contests. Five were positions for office and one was a referendum. Of the five office positions, only one was contested. However the uncontested officials still needed to be confirmed by a majority of voters according to the GSAÉD regulations. Thus these four contests consisted of a 'yes' or 'no' option. The contested office position had two candidates, and the referendum also consisted of a 'yes' or 'no' option. In essence, the election consisted of five contests, all of which had two candidates/options.

B. Polling Station

A polling station consisted of the following equipment:

- 1 full computer system,
- 1 scanner,
- 1 printer,
- 1 shredder,
- 1 clipboard and lock,
- 2 bingo daubers,
- 1 polling booth,
- 1 ballot box (for paper backups).

For this election, the polling station equipment was provided by the Punchscan team to GSAÉD at no charge. However we will estimate the cost of purchasing this equipment: computer ($200[5]), scanner ($100), printer ($200), shredder ($50), clipboard and lock ($20), daubers ($2), polling booth ($10), and ballot box ($10). This puts the marginal cost of a polling place at around $600. While this is a crude estimate, we include it to contrast Punchscan from the cost of purchasing a DRE.

C. Issuing a Ballot

When a voter entered the polling station to vote, the poll worker performed the following steps:

1) Looked up the voter in database using student card.
2) Placed the top sheet face down on the clipboard assembly cover.
3) Placed the bottom sheet face down on the clipboard assembly cover.

[5]Only a basic computer capable of running Java Runtime Environment (JRE) is required.

4) Closed the clipboard assembly cover so that ballot was facing up, and with only the serial number showing.
5) Checked that the serial numbers on the top and bottom sheet matched.
6) Locked the ballot in place and gave the clipboard to the voter.

D. Marking a Ballot

Upon receiving the ballot and clipboard, the voter retreated to the polling booth and performed the following steps:

1) Opened the clipboard assembly to view the ballot.
2) For each contest: located the chosen candidate/option to vote for;
3) Noted the symbol beside the candidate/option;
4) Located the hole containing the same symbol;
5) Marked the hole by firmly daubing it.
6) In view of the poll workers, removed and shredded either the top or bottom sheet.
7) Returned the clipboard (with the remaining sheet still locked in) to poll workers.

E. Casting a Ballot

Upon receiving the clipboard, the poll workers performed the following steps:

1) Unlocked the clipboard and removed the remaining ballot sheet.
2) Scanned the ballot sheet and displayed detected mark positions to the voter on the computer screen.
3) If approved by the voter, cast the ballot.
4) Marked the voter as voted in database.
5) Placed the sheet in the printer for the overlays, digital signature, and paper backup.
6) Cut the paper backup off the bottom corner of the sheet.
7) Gave the sheet to the voter as a receipt.
8) Placed the paper backup in the ballot box.

VI. After the Election

A. Election Results

After the polling stations were closed, the Punchscan team, the CRO, and a scrutineer assembled to tally the results. The encrypted ballot receipts were uploaded to the Punchscan server. The CRO then entered her passphrase to decrypt and tally the results. With 154 ballots cast, the results are summarized in Table I.

B. Post Election Audit

After the completion of the election, the tallying procedure was audited. The tallying procedure takes the ballot receipts as inputs, applies an inverse permutation, and returns a list of the candidates that were voted for using a decryption table. In order to preserve voter privacy, the tallying function shuffles the order of the data in the table to decorrelate a specific inputted receipt from a specific outputted result. If the complete decryption table were opened for inspection, votes could be traced back to individual ballot receipts. As a result, tallying is broken into two sequential phases. In both phases, a permutation is applied to the receipt information and then the data order is shuffled (see [6] for details). Revealing the results of either of these phases (i.e., halves of the decryption table) without revealing the other preserves privacy and allows again for a cut-and-choose protocol.

The 10 decryption tables were each partially reveled, again depending on selections made by stock market data in similar form to the pre-election audit. The CRO entered her passphrase in order to recover the data to be published. The published data was then audited with a software tool provided by Punchscan—the audit verifies that data in the decryption table matches the commitments published after the first meeting and that the permutations were applied correctly. As with the pre-election audit, the auditing is software independent. In whole, this procedure guarantees the accuracy of the tally to a high probability.

VII. Results and Discussion

A. Technical Issues

There were a number of issues that the pollworkers experienced. For example, of the 154 ballots cast, only 145 were recorded in the electronic pollbook. This means that nine instances occured in which ballots were cast without the voter's eligibility being verified. However this is a matter of pollworker procedural compliance, and therefore is no different than any existing voting system in this regard.

More interesting is how the pollworkers reacted (uncued) to technological failures. On the software end, there were several instances in which either the polling place or electronic pollbook software froze. On the hardware end, there were several instances in which the printers jammed or the wireless internet signal was lost. During these instances we found that all the pollworkers undertook to create a handwritten account [6] of the cast ballot, interestingly in some cases without having been instructed how to do so. In the absence of the printer to create overlays and digital signature, the poll workers signed the ballots manually. These handwritten records were later recorded electronically, and transcription errors would be subject to detection through the online receipt verification check. This demonstrated an unexpected robustness of Punchscan against a host of denial of service scenarios.

B. Psychological Acceptability

The act of marking a Punchscan ballot by matching shuffled letters has been referred to as "indirection". Opinions of voters varied widely over the degree to which indirection was an issue. Some voters actually claimed that Punchscan was "no harder" to mark than a traditional ballot, whereas others found it irritating. Here the voters were more concerned with their personal experience than in their ability to correctly transcribe their voting intent on a Punchscan ballot. In order to gague the usability of Punchscan however, the rate of occurance of intent transcription errors would have be measured. However in the

[6]Serial number plus a numeric code for each mark (or absence of mark) made by the voter.

TABLE I
RESULTS

	President (Uncontested)	VP Communications (Uncontested)	VP Internal (Uncontested)	VP Services (Uncontested)	VP Finance (Contested)	Referendum
Yes / Candidate A	118	120	117	121	81	98
No / Candidate B	28	21	31	26	54	43

context of voting systems, a proper user-study would require the ability to maintain a linkage between individual voters and their vote in order to gague their ability to successfully cast their vote as they had intended. Obviously this is not possible during a live secret-ballot election. However the reactions of the participants during this case-study do point to important questions a user study should undertake to answer. The design principal of psychological acceptability can be applied to voting technology [8], postulating that the security mechanisms of the system should be congruent with the voter's mental model of the system. Our case study confirmed that the security mechanisms of Punchscan were not well understood by the voters.

C. Ballot marking

The fact that the order of the letters on the ballots were randomized was not clearly indicated on the ballot, leaving the voter with only their intuition for forming a proper understanding of why a receipt does not contain adequate information for determining their vote. The randomized lettering was likely the predominant barrier in understanding. However once this was explained to the voters, many understood in a flash of insight. However understanding did not necessarily precipitate willingness and many voters indicated their sense of burden with the voting process. What was clear from the election was that voters did not intuitively understand the ballot marking process. Most became satisfied of the requirements after a verbal explanation. However the quality of explanations and (therefore their effectiveness) varied between the pollworkers. As of yet, there still is not an standard script on how best to educate new voters.

D. Shredding

Also at issue was the process of shredding one of the ballot sheets. Concern was expressed by several voters over what they perceived as the destruction of their vote. Many were also confused by being given the option to shred either page, in some cases asking several times to ensure they understood correctly. This might be attributed to the fact that typical administrative tasks (such as filling out a government form) do not offer options as to how the task should be completed. Therefore we might reasonably conclude the choice component of ballot completion is contrary to the voter's mental model. The original design intention was for voters to retain top and bottom sheets with near equal probabilities to reveal ballot tampering. Interestingly however given the choice approximately 85% of the voters chose to retain the bottom page as their receipt. On the third day of polling, after this trend had been established, the Punchscan team questioned voters leaving the polling station why they chose the sheet to shred that they did. The majority explained that because the ballot was still locked to the clipboard, they found that ripping off the top sheet to be the easiest way to complete the action. However in a subsequent Punchscan election, held at the 2007 Computers, Freedom and Privacy (CFP) conference, where clipboards were not used, approximatly 85% of the 36 voters still chose to keep the top sheet. The implication of receipt skew makes it less probable for attacks to be detected during the post-election audit. Although the degree of receipt skew in this election didn't significantly affect a reasonable assurance of integrity, it is something that should be better understood for future elections.

E. Conveying the Purpose of Punchscan

Another source of dissonance between the ideals of Punchscan and voters' mental model concerned the purpose of ballot receipts. Many did not realize they were going to receive a receipt and wanted to leave immediately after marking their ballots. Furthermore, when it was explained to them that would receive a receipt, and what the receipt allowed them to do with respect to independent verification, a number of voters were still indifferent. It is important to the integrity of the election that voters at least take the receipt—a left receipt means that receipt will not be checked, opening a window of opportunity for an attacker to modify that ballot.

Voters' reactions demonstrated that E2E has a long way to go in making its benefits clear to voters. Its not necessary for the voters to have a perfect mental model of the system. It is difficult to gauge how fundamental these problems are to Punchscan, and E2E systems in general. They could be simply rooted in the novelty of this kind of system, in which case voter education would go a long way to resolving these issues. Our case study does indicate that as some voters became aware of the verification ability afforded to them with Punchscan, they were accepting of the extra measures required for voting.

The ballot receipts were hosted on the Punchscan webserver and the server logs show that the image files of 83 of the 154 ballots got at least one hit. Whether this means they were actually checked against the receipt cannot be determined from available data, but it is suggestive of an interest by voters in checking their ballot receipts online.

F. Poll Worker Feedback

The poll workers opinions varied as to the difficulty of the ballot casting procedure. All polling stations encountered minor computer glitches at some point during the election, as well as the occasional printer jam. The polling place software also experienced some buggy behavior after being

left running for long periods. However recalling from earlier, on the occasions that there were technical complications, the poll workers were able to record the ballots manually. Their comments include the need for "a more detailed explanation of the voting process on the ballots." One poll worker explained to us that a voter daubed the serial number holes, which in turn caused a serial number recognition error at the polling station. Also echoing the comments of voters was the need for an explanation as to "why the voters have to destroy a page." They also suggested visual instructions posted inside the polling booth, a brochure explaining Punchscan's "security features."

Perhaps the most widespread concern however was the time to cast. Obviously 154 voters across 5 stations and 3 days allowed for the poll workers to take their time casting votes. However, as one poll worker described it, "I don't know how it will work if we have people lining up." We did not conduct stopwatch-timed trials during the actual election, however generally we found unlocking the clip board, scanning and casting the ballot, and then printing overlays took around 60 and 90 seconds. On the other hand, at the subsequent CFP election we used Punchscan in what we call "mail-in" mode, which in this context meant that instead of shredding one half of the receipt, the voter kept one half, and gave the other half to the poll worker. The collected receipts are scanned at a later date. Therefore the time to cast for this variant was the time it took to hand the poll worker one of the ballot sheets (effectively zero). However the mail-in version does not offer the same degree of privacy or integrity as the full-scale version used in the GSAÉD election.

VIII. Conclusion

Ultimately this election was successful in the sense that its participants (the Punchscan team, GSAÉD, the voters) were sufficiently satisfied in Punchscan's ability to fairly and accurately conduct this election. The elected candidates became ratified by the council, and no challenges were made against the election results. The election was modest and not hotly contested, but it is our position that this case study represents an important milestone for Punchscan and E2E elections in general. Stu Feldman has outlined a roadmap for technical maturity (as quoted in [4]):

1) You have a good idea.
2) You can make your idea work.
3) You can convince a (gullible) friend to try it.
4) People stop asking why you are doing it.
5) Other people are asked why they are not doing it.

The recent and current research in cryptographic voting demonstrates that E2E has reached the first step. The development and release of E2E systems, like Punchscan 1.0, proves that the second step has been reached by a smaller set of proposed E2E systems. This case study demonstrates that Punchscan has advanced E2E another step in the direction of technical maturity. Through the study, we have found that further work needs to be done to address the fourth step, particularly in the area of usability studies. It is our hope that E2E systems will eventually reach the fifth and final stage, and become the de facto method of voting in public elections.

A. Future Direction

As outlined, one solid avenue for future work is in user-studies. However it seems evident from this study that the future direction of E2E elections should focus instead on decoupling the integrity aspects from the voting process itself. Voter response suggests that the indirection inherent in marking the ballot was not an insurmountable obstacle for the voters polled. However given that the voters were provided with one-on-one instructions, as well as the unrepresentative educational background of the sample population (i.e., graduate students), there remains a limited indication that indirection in marking will produce successful results for the general public. On these grounds we hope to see a thorough user study be performed on the effect of indirection on intent transcription errors. That said, our own future work into E2E systems will seek to investigate the possibility of providing integrity without the use of ballot indirection. Given that many of the voters reacted with either indifference or contempt to their new ability of independent verification, we would suggest that future systems consider making the receipt issuing process fully available but non-mandatory. In that sense one might regard the future of E2E systems acceptable by the voting public (and therefore realizable) once the E2E benefits can be isolated and modularized within the actual voting process. Although such schemes may not offer the same degree of integrity as Punchscan, they will be far more palatable to voters, and ultimately more likely to be used in real-world elections.

Acknowledgments

The authors wish to thank David Chaum, chief returning officer Angelika Welte, and GSAÉD for making this election possible.

References

[1] Voting system performance guidelines. *2005 Voluntary Voting System Guidelines*, Volume 1, United States Election Assistance Commission, Version 1.0, 2005.

[2] J. Clark, A. Essex, and C. Adams. Secure and observable auditing of electronic voting systems using stock indices. *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2007*.

[3] K. Fisher, R. Carback and A.T. Sherman. Punchscan: introduction and system definition of a high-integrity election system. *Workshop on Trustworthy Elections (WOTE) 2006*.

[4] D. Geer. Technical maturity, reliability, implicit taxes, and wealth creation. *;login: The magazine of Usenix & Sage*, 26:8, December 2001.

[5] D.W. Jones. Chain voting. *Workshop on Developing an Analysis of Threats to Voting Systems*, National Institute of Standards and Technology, 2005.

[6] S. Popoveniuc and B. Hosp. An introduction to Punchscan. *Workshop on Trustworthy Elections (WOTE) 2006*.

[7] S. Reiss. The Wired 40. *Wired*, 14.07, July 2006.

[8] J. Saltzer and M. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63:9, 1975.

[9] Adi Shamir. How to share a secret, Communications of the ACM, 22(1), pp 612-613, 1979.