Name: Benjamin Ryan Goldsworthy BSc (Hons)

Student ID: 32098584

Dissertation Title: Mobile-based auditing of the DẼMOS 2 e-voting system

Module: SCC.420 MSc Dissertation

Date: 2018-09-20

Date:

Signed:

Abstract

DẼMOS 2 is a proposed e-voting system that is end-to-end verifiable, meaning that its results can be verified for correctness from end-to-end. This is clearly an important characteristic of any e-voting system that aspires to widespread adoption, but it requires that voters and third parties possess the means to perform such verifications. It is in this area that the current DẼMOS 2 implementation is lacking, and this project is an attempt to rectify this through the development of a mobile phone app. for Android devices. The design of such an app. is here presented, but the implementation remains incomplete. Despite this, an evaluation of the app. as it currently stands is presented, followed by a detailing of avenues of further development. Ultimately, however, the supposed benefits of e-voting are brought into question.

Acknowledgments

I would like to acknowledge the help of my supervisor, Dr Bingsheng Zhang, over the course of this whole project, Vincent de Almeida for his assistance with understanding and developing the DẼMOS 2 software and Zengpeng Li for his presentation on the Estonian election system architecture.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter contains an introduction and outline of the aims of the project.

1.1 Problem Specification

E-voting is becoming increasingly popular both as a addition to, and a replacement for, traditional voting methods, both at the lowest (e.g. club executive elections) and highest (e.g. national elections) levels. In order to maintain trust in the electoral process, these systems must be considered secure and the results they generate considered accurate. One way of demonstrating this is to demonstrate the property of E2E verifiability in the system design, which ensures that one can verify various properties about the election, such as that the ballot sent to the election server contains the vote cast by the voter and has not been modified en route. Providing interested parties the theoretical assurance that they can verify aspects of the election is one thing; without the means to actually do so, any such system will remain of interest only to the world of academia.

One such proposed E2E-verifiable system is the DẼMOS e-voting system, and the subsequent DẼMOS 2 E2E-verifiable Internet voting (E2E-VIV) system. Whilst this system design has been proven to be E2E-verifiable, there exists as yet no independent auditing software to actually enable this verification in practice, nor even the functionality within the current version of the system needed for such a piece of software. So that DẼMOS 2 can begin to be adopted by regular, non-academic users, this gap must be filled. This dissertation shall detail the attempt to fill this gap, through the development of a smartphone application that will allow any interested party to verify that an election conducted via DẼMOS 2 has been conducted fairly, whilst still maintaining the secrecy necessary to deter vote coercion or selling.

1.2 Project Aims

The aims of this project are as follows:

- to add functionality to DẼMOS 2 to allow a voter to check that their vote has been recorded-as-intended;
- to through adding functionality to DẼMOS 2 and the development of a separate Android mobile application, allow a voter (or a third party to whom they have delegated the task) to verify that their ballot has been cast-as-recorded;

- to as before, allow a voter or third party to verify that the final calculated result of a given election has counted-as-cast all given ballots;
- to ensure that these three features are implemented in such a way that ensures a voter's privacy is protected;
- to ensure that these three features are implemented in such as way as to deter voter coercion, vote buying and vote selling; and
- to document the ways in which which these features are implemented so that future developers are able to easily develop their own independent election auditing tools.

1.3 Overview

This dissertation is divided into 8 sections. Following the present introductory section is an account of the historical development of voting, the recent rise of e-voting and a number of real-world case studies. In the third section, the requirements of an election auditing app for mobile are listed. In the fourth, the design of the app, including justifications of the attempts made to address the aforementioned requirements, is presented. In the fifth, a report on the technical implementation of the design are shown. In the sixth, the process of using the app is detailed. In the seventh, the results of testing on the finished application are demonstrated and the success of the project evaluated. Finally, the paper concludes with a reflection on the project as a whole and proposals for potential future avenues of research and development.

Chapter 2

Background

This section provides a background to the topic at hand. First, an analysis is made of the difficulties of group decision-making and the solutions to the problem. In order to help classify these solutions, a two-level taxonomic system is described. Following this, formal definitions of the terms *voting* and *election* are given, with regard to these taxonomies. The terms *traditional voting*, *e-voting* and *I-voting* are then defined, along with details of their historical development and contemporary case studies of each. Finally, the DĒMOS 2 e-voting system is described.

2.1 Group decision-making

Life is a constant process of decision-making and all organisms are constantly doing so, be it consciously or unconsciously. The majority of these daily decisions leave little room for debate—whether one shall eat some food is easily answered by whether one is currently hungry—but there are times when the correct, or at least the least bad, decision not so clear. This becomes more common whenever those organisms are tasked with making decisions that affect more than just themselves—that is, decisions on the level of the *group*. For example, questions like 'should x lead us?' or 'should we fight with group z?' affect multiple lives and—when talking about human decision-making—the answers can have as much to do with personal, individually-variate beliefs—about x's character, for example, or about the particular egregiousness of group z—than they do with factual observation. For groups to function, therefore, various group decision-making processes have developed over the span of evolutionary history.

Though the remainder of this chapter shall focus exclusively on group decision-making within exclusively human groups—*anthrocracies*—but it is worth nothing that elements of the following proposed taxonomies map equally onto the 'government' of non-human groups, such as the lykocracy of a wolf pack. Within anthrocracies of a certain level of development, these processes are often codified into systems of government. Note, however, that whilst these terms are usually encountered in political discourse and reference the 'state', or any group which maintains a monopoly on the legitimate use of violence within its territory [104], they are equally applicable to *all* groups—i.e., whenever two or more people operate as a single unit. Whether one is talking about NATO, the UK, the Catholic Church, a university society, a Masonic lodge or a family of six or a couple of newlyweds, all have the necessity of group decision-making in common.

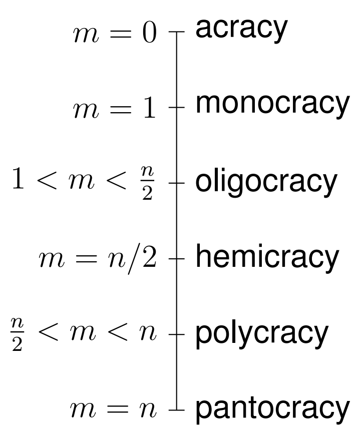


Figure 2.1: The Kratic Scale

2.2 A taxonomy of group decision-making processes

One must ask two questions of a group's decision-making processes in order to classify them. The questions are:

- what proportion of those people take part in the decision-making? and
- how are those who take part distinguished from those who do not?

Both of these questions shall be dealt with in turn, but both necessitate that we first distinguish between *direct* and *indirect* participation in group decision-making. Direct participation is here defined as any act which has a completely- or partially-binding effect on the group's decision. For example, as we shall see later, casting a vote that will be added to a tally for a particular course of action, where the resulting most-popular course will be undertaken, is an example of direct participation. Indirect participation is any act which has a non-binding effect. For example, advising the direct decision-makers or spreading awareness about one's preferred course of action.

2.2.1 The Kratic Scale

The first dimension along which one may categorise a group's decision-making processes is that of division. Where a group may consists of n members ($n \geq 2$), what proportion m of n directly participate in group decision-making? The values of m can be plotted on a $[0 - n]$ scale (see fig. 2.1).[1] Alternatively, each possibility can be described as such:

- acracy, wherein none of the population make decisions for others;
- monocracy, wherein one of a population makes decisions for the whole;
- oligocracy, wherein a minority subset of the population makes decisions for the whole;
- hemicracy, wherein one half of the population makes decisions for both;
- polycracy, wherein a majority subset of the population makes decisions for the whole;
- pantocracy, wherein the entire population makes decisions for itself.

[1]In the interests of specificity and consistency, I have chosen to replace the more common terms 'anarchy' and 'oligarchy' with their '-cracy' forms.

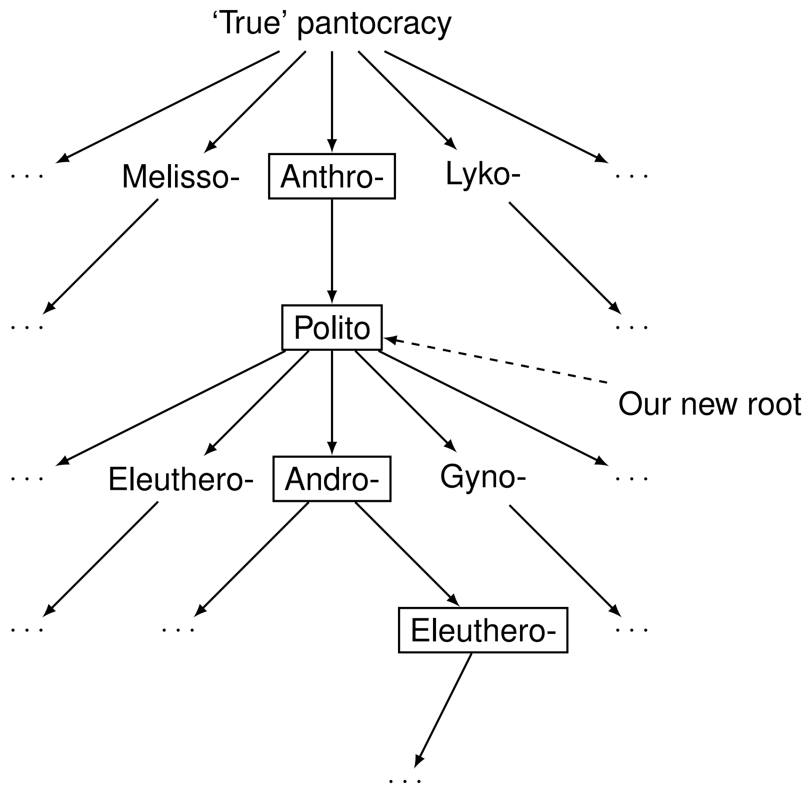


Figure 2.2: A Kratic Tree, with a path taken to describe a group in which direct decision-making power rests with free men (a *polyeleuthero-hemiandrocracy*)

This taxonomy is not perfect. Even putting aside acracy, which draws into question whether the resulting 'group' is any longer even that, things in reality are never so neatly-defined. None of these terms bring into account how much each member's participation is worth, relative to one another—i.e., if everyone in the population directly participates in group decision-making, but one minority subset's votes are counted twice, is that an oligocracy or a pantocracy? As the closest approximation of a true pantocracy might be something on level of an entire natural ecosystem, the human capacity for outsized environmental impact raises this very question.

2.2.2 Kratic Trees

Having specified the proportion of a group's population that participate directly in decision-making, we may now describe the basis by which those participants are selected. Rather than a scale, this element of the taxonomy takes the form of a tree, as seen in fig. 2.2. At the root node is the 'true' pantocracy of the ecosystem. Each branch consists of a different quality by which decision-making participants are distinguished from non-participants. For example, we have previously stated that we are here interested only in a study of human groups. Thus we first specify the oligoanthrocracy within the pantocracy as our scope of study. In addition, many human groups divide all those people impacted by the group's decisions—i.e., all of its *stakeholders*—into either the *polis* or not (e.g., dividing people living with the UK into 'citizens' or 'non-citizens'). Thus, we are now referring to a polypolitocracy (assuming that the polis represents the majority) within this broader oligoanthrocracy, and we may say that in the territory of the United Kingdom we have:

- the 'true' pantocracy of all stakeholders present (i.e., the ecosystem); which includes

- the oligoanthrocracy of all humans present (e.g., UK citizens, EU citizens, stateless persons, etc.); which includes
- the polypolitocracy of all UK citizens.

Say we want to discuss a nation in which a minority make decisions for the whole on the basis of their wealth. Starting from the root of the tree, this would produce the rather unwieldy 'oligopluto-polypolito-oligoangthrocracy'. Clearly, it is better that we identify in advance a node on the tree to serve as the root within our scope, and then elide the preceding terms, as we have done here by declaring our root to the polypolito-oligoanthrocracy node. If we later wished to contrast a given anthrocracy with a melissocracy of bees [87], for example, it would be necessary to move our root one node up, to the species distinction level.

The subset of the polis that are allowed to directly participate in group decision-making varies still further. For the present taxonomy, we shall abandon common but not-particularly-informative terms such as 'democracy'. Classical Athenian 'democracy', for example, allowed only male citizens over 30 years old to run for office, making it under this more precise taxonomy an oligo-geronto-androcracy, or rule of the few on the basis of their being sufficiently old and male. Even a modern 'democracy' like the UK restricts certain citizens—e.g., those in prison or under 18 years old—from participating directly [108], and is thus something of a poly-eleuthero-gerontocracy—rule of most, on the basis of their being sufficiently old and free. For the purpose of brevity, the term 'democracy' shall be used in the place of this last term for the remainder of this work.

Most historical societies have aligned primarily with either the monocratic or oligocratic ideal. Monocracy has traditionally been supplemented with some sort of theological basis for the monocrat's superiority, as in the idea of absolute monarchy and the 'divine right of kings', paired with a patrilineal descent—thus, a monotheopatricracy, or rule of one on the basis of God and their father. An oligocracy will similarly justify the superiority of its decision-making class, whether on the grounds of: their wealth (as in a plutocracy); their wisdom (as in an noocracy); their talent or ability (as in a meritocracy); or any number of other properties, both quantifiable or otherwise.

Democracy relies on the idea that by allowing all members of the population an equal say in the group's decisions, even someone whose preferred choice of action is not the most popular is nonetheless satisfied that the decision thus chosen has been chosen fairly, and will submit to it in the hopes that if the situation is later reversed and their preferred course of action turns out to be the most popular, the supporters of the less-popular choices will similarly accept the result. This is best encapsulated in the Enlightenment-era idea of the social contract, by which governments derive their right to exist from the consent of the governed. However, even in many democracies the distinction is unclear. In representative democracy, for example, all (or almost all) of the members have a say in the election of a group of people to whom the actual decision-making powers fall. In this case, representative democracy is more accurately described as democratically-elected oligocracy.

2.3 How each group type makes group decisions

Having described the different systems of government in terms of how they distribute participation in group decision-making amongst their membership, we shall now look at different means by which theses decisions are then made. Again, acracy shall be left aside.

In an monocracy, the monocrat makes group decisions the same way they would individual ones—they choose their preferred course of action, and that is the course of action that is followed. In a true monocracy there is nobody to provide input on the decision-making, so there is no debate except that internally within the monocrat. In real monocracies, such as many monarchies, the monocrat is given a team of advisers who can attempt to sway them towards one course of action or another, but who ultimately do not have a direct impact on the decision chosen—they are the monocratic equivalent of the powerless majority in an oligocracy.

Despite differing in *who* they allow to take part in the group decision-making, all non-monocratic types of group use the same means of determining the group's course of action. At its simplest, this is simply choosing at random from the available options, such as the sortition system of ancient Athenian democracy or, contemporarily, selection for jury service in the US and UK. Generally, however, groups assume (not *necessarily* correctly; see [77]) that reasoned-about and consciously-selected choices will generally produce more positive outcomes than random ones. As such, non-autocratic group decision-making methods involve around letting decision-makers indicate their preference(s) on a range of courses of action, recording those preferences and then determining which course of action to follow from this information.

Voting is the term for this preference-indicating.[2] In voting each participant, or 'voter', indicates their choice(s) of preference out of a number of discrete options—the ancient Greeks used pottery shards to record choices, most contemporary elections use paper ballots and smaller groups may make do with raised hands—and the winner is that option or those options that achieve some predetermined criteria, such as having received the greatest number of votes after a specified end date for the election. The whole process is called an 'election', and there are different ways of handling the results thus generated. *Winner-takes-all* means that the winning choice is enacted as it was proposed, regardless of how small a majority it may have garnered or how detrimental to the minority it may be. *Consensus-based decision-making*, on the other hand, requires the minority of voters to nonetheless consent to the course of action favoured by the majority before it may be enacted, and subsequently that the supporters of such a course of action must negotiate with those of the opposed minority in order to produce a compromise that all can accept.

How many cycles are taken before reaching the final result can also differ. The *Delphi method*, for example, invites the participants to anonymously present their judgements on the matter at hand to one another. They are then invited to revise their judgements in light of hearing others' arguments. This process repeats until a predetermined stopping point is reached, such as a set number of cycles having elapsed or a certain threshold of experts concurring with one another [25]. The privacy of one's choice(s) of preference also varies, from the use of secret ballots with no link back to the voter who cast them to voting via public announcement in favour of a given course, as does their mutability.

[2]Note that voting has been observed in a number of non-human species; see [58, 22, 87].

2.4 A brief history of voting

Voting has likely existed in some form or another for as long as human groups have had to make decisions on matters that may divide them, but democratic society is traditionally assumed to have begun with the reforms of Solon in Athens, between the 7th and 6th centuries B.C. (this traditional assumption has, however, been challenged by recent scholarship claiming evidence of the emergence of similar systems outside of the 'West' around a similar time period [65]). Nowadays, democracy is in vogue at the nation state-level—the majority operate systems that are (to some degree) democratic [114], and the US Department of State can be assumed to speak for all of the democratic world when it states its belief that '[d]emocratically governed nations are more likely to secure the peace, deter aggression, expand open markets, promote economic development, protect [our] citizens, combat international terrorism and crime, uphold human and worker rights, avoid humanitarian crises and refugee flows, improve the global environment, and protect human health.' [100]

Unlike most other aspects of society, the means of conducting an election have remained largely unchanged in the face of technological advancements over the centuries. Modern national elections continue to use paper ballots for recording of voter preferences, which are then counted by hand—methods that would be just as familiar to ancient Athenians with their shards of pottery as they are to us. Despite this, one characteristic of most contemporary voting systems would appear utterly alien to an early mindset: *who* is granted the vote. The expansion of suffrage, or voting rights, to greater and greater numbers of people is the only way in which voting has substantively changed over thousands of years.

2.4.1 An example election: UK general election 2015

In this section, we shall examine how a traditional election is generally run. We shall use the 2015 UK general election as our example [35].

Prior to the election day, all eligible voters (i.e. those who have previously registered for the electoral roll) are mailed a poll card detailing what the election is for, as well as when and where they are to cast their votes. On the day itself, they arrive at their allocated polling station and announce their name to the polling clerk, who strike them off of a list of eligible voters for the area. They then proceed to take their ballot paper to a private or semi-private voting station, where they place a cross next to the candidate of their choice. Finally, they fold the ballot paper in two and deposit it in a ballot box, before leaving the polling station. Some voters may instead submit postal ballots, the process for which shall not be detailed here.

The ballot boxes are then transported in vans to the count, which is presided over by an entirely ceremonial Returning Officer (RO)—usually the county high sheriff, mayor or council chairman—and the non-ceremonial Acting Returning Officer (ARO)—usually a senior local authority officer—who is ultimately '...legally and potentially financially responsible for screw-ups by any election staff.' Also present are the teams of counters, '...election agents, protecting the interests of their candidate...' and impartial, accredited observers, who are expected to report any suspicions they have about the fairness of how the count is being conducted to the election officials.

First, the ballot boxes are unsealed, the ballots removed and the boxes shown to all present to be genuinely empty. All of the ballot papers are then counted, and the totals compared to the ballot paper account provided by the Presiding Officer of each ballot station. If there is a mismatch, they are recounted until they match or until the same number is recorded twice in a row. The ballot papers are then mixed and allocated to teams of counters, who sort them into piles based on which candidate they display a vote for. Unclear ballots are deferred to the ARO or their deputy to make a decision on. After all of the ballot papers have been sorted, each candidate's votes (and any rejected votes) are again counted and compared to the total expected number of ballot papers.

The ARO shares the results with the candidates and their agents, who may request a recount if they consider it necessary. There are no limits on the number of recounts that can be held, but it is the ARO's decision to allow or deny a request. If the final result is a tie, the ARO may decide the winner via any method that all candidates approve, such as a coin toss.

2.5 Advantages and limitations of traditional voting

Traditional voting refers to those methods of running an election which have existed in some form or another for centuries, such as the aforementioned pottery shards, paper ballots and raised hands. The paper ballot system, whilst certainly labour intensive, is reliably robust and technologically immune to cyber-subterfuge. There are some fundamental limitations, however. Some are unique to certain countries' systems. For example, the UK, Singapore and a handful of others print unique serial numbers on each ballot paper as a means to combat fraud, with the trade-off of making each ballot potentially traceable to the voter who cast it [40], and there are allegations that British intelligence agencies may have used these methods to identify those voting for the Communist Party of Great Britain during the 1960s and '70s [112, 106]. The only protection against such an attack—besides the prohibitive amount of time and effort it would take when many contemporary voters seem happy to display their votes on social media [98, 10]—is that it is illegal to do so (or, more cynically, to be caught doing so). Additionally, the lack of a photographic ID requirement in UK polling stations theoretically allows any person who knows another's registered polling station to vote in their stead—one could use another's vote for a party contrary to their own preference, or use a person one knows to be unlikely to vote themselves as a means to increase one's own voting power, via an additional vote for their own candidate of choice.

Other limitations are inherent in the technology, or lack thereof, used. Paper-based and postal ballots, for example, '...don't allow voters to confirm their votes were actually counted.' [82] This means that the system demands that a voter trust the election officials to carry out a fair count. In addition, the present system requires that a voter physically show up at a polling station. This may deter those members of the electorate who would struggle to make their way to a local polling station, such as the disabled or expatriates, from participating. In addition, the impact of bad weather on voter turnout continues to be debated [76, 9, 48, 72, 52, 30, 6]. These restrictions may, in part, explain the 10% drop in global voter turnout over the last 25 years [113] (along with an over 20% drop in British voter turnout since 1950 [29]). Whilst postal votes and voting-by-proxy have been introduced to remedy these issues, they bring with them unique vulnerabilities of their own. Finally, the only element of the British election in which IT is utilised—the management of the electoral registers—has proven to be less-than-reliable, with mismanaged registers leading to voters being turned away from their polling stations [34, 102, 47]. During one mayoral election in 2016, '[o]ne station said that of the first 30 voters to show up, only three were on the register.' [47]

2.6 E-voting

One proposed solution to these limitations is the introduction of *E-voting*, either as an alternative to traditional voting or (more often) as a supplementary option. E-voting, as contrasted with traditional voting, is the introduction of any electromechanical machinery to the electoral process, be it in the casting of votes or simply the tabulation of results. The term encompasses '...systems such as DRE voting machines, ballot scanners, digital pens and internet voting systems.' [23]

An e-voting system can be classified as either a *Paper-Based Electronic Voting System*, where a paper ballot is produced by or marked using electronic means, or a *Voting System*, where no paper ballot is involved and the ballot is entirely electronic. Direct Recording Electronic (DRE) voting systems can be classified as either *on-site* or *remote*. In the former case, whatever electronic systems are used are used within the confines of an official polling station, under supervision. In the latter, votes may be cast remotely. This is usually over the Internet, and such a system is called an *Internet voting (I-voting)* system.

The Council of Europe has identified eight conditions which any e-voting system must aim to satisfy. These are:

1. universal suffrage;
2. equal suffrage;
3. free suffrage;
4. secret suffrage;
5. regulatory and organisational requirements;
6. transparency and observation;
7. accountability; and
8. reliability and security of the system [24].

There are a number of possible advantages to introducing e-voting (either as the sole or a supplementary means of voting) that have been argued for. One is the hope that by making voting something that can be done from home, perhaps with a simple app on the mobile phone already in the potential voter's pocket, it may better fit in to a changing contemporary culture of civic participation and help to reverse the trends of declining voter turnout, particularly amongst young voters. The evidence for this, however, is by no means conclusive—whilst one report claimed that the introduction of e-voting in the UK could increase 18–24-year-old turnout by up to 70%, and overall turnout by up to 79% [105], a working group appointed by the Finnish Ministry of Justice to review their own e-voting pilot concluded that '...enabling online voting would probably not increase voting turnout in any significant manner' [33]. Other research has found that whilst there was no significant general increase in voter turnout [13, 109], and that the middle-aged voter turnout actually increased more than the youth turnout [13].

Through the use of cryptography, ballots can be made physically unreadable for any unauthorised party, providing a stronger defence than a mere legal one. Additionally, the anonymity of voters may be better preserved trough the use of cryptographic means. Most promisingly, these additional layers of verifiability make it possible to create a system in which any part can be verified at any time, by any party, provided they have the technical means to do so. E-voting can also be implemented in such a way as to allow for a voter to change their vote at any time before the end of the election, as Estonia's system does [82], which may serve to help voters to make more informed decisions about their preferences over a longer period of time than the single day of voting that is typical of UK elections, as well as making vote buying unappealing (as voters can no longer be trusted not to change their vote after pocketing their fee).

Another argument often made in favour of e-voting is the proposed reduction in cost, with one report claiming that switching over to an e-voting system could save the UK £12.8 million annually [105]. This would, of course, only be the case if a government were to take the radical step of running an election solely using e-voting—otherwise, there would be an additional cost of running an e-voting system alongside the traditional means, no matter how small it may be. Finally, e-voting may allow expatriates to easily participate in elections in their home country, although as the means to enable this already exist without the introduction of e-voting, objections to doing so are likely to be more political than technical.

2.7 E2E-verifiability

Traditionally, the security of voting systems has been assessed as a result of the assessed security of each part of the process by which elections are undertaken in the system. However, this granular level of security assessment can be less than ideal, leading to the ignoring of some vulnerabilities and the overcorrection of others. The alternative is for a system to be *E2E-verifiable*, in which the security of the entire chain is assessed holistically. In addition, '[s]ecurity experts advise that end-to-end verifability—lacking in current systems—is one of the critical features needed to guarantee the integrity, openness, and transparency of [I-voting] election systems.' [27]

E-voting systems that are considered to be E2E-verifiable when the following integrity properties are satisfied for each vote cast:

1. recorded-as-intended, i.e. the vote made accurately reflects the intention of the voter;
2. cast-as-recorded, i.e. the ballot cast accurately reflects the recorded vote; and
3. counted-as-cast, i.e. the eventual election tally includes all cast ballots in the totals for their respective candidates. [83]

Each of these steps *must* be verifiable, but the parties that can verify each step varies. In the case of recorded-as-intended, for example, '[o]nly [the] voter knows [their] intentions, and these should be kept private, so only [they] can verify the record of their vote'. On the other hand, satisfying cast-as-recorded verifiability necessitates that the '[c]ollection of cast ballots needs to be *public*, and cast ballots need to be *identifiable*, so [a] voter can find [their] ballot and check that it is correctly included [emphasis theirs]' in the collection, but in such a way that a voter is unable to sell their vote to an interested other party, such as through the encryption of ballots and the issuing of the resultant ciphertexts as receipts. Finally, despite this ballot encryption, a number of approaches exist to allow any party to verify counted-as-cast for the final election result.

One of the earliest proposals for an E2E-verifiable began with a lamentation that '[c]urrent electronic voting machines at polling places don't give receipts [and that, r]ather, they require prospective voters to trust them—without proof or confirming evidence—to correctly record each vote and include it in the final tally' [16], going on to introduce a system that could provide a '...a fundamentally new kind of receipt' based on visual cryptography. In this case, the visual cryptography involved the use of translucent printed ballots that must be overlaid in order to reveal the complete receipt. This was followed in short order by a proliferation of E2E-verifiable systems such as Punchscan [18], Prêt à Voter [85], Scratch & Vote [2], Scratch, Click & Vote [55], ThreeBallot [84], all of which rely on paper ballots of varying degrees of complexity. Scantegrity distinguished itself by proposing an add-on for existing optical scan voting systems, rather than a replacement for the system itself [17]; STAR-Vote was intended to do something similar a few years later when it was introduced in Travis County, Texas, in the hopes of prompting a shift '...away from the old model, in which counties principally buy unique hardware from vendors...[towards one that is] county-owned and -operated, relying on open-source software that can be shared across jurisdictions and only requires equipment that can be bought commercially off the shelf, like tablets and scanners.' [11, 74] However, the STAR-Vote project was ultimately unsuccessful [80].

2.8 E-voting in practice

E-voting and I-voting have been trialled or implemented, whether for select populations or nationwide, in a number of countries, as varying levels within each, and to varying degrees of success and longetivity (see table 2.1, although bear in mind that 'e-voting' encompasses quite a range of systems and thus direct comparisons between just what it is that each country has implemented under that name are difficult to make). In this section, a brief summary of the experience of each country that has experimented with such measures shall be given. The countries are here listed in alphabetical order.

Australia

Australia trialled e-voting for vision-impaired voters in their 2007 federal election, which later '...evolved into the current method of telephone voting for this group of voters'. Electronically-certified lists were first introduced for the 2013 federal election and have continued in use since; ballot paper scanning and voter preference recording were also performed with electronic means in their 2016 federal election [60]. At the state level, the Australian Capital Territory (ACT) first introduced e-voting options for their 2001 elections, with the technology '...applied to both the casting and counting of votes.' By the 2008 ACT elections, 1 in 5 voters' votes were in some way electronically-processed. Victoria and Western Australia have introduced some e-voting measures, Tasmania and South Australia use e-voting solely for logistical purposes and Queensland and the Northern Territory have no implemented no e-voting measures, with no plans to do so in the future [42].

Australia trialled I-voting for Australia Defence Force personnel deployed overseas in their 2007 federal elections. The trial cost amounted to $1,159 per I-voter, compared to a cost of $8.86 per traditional voter, and the trial was thus discontinued on a cost basis [49]. The Parliament of Australia Web site declares that, '[o]verall, the Australian states and territories have not embraced electronic voting to any great degree', although New South Wales has allowed certain voters to vote over the Internet in state-level elections since 2011 [42, 60].

Country	E-voting introduced	Ongoing?	I-voting introduced	Ongoing?
Australia	2007	Yes	2007	No
Belgium	1991	Yes		
Brazil	1996	Yes		
Estonia	2005	Yes	2007	Yes
Finland			2008	No
France			2003	No
Germany	2005	No		
India	1982	Yes		
Ireland	2003	No		
Italy	2008	Yes		
Kazakhstan	2004	No		
Namibia	2014	Yes		
The Netherlands	1966	No		
Norway	2003	No	2011	No
The Philippines	2010	Yes		
Romania	2003	Yes		
South Korea	2012	Yes		
Switzerland			2009	Yes
United Arab Emirates	2011	Yes		
United Kingdom	2000	No		
United States[3]	1964	Yes	1997	Yes
Venezuela	1998	Yes		

Table 2.1: The introduction of e-voting and I-voting by country. For countries operating federal systems of government, only details of federal adoption are given.

Belgium

Belgium introduced e-voting experimentally in 1991. Coverage was extended to 22% of the population in 1994 and to 44% in 1999, a level at which it has remained steady since [26].

Brazil

Brazil introduced e-voting in 1996 and, since 2000, has conducted all elections entirely electronically [21].

Canada

E-voting has been used in some municipal elections in Canada since 1988 [20], and I-voting since 2003 [39]. E- and I-voting are not used at the provincial or federal levels.

Estonia

As part of its E-Estonia scheme, Estonia became the first country in the world to host a legally-binding general election using I-voting in their 2005 local elections, and the first to allow I-voting in a national election in their 2007 election [15]. By the 2015 parliamentary elections, 30.5% of participants cast their votes over the Internet [101].

Finland

Finland piloted I-voting in three municipalities in 2008. A working group was set up in 2017 to assess the feasibility of introducing nationwide I-voting, only to conclude that the risks outweighed the benefits [33].

France

France had allowed French citizens living abroad to vote in Assembly of French Citizens Abroad elections over the Internet since 2003 [46]. This was rescinded in 2017 due to cybersecurity concerns [96].

Germany

Germany introduced e-voting for their 2005 Bundestag elections. Plans to extend the system were cancelled in 2009 after the Federal Constitutional Court ruled e-voting unconstitutional [70].

India

India introduced e-voting in 1982 on an experimental basis, although the election was subsequently struck down by the Supreme Court of India [94]. After amendments were made to the relevant legislation, all state elections and by-elections used e-voting in 2003. In 2011, Gujarat state became the first to experiment with I-voting [90].

Ireland

Ireland bought a number of e-voting machines to be piloted in their 2003 general election. The machines were unpopular and mothballed, until Taoiseach Brian Cowen announced in 2010 that the machines were to be disposed of, with the overall cost of the project amounting to over €54 million [73].

Italy

A handful of municipalities in Italy trialled partial e-voting in 2006, before the experiment was halted over tampering concerns by the then-Prime Minister. Another attempt was made in 2008 that ran a paper ballot system alongside an e-voting system in order to allow the result to be verified, and in 2013 an all-e-voting system was introduced for the first time [81].

Kazakhstan

Kazakhstan introduced e-voting as an option for their 2004 Parliamentary elections and retained the system for their 2005 and 2007 elections [50], before abandoning it for subsequent elections on the grounds that '...the system turned out to be unpopular among voters...political parties still do not trust it...[and] significant funds are required to update this system and make it work' [86].

Lithuania

Lithuania has a target for 20% of votes to be cast via I-voting by 2020, but there does not appear to be as yet any means to do so [59].

Namibia

Namibia became the first African nation to introduce e-voting in 2014, despite '...an 11th-hour court challenge to stop the vote from going ahead, saying the use of the Indian-made e-voting machines could facilitate vote rigging.' [68]

The Netherlands

The Netherlands used e-voting until 2007, when security concerns raised by the pressure group Wij Vertrouwen Stemcomputers Niet ('We Do Not Trust Voting Machines') led to the machines' withdrawal [57, 110].

Norway

Norway carried out limited e-voting pilots in 2003. 'Experiments with [I-voting] were carried out during elections held in 2011 and 2013' [28], but these were unsuccessful in increasing voter turnout. This, combined with security concerns, led to the dissolution of the trials in 2014 [99].

The Philippines

The Philippines invested $160 million into e-voting in 2010. Despite issues discovered with the machines during pre-tests that necessitated their recall and reissuing nationwide [79], as well as the placing of 250,000 troops on high alert during the election in order to quell violent protests against the system [61], the Phillipines held its first e-voting-facilitated presidential election that year [62].

Romania

Romania trialled e-voting for military personnel deployed overseas in 2003 [93] and pledged for their 2016 local elections that '[t]here will not be even one polling station that does not have an electronic contact device' [31].

South Korea

South Korea introduced electronic means of vote-counting in 2012, and the resulting system has been praised as 'best practice' [63].

Switzerland

Switzerland becam I-voting trials in 2009, opening the option up to all Swiss expatriates in 2014.

The United Arab Emirates

E-voting was introduced for the 2011 Federal National Council elections in order to encourage participation amongst an electorate unfamiliar with elections.

The United Kingdom

The first e-voting pilot in the UK took place in 2000, limited to vote-counting technology. Despite the Digital Democracy Commission's 2015 report stating that '[b]y 2020, secure online voting should be an option for all voters' [92], a 2016 government response stated that the governmetn '...do not have any plans to introduce electronic voting for statutory elections, either using electronic voting in polling booths or remotely via the internet.' [75]

Despite errors in implementation producing as many as 150,000 unintentionally-spoilt ballots after it's introduction in 2007, Scotland has used e-voting in subsequent elections without issue.

The United States of America

Electronic counting of ballots was first introduced in California in 1964, and the first DRE machine in Illiois in 1974. This trend has continued despite a number of embarassing incidents, such as the contested 2000 presidential election [**fahrenheit911**], turnouts of over 200% [36] and leaked elector details [67], and many states now use some form of electronic assistance in their election-running (although specific figures do not appear available) [41].

32 states allow for I-voting by overseas military personnel and citizens via one or more electronic means, in line with the requirements of the 2007 Military and Overseas Voter

Empowerment Act. The first example of I-voting provision was Texas's decision to allow astronauts in orbit the ability to vote over email in 1997. In addition, four states allow I-voting for non-overseas citizens, with Alaska giving the option to *all* registered voters.

Venezuela

E-voting was introduced for the 1998 presidential election.

2.9 DĒMOS 2

One further proposal for an E2E-verifiable e-voting system is DĒMOS. The system, which can also be used for I-voting, aims to provide 'end-to-end verifiable elections' whilst, as opposed to popular alternatives such as Helios [1], not relying on '...any additional setup assumptions or access to a random oracle' [54]. Subsequent to the original DĒMOS, two variants exist. DĒMOS 2 was developed in order to address the scalability issues of the original system design [53], whilst D-DĒMOS is a distributed version of the same system [19].

The proofs presented in the original DĒMOS paper demonstrate that the system is E2E-verifiable, and this property holds true for each subsequent iteration of the system. As a reminder, this '...mandates that the voter can obtain a receipt at the end of the ballot casting procedure that can allow her to verify that her vote was (i) cast as intended, (ii) recorded as cast, and (iii) tallied as recorded.' The receipts in any such E2E system must also be delegatable, in that '...the voter may delegate the task of verifiability to any interested third party...'. Finally, however, is the complication that '...it should be infeasible for the voter to use her receipt as proof of the way she voted...', the risks of encouraging vote selling/buying being obvious.

Development on a usable implementation of the DĒMOS 2 system began in 2017 as a project within the Django web app development framework [111]. Ownership of the code-base was transferred in 2018 and the implementation was expanded to use Node.js and Celery for the server-side processing [5].

Whilst this software implementation of the DĒMOS 2 system is for the most part entirely functional, allowing the running of elections using the secure E2E-verifiable cryptographic principles outlined in the original paper, one component is missing—the ability for a voter or an interested third-party to independently audit a given election to ensure that the ballots are being recorded-as-intended, cast-as-recorded and, ultimately, counted-as-cast. It is this final puzzle piece that the remainder of this dissertation shall detail the development of.

Chapter 3

System Architecture

This chapter contains an overview of a typical e-voting system architecture, taken primarily from the example of the Estonian model [56] with some adaptations for generalisability, followed by an overview of the specific architecture of the DẼMOS 2 system.

3.1 E-voting

Participants in an e-voting election can act in one of four positions. These positions are:

- voter, responsible for the casting of votes using a Vote Support Device (VSD);
- tallier, responsible for computing final election results;
- auditor, responsible for verifying the correctness of each stage of the election; and
- trustee, responsible for securing the election using a Trustee Support Device (TSD).

The VSD and TSD can be any applicable device used by the voter or trustee, such as a laptop, mobile phone application, etc. Each position can have any number of participants perform it.

An e-voting system consists of four elements. These elements are:

- the Bulletin Board (BB);
- the Election Authority (EA);
- the Registration Authority (RA); and
- the I-ballot box.

The architecture of an e-voting system is shown fig. 3.1. When an election is being set up by the system, the trustee(s) first share amongst themselves a secret s which is used in order to generate a public-key cryptography keypair (PK, SK). The public key PK is published to the election server, whilst the secret key SK is divided into n pieces, where n is the number of trustees, using an (n, t)-threshold secret sharing scheme [88, 12]. Each trustee is given one share of SK, and the complete SK can only be reconstructed when a sufficient number of shares (the threshold t) are combined.

The BB for an election stores the candidate and voter lists. Once the election has been set up, the system uses the contact details stored in the voter list to contact all eligible voters

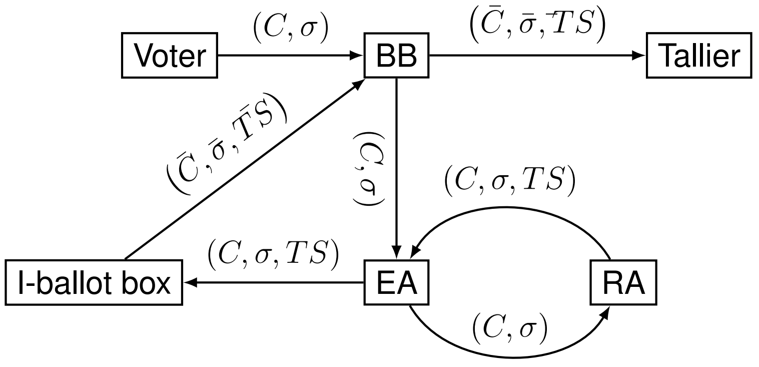


Figure 3.1: Typical e-voting system architecture

and provide them with a link through which they may vote. Following the link generates a unique voter ID for the voter. Using their VSD, the voters choose their preferred choice(s) from the candidate list. These ballots are encrypted with the election's PK to produce the ciphertext C. The voter sends C, along with the signature σ, to the BB.

The BB sends (C, σ) to the EA's Vote Forwarding Server (VFS). The EA's VFS, in turn, passes the ballot to the RA, which adds a timestamp TS. This timestamp is used in schemes that allow voters to change their votes to determine which votes to count and which to discard. The RA returns (C, σ, TS) to the EA, which stores it in its Vote Storage System (VSS).

The EA then sends the ballot to the I-ballot box, which contains a mix-net. A mix-net takes n inputs and, through a sufficiently complex series of operations to each so that the process is not reversible, produces n ciphertext outputs. Whilst zero-knowledge proofs can be presented in order to show the correctness of the outputs, but these are beyond the scope of the current inquiry. The encrypted ballot $(\bar{C}, \bar{\sigma}, TS)$ is then sent back to the BB.

When the election has expired, the BB passes its ballots to the tallier, using the timestamp to resolve priority conflicts where relevant. Given n ballots, the tallier computes $Tally\left(C_1, C_2, \ldots, C_n\right)$ to determine the number of votes for each candidates, where $Tally\left(\right)$ is the function required to tally votes based on the encryption scheme used. The result can then be shared through whatever means are appropriate.

At any point during this process, the auditor can test the three properties required for E2E-verifiability, as discussed in § 2.7, as well as the correctness of the tallier.

3.2 DẼMOS 2

3.2.1 Software Stack

The currently-existing DẼMOS 2 implementation [111, 5] runs as a web application, consisting of three server-side elements:

- a Node.js web server [71];
- a Celery distributed task queue [91]; and
- a MySQL database [66]

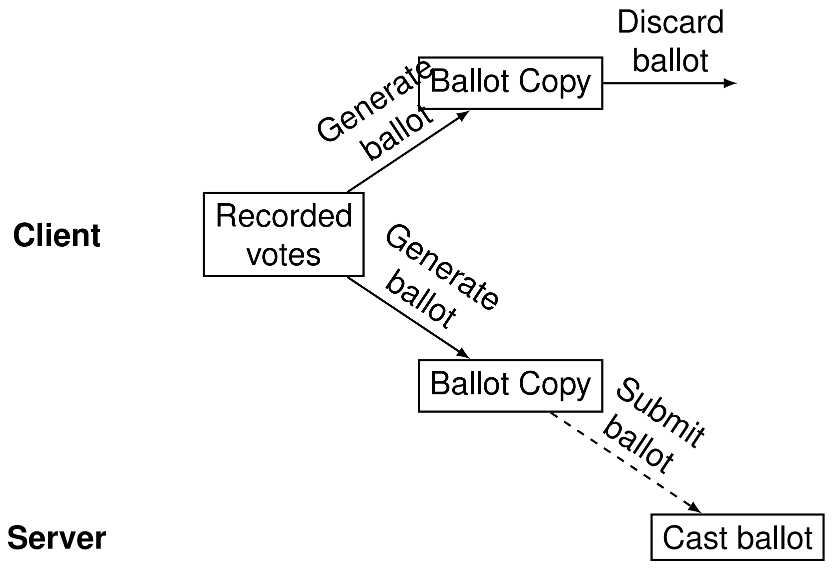


Figure 3.2: From recorded votes to cast ballots

The Node.js server serves a web app built using the Python-based Django Web framework [95] in the Model-View-Template (MVT) pattern, using models stored in the MySQL database.

Unencrypted ballots are never passed to the server, with encryption being performed using client-side Javascript. The Milagro Crypto Javascript library is used for all cryptographic tasks [64], CSV parsing is handled by the Papa Parse module [43] and jQuery is used throughout [51]. Fig. 3.2 shows the path taken to submit a user's ballot to the server.

3.2.2 Running an Election

When the server is running, the web app may be reached on port 8000. The user is prompted to log in or register. Having done so, they may now set up an election. Along with details of the start and finish times, the polls and their candidates, the user is also prompted to upload a number of email addresses for trustees and voters.

Once completed (and after all entered details have been validated), the election is created with a 'Pending' status. Emails are then sent to all addresses in the trustees list containing a link to a page on the web app in which they are prompted to generate and save a private key before submitting the corresponding public key to the server. Once enough public keys have been returned, the election status is changed to 'Prepared'.

Emails are now sent to all entered voters containing their unique voting links. If the election start time has not yet passed, these links will display an error. Once the start time has passed, the election status is changed to 'Active' and the links instead direct voters to their voting page, where they are able to place their vote on each poll presented.

When all votes are recorded, two copies of the resulting ballot are generated. The user is prompted to choose one ballot to submit to the server, and the other is discarded. A voter is free to return at any time before the end of the election and change their votes, which will necessitate the generation and submission of a new ballot.

Once the election is complete the trustees are again emailed, this time with a link at which they can submit their private keys. Onces a sufficient number of these keys have been submitted, the election results are decrypted server-side and the results tallied.

Chapter 4

Design

All versions of DẼMOS are proven to be end-to-end verifiable. However, as yet no third-party auditing software exists, without which any such verifiability remains purely academic. Thus, there is a need for such a piece of software in order to allow any interested party to verify the results of an election with ease. In this chapter, the design of such a piece of software, in the form of an Android mobile application, is described.

4.1 Project Requirements

The requirements for this project can be divided into two groups: requirements for the DẼMOS 2 software as currently exists; and requirements for the auditing app that must be created. The key words 'MUST' and 'MUST NOT' in this document are to be interpreted as described in RFC 2119 [14].

4.1.1 DẼMOS 2 Requirements

Modifications to the existing DẼMOS 2 system must be made so that:

1. voters MUST be presented with confirmation of their recorded vote in order to allow for *recorded-as-intended* verification;
2. voters MUST be presented with a secure copy of their ballot in a format that can be easily transferred to another device (such as a mobile smartphone);
3. options MUST be provided for the format of this data transfer in order to ensure that all voters are able to transfer it regardless of the technology they possess;
4. voting information MUST NOT pass unencrypted to the server at any point;
5. voters MUST be able to check that their ballot exists on the server;
6. voters and third parties MUST be presented with a means of auditing ballots held on the server in order to allow for both *cast-as-recorded* and *tallied-as-cast* verification; and
7. voters and third parties MUST NOT be able to discern which option(s) a ballot contains a vote for.

4.1.2 App. Requirements

The application must be designed in such a way that:

1. voters MUST be able to receive data presented by the DẼMOS 2 system using their mobile phone;
2. voters MUST be able to store secure copies of their ballots locally;
3. voters MUST be able to transfer their secure ballot copies to third parties;
4. voters and third parties MUST be able to compare their secure ballot copies with the ballot copies stored on the server; and
5. voters and third parties MUST NOT be able to discern which option(s) a ballot contains a vote for.

4.2 DẼMOS 2 Modifications

Req. 1 is easy to implement without breaching req. 4 by using client-side Javascript—during the voting process, before the ballot is produced, a confirmatory window is presented to the voter featuring the candidate(s) for whom they have voted.

There are a handful of options by which req. 2 could have been satisfied, but the use of a Quick Response (QR) code—a matrix-based form of barcode—seemed the simplest. 83% of mobile smartphones in 2012 incorporated a camera [3], and it seems reasonable to assume that this proportion is by now far higher, so the use of a QR code should also mostly satisfy req. 3, although a simple text representation of the content of the QR code is also presented alongside it in order to ensure that all users are able to transfer their data.

The original plan was for DẼMOS 2 to display a QR code containing the entire unselected ballot. However, the ballots (which are JSON objects) are very lengthy due to the fact that they contain cryptographic values—a standard QR code can only fit 3 KiB of data [**qr**]—and so they had to be shortened via Base64-encoding. Even this produced very large values, so in the final version the ballot is encrypted using symmetric encryption and the secret key SK (over the original asymmetric encryption using the election's public key PK, resulting in $Enc\,(Enc\,(ballot)_{PK})_{SK}$). A Hash-based Message Authentication Code (HMAC) tag is then computed for this ballot and the whole thing sent to the server.

By presenting SK and the HMAC to the voter on the client side, and therefore never sending them to the server, it can be ensured that the server cannot decrypt, and so read or alter, the ballot, satisfying req. 4. The use of an HMAC ensures that the server cannot replace the ballot with its own altered version, as the HMAC would then be different. Finally, by giving the ballot a unique ID and passing that to the voter as well, the voter (or a third party) can request the ballot from the server and decrypt the layer of symmetric encryption using SK.

To fulfil req. 5, a 'Find Ballot' page was added to DẼMOS 2. This page takes a number of values—the voter ID and poll ID—and two hashes. It uses the IDs to retrieve a matching ballot from the server database (if found), hashes it and compares the hash to the two provided in order to check that the ballot on the server matches one of the two generated for the voter. MD5 hashes were chosen for the level of speed and security they provide compared with alternatives such as SHA-1 [103].

For req. 6, a new ‘Audit’ page will be added to the DẼMOS 2 system. This page can be passed the unique handles of two encrypted ballots in order to retrieve them from the server. Alternatively, ballot record files can be uploaded using a file browser in order to bypass the server, if need be. After validating the ballots’ HMACs, they can then be decrypted using an AES secret key entered by the voter and compared in order to check for equality, without showing any information about who the ballot contains a vote for.

The ability to compare arbitrary ballots might have made it possible for an adversary to determine which option(s) a ballot contained a vote for by making ballots corresponding to different votes and comparing them with genuine ballots until a match was found. However, req. 7 should be covered by the fact that all handled ballots remain encrypted with the election’s PK.

4.3 Auditor App Design

As mentioned above, QR codes were chosen as the primary means of data transfer, thus satisfying req. 1. As such, the app uses the mobile device’s camera in order to scan the code and store the resulting ballot ID. By comparing the stored HMAC hash with that attached to the downloaded ballot, it can be confirmed that the ballot has not been tampered with.

Android apps are divided into Activities, each of which represents a function of the app and may or may not have a UI display associated with it. The main menu of the app., to which it would initially load, should present options for whether the user is casting a vote, and thus wishes to record their ballot information using the camera, or if they are auditing an election instead.

Choosing the ‘cast a vote’ option, the user will be taken to an Activity in which they use their mobile phone camera to scan a series of QR codes presented by DẼMOS 2, storing the transferred data in a LBRF. Once the last QR code has been scanned, the voter will be returned to the main menu.

Choosing instead to ‘audit an election’, the user will be asked if they wish to check that a ballot exists on the server, compare two ballots or audit an election result. In the first case, they will be presented with a file browser and asked to locate the LBRF that they wish to use. Having done so, they will be taken to an Activity with a WebView—which acts similarly to an HTML `<iframe>` element in that it embeds a web page within the app.—pointing to the DẼMOS 2 ‘Find Ballot’ page mentioned previously, with the relevant details retrieved from the LBRF submitted to the page as URL parameters. This would check that the hash of the ballot submitted to the server matches one of the two presented to the voter at ballot generation and display a message in the app. to say so.

When choosing to compare two ballots, the user will again be presented with a WebView featuring the DẼMOS 2 ‘Audit’ page. The user will be able to specify their two ballot handles by either entering them manually, or by selecting two LBRFs using a file browser, from which the handles can be extracted. The page will then retrieve both ballots, computer their HMACs and prompt the user to enter their secret key. After decrypting both ballots, the

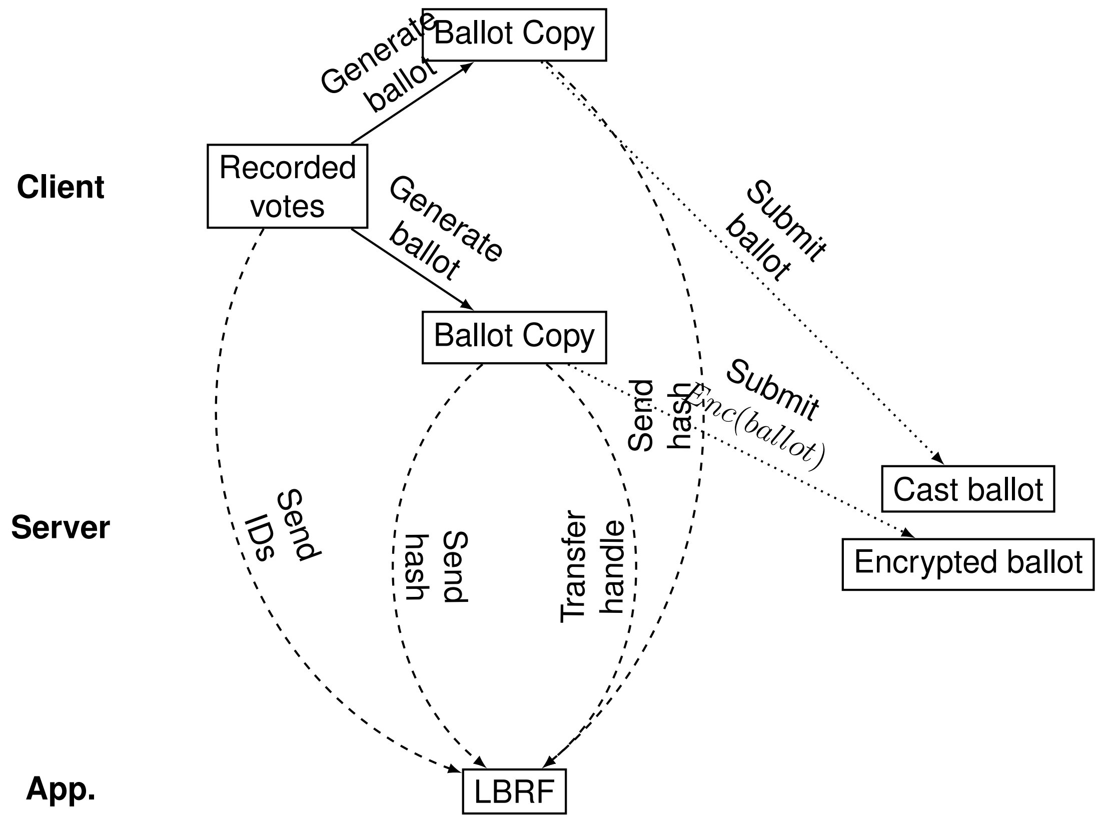


Figure 4.1: From recorded votes to cast ballots, encrypted ballots and LBRFs

HMACs within can be compared to those just generated in order to detect any modifications. Following this, the ballots can be compared for equality, and the result presented to the user.

For the election result auditing option, the user will be presented with a file browser and asked to select a directory containing multiple LBRFs. After checking that all files contain the same election ID, each ballot will be checked for presence on the server.

Use of WebViews makes it possible to to load the DĒMOS 2 pages mentioned previously and to run their Javascript client-side on the device, which easily satisfies reqs 4 & 5. To satisfy reqs 2 & 3, the LBRFs will be saved to the phone's External Storage and so are accessible from outside of the app, such as by a file viewer or email application.

Finally, it was decided to develop the app for the Android Operating System (OS) for two reasons. First, Android has by far the largest market share out of all mobile device OSes at around 85% to iOS, which has the second-biggest share at around 15% [45]. Second, Android development is free and requires no license, whereas iOS development requires that one join the Apple Developer Program at a rate of $90 per year.

4.3.1 Local Ballot Record File (LBRF)

The LBRFs need to contain all the information required to identify ballots on the server and validate their integrity. For the first condition, the LBRF will contain the voter ID, the event ID for the election and the ID of the poll which the ballot contains votes for in order to retrieve the submitted ballot later. In addition, the LBRF will contain the hashes of both ballot copies generated client-side in order to later verify that one of them is present on the server. The

LBRF will also contain the unique handle of the encrypted, unselected ballot sent to the server in order to retrieve it later. Finally, the LBRF will contain the HMAC computed over the encrypted ballot before it is sent to the server. This can later be compared with the HMAC computed over the encrypted ballot retrieved from the server in order to verify that it has not been tampered with.

Chapter 5

Implementation

This chapter shall detail the implementation thusfar of the designed app., including modifications made to the DĒMOS 2 system. Sources for both can be found on GitHub [37, 38].

5.1 DĒMOS 2

5.1.1 General

`/allauthdemo/polls/urls.py`

URL routing instructions were added to this file for both newly-added pages:

```
url(r'^audit/$', views.vote_audit, name='vote_audit'),
url(r'^find_ballot/$', views.find_ballot, name='find_ballot')
```

`/allauthdemo/polls/models.py`

Firstly, a table had to be added to the DĒMOS 2 database model to hold the AES-encrypted ballots transferred via the server. This was added to the `/allauthdemo/polls/models.py` file:

```
class EncBallot(models.Model):
  handle = models.CharField(primary_key=True, default=uuid.uuid4,
     editable=False, max_length=255)
  ballot = models.CharField(max_length=10240)
```

5.1.2 'Vote' page

Modifications were required to the event voting page in order to provide the means for auditing.

`/allauthdemo/polls/views.py`

The `event_vote()` method was amended to allow for the saving of the unselected, encrypted second ballots to the MySQL database:

```
enc_ballot_json = request.POST.get('encBallot')
handle_json = request.POST.get('handle')
```

```
# Adds or replaces the encrypted un-submitted ballot to the database for
    the auditor app to pick up later
if EncBallot.objects.filter(handle=handle_json).exists():
   b = EncBallot.objects.get(handle=handle_json)
   b.ballot = enc_ballot_json
   b.save()
else:
   b = EncBallot(handle=handle_json, ballot=enc_ballot_json)
   b.save()
```

`/static/js/event_vote.js`

The next step was to amend the static client-side voting page to present the voter with their ballot hash, encrypted ballot handle and QR codes of each. This was added to the `/static/js/event_vote.js` file. r had to be added to the encrypted ballot fragments in the `generateBallots()` method for later use in ballot verification:

```
var rBytes = [];
cipher.r.toBytes(rBytes);

encFragments.push({
   C1 : c1Bytes.toString(),
   C2 : c2Bytes.toString(),
   r : rBytes.toString()
});
```

A `showIDsQRCode()` method was added, called after ballot generation, which displays the voter ID, event ID and poll ID for a given ballot:

```
function showIDsQRCode(ballotA, ballotB, selectedOption) {
   var voterID = window.location.search.slice(1).split(/=(.+)/)[1];
   var eventID = window.location.href.split('/')[4];
   var pollID = $('#poll-num').text();

   // Display a QR code with ID details
   var modalDialog = $('#modalDialog');
   var title = modalDialog.find('.modal-title');
   var body = modalDialog.find('.modal-body');

   body.empty();
   title.text('Step 0 of 3: Scan this');

   let pleaseScanP = document.createElement('p');
   pleaseScanP.innerHTML = "Please scan the following QR code from your
       DEMOS 2 mobile application:";

   let QRDiv = document.createElement('div');
   var QRCodeImg = document.createElement('img');
   QRCodeImg.setAttribute('class', 'QR-code');
   QRCodeImg.setAttribute('id', "qr-img");
   new QRCode(QRCodeImg, voterID+";"+eventID+";"+pollID);
   QRDiv.append(QRCodeImg);
```

```
        body.append(pleaseScanP);
        body.append(QRDiv);

        // Prepare the appropriate dialog buttons
        updateDialogButtons(DIALOG_BTN_STATES.STEP_1);

        if(!dialogOpen) {
            modalDialog.modal('toggle');
            dialogOpen = true;
        }

        $('#nextDialogBtn').click(function(e) {
            showHashQRCode(ballotA, ballotB, selectedOption);
        });
    };
```

Following this, `showHashQRCode()` hashes the two generated ballots and presents the result to the user in both text and QR code form:

```
    function showHashQRCode(ballotA, ballotB, selectedOption) {
        var ballots = new Array(ballotA, ballotB);
        var ballotHashes = new Array(2);

        // Hash both ballots and store
        for (let i = 0; i <= 1; i++)
            ballotHashes[i] =
                SHA256Hash(stringtobytes(JSON.stringify(ballots[i])), true);
```

After the selection of a ballot to submit to the server, and after the choice has been confirmed by the voter, the `sendBallotsToServer()` method was modified. This method originally only handled the submission of the selected ballot to the server and discarded the unselected ballot. This was amended to instead generate a unique handle for the unselected ballot:

```
    var voterID =
        window.location.search.slice(1).split(/=(.+)/)[1];//.slice(0, -2);
    var eventID = window.location.href.split('/')[4];
    var pollNum = $('#poll-num').text();
    var ballotID = encodeURIComponent(btoa(JSON.stringify({voterID: voterID,
        eventID: eventID, pollNum: pollNum})));
```

Then, the unselected ballot is AES-encrypted, an HMAC generated and the whole thing submitted to the server, identified by the aforementioned handle:

```
    // TODO: Generate a SK rather than using a static one. UUID generated
        server side and then injected JS side?
    JSON.stringify(otherBallot)
    var SK = "temporary";
    var encAlt = sjcl.encrypt(SK, JSON.stringify(otherBallot));
    var out = (new sjcl.misc.hmac(key, sjcl.hash.sha256)).mac(encAlt);
    var hmac = sjcl.codec.hex.fromBits(out);
    let selectedBallotAsStr = JSON.stringify(selectedBallot);

    $.ajax({
```

```
    type : "POST",
    url : window.location,
    data : { handle: ballotID, encBallot: encAlt, ballot:
        selectedBallotAsStr, selection: selection },
    success : function(){
       onAfterBallotSend(ballotID, SK, hmac);
    }
});
```

Following this, the `onAfterBallotSend()` method was modified to present the handle of the unselected ballot, in both text and QR form:

```
// Add the second section: QR code that contains the ballot identifier
    and HMAC
var QRCodeImg = document.createElement('img');
QRCodeImg.setAttribute('class', 'QR-code');
new QRCode(QRCodeImg, ballotID+";"+btoa(hmac));

body.append(QRCodeImg);
```

SK is also displayed:

```
let SKContainerDiv = document.createElement('div');
SKContainerDiv.setAttribute("class", "containerMarginTop");

let SKDiv = document.createElement('div');
SKDiv.setAttribute("class", "skDIV");

let SKP = document.createElement('p');
SKP.innerHTML = SK;
SKDiv.append(SKP);

SKContainerDiv.append(SKDiv);
body.append(SKContainerDiv);
```

5.1.3 'Find Ballot' page

This page allows a user to check whether the election server contains a ballot whose hash matches one of the two saved by the voter during the voting process.

`/allauthdemo/polls/views.py`

A method was added to the Django view controller that takes ballot identification details and two hashes, delimited with a semicolon, as URL parameters:

```
def find_ballot(request):
   voter_id = request.GET.get('vid', None)
   poll_id = request.GET.get('pid', None)
   ballot = get_object_or_404(Ballot, voter=voter_id, poll=poll_id)
   hash1 = request.GET.get('hash1', None)
   hash2 = request.GET.get('hash2', None)

   if not ballot.cast:
```

```
            messages.add_message(request, messages.WARNING, "This ballot has not
               been cast.")
            return HttpResponseRedirect(reverse("user_home"))

        return render(request, "polls/find_ballot.html",
            {
               "ballot": ballot.json_str,
               "hash1": hash1,
               "hash2": hash2
            })
```

/allauthdemo/templates/polls/find_ballot.html

This Django template renders the ballot and hashes in hidden elements:

```
{% extends "bases/bootstrap-with-nav.html" %}
{% load staticfiles %}
{% load bootstrap3 %}

{% block content %}

<pre hidden id="ballot_found">{{ ballot }}</pre>
<pre hidden id="ballot_hashes">{{ hash1 }};{{ hash2 }}</pre>
<h1 id="ballot_result"></h1>

{% endblock %}
```

/static/js/find_ballot.js

The client-side Javascript compares the ballot hash with both of the provided hashes:

```
$( document ).ready(function() {
   ballotHash =
      SHA256Hash(stringtobytes(JSON.stringify($('#ballot-found').text())),
      true);
   var hashes = $('#ballot_hashes').text().split(';');
   $('#ballot_result').text("Ballot not found!");
   for (var i = 0; i <= 1; i++) {
      if (hashes[i] == ballotHash) {
         $('#ballot_result').text("Ballot found!");
      }
   }
});
```

5.1.4 ‘Audit Ballot’ page

This page was intended to allow a voter or a third-party auditor to compare two ballots for equality.

/allauthdemo/polls/views.py

A `vote_audit()` method was added to the view controller which takes two encrypted ballot handles passed as URL parameters and attempts to retrieve them from the MySQL database:

```
def vote_audit(request):
   handle1 = request.GET.get('handle1', None)
   handle2 = request.GET.get('handle2', None)
   vars = {}

   if handle:
      encrypted_ballot1 = get_object_or_404(EncBallot,
          handle=''+urllib.quote_plus(handle1))

   if handle2:
      encrypted_ballot2 = get_object_or_404(EncBallot,
          handle=''+urllib.quote_plus(handle2))

   return render(request, "polls/vote_audit.html",
      {
         "handle1": handle1,
         "ballot1": encrypted_ballot1.ballot,
         "handle2": handle2,
         "ballot2": encrypted_ballot2.ballot
      })
```

/allauthdemo/templates/polls/vote_audit.html

This template produced the page containing any ballots requested and retrieved from the MySQL database:

```
{% extends "bases/bootstrap-with-nav.html" %}
{% load staticfiles %}
{% load bootstrap3 %}

{% block content %}

<label class="gp-1" for="handle1">Ballot #1 handle:</label>
<input class="gp-1" id="handle1" value="{{ handle1 }}" type="text"/>
<label class="gp-1" for="handle2">Ballot handle:</label>
<input class="gp-1" id="handle2" value="{{ handle2 }}" type="text"/>
<button class="gp-1" id="retrieve-ballots">Retrieve Ballots</button>

<hr>

<label class="gp-2" for="ballot1">AES-encrypted ballot #1 from
    server</label>
<pre class="gp-2" id="ballot1">{{ ballot }}</pre>
<input class="gp-2" id="SK1" value="temporary" type="text"/>
<button class="gp-2" id="decrypt-ballot1">Decrypt Ballot</button>
<label class="gp-3" for="ballot-content1">Decrypted ballot</label>
<pre class="gp-3" id="ballot-content1"></pre>
```

```
<label class="gp-3" for="ballot-result1">Ballot encoding</label>
<pre class="gp-3" id="ballot-result1"></pre>

<hr>
<label class="gp-2" for="ballot2">AES-encrypted ballot #2 from
   server</label>
<pre class="gp-2" id="ballot2">{{ ballot2 }}</pre>
<input class="gp-2" id="SK2" value="temporary" type="text"/>
<button class="gp-2" id="decrypt-ballot2">Decrypt Ballot</button>
<label class="gp-3" for="ballot-content2">Decrypted ballot</label>
<pre class="gp-3" id="ballot-content2"></pre>
<label class="gp-3" for="ballot-result2">Ballot encoding</label>
<pre class="gp-3" id="ballot-result2"></pre>

{% endblock %}
```

/static/js/vote_audit.js

If the page is requested without two ballot handles, the option to enter them is displayed. Otherwise, both ballots are displayed and the option to enter an SK for each is offered:

```
$( document ).ready(function() {
   $('.gp-2, .gp-3').hide();
   var n = 0;
   for (var i = 0; i <= 1; i++) {
      if ($('#handle'+i).val() != "") {
         n++;
      }
   }
   if (n == 2) {
      $('ballot-group'+i+' .gp-1').css('opacity','0.6');
      $('#retrieve-ballots').prop('disabled', true);
      $('ballot-group'+i+' .gp-2').show();
   }
});
```

If two handles have not been passed to the page, handles can be submitted by the user. This reloads the page with them inserted into the URL:

```
$('#retrieve-ballots').click(function() {
   window.location =
       "/audit?handle="+$('#handle1').val()+'&handle2='+$('#handle2').val();
});
```

With two ballots loaded and the SKs of each inputted, they can be decrypted in order to determine which candidate they contain a vote or votes for. This code does not work, but the intention was to compute whether $C_2 \div {C_1}^r$ equals g^0 or g^1, with the exponent m being a 1 for a cast vote or a 0 otherwise:

```
$('#begin-test').click(function() {
   var ctx = new CTX("BN254CX");
   var ciphertext = {
      C1: null,
      C2: null,
   r: null
}

var ballot = JSON.parse(sjcl.decrypt($('#SK').val(),
    $('#ballot').text()));

var votes = ballot['encryptedVotes'];
$('#ballot-content').text(JSON.stringify(votes));
var voteNum = 0, optionNum = 0;

// For each encrypted vote within the ballot...
votes.forEach(function(vote) {
   voteNum++;
   $('#ballot-result').text($('#ballot-result').text() + "Vote " +
       voteNum + ": \n ");

   // For each encrypted fragment within the vote (i.e. the encoded
       vote for one option)...
   vote['fragments'].forEach(function(fragment) {
      optionNum++;
      $('#ballot-result').text($('#ballot-result').text() + "Option " +
          optionNum + ": \n ");

      var encoding = "";

      var C1Bytes = getBytes(fragment['C1'].split(","));
      var C2Bytes = getBytes(fragment['C2'].split(","));
      var rBytes = getBytes(fragment['r'].split(","));

      ciphertext.C1 = new ctx.ECP.fromBytes(C1Bytes);
      ciphertext.C2 = new ctx.ECP.fromBytes(C2Bytes);
      ciphertext.r = new ctx.BIG.fromBytes(rBytes);

      // For each pair of C1,C2 values (i.e. one ballot's ciphertext)
          and the randomness used in its encryption r,
      // test whether C2/(C1)^r = g^0 or g^1, and record g's exponent.
      //var c1 = ctx.PAIR.GTpow(ciphertext.C1, ciphertext.r);

      var B;
      var j;
      for (j = 0; j <= 1; j++) {
         //use D as temp var
         B = new ctx.BIG(j);
         D = ctx.PAIR.G1mul(params.g1,B);
         if (D.equals(gM)) {
            return {
               M:j
            }
         }
      };

      encoding += (m) ? "1" : "0";
```

```
            // Somehow, this string of 1s and 0s here needs to become _one_ 1
               or 0 to signify whether the option was
            // voted for or not.
            $('#ballot-result').text($('#ballot-result').text() + encoding +
               "\n ");
        });
    });
});
```

5.2 Android app.

The Android app. was unfortunately not finished, but the broad strokes of the design are present.

5.2.1 Project files

`/app/src/main/AndroidManifest.xml`

The app. manifest declares the package to be `uk.ac.lancaster.auditor` and demands the `CAMERA`, `WRITE_EXTERNAL_STORAGE` and `INTERNET` permissions:

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="uk.ac.lancaster.auditor">
   <uses-permission android:name="android.permission.CAMERA" />
   <uses-permission
       android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
   <uses-permission android:name="android.permission.INTERNET" />

   <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:supportsRtl="true"
      android:theme="@style/AppTheme">

   <activity android:name=".MainActivity">
      <intent-filter>
         <action android:name="android.intent.action.MAIN" />
         <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
   </activity>
   <activity
      android:name=".barcode.BarcodeScanActivity"
      android:label="@string/title_activity_barcode_scan"
      android:theme="@style/AppTheme.NoActionBar"
   />
   <activity
      android:name=".barcode.BarcodeCaptureActivity"
      android:label="@string/title_activity_barcode_capture"
```

```
            android:theme="@style/Theme.AppCompat.NoActionBar"
        />
        <activity
            android:name=".BallotVerifyActivity"
            android:label="@string/title_activity_ballot_verify"
            android:theme="@style/AppTheme.NoActionBar"
        />
        </application>
    </manifest>
```

`/app/build.gradle`

The app. has the following dependencies:

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:design:26.1.0'
    testImplementation 'junit:junit:4.12'
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.google.android.gms:play-services-vision:15.0.2'
    implementation
        "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
}
```

5.2.2 Main Menu

`/res/layout/activity_main.xml`

The main menu uses a `ConstraintLayout` and places two buttons on the screen—one for voters, and one for auditors:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/cast_vote"
        android:layout_width="218dp"
        android:layout_height="64dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="I'm casting a vote"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/audit_election"
        android:layout_width="218dp"
        android:layout_height="64dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="336dp"
        android:text="I'm auditing an election"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/cast_vote" />
</android.support.constraint.ConstraintLayout>
```

/java/.../MainActivity.kt

The main menu code is limited to sending the user to the appropriate Activity for the button they have pressed:

```
package uk.ac.lancaster.auditor

import android.content.Intent
import android.os.Bundle
import android.support.v7.app.\@AppCompatActivity
import android.widget.Button
import uk.ac.lancaster.auditor.barcode.BarcodeScanActivity

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        findViewById<Button>(R.id.cast_vote).setOnClickListener {
            val intent = Intent(applicationContext,
                BarcodeScanActivity::class.java)
            startActivity(intent)
        }
        findViewById<Button>(R.id.audit_election).setOnClickListener {
            val intent = Intent(applicationContext, AuditActivity::class.java)
            startActivity(intent)
        }
    }
}
```

5.2.3 Vote recording

The QR code-scanning vote recording Activity is based on Daniell Algar's BarcodeReader-Sample code, which is released under an MIT license [4].

`/res/layout/activity_barcode_scan.xml`

The barcode scanner layout file is unchanged from the original, presenting a `Button` to begin scanning and a `TextView` to show the scanned text.

`/java/.../barcode/BarcodeScanActivity.kt`

Added to the start of the `Activity` is the structure that will be used to create the LBRF:

```
private lateinit var mResultTextView: TextView
private var stage = 0

data class Record(val recordID: Int) {
   var voterID: String = ""
   var eventID: String = ""
   var pollID: String = ""
   var hashes: String = ""
   var handle: String = ""
   var hmac: String = ""
}

private val record = Record(0)
```

The `onActivityResult()` method is modified to save the relevant scanned data in the `Record` object as each code is scanned:

```
// Store the relevant info from whatever this QR code is for
when (stage) {
   // Stage 0 is getting the voter ID, poll ID and event ID
   0 -> {
      val barcodeVal = barcode.displayValue.split(";")
      record.voterID = barcodeVal[0]
      record.eventID = barcodeVal[1]
      record.pollID = barcodeVal[2]
   }
   // Stage 1 is getting the hashes of both ballots
   1 -> {
      record.hashes = barcode.displayValue
   }
   // Stage 2 is getting the handle of the unselected ballot on the server
   2 -> {
      record.handle = barcode.displayValue.split(";")[0]
      record.hmac =
         String(Base64.decode(barcode.displayValue.split(";")[1],
         Base64.DEFAULT))
      saveRecord()
      finish()
   }
   else -> {
      //something's gone wrong if this gets called
   }
}
stage++
```

The `saveRecord()` method produces a file in the mobile phone's internal memory containing the LBRF:

```
private fun saveRecord() {
   // save the ballot
   val filename = record.handle
   val fileContents = record.toString()
   val outputStream: FileOutputStream

   try {
      outputStream = openFileOutput(filename, Context.MODE_PRIVATE)
      outputStream.write(fileContents.toByteArray())
      outputStream.close()
   } catch (e: Exception) {
      e.printStackTrace()
   }
}
```

Chapter 6

Process Description

This chapter shall demonstrate the process by which a user interacts with the system.

6.1 Voting

When voting, the voter follows the unique link emailed to them to the voting page for the appropriate election event. They indicate their preference(s) as shown in fig. 6.1, then click the ‘Begin Voting’ button to generate the resulting ballots (fig. 6.2). The first QR code displayed (fig. 6.3) contains the voter ID, event ID and poll ID. The second (fig. 6.4) contains the hashes of both ballot copies. The voter is then prompted to choose one of the ballot copies to submit to the server (fig. 6.5). Before this ballot is sent, a confirmation window is shown which allows the voter to audit that their vote has been *recorded-as-intended* (fig. 6.6). Finally, the selected ballot is submitted to the server, the unselected ballot is AES-encrypted with SK and sent to the server and a third QR code is displayed alongside SK which contains the unique handle for the encrypted ballot on the server (fig. 6.7).

6.2 Auditing

On the main menu, the user can choose whether they are casting a vote or auditing an election (fig. 6.8). Choosing the former option, they are shown the QR reader screen (fig. 6.9). For each QR code displayed on the web app., they scan with their mobile device camera (figs 6.10, 6.12 & 6.14) and the results are shown on-screen (figs 6.11, 6.13 & 6.15). At the end, the complete LBRF is saved to the devices internal memory.

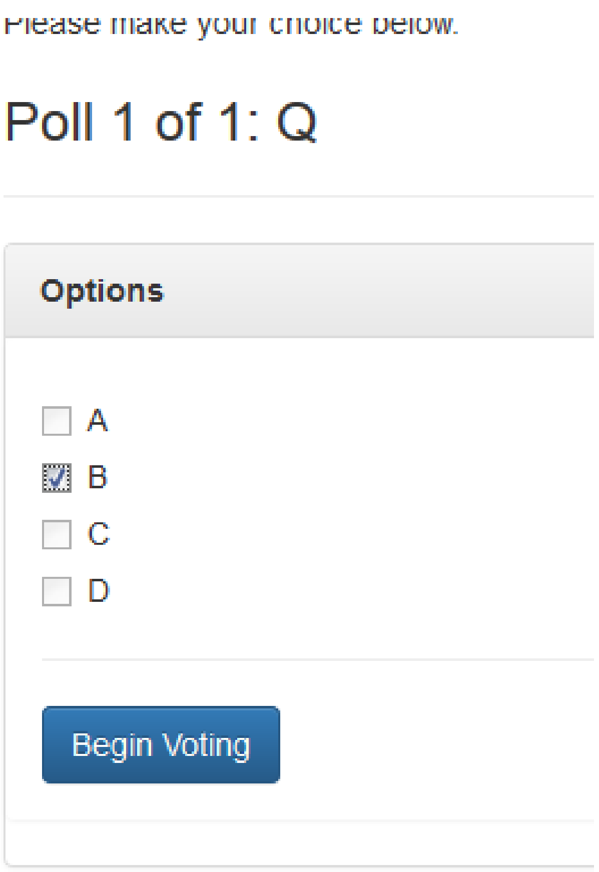
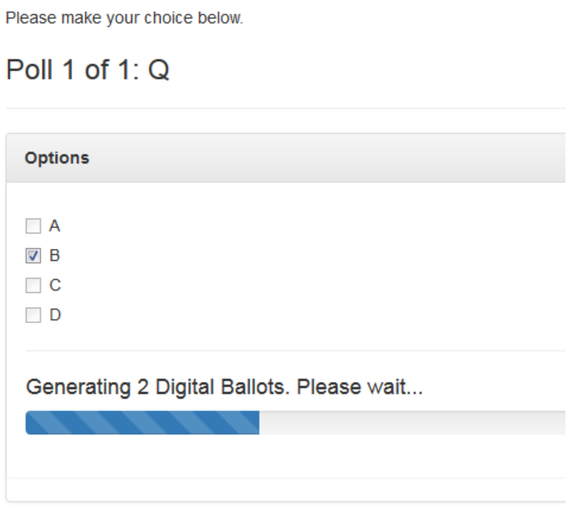
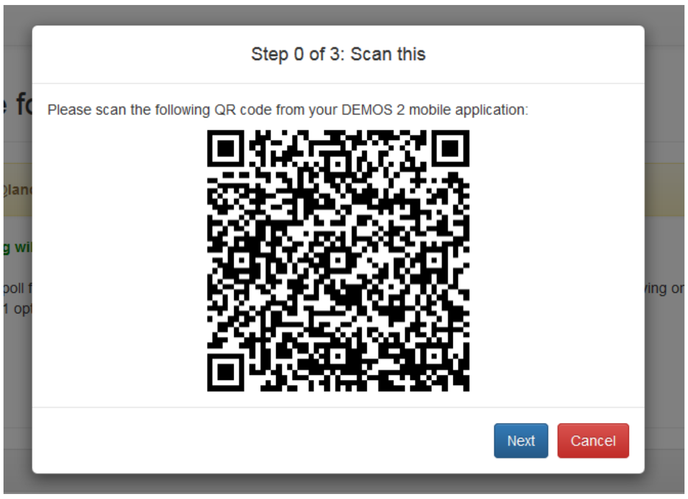


Figure 6.1: Recording a vote



Generating 2 Digital Ballots. Please wait...

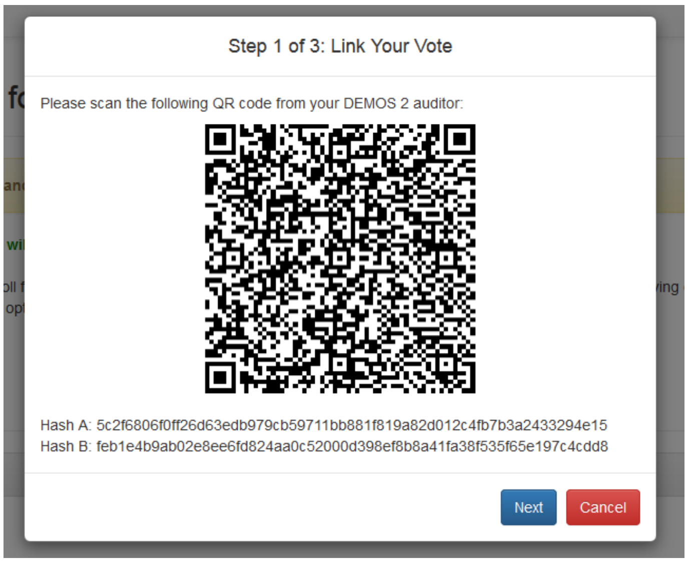
Figure 6.2: Generating ballots



Step 0 of 3: Scan this

Please scan the following QR code from your DEMOS 2 mobile application:

Figure 6.3: QR code 1/3: IDs

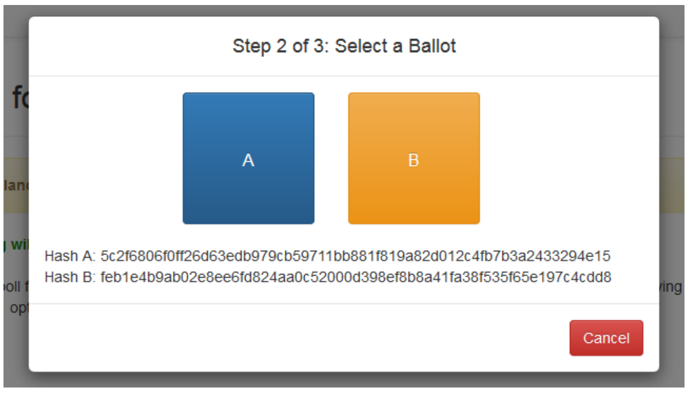


Step 1 of 3: Link Your Vote

Please scan the following QR code from your DEMOS 2 auditor:

Hash A: 5c2f6806f0ff26d63edb979cb59711bb881f819a82d012c4fb7b3a2433294e15
Hash B: feb1e4b9ab02e8ee6fd824aa0c52000d398ef8b8a41fa38f535f65e197c4cdd8

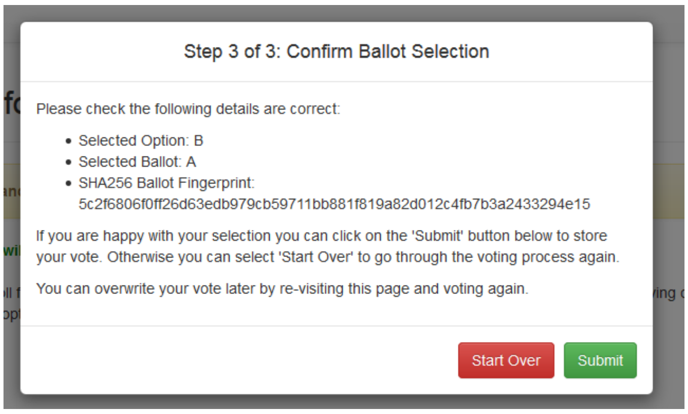
Figure 6.4: QR code 2/3: Ballot hashes



Step 2 of 3: Select a Ballot

Hash A: 5c2f6806f0ff26d63edb979cb59711bb881f819a82d012c4fb7b3a2433294e15
Hash B: feb1e4b9ab02e8ee6fd824aa0c52000d398ef8b8a41fa38f535f65e197c4cdd8

Figure 6.5: Choosing a ballot to submit


Step 3 of 3: Confirm Ballot Selection

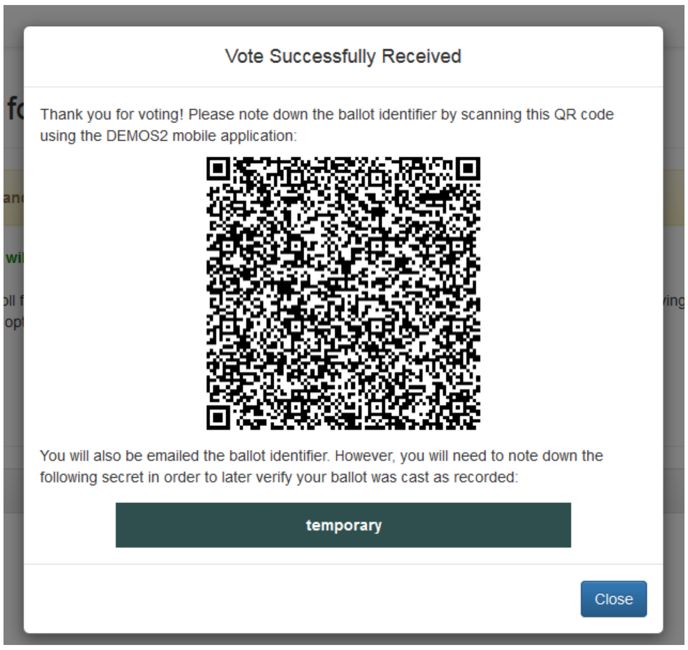
Please check the following details are correct:

- Selected Option: B
- Selected Ballot: A
- SHA256 Ballot Fingerprint: 5c2f6806f0ff26d63edb979cb59711bb881f819a82d012c4fb7b3a2433294e15

If you are happy with your selection you can click on the 'Submit' button below to store your vote. Otherwise you can select 'Start Over' to go through the voting process again.

You can overwrite your vote later by re-visiting this page and voting again.

Figure 6.6: Ballot confirmation


Vote Successfully Received

Thank you for voting! Please note down the ballot identifier by scanning this QR code using the DEMOS2 mobile application:

You will also be emailed the ballot identifier. However, you will need to note down the following secret in order to later verify your ballot was cast as recorded:

temporary

Figure 6.7: QR code 3/3: Encrypted ballot handle

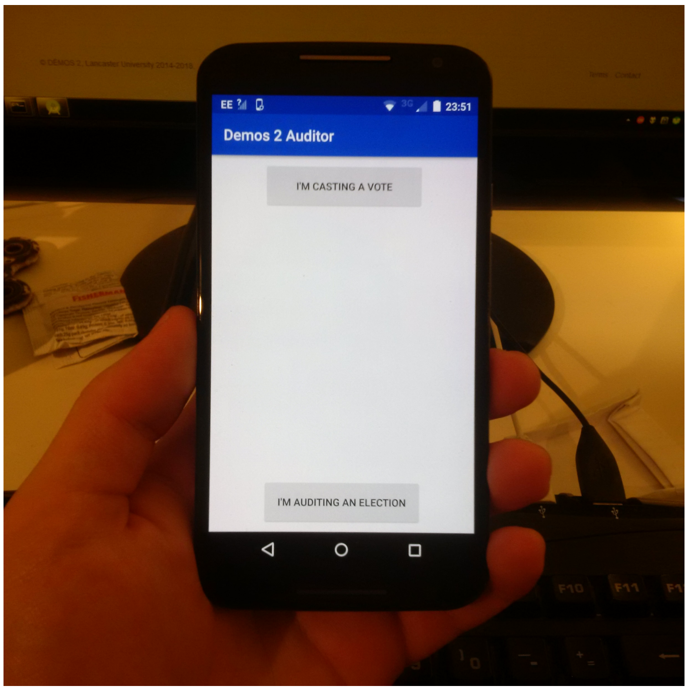


Figure 6.8: App. main menu

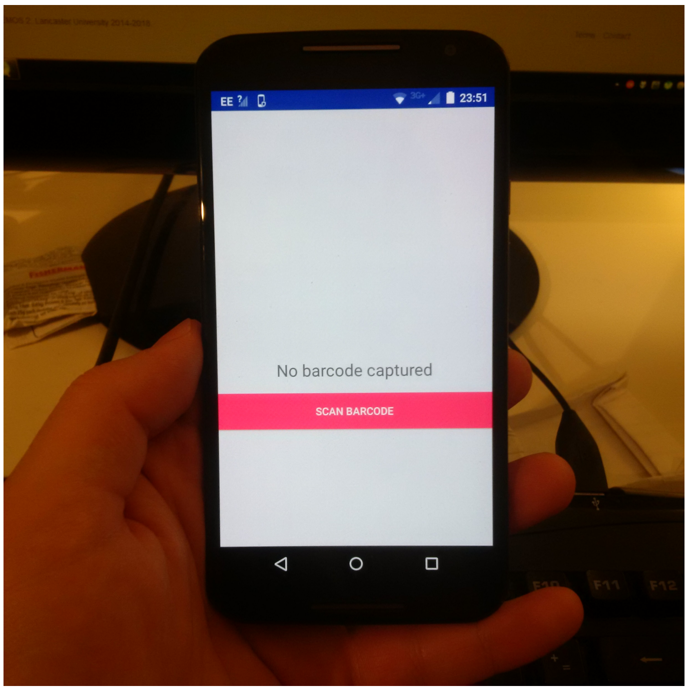


Figure 6.9: ‘Casting a vote’ screen

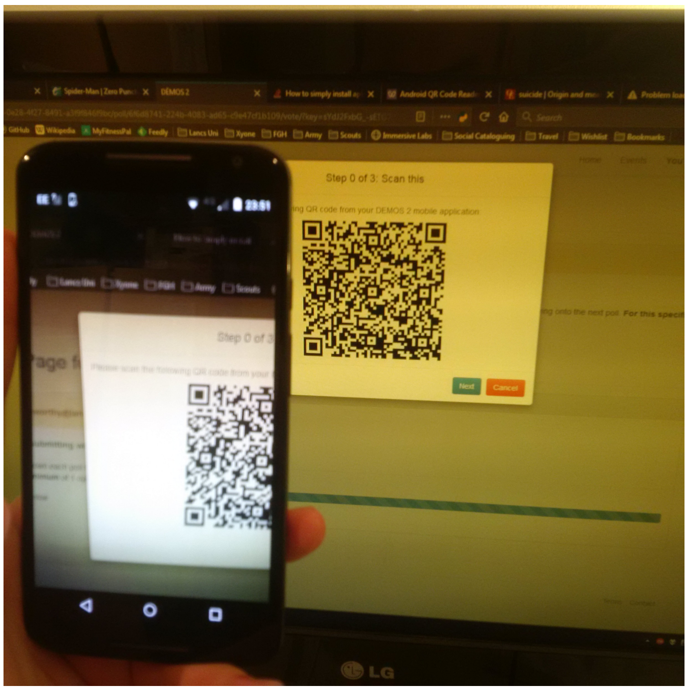
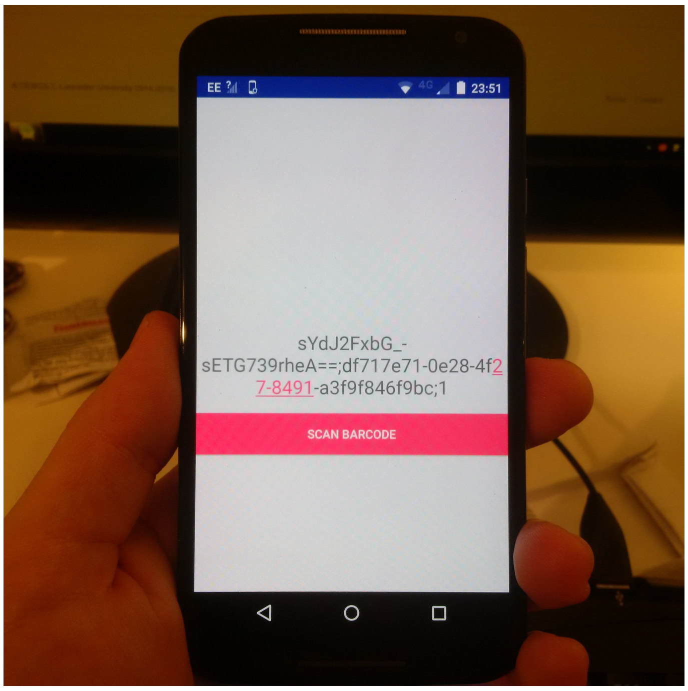


Figure 6.10: Recording QR code 1/3



```
sYdJ2FxbG_-
sETG739rheA==;df717e71-0e28-4f2
7-8491-a3f9f846f9bc;1
```

Figure 6.11: QR code 1/3 result

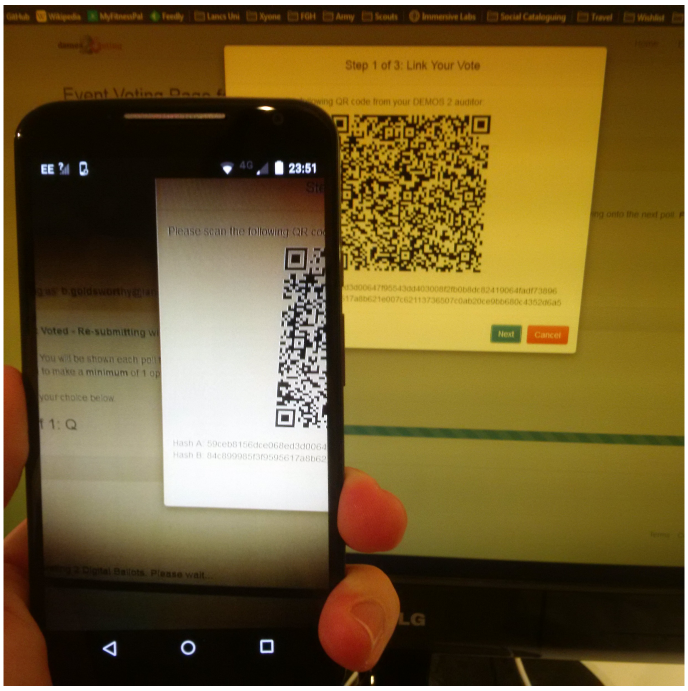
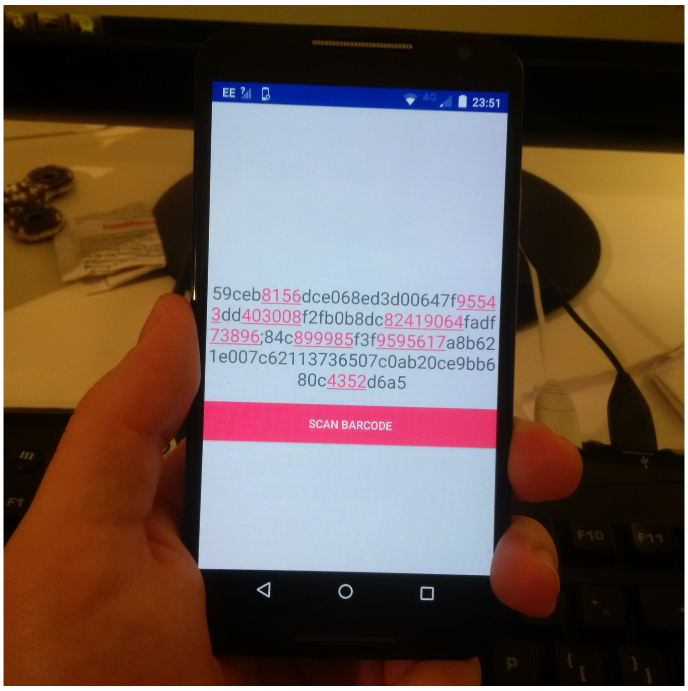

Step 1 of 3: Link Your Vote
Please scan the following QR code
Next
Cancel
23:51

Figure 6.12: Recording QR code 2/3



```
59ceb8156dce068ed3d00647f9554
3dd403008f2fb0b8dc82419064fadf
73896;84c899985f3f9595617a8b62
1e007c62113736507c0ab20ce9bb6
80c4352d6a5
```

Figure 6.13: QR code 2/3 result

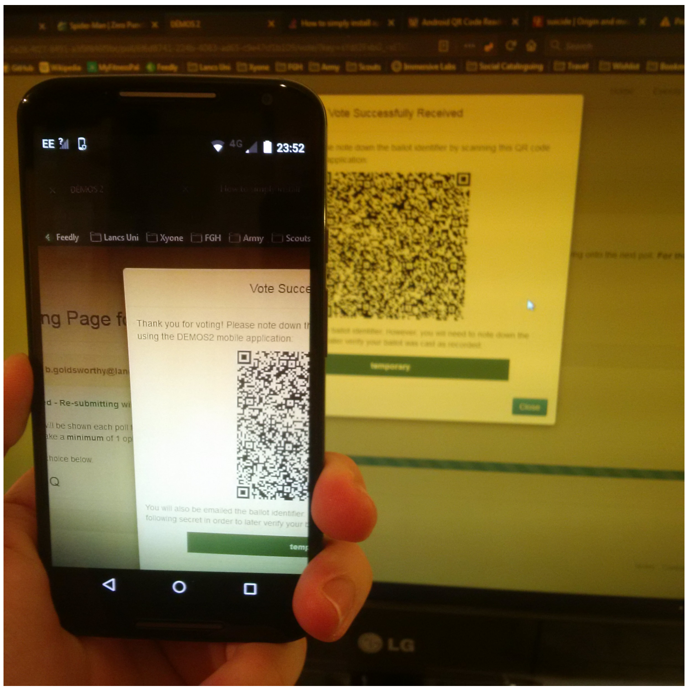
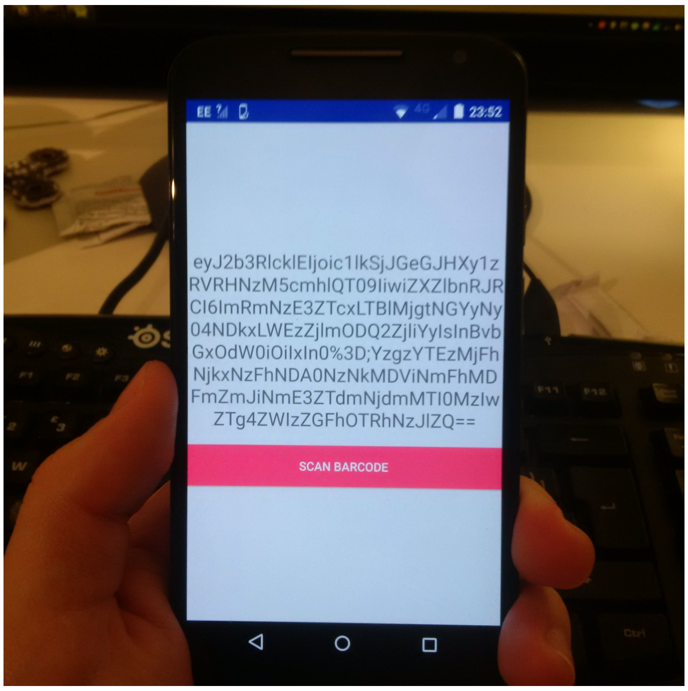

Vote Successfully Received
Vote Succe
Thank you for voting! Please note down t
using the DEMOS2 mobile application
temporary
Close

Figure 6.14: Recording QR code 3/3



```
eyJ2b3RlcklEIjoic1lkSjJGeGJHXy1z
RVRHNzM5cmhlQT09IiwiZXZlbnRJR
CI6ImRmNzE3ZTcxLTBlMjgtNGYyNy
04NDkxLWEzZjlmODQ2ZjliYyIsInBvb
GxOdW0iOiIxIn0%3D;YzgzYTEzMjFh
NjkxNzFhNDA0NzNkMDViNmFhMD
FmZmJiNmE3ZTdmNjdmMTI0MzIw
ZTg4ZWIzZGFhOTRhNzJlZQ==
```

Figure 6.15: QR code 3/3 result

Chapter 7

Testing & Evaluation

Unfortunately, this project did not succeed in producing a fully-functioning implementation of the designed app. in the time alloted. As such, this chapter shall instead feature an overview of the tests that would have been implemented in order to test that the system was functioning correctly, had it been completed.

7.1 Is DẼMOS 2 E2E-auditable?

DẼMOS 2 has been proven to be E2E-verifiable, but without the means by which to do so this means little. Any proposed means (such as a mobile device app.) must be able to do the following;

7.1.1 Verify *recorded-as-intended*

The voter must be able to ensure that the system has recorded their vote as having been for the candidate(s) that they intended. Clearly, only they are able to do so. This has been implemented in the confirmatory window which is displayed during the ballot casting process, which shows them the recorded vote(s) and allows them to approve or reject them.

7.1.2 Verify *cast-as-recorded*

The part of the process in which the recorded vote(s) are turned into a ballot is one in which a malicious actor could attempt to interfere with the process. Therefore, the second property that must be verifiable is that a ballot represents a vote for the candidate(s) the voter recorded votes for. This can be done by allowing them to specify their choice(s), decrypting the ballot and determining whether the vote(s) align.

7.1.3 Verify *counted-as-cast*

The final election result must be auditable to ensure that all ballots have been counted, and that all ballots have been counted *correctly*. One way of doing so may be to allow all ballots to be downloaded by the auditor, who can then decrypt them and tally the result themselves.

7.1.4 Maintain ballot secrecy

As the previous two solutions both involve ballot decryption, means must be implemented to make sure that a malicious actor cannot identify a voter's ballot and choice(s), and that a voter may not share with another party who their vote was for in order to deter vote selling or coercing. However, the fact that a voter can change their vote as many times as they wish may render the second issue moot, as nobody would buy votes that could easily be changed after proof was received.

7.2 Validation

Validation is necessary at many stages of the process, and must be robustly tested with edge cases to detect both errors and malicious tampering.

7.2.1 Cast equals recorded

After showing the voter the confirmation window containing their recorded vote, it must be tested whether malicious client-side Javascript can change the recorded vote in time for ballot generation.

7.2.2 Detect encrypted ballot tampering

An HMAC must be computed on any received encrypted ballots and compared with that stored within the app. in order to detect if any element of the ballot has been tampered with on the server.

7.2.3 Detect missing ballots

The hashes of both generated ballots must be tested to ensure that one of them (and only one) matches that of a ballot submitted to the server.

Chapter 8

Conclusion

Unfortunately, this project experienced limited success, failing to produce a fully-functioning mobile app. that could be used to verify the three properties of E2E-verifiability. In this chapter, the progress that was made is assessed in relation to the original stated requirements. Then, the project as a whole is dicussed and the lessons learned from its failure detailed. After this, suggestions for further development are made, being largely those features that were not successfully implemented here. Finally, we end on a positive note by examining the succeses of the project, such as they are.

8.1 Review of aims & requirements

We recall here the aims outlined in § 1.2:

- to add functionality to DẼMOS 2 to allow a voter to check that their vote has been recorded-as-intended;
- to through adding functionality to DẼMOS 2 and the development of a separate Android mobile application, allow a voter (or a third party to whom they have delegated the task) to verify that their ballot has been cast-as-recorded;
- to as before, allow a voter or third party to verify that the final calculated result of a given election has counted-as-cast all given ballots;
- to ensure that these three features are implemented in such a way that ensures a voter's privacy is protected;
- to ensure that these three features are implemented in such as way as to deter voter coercion, vote buying and vote selling; and
- to document the ways in which which these features are implemented so that future developers are able to easily develop their own independent election auditing tools.

Aim 8.1 was successfully achieved, and some progress was made towards aim 8.1. Unfortunately, however, the functionality was not implemented successfully, which in turn led to aim 8.1 not even being started. As a result, there was no opportunity presented to fulfil aims 8.1 & 8.1. Finally, aim8.1 was achieved for the functionality that *was* successfully implemented, with suggestions on how such future developers may continue the work begun here to be presented in a subsequent section.

We also recall here the requirements outlined in § 4.1:

8.1.1 DẼMOS 2 Requirements

Modifications to the existing DẼMOS 2 system must be made so that:

1. voters MUST be presented with confirmation of their recorded vote in order to allow for *recorded-as-intended* verification;
2. voters MUST be presented with a secure copy of their ballot in a format that can be easily transferred to another device (such as a mobile smartphone);
3. options MUST be provided for the format of this data transfer in order to ensure that all voters are able to transfer it regardless of the technology they possess;
4. voting information MUST NOT pass unencrypted to the server at any point;
5. voters MUST be able to check that their ballot exists on the server;
6. voters and third parties MUST be presented with a means of auditing ballots held on the server in order to allow for both *cast-as-recorded* and *tallied-as-cast* verification; and
7. voters and third parties MUST NOT be able to discern which option(s) a ballot contains a vote for.

8.1.2 App. Requirements

The application must be designed in such a way that:

1. voters MUST be able to receive data presented by the DẼMOS 2 system using their mobile phone;
2. voters MUST be able to store secure copies of their ballots locally;
3. voters MUST be able to transfer their secure ballot copies to third parties;
4. voters and third parties MUST be able to compare their secure ballot copies with the ballot copies stored on the server; and
5. voters and third parties MUST NOT be able to discern which option(s) a ballot contains a vote for.

We shall begin with the DẼMOS 2 requirements, wherein much of the successes of the project lie. Req. 1 was achieved, as was req. 2 (via the LBRFs). Req. 3 was also achieved by adding text displays alongside QR codes, ensuring that all voters would be able to transfer the data they needed for auditing without the need for a smartphone camera.

Req. 4 was adhered to throughout, and req. 5 almost implemented. Req. 6 was designed but not implemented, which meant the circumstances that would have required req. 7 were never present.

Moving on to the app. requirements, req. 1 was achieved via the use of QR codes, but no option was implemented for those without the ability to read them. All other requirements were not implemented.

8.2 Review of project

This project was one part of a one-year, full-time Master's course. Ultimately, the project has not been successful and has failed to produce the planned, fully-functioning software. What *has* been produced is a version of D EMOS 2 modified such that it should be possible for a future developer to produce the planned app. with minimal further changes, as well as the basis of such an app. with which they could begin their project. In this section, the course of the project shall be reviewed and the reasons for its failure—predominantly a consistent lack of focus and motivation on the part of myself—identified.

The project began promisingly. Myself and my supervisor had developed a strong plan in which I would learn about Android app. development and Django over the first term of the Master's course during which other modules were to take priority. This would ensure that I would be able to dive straight into development once the majority of the course had been completed. Unfortunately, deferring the start of the project proper to so far in the future proved to be a bad idea, leading to a very slow start. In addition, for the initial few months of of the second term I remained uncertain as to the project's goal, having believed that the purpose of the app. was to provide a side channel by which a voter could verify their ballot having been cast independently of the device used to cast it, if they believed that the device may have been untrustworthy—i.e., a public library computer.

Ultimately, work on the project did not start in earnest until the end of the course's second term, leaving only a few months for its completion. From this inauspicious beginning, the project never truly recovered. In addition, whilst the completion of the last remaining non-dissertation module should have provided the ideal opportunity for me to devote my full attention to the project, I shortly thereafter began full-time employment, further sapping both the motivation to complete the project and the time available to do so.

High-stakes, but ultimately successful, last-minute project completions are nothing new to me, having been my preferred work style during both my undergraduate and postgraduate study. For this project, however, there was a crucial obstacle to overcome—a lack of motivation, borne of the belief that the project as a whole represented not just a field that was not beneficial for democracy, but may in fact be actively detrimental. I fully accept the responsibility for not having raised these issues at an earlier juncture, perhaps soon enough to change project, and I shall here detail my objections to both e-voting and I-voting.

8.2.1 DRE voting systems considered harmful

Proposed advantage	**Expected tesult**	**Observed result**
Allows everyone to audit elections	Through E2E auditing, any voter or interested third party so delegated can verify an election result to be fair	The number of people able to actually *understand* the verification, and therefore to trust it, is miniscule. In addition, various checks and balances in traditional voting systems produce a system that is *almost* as secure, allow anyone to register to observe the count (or to delegate this responsibility to trusted parties) whilst also being mechanically comprehensible to all.
Reduces cost	DRE voting entails less cost than the use of masses of paper ballots—the UK could apparently stand to save £12.8 million annually [105]	Those savings assume a shift to a completely e-voting system, with no concurrent paper system for auditing purposes. In addition, Australia found the cost to be $1,159 per I-voter, compared to $8.86 per traditional voter [49].
Increases turnout	By making voting more convenient, DRE voting will counter long-term decreasing trends in voter turnout, particularly amongst youth demographics more familiar with live lived through a smartphone—in the UK by as much as 70–79% [105].	Finland found no significant increase in turnout [33]. Other research has concurred with this [13, 109], with one study even finding the greatest increase to have been amongst middle-aged voters.
Protects against corruption	Through the use of cryptography, corrupt election officials can no longer interfere with the running of an election—you can't argue with math.	DRE voting requires that one choose the make and model of system to use, or to make their own. With only a tiny proportion of the electorate able to understand the differences, DRE voting serves to empower corrupt officials to choose known-vulnerable systems or to introduce their own backdoors. Paper-based elections are performed in much the same, vendor-neutral way all over the world, so this is not a possibility.
Allows more eligible voters to vote	As voting can be done remotely, this enables those who cannot make it to a polling station—e.g., the disabled, military personnel overseas, astronauts—to vote.	This can similarly be achieved for most such voters via postal ballots, and the selective introduction of I-voting if need be for tiny, easily verifiable groups such as in-orbit astronauts.

Table 8.1: An overview of the proposed advantages of DRE voting systems in light of real-world experiences

As mentioned in § 2.6, ‘e-voting’ may refer to either a paper-based e-voting system or a DRE voting system. The former category includes voting systems in which only some minor role is played by electronic means, such as marking and counting paper ballots—with the possible advent of self-driving cars in the near future, this could also include the transportation of ballots from polling station to official count in driverless vans. These paper-based e-voting systems are, by and large, benign, although the use of automated ballot marking and counting devices must be paired with a robust means of detecting errors or tampering in either, such as a miscounting machine or an unmarked ballot.

DRE voting systems, on the other hand, are fraught with peril. During the research undertaken for § 2.8, it became clear that almost every country that had experimented with DRE voting systems had run into trouble. Documented issues abound with *specific* (and nonetheless popular) makes of DRE voting systems [41, 36, 67, 32, 44, 7, 80, 61, 79, 57, 110, 86, 73, 94, 70, 96]. I-voting fares even worse, with most pilot schemes having been cancelled shortly following their introduction [99, 49, 33] citing reasons of prohibitive cost or insecurity. Only three countries have continued to run I-voting systems for any considerable length of time, and Switzerland and (most of) the United States offer I-voting only to overseas and in-orbit citizens.

This leaves only Estonia flying the flag for I-voting. Estonia, however, is not like many other countries—through it’s e-Estonia project it has produced a highly networked nation, in which—and, crucially, unlike in the United Kingdom and United States—each citizen possesses a unique and mandatory ID card used for elections, amongst other things. These ID cards give the Estonians a great advantage when attempting to implement I-voting, allowing them to easily authenticate eligible voters and making it harder to cast fraudulant ballots. The level of resistance to the introduction of governmental ID cards in both the UK and the US, however, is as high as it is consistent. Without these, the prospects of successfully introducting nationwide I-voting in either of these countries—the primary foci of this project—are low. Even these ID cards, however, have not been without issue [78].

Implementations may be vulnerable, but the most damning problems with DRE voting systems are fundamental [8]—as table 8.1 lays out. Though traditional paper elections may not be *cryptographically* secure, they are for all intents and purposes *practically* secure. There is no way to interfere noticably with the outcome except with large-scale ballot theft, stuffing or coercion efforts, and if men with guns are stood at the polling station telling you how you have to vote, whether the result is auditable or not is the least of your concerns. Whilst comparing ballot totals across different stages of the tallying, allowing neutral observers to oversee the count and offering candidates the right to request a recount if they believe the count to have been unfair are not *perfect* measures of security, they are certainly *good enough*.

This leads into the most important trade-off with traditional vs DRE voting: an election ‘...must not only *be* fair, but also *be seen* to be fair[, w]ithout a BSc.’ [97] The presence of trust in the electoral system is as vital as that of the electors themselves, and within traditional voting this trust is maintained by virtue of the fact that, even if they are not personally present at the count, the average voter has no problem understanding how the process of recording, collecting and counting paper ballots works, as well as the fact that a range of people from all walks of life—members of all parties and neutral observers—are present, overseeing the count.

Contrast this with DRE voting systems, which rely on complex cryptography to produce a higher level of security at the cost of drastically limiting who can understand the system. Unless the introduction of glsdre voting is accompanied with a scheme to teach the entire electorate the basic tenets of cryptography and zero-knowledge proofs, they will not understand and may consequently lose trust in the system. Though experts may be trusted to verify election integrity, as count observers are currently entrusted to oversee paper counting, the number of cryptographic experts is far smaller than the number of people with functioning eyes. As such, what to do if party x has no supporters with the requistite technical knowledge? Are they to trust party y's experts?

As such, the introduction of DRE voting systems turns the electoral process into an inpenetrable black box to the vast majority of the electorate. Eroding trust is one issue with this; a more pernicious one emerges when one considers that one potential adversary in an election setting is that of a malicious election authority. DRE voting systems accept this risk, and propose as a result that voters should be able to audit that an election has been conducted fairly. However, when one has no understanding of their internal workings, one black box appears much the same as any other. Consider a malicious election authority producing their own 'auditing' app. that does nothing of the sort—who would know? Experts, no doubt, but their warnings regarding commonly-used DRE voting machines (such as DieBold) have thusfar gone largely unheeded, so why should this situation be different?

The theoretical corrupt election authority is further emboldened by the fact that the introduction of a DRE voting system necessarily means the introduction of a *specific* DRE voting system. It requires the decision to use consumer product x rather than consumer product y, rather than a vendor-neutral method of counting marked objects using one's eyeballs. Not only does this mean that each system must have its security verified separately, but it could also allow the supposed corrupt election authority to intentionally choose to implement a DRE voting system known to be vulnerable, or even to produce one itself with intentional backdoors. One cannot design a backdoor into a system that relies on people sorting paper ballots into piles.

So, DRE voting systems appear to be both harmful to the trust on which democracy relies in the name of security, whilst simulteneously enabling corrupt election officials to better interfere with the running of their elections. Additional proposed benefits are those of cost reduction and increased voter turnout, neither of which appear to be supported by the experiences of those countries that have thusfar run pilots [99, 49, 33]. The sole remaining argument in favour of DRE voting is that it allows voters who are otherwise unable to attend a polling station—the disabled, the expatriate, the military, the astronaut—to nonetheless exercise their democratic rights. However, almost all of these voters' needs can be similarly served by the introduction of a postal ballot.

This leaves only those voters currently in orbit—all six of them [107]. Even expanding this to include voters under the sea or in the middle of a desert, the number is miniscule, and the benefit of ensuring them a vote must be weighed against the detriment of undermining the entire electoral system. Even if the decision is made that ensuring their ability to vote must be paramount, there is no argument for extending such an option to all voters. As Texas has shown, astronauts can be granted a unique allowance to vote by email—the tiny number of them should serve to both make tampering with their votes easily detectable and woefully unproductive for the attacker.

In conclusion, then, I concur with the National Academies of Science who wrote in their 2018 report on e-voting that '[q]lections should be conducted with human-readable paper ballots [which] may be marked by hand or by machine (using a ballot-marking device) [and] may be counted by hand or by machine (using an optical scanner)', but no more [69].

8.3 Suggestions for further development

Despite all of these reservations regarding the benefits of DRE voting systems, others are of course welcome to work on them if they disagree. In this section, therefore, suggestions shall be made for how a future developer may build on the work begun here to create the DĒMOS 2 auditing app. that was originally intended.

8.3.1 Add *cast-as-recorded* functionality

Currently, the *recorded-as-intended* functionality shows the voter the option they have recorded a vote for and asks them to confirm it as correct before generating any ballots. However, if the voter was using the system on a malicious device that injected Javascript into the page, it would be possible to change the value of the recorded vote after the voter has confirmed it, in time to generate the ballots. This necessitates the development of *cast-as-recorded* auditing functionality, as detailed in the Design chapter.

8.3.2 Add *counted-as-cast* functionality

Currently, no start has been made on the implementation of *counted-as-cast* functionality, the third element of E2E verifiability.

8.3.3 Introduce validation to QR codes

Currently, there is no validation embedded in the QR codes. This means that whilst the user must scan three QR codes for each section of the LBRF, there is no way of detecting if they have accidentally scanned the wrong one for the current step (e.g., scanning the ballot hash QR code when the app expects the encrypted ballot handle code). This should be added.

8.3.4 Extend browser support for DĒMOS 2

Currently, the qrcode.js library [89] is used to generate all QR codes. Despite its claims to be a '[c]ross-browser QRCode generator', the QR codes do not appear to display on non-Firefox Web browsers. This is obviously an obstacle to the widespread support of DĒMOS 2 and should be resolved, through the use of a different QR code library if need be.

8.3.5 Redirect the user to the app. main menu after scanning all QR codes

After scanning the third and final QR code, the user should be directed back to the main menu.

8.3.6 Save LBRFs to external memory

Android uses the terms ‘internal memory’ to refer to files that are accessible only by the app. that has created them and ‘external memory’ to refer to files that are accessible by other apps, such as file browsers. Currently, LBRFs are saved to the former, but if the intention is to enable them to be passed between interested parties for auditing purposes, this should be changed.

8.4 In closing

In closing, this project has been unsuccessful, but not unproductive. From it I have gained experience in both the Django Web framework and Android app. development, including in the Kotlin language, where previously I had none in either. I have also had a chance to work on software simulteneously with another, becoming more familiar with Git version control in the process. Through the typesetting of this dissertation I have broadened my understanding of LaTeX. Moreover, I have become intimately familiar with the pros and cons of e-voting, which has in turn shifted my opinion on such systems from a cautiously positive one to an overwhelmingly negative one.

Bibliography

[1] Ben Adida. ‘Helios: Web-based Open-Audit Voting.’ In: *USENIX security symposium*. Vol. 17. 2008, pp. 335–348.

[2] Ben Adida and Ronald L Rivest. ‘Scratch & Vote: self-contained paper-based cryptographic voting’. In: *Proceedings of the 5th ACM workshop on Privacy in electronic society*. ACM. 2006, pp. 29–40.

[3] Tomi Ahonen. ‘Almanac 2012: The Mobile Telecoms Industry Annual Review for 2012’. In: *TomiAhonen Consulting* (2013).

[4] Daniell Algar. *BarcodeReaderSample*. 2016. URL: https://github.com/varvet/BarcodeReaderSample (visited on 09/20/2018).

[5] Vincent de Almeida. *DEMOS2*. 2018. URL: https://github.com/vincentmdealmeida/DEMOS2 (visited on 09/03/2018).

[6] Bailrigg F.M. ‘Question Time, Grad Ball, Founders’. *You Ask the SU.* Radio. May 2018. URL: https://www.mixcloud.com/BailriggFM/question-time-grad-ball-founders-you-ask-the-su/ (visited on 07/15/2018).

[7] Jonathan Bannet et al. *Hack-a-vote: Demonstrating security issues with electronic voting systems*. Tech. rep. 2003.

[8] Hagai Bar-El. *Why secure e-voting is so hard to get*. 2015. URL: https://www.hbarel.com/analysis/cyber/secure-e-voting-is-hard-to-get (visited on 09/20/2018).

[9] BBC News. *Does the weather influence elections?* 2010. URL: http://news.bbc.co.uk/2/hi/uk_news/politics/election_2010/8659913.stm (visited on 07/15/2018).

[10] Martin Belam. *Is it illegal to take a selfie while voting in a polling station?* 2017. URL: https://www.theguardian.com/politics/2017/jun/08/is-it-illegal-to-take-a-selfie-while-voting-in-a-polling-station (visited on 07/15/2018).

[11] Susan Bell et al. ‘STAR-Vote: A secure, transparent, auditable, and reliable voting system’. In: *USENIX Journal of Election Technology and Systems (JETS)* 1.1 (2013), pp. 18–37.

[12] George Robert Blakley. ‘Safeguarding cryptographic keys’. In: *Proceedings of the National Computer Conference*. Vol. 48. 1979, pp. 313–317.

[13] Daniel Bochsler. *Can Internet Voting Increase Political Participation? Remote Electronic Voting and Turnout in the Estonian 2007 Parliamentary Elections, Presentation at Internet and Voting Conference Fiesole, 2010*. 2013.

[14] Scott Bradner. *RFC 2119: Key words for use in RFCs to Indicate Requirement Levels*. 1997. URL: https://tools.ietf.org/html/rfc2119 (visited on 09/20/2018).

[15] Anne Broache. *Estonia pulls off nationwide Net voting*. URL: `https://archive.is/20120713045721/http://news.com.com/Estonia+pulls+off+nationwide+Net+voting/2100-1028_3-5898115.html#selection-875.1-875.40` (visited on 09/03/2018).

[16] David Chaum. 'Secret-ballot receipts: True voter-verifiable elections'. In: *IEEE Security & Privacy* 2.1 (2004), pp. 38–47.

[17] David Chaum et al. 'Scantegrity: End-to-end voter-verifiable optical-scan voting'. In: *IEEE Security & Privacy* 6.3 (2008).

[18] Steven Cherry. 'Making every e-vote count'. In: *IEEE Spectrum* 44.1 (2007), pp. 13–14.

[19] Nikos Chondros et al. 'D-DEMOS: A distributed, end-to-end verifiable, internet voting system'. In: *Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on*. IEEE. 2016, pp. 711–720.

[20] City of St. Catharines. *How to Vote*. 2003. URL: `https://web.archive.org/web/20031121211754/http://www.stcatharines.ca/Elections/electionsections/howtovote.asp` (visited on 09/03/2018).

[21] Rodrigo Carneiro Munhoz Coimbra. *O que faz a urna funcionar?* 2018. URL: `http://www.tse.jus.br/o-tse/escola-judiciaria-eleitoral/publicacoes/revistas-da-eje/artigos/revista-eletronica-eje-n.-5-ano-5/digressoes-sobre-as-doacoes-de-campanha-oriundas-de-pessoas-juridicas` (visited on 09/03/2018).

[22] Larissa Conradt and Timothy J Roper. 'Democracy in animals: the evolution of shared group decisions'. In: *Proceedings of the Royal Society of London B: Biological Sciences* 274.1623 (2007), pp. 2317–2326.

[23] Council of Europe, The. *E-voting*. 2016. URL: `https://www.coe.int/en/web/electoral-assistance/e-voting` (visited on 07/15/2018).

[24] Council of Europe, The. *Recommendation CM/Rec(2017)5[1] of the Committee of Ministers to member States on standards for e-voting*. Adopted by the Committee of Ministers on 14 June 2017 at the 1,289th meeting of the Ministers' Deputies. 2017. URL: `https://search.coe.int/cm/Pages/result_details.aspx?ObjectId=0900001680726f6f` (visited on 07/15/2018).

[25] Norman Dalkey and Olaf Helmer. 'An experimental application of the Delphi method to the use of experts'. In: *Management science* 9.3 (1963), pp. 458–467.

[26] Danny De Cock and Bart Preneel. 'Electronic voting in belgium: Past and future'. In: *International Conference on E-Voting and Identity*. Springer. 2007, pp. 76–87.

[27] S Dzieduszycka-Suinat et al. 'The future of voting: end-to-end verifiable internet voting-specification and feasibility study'. In: *US Vote Foundation* (2015).

[28] *E-voting experiments end in Norway amid security fears*. 2014. URL: `https://www.bbc.com/news/technology-28055678` (visited on 09/03/2018).

[29] Economist, The. *Paper cuts*. 2012. URL: `https://web.archive.org/web/20180710121956/https://www.economist.com/international/2012/10/27/paper-cuts` (visited on 07/15/2018).

[30] Rob Eisinga, Manfred Te Grotenhuis, and Ben Pelzer. 'Weather conditions and voter turnout in Dutch national parliament elections, 1971–2010'. In: *International journal of biometeorology* 56.4 (2012), pp. 783–786.

[31] *Electronic voting system to be used in this year's local elections*. 2016. URL: http://business-review.eu/news/electronic-voting-system-to-be-used-in-this-years-local-elections-105256 (visited on 09/04/2018).

[32] Ariel J Feldman, J Alex Halderman, and Edward W Felten. 'Security analysis of the Diebold AccuVote-TS voting machine'. In: (2006).

[33] Finnish Ministry of Justice, The. *Working group: Risks of online voting outweigh its benefits*. 2017. URL: https://oikeusministerio.fi/en/article/-/asset_publisher/tyoryhma-nettiaanestyksen-riskit-suuremmat-kuin-hyodyt (visited on 07/15/2018).

[34] Katie Forster. *UK election: Registered voters turned away in one of the most marginal seats in country*. 2017. URL: https://www.independent.co.uk/News/uk/politics/uk-election-voters-turned-away-marginal-seat-plymouth-tory-labour-seat-postal-vote-a7780146.html (visited on 07/15/2018).

[35] Chris Game. *Explainer: how Britain counts its votes*. 2015. URL: https://theconversation.com/explainer-how-britain-counts-its-votes-41265 (visited on 07/15/2018).

[36] Jonathan Gitlin. *Frozen machines, 243-percent turnout, and other woes in Georgia voting*. 2018. URL: https://arstechnica.com/tech-policy/2018/08/georgia-defends-voting-system-despite-243-percent-turnout-in-one-precinct/ (visited on 09/18/2018).

[37] Ben Goldsworthy. *DEMOS2*. 2018. URL: https://github.com/Rumperuu/DEMOS2 (visited on 09/20/2018).

[38] Ben Goldsworthy. *DEMOS2Auditor*. 2018. URL: https://github.com/Rumperuu/DEMOS2Auditor (visited on 09/20/2018).

[39] Nicole Goodman. *Issues Guide: Internet Voting*. 2012. URL: https://www.edmonton.ca/city_government/documents/PDF/Internet_Voting_Issues_Guide_December_21_2012.pdf (visited on 09/03/2018).

[40] Martin Hallberg and Maija Karjalainen. *Serial numbers on ballots*. 2010. URL: http://aceproject.org/electoral-advice/archive/questions/replies/912993749 (visited on 07/15/2018).

[41] Paul S Herrnson et al. 'The current state of electronic voting in the United States'. In: *Digital Government*. Springer, 2008, pp. 157–180.

[42] Brenton Holmes. *E-voting: The promise and the practice*. 2012. URL: https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/pubs/BN/2012-2013/EVoting%5C#_Toc338074390 (visited on 09/03/2018).

[43] Matt Holt. *Papa Parse - Powerful CSV parser for JavaScript*. 2013. URL: https://www.papaparse.com/ (visited on 09/18/2018).

[44] Harri Hursti. *Critical security issues with Diebold optical scan design*. Tech. rep. Black Box Voting, Inc., 2005.

[45] IDC. *Smartphone OS Market Share*. 2017. URL: https://www.idc.com/promo/smartphone-market-share/os (visited on 09/13/2018).

[46] Internet Rights Forum, The. *What is the Future of Electronic Voting in France?* 2003. URL: https://web.archive.org/web/20070927230856/http://www.foruminternet.org/telechargement/documents/reco-evote-en-20030926.pdf (visited on 09/03/2018).

[47] Martin James. *Managing UK elections - could technology help prevent voting issues?* 2017. URL: https://www.itproportal.com/features/managing-uk-elections-could-technology-help-prevent-voting-issues/ (visited on 07/15/2018).

[48] Terrell Johnson. *Bad Weather on Election Day? Many Won't Vote*. 2012. URL: https://weather.com/news/news/bad-weather-election-turnout-20120924 (visited on 07/15/2018).

[49] Joint Standing Committee on Electoral Matters. *Report on the 2007 federal election electronic voting trials: Interim report of the inquiry into the conduct of the 2007 election and matters related thereto*. URL: https://www.aph.gov.au/Parliamentary_Business/Committees/House_of_Representatives_Committees?url=em/elect07/report.htm (visited on 09/03/2018).

[50] Douglas W Jones. 'Kazakhstan: The Sailau e-voting system'. In: ed. by Michael Yard. International Foundation for Electoral Systems, 2010, pp. 74–95.

[51] *jQuery*. 2008. URL: https://jquery.com/ (visited on 09/18/2018).

[52] Alexander Kendall. *The Partisan Effects of Voter Turnout*. 2004.

[53] Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. 'DEMOS-2: scalable E2E verifiable elections without random oracles'. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2015, pp. 352–363.

[54] Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. 'End-to-end verifiable elections in the standard model'. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2015, pp. 468–498.

[55] Mirosław Kutyłowski and Filip Zagórski. 'Scratch, Click & Vote: E2E voting over the Internet'. In: *Towards trustworthy elections*. Springer, 2010, pp. 343–356.

[56] Zengpeng Li. *Overview of the Estonian e-voting architecture*. Personal conversation. 2018.

[57] Jan Libbenga. *Dutch pull the plug on e-voting*. 2007. URL: https://www.theregister.co.uk/2007/10/01/dutch_pull_plug_on_evoting/ (visited on 09/03/2018).

[58] Christian List. 'Democracy in animal groups: a political science perspective'. In: *Trends in Ecology & Evolution* 19.4 (2004), pp. 168–169.

[59] *Lithuanian government seeks to introduce online voting this year*. 2017. URL: https://www.baltictimes.com/lithuanian_government_seeks_to_introduce_online_voting_this_year/ (visited on 09/03/2018).

[60] Rob Lundie. *Electronic voting at federal elections*. 2017. URL: https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/pubs/BriefingBook45p/ElectronicVoting (visited on 09/03/2018).

[61] Alastair McIndoe. *Philippines on edge*. 2010. URL: https://web.archive.org/web/20100509211221/http://www.straitstimes.com/BreakingNews/SEAsia/Story/STIStory_523723.html/ (visited on 09/03/2018).

[62] John McLean. *Philippine election results: Voters test new electronic voting system*. 2010. URL: https://www.csmonitor.com/World/Asia-Pacific/2010/0510/Philippine-election-results-Voters-test-new-electronic-voting-system (visited on 09/03/2018).

[63] Tim Meisburger. *Korean Elections: A Model of Best Practice*. 2016. URL: https://asiafoundation.org/2016/04/20/korean-elections-a-model-of-best-practice/ (visited on 09/18/2018).

[64] *milagro-crypto-js*. 2016. URL: https://github.com/milagro-crypto/milagro-crypto-js (visited on 09/18/2018).

[65] Steve Muhlberger. *Democracy in ancient India*. 1998.

[66] *MySQL*. 1998. URL: https://www.mysql.com/ (visited on 09/18/2018).

[67] Ellen Nakashima. *In Georgia, a legal battle over electronic vs. paper voting*. 2018. URL: https://web.archive.org/web/20180916184236/https://www.washingtonpost.com/world/national-security/in-georgia-a-legal-battle-over-electronic-vs-paper-voting/2018/09/16/d655c070-b76f-11e8-94eb-3bd52dfe917b_story.html (visited on 09/18/2018).

[68] *Namibian election first in Africa to use electronic voting machines*. 2014. URL: http://www.abc.net.au/news/2014-11-28/namibian-election-first-in-africa-to-use-electronic-voting/5927206 (visited on 09/03/2018).

[69] National Academies of Sciences, Engineering, and Medicine. *Securing the Vote: Protecting American Democracy*. Tech. rep. National Academies Press, the, 2018. URL: https://www.nap.edu/catalog/25120/securing-the-vote-protecting-american-democracy (visited on 09/20/2018).

[70] National Democratic Institute. *The Constitutionality of Electronic Voting in Germany*. 2016. URL: https://www.ndi.org/e-voting-guide/examples/constitutionality-of-electronic-voting-germany (visited on 09/03/2018).

[71] *Node.js*. 2009. URL: https://nodejs.org/en/ (visited on 09/18/2018).

[72] NPR. *Exploring The Link Between Weather, Elections*. 2008. URL: https://www.npr.org/templates/story/story.php?storyId=96476912 (visited on 07/15/2018).

[73] Marie O'Halloran and Michael O'Regan. *E-voting machines to be disposed of*. 2010. URL: https://www.irishtimes.com/news/e-voting-machines-to-be-disposed-of-1.865193 (visited on 09/03/2018).

[74] Eli Okun. *Travis County Forges New Territory in Creating Voting Machine*. 2014. URL: https://www.texastribune.org/2014/07/09/travis-county-forges-new-territory-voting-machines/ (visited on 08/30/2018).

[75] John Penrose. *UK Government response: Full text*. 2016. URL: https://webrootsdemocracy.org/uk-govt-response-full/ (visited on 09/18/2018).

[76] Mikael Persson, Anders Sundell, and Richard Öhrvall. 'The weather does not affect voter turnout, but only if voting is convenient for the public'. In: *LSE American Politics and Policy* (2014).

[77] Alessandro Pluchino, Andrea Rapisarda, and Cesare Garofalo. 'The Peter principle revisited: A computational study'. In: *Physica A: Statistical Mechanics and its Applications* 389.3 (2010), pp. 467–472.

[78] *Possible security risk affects 750,000 Estonian ID-cards*. 2017. URL: http://estonianworld.com/technology/possible-security-risk-affects-750000-estonian-id-cards/ (visited on 09/20/2018).

[79] *Presidential poll under threat from faulty voting machines*. 2010. URL: http://en.rfi.fr/asia-pacific/20100505-presidential-poll-under-threat-faulty-voting-machines/ (visited on 09/03/2018).

[80] Caleb Pritchard. *STAR-Vote collapses*. 2017. URL: https://www.austinmonitor.com/stories/2017/10/star-vote-collapses/ (visited on 09/13/2018).

[81] Cristina Prina Ricotti. *Italy kicks off all e-voting pilot in Salento*. 2013. URL: https://www.zdnet.com/article/italy-kicks-off-all-e-voting-pilot-in-salento/ (visited on 09/03/2018).

[82] Harry Ridgewell. *Why aren't we all voting online?* 2018. URL: https://www.wikitribune.com/story/2018/06/26/internet-tech/why-arent-we-all-voting-online/75127/ (visited on 07/15/2018).

[83] Ronald Rivest. *Perspectives on "End-to-End" Voting Systems*. 2009. URL: https://csrc.nist.gov/CSRC/media/Events/End-to-End-Voting-System-Workshop/documents/Rivest-NIST-E2E-keynote.pdf (visited on 08/31/2018).

[84] Ronald L Rivest. 'The ThreeBallot Voting System'. In: (2006).

[85] Peter YA Ryan et al. 'The Prêt à Voter verifiable election system'. In: *IEEE transactions on information forensics and security* 4.4 (2009), pp. 662–673.

[86] *Sailau e-system will not be used at Kazakhstan parliamentary elections in 2012*. 2011. URL: https://en.tengrinews.kz/politics_sub/Sailau-e-system-will-not-be-used-at-Kazakhstan-parliamentary-5678/ (visited on 09/03/2018).

[87] Thomas D Seeley. *Honeybee Democracy*. Princeton University Press, 2010.

[88] Adi Shamir. 'How to share a secret'. In: *Communications of the ACM* 22.11 (1979), pp. 612–613.

[89] David Shim. *qrcode.js*. 2012. URL: https://davidshimjs.github.io/qrcodejs/ (visited on 09/20/2018).

[90] Anuradha Shukla. *Indians Vote Via Web with Scytl Technology*. 2011. URL: http://www.biztechreport.com/story/1318-indians-vote-web-scytl-technology (visited on 09/03/2018).

[91] Ask Solem. *Homepage | Celery: Distributed Task Queue*. 2007. URL: http://www.celeryproject.org/ (visited on 09/18/2018).

[92] Speaker's Commission on Digital Democracy, The. *Open Up!: Report of the Speaker's Commission on Digital Democracy*. 2015.

[93] Marian Stoica and Bogdan Ghilic-Micu. 'E-Voting Solutions for Digital Democracy in Knowledge Society'. In: *Informatica Economica* 20.3 (2016).

[94] Supreme Court of India. *A.C. Jose vs. Sivan Pillai and ors.* 1984. URL: legalcrystal.com/654583 (visited on 09/03/2018).

[95] *The Web framework for perfectionists with deadlines | Django*. 2005. URL: https://www.djangoproject.com/ (visited on 09/18/2018).

[96] Leigh Thomas. *France drops electronic voting for citizens abroad over cybersecurity fears*. 2017. URL: https://www.reuters.com/article/us-france-election-cyber-idUSKBN16D233 (visited on 09/03/2018).

[97] thomasfedb. *Voting must not only be fair, but also able to be seen to be fair. Without a... | Hacker News*. 2018. URL: https://news.ycombinator.com/item?id=17976054 (visited on 09/18/2018).

[98] James Titcomb. *Is it illegal to post my election vote on Facebook and Twitter?* 2017. URL: https://www.telegraph.co.uk/technology/2017/06/08/illegal-post-election-vote-facebook-twitter/ (visited on 07/15/2018).

[99] Alice Truong. *Norway Ends Online Voting For Local And National Elections*. 2014. URL: https://www.fastcompany.com/3032497/norway-ends-online-voting-for-local-and-national-elections (visited on 09/03/2018).

[100] US Department of State. *Democracy*. 2012. URL: https://www.state.gov/j/drl/democ/ (visited on 07/12/2018).

[101] Valimised. *Statistics about Internet Voting in Estonia*. URL: `https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia` (visited on 09/03/2018).

[102] Peter Walker and Matthew Weaver. *Anger and confusion as voters turned away during ID trial*. 2018. URL: `https://www.theguardian.com/politics/2018/may/03/uk-voters-turned-away-polling-stations-id-trial` (visited on 07/15/2018).

[103] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. 'Finding collisions in the full SHA-1'. In: *Annual international cryptology conference*. Springer. 2005, pp. 17–36.

[104] Max Weber. *Weber's Rationalism and Modern Society*. Ed. by Tony Waters and Dagmar Waters. Palgrave Books, 2015, pp. 129–198.

[105] WebRoots Democracy. *Viral Voting: Future-proofing UK elections with an #onlinevoting option*. URL: `https://webrootsdemocracy.files.wordpress.com/2014/05/webroots-democracy-viral-voting.pdf` (visited on 07/15/2018).

[106] *What happens to the voting slips used in British elections after they have been counted?* 2013. URL: `https://www.theguardian.com/notesandqueries/query/0,,-1051,00.html` (visited on 07/15/2018).

[107] *Who Is Currently In Space?* 2017. URL: `https://www.worldspaceflight.com/bios/currentlyinspace.php` (visited on 09/20/2018).

[108] *Who is eligible to vote at a UK general election?* 2014. URL: `http://www.electoralcommission.org.uk/faq/voting-and-registration/who-is-eligible-to-vote-at-a-uk-general-election` (visited on 09/06/2018).

[109] Tove Wigartz. 'Does internet voting in Estonia affect voter turnout?' In: (2017).

[110] Wij Vertrouwen Stemcomputers Niet. *The Netherlands return to paper ballots and red pencils*. 2007. URL: `http://wijvertrouwenstemcomputersniet.nl/English` (visited on 09/03/2018).

[111] Carey Williams. *DEMOS2*. 2017. URL: `https://github.com/CareyJWilliams/DEMOS2` (visited on 09/03/2018).

[112] Gordon Winter. *Inside BOSS: South Africa's Secret Police*. Penguin Books Ltd, 1981.

[113] World Bank, The. *World Development Report 2017: Governance and the Law*. URL: `http://www.worldbank.org/en/publication/wdr2017` (visited on 07/15/2018).

[114] Kelsey Ziomek. *How many democratic nations are there?* 2013. URL: `http://www.borgenmagazine.com/many-democratic-nations/` (visited on 07/12/2018).