Is your vote overheard? A new scalable side-channel attack against paper voting

Kristjan Krips*†, Jan Willemson*‡, Sebastian Värv*
**Cybernetica*, Ülikooli 2, Tartu, Estonia
Email: {kristjan.krips, jan.willemson}@cyber.ee, sebastian.varv@gmail.com
†*Institute of Computer Science*, University of Tartu, J. Liivi 2, Tartu, Estonia
‡*STACC*, Ülikooli 2, Tartu, Estonia

***Abstract*—In an ongoing discussion comparing the security properties of electronic and paper voting, decreased privacy is often presented as an argument against remote Internet voting. We contribute to this discussion by presenting a side-channel attack against the physical environment of traditional paper-based elections. More precisely, we build a device based on an Arduino development board and cheap electret microphones, capable of triangulating the locations of marks made on wooden tables with high precision. In the best configuration, we are able to determine the correct cell having dimensions 4×5 cm with more than 90% accuracy. This will allow breaching privacy of ballot sheet designs that rely on the voter marking her choice(s) between a potentially high number of candidates printed on one large sheet. We complement our attack with a study on various aspects of deployment of facial recognition. This gives rise to the setup where the attacker installs cameras in the polling stations, aiming at automated detection of people leaving the voting booths. Combining the two approaches, we will have a completely automated (and hence relatively well scalable) attack against the privacy of paper-based voting.**

***Index Terms*—Voting, privacy, sonic testing, facial recognition.**

I. Introduction

Voting is a form of public opinion polling that forms the core of democratic society. Throughout the millennia of experiences, a number of requirements have been established that a voting system should satisfy in order to accurately capture the societal preferences. The exact legal framework varies from country to country, but the requirements typically include

- eligibility (only the persons with a right to vote should be allowed to do so),
- uniformity (everyone has the same number of allowed votes),
- generality (eligible voters should have access to voting capability),
- freedom (voters should be able to express their true preferences without being coerced or otherwise illegally influenced).

These requirements, in turn, are translated into the technical properties of the voting system being utilised. For example, checking eligibility assumes reliable voter lists and person identification mechanisms. Uniformity and generality rely on integrity measures like securing the ballot boxes and verifying the final tally by recounts or post-election audits. Voting freedom, on the other hand, is typically implemented via ballot secrecy.

Different voting methods are able to meet these requirements to varying levels. Remote Internet voting, for instance, has been criticised repeatedly for inability to provide a coercion-free vote casting environment, since breaking the vote secrecy is relatively easy in an uncontrolled remote location like voter's home [1]–[6]. Physical polling stations, on the other hand, are designed to enforce privacy by means of shielded booths.

Vote casting within the booth can happen in several ways. The most established method is marking one's choice(s) on a sheet of paper and putting it into a ballot box. However, the ballot sheets may sometimes be rather large and the logic of filling them may be complicated [7]. Also, marking ballots by hand can be rather error-prone [8].

To address these issues, many assisting technical tools have been developed and tried throughout the years. Such assistants include punch card devices and lever machines [9]; more advanced apparatus can be used to prepare the ballot sheets to be printed out [7] or record votes digitally (so-called direct-recording electronic (DRE) machines).

The history of DRE equipment is rich with unintended vulnerabilities, poor design choices and the resulting attacks [10]–[16]. To a certain extent, such problems have discredited the whole idea of using machinery in the process of vote casting. As a result, there exist entire communities devoted to promoting paper-based elections over electronic alternatives (like https://www.verifiedvoting.org/ and http://handcountedpaperballots.org/).

However, it is the belief of the authors of the current paper that the members of such communities tend to underestimate the vulnerabilities of paper voting and the effect that technological advancements have in terms of its security level.

There are many well-documented ways to attack integrity of the results of paper voting, including ballot box stuffing [17], disappearing ink [18], setting up fake ballot boxes or stealing genuine ones [19], etc. Also, many generic election attacks like gerrymandering or confusing the voters by setting up similarly-named candidates apply to paper voting [18].

Alvarez and Hall show that the need to delegate operating polling stations to numerous semi-reliable local agents may lead to problems with both integrity and availability (i.e. generality) of paper-based elections [20].

However, in this paper we are going to concentrate on vote privacy which is the key measure to achieve voting freedom and coercion-resistance. We will present a new attack that can be implemented against the ballot sheet designs where the voter is asked to mark her preference(s) from a large number candidates printed on one sheet. The attack works both in case of simple markings and preferential voting.

The paper is organised as follows. First, Section II gives a short overview of side channel attacks against paper voting, followed by the description of the threat model in Section III and the discussion of some ballot sheet designs leading to the idea of our attack in Section IV. Section V describes the experimental setup, followed by the description of our experiments and experimental results in Sections VI and VII, respectively.

Second part of the paper is devoted to complementing our (this far anonymous) attack with a mechanism to identify voters using facial recognition. Section VIII discusses the general framework for such a mechanism, followed by the descriptions of the options for deploying the actual hard- and software in Section IX, and the methods of obtaining the necessary facial image database in Section X.

Finally, Section XI discusses possible countermeasures, and Section XII draws some conclusions and sets directions for future work.

II. State of the Art

First evidence of secret ballot as a democratic mechanism goes back to ancient civilisations. For example, the respective legislation was passed in the late Roman Republic in 139 BC. In today's understanding, this is seen as a measure for lessening the control of the upper classes over the electorate, and enhancing the voters' effective freedom of choice [21].

Wider acceptance of secret vote in Western democracies took place throughout the 19th century, and today this principle is considered so fundamental that it is even stated in Article 21.3 of the Universal Declaration of Human Rights.

The choice for the actual technique of ballot filling is left to the election organisers. As mentioned above, various methods for that have been experimented with throughout the history of elections. Unfortunately, not all of them are equally resistant to privacy violations.

Of course, an attacker can always coerce the voter to take a *stemfie* (selfie together with one's filled ballot sheet) [22], perform the so-called Italian attack where the voter is required to vote in a pre-determined pattern [23], or set up chain voting [24]. However, these scenarios assume voter's knowledge and active participation in the attack. We argue that stealthy privacy violations pose potentially even a greater risk to democratic freedom. When the voter learns about breaking the privacy of her true preference only after the voting, she has no real options to prevent or fix the problem as she can not take the vote back or decide not to go voting at all.

On the other hand, even in case of informed voter collaboration (e.g. for vote buying/selling), a stealthy attack has noticeable benefits for the attacker. If he can just set the attack up once and later observe the voting process without requiring the voters do deviate from the standard actions in any way, he will decrease the risk of being detected.

Perhaps one of the best documented stealthy privacy vulnerabilities from the recent years is the side channel attack implemented by Gonggrijp and Wessling against the voting machines used in the Netherlands [25]. In their attack, voter's party preference was in some cases leaked via electromagnetic emanations from the machine. We refer to [26], [27] for a thorough description of the problem along with the political context and aftermath.

However, paper voting is not free from side channel leakages either. Fingerprinting the ballot sheets using a high-resolution scanner or photo camera was proposed by Calandrino *et al.* [28]. The basic technology needed for that was recently improved by Toreini *et al.*, making the resulting attack more efficient and accessible for even a moderately-resourced attacker [29]. We refer to the recent paper of Krips *et al.* [30] for a comprehensive overview of side-channel attack against paper voting.

In the same paper, Krips *et al.* propose a new audio side channel of ballot marking where the voter is expected to fill the ballot sheet by writing numbers (say, a candidate number or preferential order). Even though their rate of detecting the numbers from the sound of writing was rather good, applicability of this attack is limited to rather specific types of ballots. Such ballots form a relatively small portion of the designs used around the world.

In this paper, we are going to develop a side channel against a considerably larger selection of ballot sheets.

III. Threat model

As stated above, vote privacy breaches play a key role in various coercion attacks. In this paper, we have two types of coercive attacks in mind as use cases.

- *Vote buying:* In this scenario, the voter has had a prior contact with the coercer and has knowledge about the ongoing attack. However, the exact method of verifying compliance to the coercer's demands is not necessarily known to the voter.
- *Political persecution:* In this scenario, the attacker gathers vote information without the voter's prior knowledge and uses it later to harass her because of her political preferences.

We see that in both of the cases it is important for the attacker to keep the attack apparatus stealthy.

IV. Ballot sheet designs and attack idea

Design of ballot sheets is on one hand a well-studied, but on the other hand also a very sensitive topic. For example, it has been observed that the candidates listed on top of the sheet are more likely to score higher results [31]–[33].

In our research, however, we are more interested in the way the voter expresses her preference. As the number of candidates is typically large (easily reaching hundreds), the voter must be able to select from them. Ballot design-wise, the simplest option would be to have the ballot blank and letting the voter to write in a name or number for the party or candidate, but such designs are relatively rare, being found in less than ten countries around the world [34].

A considerably more popular option is listing all the running candidates on one sheet and asking the voter to select one (or several) of them. Combined with the potentially large number of candidates, we will obtain potentially large ballot sheets, too.

For example, the sheet from Dutch elections of 2017 being about 1.5 meters wide is shown in Figure 1.

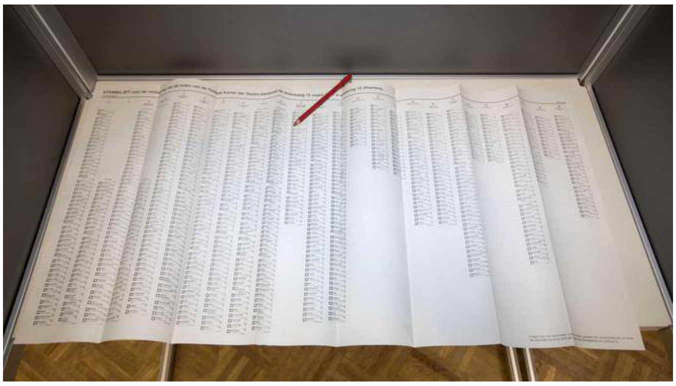


Fig. 1. Dutch ballot sheet (image source: https://imgur.com/gallery/F8EmD)

A similar situation occurs in Australia where e.g. in 2013 Victorian Senate elections 39 parties (plus independents) were running and the ballot paper extended to 102 cm in width.[1]

As a third example, some local elections in Germany feature huge ballot sheets as well [7] (see Figure 2).

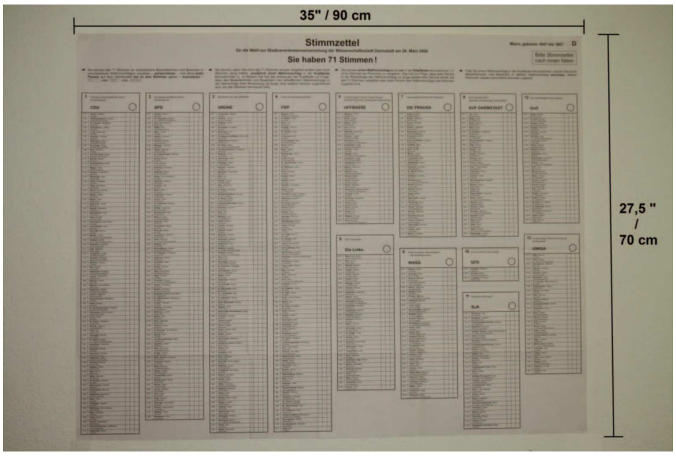


Fig. 2. German local election ballot sheet [7]

These cases may be considered as quite extreme, but they lead us naturally to the main idea of our attack. Namely, if the whole table provided in the polling booth is covered by the ballot sheet from edge to edge, then we could reveal the voter's preference if we would be able to detect the location of the sound the pen makes when marking the ballot. As we will see below, such a detection may be implemented very cheaply, but at the same time with high accuracy making use of the physical properties of the table plate. Also, all of the attack apparatus can be hidden from the voter's view under the table.

Note that for the preference disclosure, it is not necessary that the ballot sheet extends from edge to edge in both dimensions. E.g. when the candidates are listed vertically and the resulting ballot sheet spans from the top to the bottom of the table, just getting the y-coordinate of the detected location reveals the voter preference.

Of course, if the ballot sheet is smaller than the table, the amount of the leakage will be reduced, but it will not become zero. For example, if the sheet is longer than half of the height of the table and the mark location was detected as being close to the centre, the attacker knows that candidates close to the top and the bottom of the sheet were not marked. In case of even smaller ballot sheets the attacker may make some probabilistic assumptions (e.g. that the ballot was placed close to the centre of the table) and still obtain a non-zero leakage.

Note that a systematic attacker can make the ballot sheet larger by putting forward a number of extra candidates, arguing that it is a democratic right of every eligible citizen to run for an elected position even if they have only a marginal chance of success. As a result, the leakage from detecting the marking location will be increased.

An informed voter may counter the attack by purposefully placing the ballot sheet off the centre of the board (even partly folding it if necessary). However, since our attack apparatus can operate stealthily, the attacker does not need to reveal the exact vote detection mechanism even in the vote buying scenario. In the persecution scenario, the voter does not even know that the attack is going on at the time of voting (see Section III).

There are also specific ballot sheet design cases where the placement of the sheet on the table does not play that big of a role.

For example, a relatively large privacy leak will occur in case of approval voting, where the voter is allowed to pick several candidates from the list (or even as many as she likes) [35]. Such ballots have e.g. been used in France [36] and Germany [37]. The candidates/parties can be expected to be quite well distinguishable by their political views, hence some pairs of them are more likely to be marked together by the voters than others. Thus, observing the pattern of relative locations of the markings, the attacker may draw conclusions based on where this pattern would fit the best on the ballot sheet.

Another example of a design potentially vulnerable to our attack is preferential voting [38]. For example, in case of

[1] https://www.smh.com.au/politics/federal/metre-long-ballot-paper-means-voters-will-need-to-read-the-fine-print-20130817-2s3yw.html

ranked voting systems the voter is expected to write numbers of preference next to the candidates. Such systems enjoy popularity in various English-speaking countries [30]. Assuming that the voter writes numbers in increasing order (i.e. likely in some other order than from top to the bottom on the sheet) and that the order of candidates is the same on all ballot sheets, the attacker obtains a pretty complete view of the voter's preference ranking.

In the next Section, we will describe the construction of our mark detection device.

V. Attack experiment setup

As stated above, we are not going to attack the paper ballot directly, but rather the table that is used to lay the ballot onto while filling it. Our core observation is that the sound the pen makes during ballot marking travels through the table material at a relatively well-predictable speed, making it in principle possible to triangulate the location of the sound source (i.e. the marking).

Even though the main idea of our attack is simple, there are several challenges to address when implementing and evaluating it.

First, different kinds of tables are being used in polling stations. We have conducted no studies on the materials that one can find in real-life polling station tables. However, we chose two different kinds of materials for our experiments, namely melamine-covered chipboard and glued timber. Plates of size 80×60 cm from both of the materials were selected to model the tables.

Even though we can not claim representativity of our results, we argue that they still give a good understanding on how well our approach works for some pretty common materials. Also, the method we develop is general and can easily be adjusted to other types of tables, too.

The second decision to make in our setup design was the choice of the sound detection devices. We experimented with a number of different microphones and piezoelectric elements, and found that the most stable results were obtained using MAX4466 adjustable gain electret microphones. They can be bought as for 5-10$ as manufactured by the original producer, or for about 1-1.5$ from online Chinese producers and resellers. We tried both, and at least with the items we got, the originals were working more reliably.

Third, we needed a device to capture and process the signals. As we were targeting the raw signal data, we needed a device with analog input capability. Development boards from the Arduino family are perfect for this task, with the additional bonus of being relatively cheap. We chose Arduino Due due to its ATMEL SAM3X8E ARM Cortex-M3 CPU which runs at 84MHz, the highest currently available for the Arduino family. It can be bought for about 15$ online.

Another benefit of Arduino Due is that its analog-digital converter (ADC) can work in the so-called free-running mode where the next conversion is started as soon as the previous one is over (as opposed to the standard mode where there is a predefined number of cycles ADC waits between two conversions). The benefit of the free-running mode is its high working frequency (about 600kHZ in our case), but the shortcoming may be decreased reading precision due to some residual charges on the chip. However, our experiments showed that this shortcoming was greatly compensated by higher speed.

Next, we determined the number and placement of the microphones. After some discussion, we settled with four microphones being placed in the corners of the table. However, this is not the only possible choice. Increasing the number of microphones could, in principle, increase triangulation accuracy. On the other hand, since Arduino Due (as Arduinos in general) only has one ADC, the reads need to be performed sequentially. Thus, increasing the number of microphones would also increase the time interval between two reads of the same microphone, hence potentially decreasing the triangulation precision.

Finding out the optimal setup is a separate research question. As can be seen from experimental results (see Section VII), our choice worked quite well already.

The microphones were connected to Arduino Due using wires and a breadboard; the connection scheme can be seen in Figure 3.

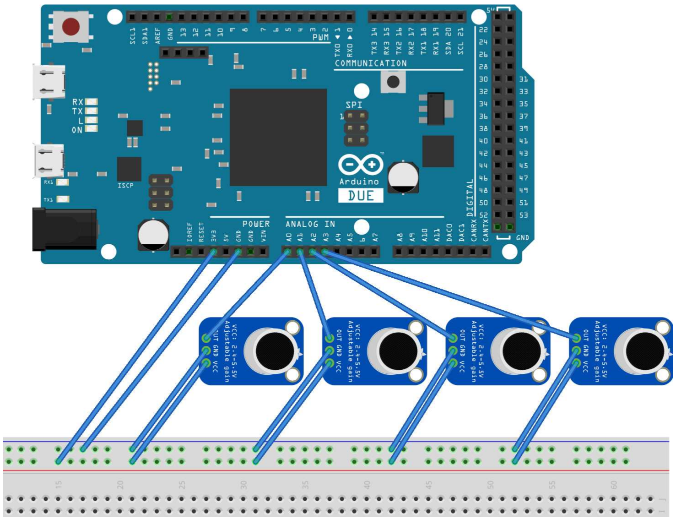


Fig. 3. Microphone connection scheme

Of course, we want the attack to work in a stealth mode, hence we need to install the whole system under the table. The overall experimental setup photographed from below is given in Figure 4.

The next decision to take concerned microphone attachment. There are several ways to do this, e.g. by taping them to the board, or drilling in. Both of these methods have their pros and cons. Drilling in would assume prior access to the tables, but has the benefit of potentially better sound detections, and also the option of hiding the microphones inside the table feet. The electret element of the microphone has a diameter less than 1 cm, and the whole circuit board of MAX4466 microphones has dimensions of about 1×2 cm (see Figure 5).

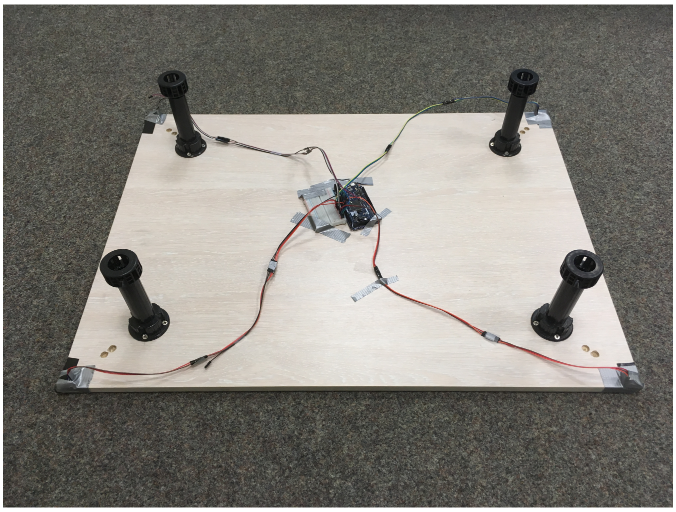


Fig. 4. Experimental setup from below (melamine-covered chipboard)

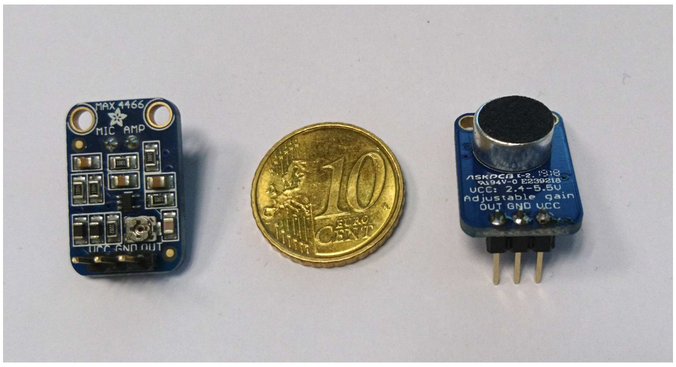


Fig. 5. MAX4466 electret microphone

If prior access to the tables is not possible, the attacker can enter the booth during the voting period and tape the microphones and Arduino under the table onsite. In this scenario, the attacker's task will be easier if curtains are used as part of polling booth construction (which often are there to assure voter's privacy).

In our experiments, we measured detection accuracy for both of the scenarios, by first taping the microphones under our plates, and then drilling them in.

The final question to answer was what exactly should be measured during our experiments. On the physical level, analog input from the microphones is sampled as a voltage the signal corresponds to at the moment of sampling. We can compare this voltage to a reference level when there is no signal, and draw several conclusions from it.

As the first approach, we can look at the difference of voltages of the reference point and current measurement, interpreting it as a difference in signal amplitude. Assuming that the signal fades in the table, the microphone closest to the marking spot should give the strongest signal, followed by other microphones in the order of distance, allowing to determine the location.

We attempted to implement this approach by measuring the total energy of the signals received by each one of the microphones. In general, energy of the signal $x(t)$ is computed as

$$\int_{-\infty}^{\infty} (x(t))^2 dt \ ,$$

and this quantity can be approximated by

$$\sum_{i=0}^{N-1} (x(i))^2$$

where $x(i)$ denotes the sampled signal amplitude at reading i.

Another approach is simply to determine the moment when the first microphone detects a signal exceeding the threshold level, and measure the time differences until the moments the other microphones detect it, too. Since the microphones are not read simultaneously, such an approach will have an inherent error, but hopefully not too much.

During our tests, we experimented with both of the approaches, and found the second one to produce considerably more reliable results. We did not study the reasons behind the poorer performance of the signal energy based triangulation. However, one of the reasons may be secondary signals that reach the microphone after bouncing back from the edge of the table [39], hence causing the signal energies to be estimated larger than they actually are. On the other hand, the timing based approach is free from such problems – when the signal first hits the microphone, it has rather likely come directly (or has had a bounce-back very close to the microphone).

VI. Experiments

As explained above, the target of our attack is determining the location where the pen is scratching the table (through the ballot sheet paper).

There are some physical limitations on the precision of this detection. The signal travels through wood at the speed of approximately $3 \ldots 4 \frac{\mathrm{km}}{\mathrm{s}}$. As we were able to get the ADC of Arduino Due working at approximately $600\mathrm{kHz}$, each one of the four microphones is read with frequency of approximately $150\mathrm{kHz}$. For reliable triangulation, we need to use all the four inputs, hence this $150\mathrm{kHz}$ is also the approximate maximal frequency of triangulation.

Thus, the distance the signal travels during the interval $\frac{1}{150000}\mathrm{s}$ of two consecutive position reads is approximately

$$\frac{3\frac{\mathrm{km}}{\mathrm{s}}}{150000\frac{1}{\mathrm{s}}} = \frac{3000\mathrm{m}}{150000} = 2\mathrm{cm} \ .$$

Thus, we can not expect the precision better than 2 cm, and probably even a bit worse, since we can not read all the microphones at the same time and the time intervals between two reads are not constant.

On the positive side, the precision of a few centimetres is sufficient to mount interesting attacks against various kinds of ballot sheet designs (see Section IV).

For our experiments, we chose to divide the table plates into 4×5 cm cells, thus forming a 20×12 grid on the 80×60 cm plate.

For each of the cells, we produced marking sound around its centre using a pencil. For each cell, about $10 \ldots 15$ marks were made and the timings of the signal reaching the microphones were recorded. An image of the glued timber table with cells and markings can be seen in Figure 6.

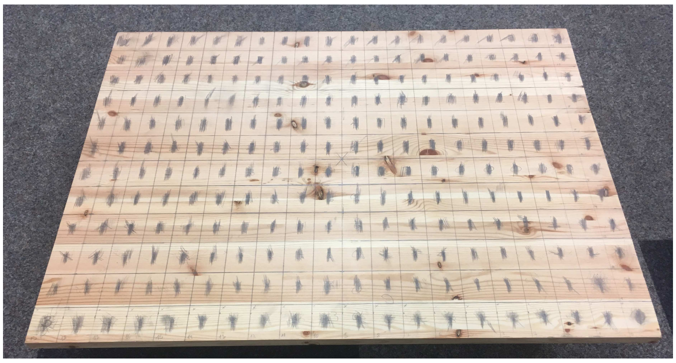


Fig. 6. Cells and markings on a table (glued timber plate)

We tested the experimental setups with both having a sheet of paper on the table and making the marks directly on the board. The results from these setups did not differ significantly. However, as the series of experiments were rather long (several thousands of marks in each series) and the sheet did not cover the whole plate, moving it from one position to another caused too much disturbing noise. Thus, we decided to drop the paper from the actual experiment and make the marks on the board directly.

In a real situation, the attacker would need to make an extra effort of distinguishing the sound of ballot marking from the noise caused by putting the sheet onto and removing it from the table. For this, the attacker may tweak the microphone sensitivity level, or study the noise patterns that the paper generates in the recording. For the latter, machine learning methods working on the audio recording similar to the one proposed by Krips *et al.* may be considered [30].

Also, in case of the vote buying scenario, the attacker can instruct the voter to handle the paper gently, leave some time between laying down the paper and marking her vote, etc.

For each scratch sound on the plate, we recorded the moments t_1, t_2, t_3, t_4 the sound was first detected by every one of the four microphones, and stored the time deltas with respect to the earliest moment. I.e., the data tuple stored for every sample was $(t_1 - t, t_2 - t, t_3 - t, t_4 - t)$, where $t = \min\{t_1, t_2, t_3, t_4\}$.

For the classification task we evaluated five different algorithms – k-Nearest-Neighbour (kNN), weighted version of kNN, Gradient Boosting Classifier, Multi Layered Perception Classifier and Random Forest Classifier [40].

Table I presents evaluation results of the five tested classification algorithms in case of the chipboard and drilled in microphones (for exact choices of the best kNN parameters, see below).

TABLE I
COMPARISON OF CLASSIFIERS

Method	Accuracy
Weighted kNN	90.4%
kNN	89.2%
Random Forest Classifier	87.3%
Gradient Boosting Classifier	84.6%
MLP Classifier	16.3%

As kNN-type classifiers worked the best, we will describe their application in our case in a bit more detail.

The basic idea behind the kNN classifier is simple. We first select a distance metric to describe how similar two samples are. Then we use some part of our dataset for learning/training, and the rest for testing. In case of kNN, learning simply means saving the first part of the dataset in a format $(sample, class)$, where $sample$ refers to the time delta quadruple, and $class$ is the corresponding 4×5 cm cell on the board.

To classify a testing sample, we select k closest (according to our distance metric) samples from the training set and select the class (i.e. the table cell) that is represented the most among those k closest samples.

For the kNN classifier, we experimented with k values 3, 5 and 7, three distance metrics – Canberra, Euclidean and Bray Curtis –, and six possible weights.

For two sample tuples $u = (u_1, u_2, u_3, u_4)$ and $v = (v_1, v_2, v_3, v_4)$, their Bray Curtis distance is defined as

$$d_{BC}(u, v) = \frac{\sum_{k=1}^{4} |u_k - v_k|}{\sum_{k=1}^{4} (u_k + v_k)} .$$

For the weighted version of the kNN, also the weights need to be defined. We experimented with the weights of the form $1/d^e$, $e \in \{1, 2, 3, 4, 5, 6\}$, where d is the selected distance metric (say, Bray Curtis). The resulting weighted kNN classifier then works as follows.

We first have a set S of training data where each sample tuple u^i has a corresponding cell with coordinates (x_i, y_i). When a new tuple v needs classification, we first select k closest (according to our distance metric) samples $u^1, u^2, \ldots, u^k$ to it from our set S. We then group the selected k samples by the corresponding cell and define the weight of the ith group to be

$$\sum_{j=1}^{n_i} \frac{1}{(d(u^{i,j}, v))^e} ,$$

where $\{u^{i,1}, u^{i,2}, \ldots, u^{i,n_i}\}$ is the set of tuples in the ith group and e is the selected weight exponent. The group with the largest weight wins, and the corresponding cell is declared to be the outcome of classification.

Testing classification accuracy across all our experimental setups, we concluded that the choices of $k = 5$, Bray Curtis distance and weight exponent $e = 3$ worked the best on average. However, there is room for some marginal improvement if the polling booth environment (most notably, table material) can be well predicted and/or studied beforehand.

VII. Experimental results

For each of the four test configurations (melamine covered chipboard or glued timber, combined with taping or drilling the microphones), we measured three kinds of accuracies.

1) The accuracy of predicting the exact 4×5 cm cell.
2) The accuracy of predicting the area of the cell with an allowed mistake of 1 cell in each of the eight directions (i.e. 3×3 cells forming a 12×15 cm rectangle).
3) The accuracy of predicting the correct 4 cm wide column.

The last measurement corresponds to the attack scenario where the attacker is interested not in the particular candidate, but rather the party preference of the voter, since on the ballot sheets the candidates of the same party are often gathered into the same column (for some examples, see Figures 1 and 2).

For all the tests, we performed 10-fold cross-validation. I.e., we divided our experimental data into 10 random subsets and used every one of them for validation against the training set formed from the remaining 9 subsets.

The average results of running the experiments 500 times are presented in Table II.

TABLE II
Experimental results

Test setup	1×1 cell	3×3 cell	Column
Chipboard, drilled mics	90.44%	98.77%	93.71%
Chipboard, taped mics	81.07%	94.14%	85.23%
Glued timber, drilled mics	71.29%	87.89%	75.42%
Glued timber, taped mics	77.61%	89.62%	80.35%

From the Table we observe that the best results were achieved on the melamine covered chipboard plate. The most likely cause is the more uniform nature of the material compared to the glued timber plate which comprises of multiple (non-homogeneous) wooden planks glued together.

Thus we may say that better performance of our attack on the chipboard plate was expected, but the overall average precision of detecting the sound source location was a great surprise to the authors. When the attacker manages to drill the microphones into a chipboard table plate, he can achieve accuracy of over 90% within the range of one 4×5 cm cell, and more than 98% within the range of one 12×15 cm cell. Even when only the taping option is available for the attacker, his prediction accuracy (depending on the area he is interested in) is about 80-90%.

Of course, the idea of using (ultra)sonic waves to study properties of different materials is not new [41]–[43], but for the achieved precision, the ease of building the final device, its low price tag and small size were quite unexpected. Creating such a device is definitely both affordable and accessible for a moderate-level attacker.

Somewhat surprisingly, in case of the glued timber plate, the results improved when attaching the microphones using tape instead of drilling the holes into the board. We attribute this effect to our inability to drill holes with flat bottom due to the shape of the drill we used. As a result, the contact of the microphones with the surface of the board was actually better in case of taping.

We conjecture that this problem had a lesser effect in case of the chipboard plate, since chipboard as a material is generally softer, and thus firm contact with the surface may have been less important.

Besides establishing the good average performance of the weighted kNN method, we were also interested in determining whether our set-up has a systematic bias in terms of maximal error rate. I.e., we wanted to find out, whether there is some part of the board that for some reason is detected considerably worse than other parts.

We computed the ratio of wrongly detected samples for all the cells across all our experiments, and the results are presented as heatmaps in Figure 7. We concluded that although sporadic cell error rates reached up to 40%, these seems to be no systematic error bias.

VIII. Building automated person identification

When the attacker is using our attack to breach vote privacy, he is only achieving part of the goal. Knowing just *what* the values of some votes are is not too interesting on its own, the attacker would typically also need to find out *who* submitted those votes. Having access to both parts of the information, he can mount a coercion attack (e.g. by attempting to buy votes, persecute the voters for their political views, etc.).

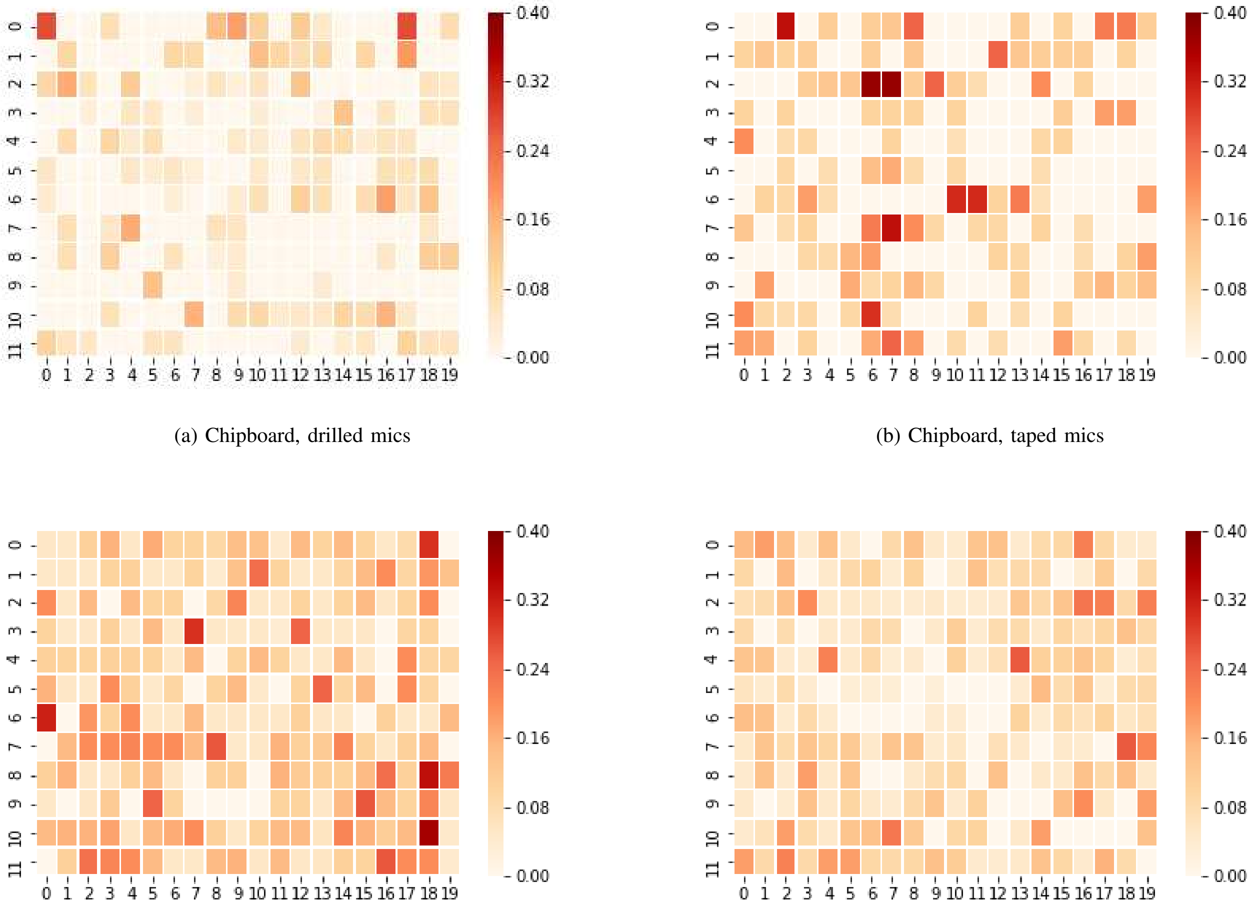
When the attacker is targeting a particular group of voters, he can first install our mark triangulation device(s) under the booth table(s), and then pretend to be a public observer. Observing the polling station, he can take notes of who exits the voting booth(s) at what times, and later cross-reference his notes with the votes recorded and time-stamped by the device.

However, the extent of such an attack would be limited by the ability of the attacker to stay in one polling station (and/or to hire collaborators who would cover several places). Thus, the key to scaling the attack is automating the detection of voters. The detection mechanism should be non-invasive, working from a distance and having relatively high precision.

The choice of such a mechanism is quite straightforward – facial recognition. The human face is highly characteristic and usually not covered. This sub-field of biometry has been well-studied, with many tools being readily available for the attacker to build upon. These aspects make facial recognition rather a well accessible tool for the attacker.

This tool is used a lot by law enforcement agencies around the world (sometimes with objectionable goals and methods [44]–[46]). However, a regular attacker does probably not have access to such resources. (Unless, of course, the government itself is interested in massive vote privacy violations – we will not consider such a scenario here further.)

Thus, a natural research question for us becomes: how easy is it for a moderately-resourced attacker to build an automated



(a) Chipboard, drilled mics

(b) Chipboard, taped mics

(c) Glued timber, drilled mics

(d) Glued timber, taped mics

Fig. 7. Heatmaps of error ratio distribution across the experiments

facial recognition system capable of operating in the polling station environment, and how well would such a set-up scale.

This task can be divided into two subtasks:

1) Build and deploy a detection system, and
2) Obtain a comprehensive personalised facial image database.

We will now discuss both of the subtasks in more detail.

IX. Building a facial detection system

When selecting the best option for building and deploying a facial detection system, there are several optimisation targets the attacker needs to consider.

- How much/little would it cost (in absolute terms or, say, per 1000 faces to detect), determining the scalability of the attack.
- What is the level of technical sophistication and competence required from the attacker.
- How easy is it to run stealthily, or with a good cover story, determining the risk of getting caught.
- What detection speed and quality can be achieved, determining efficiency of the attack.

As we will see in this Section, it will be difficult for the attacker to optimise all these parameters simultaneously (at least given the technology available in the time of this writing, i.e. late 2018). Thus, the attacker will need to make some choices depending on his attack scenario, risk aversion, budget, etc.

The task of building a facial detection system can in turn be divided into two subtasks. First, the attacker needs to install/access cameras for image capture, and second, he needs a back-end for running the actual facial recognition software.

Depending on the local traditions, the first subtask may be relatively easy to accomplish. Several countries have cameras in polling stations pre-installed as a confidence building measure[2]. We have conducted no study on the cameras found in such setups, but we consider it likely that for budgetary and setup simplicity reasons, cheap IP cameras are often used. However, IP cameras are notorious for having rather low

[2] http://aceproject.org/electoral-advice/archive/questions/replies/291099047

security standards, allowing a malicious attacker to access its streams and images, replace firmware, etc. [47]–[50]

If cameras are not a part of standard set-up in polling stations, the attacker would need to get and install them himself. IP cameras can be bought starting from 20$ a piece online, so the biggest practical problem is installing them to the polling stations.

The most straightforward approach would be placing the camera directly into the polling booth at the same time with the table plate mark triangulation device. However, contrary to the latter, the camera would need to be in the line of sight, and this may be difficult to achieve without detection in a minimalistic booth.

The next best thing is to place the camera somewhere in the polling station so that it would record the faces of people stepping out of the booth. Success of this strategy depends on the dimensions and colour of the camera as well as a general environment of the polling station, whether there is a good place to hide the video equipment, etc.

One way to solve the problem of concealing the camera is not to conceal it at all. Even if the local electoral traditions do not include confidence-building cameras, the attacker can still claim to be an election observer interested in making sure that e.g. ballot box stuffing does not happen. Since the events taking place outside of the polling booth are not really private, the election officials will have hard time arguing against such a surveillance. To support his cover story, the attacker may even share the camera streams with other, completely legitimate observers.

The subtask of running a facial recognition algorithm on the images/streams produced by the cameras is actually more demanding as these algorithms require a lot of computational power. Depending on the attack scenario, the attacker may save the camera streams during the elections and take all the time he needs to process them later on, or he may be interested in near-real-time facial recognition.

An example of the first scenario is the one of political persecution where the attacker wants, at some point in time (maybe weeks or even months later), have a list of voters who did not act according to his preferences.

In the scenario of vote buying, on the other hand, the attacker may find it easier to convince the sellers if he can offer the reward relatively soon (in minutes or hours) after the vote casting. The attacker can even automate paying out the reward by triggering, say, a Bitcoin transfer right after a successful facial recognition, making the attack to scale more easily. We will discuss one possible strategy of obtaining voters' Bitcoin addresses in Section X.

When comparing this scenario to the *stemfie* attack, we can see that taking a *stemfie*, on one hand, requires more voter involvement. On the other hand, it has to be post-processed by a human, making our attack to scale better.

For the computational face detection back-end, the attacker also has several options. Perhaps the simplest (and cheapest) to set up is using an existing cloud-based service. E.g. Microsoft is offering Face API on its Azure cloud platform[3]. For 100$, the attacker would get 100,000 detection transactions which would be sufficient for a relatively large-scale privacy violation (even considering that several transactions would probably be spent on one voter). Additionally, he would need to pay 0.25$ for every 1000 stored faces, which is still very reasonable.

The main problem with using a hosted service (besides leaving a significant digital footprint for law enforcement to work on) is its unreliability for the attacker. A large-scale vote-buying attempt will probably be detected at some point. Commercial cloud providers like Microsoft are well aware of privacy risks of such services and are likely willing to cooperate with law enforcement agencies, blocking the service on the first notice.

Note that taking all the cameras down would require more time, especially if they are hidden. On the other hand, if the cameras also have a public function of confidence building, taking them down will need to involve investigation which cameras are part of the privacy violation attack and which ones are not. In order to complicate (and hence prolong) this investigation, the attacker may install some cameras that only serve legitimate purposes.

If the voting period is short (say, one day), the risk of losing the whole computational back-end in one instant may or may not be acceptable for the attacker. As an alternative, he may attempt to build a computational face detection service himself.

In order to assess the costs, ease of setting up and the resulting detection quality of this approach, we decided to build a complete facial recognition system from generally accessible hardware and software components.

Our hardware platform included Intel i5-2310 CPU and nVidia GTX 1070 with 8BG onboard RAM as the GPU. At the time of this writing (fall 2018), GTX 1070 is a mid-high level graphics card available for about 400 euros as new, or for 200-300 euros as used. As with most of the nVidia cards, it also support CUDA framework, making it appealing for a wide range of applications from cryptocurrency mining to machine learning.

The desktop was equipped with a 720p Logitech webcam and Ubuntu 18.10 OS. The facial landmark detection routine of our choice was proposed by Kazemi and Sullivan [51], and implemented as a part of the leading open source machine learning library dlib[4].

The setup was deployed in a generic office environment, with the camera pointed towards the entrance hallway. To be able to measure detection distance, we taped markings on the floor at each full meter from the camera. We used 63 volunteers working in our office as the test subjects. One image of each subject was used to train the recognition system.

We performed two experiments. First, we were interested in the quality of live detection. Second, we also saved the video

[3] https://azure.microsoft.com/en-us/services/cognitive-services/face/

[4] http://dlib.net/

stream to determine how good of a detection accuracy we can achieve when we allow some extra time for post-processing.

The accuracy of the first experiment was measured manually by an observing researcher who wrote down the result of the detection for the listed volunteers who passed by in their natural way. Also, the approximate distance of the subject from the camera at the moment of detection was recorded.

The facial recognition script consisted of three main components. First, it had to process a frame from the video feed and find the locations of faces. Next, it had to create encodings for the faces found. Finally, the found encodings had to be matched with the encodings corresponding to the faces of the volunteers. We used Python's `face_recognition` toolkit[5] for creating and comparing the encodings. Faces were encoded by the underlying dlib library that returns a 128 element real-valued feature vector representing the corresponding face.

Once the face is located and encoded, it is trivial to compare the encodings, e.g. by their Euclidean distance to find the closest match as done by the `face_recognition` toolkit.

However, the difficult step is to find the locations of the faces on the images. There are several approaches to solve this task, and they are all computationally expensive. The dlib library allows one to use a GPU-based convolutional neural network (CNN) to detect the locations of faces. The other well-established option is to use Haar Cascades method from OpenCV toolkit.[6]

Both approaches have their positive and negative sides. CNN from dlib allows to detect faces even when they are tilted. However, it is computationally demanding, and in our case it allowed us to use only about half of the frame rate compared to Haar Cascades. When using CNN in the real time setting, we were able to detect faces from a distance of at most 3-4 meters, while Haar Cascades identified locations of faces from a distance about twice as far.

The downside of Haar Cascades is the false positive rate of face locations, but in our case this was not so significant as we were manually measuring the accuracy and we were interested only in the face encodings that have a match. Thus, we selected Haar Cascades as our method of face location detection. Note that, as a result, this part of the experiment does not really use GPU acceleration at all, making the corresponding attack considerably cheaper.

When running the second experiment with post-processing, performance limitations were not so strict. We were able to utilise the full power of the GPU running a CNN implementation from dlib for facial detection. However, depending on the exact resolution of the video, detection took about 5-10 times longer than the duration of the video itself. The attacker can speed up the wall clock time by splitting the stream into smaller chunks and processing them in parallel. On the other hand, this requires a larger investment into the GPU hardware.

The output of facial detection post-processing script was evaluated by a human operator for accuracy comparing the script output to the video. Also, the approximate distance of detection was recorded based on the meter markings visible in the stream.

Accuracy evaluation was only performed when the method identified a person from our facial database, and not when it was just able to find a location of some face on the image. The accuracy was defined to be the share of correct identifications from all the person identifications. Overall, we were able to obtain 101 identification samples when testing CPU-based live detection, and 151 samples for GPU-based post-processing.

The results of our facial recognition experiments are shown in Figure 8. The Figure displays the relationship between the accuracy and the distance of detection.

The overall average accuracy in case of CPU-based live detection was 67.33%, and the average distance of correct detection was 3.51 meters. With GPU-based post-processing, the average accuracy improved to 74.17%. The average distance of correct detection increased to (only) 3.76 meters, but we see from Figure 8 that the detection quality is more stable across the distance spectrum, with maximal distance of detection improved to 8 meters (compared to 6 meters in the live detection case). We emphasise again that these results were obtained based on one image per test subject. Increasing the number (and quality) of the images would likely improve the detection accuracy as well.

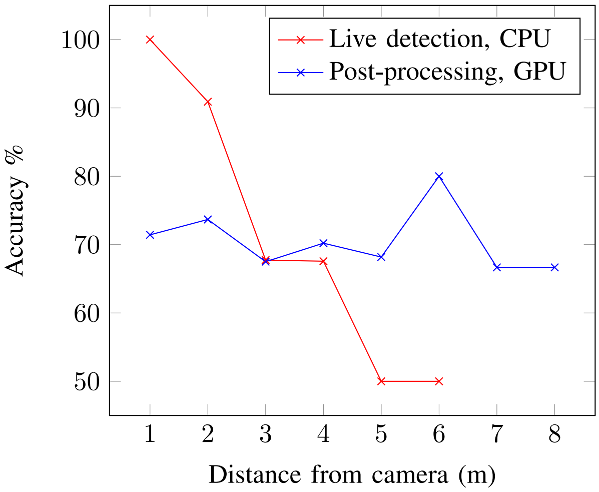


Fig. 8. Accuracy of facial recognition

If building and running a dedicated facial detection back-end is not an option for the attacker, he may try to distribute the task by attaching a computer locally to each camera. In case he would prefer a stealthy setup, using a large desktop or even a laptop may be undesirable.

As an alternative, the attacker may consider using small single-board computers attached to every camera. We tried such a scenario out on ODROID C2, featuring ARM Cortex-A53 1.5Ghz quad core CPU, Mali-450 GPU and a price tag

[5] https://github.com/ageitgey/face_recognition

[6] https://docs.opencv.org/3.4.3/d7/d8b/tutorial_py_face_detection.html

of 46$.[7] Its main competitor Raspberry Pi 3 has a somewhat better GPU, but weaker CPU, and as neither of the GPU-s is supported by dlib and OpenCV libraries, the computations would need to take place in CPU anyway. As we were able to obtain reasonable live detection performance using a CPU-based setup in our main experiments, there was some hope of getting good results on a single-board platform as well.

Testing with volunteers showed that in order to get reasonable processing time, the resolution and frame rate would need to be so low that successful recognition would only happen within 1-2 meters assuming that a person stays still in front of a camera for at least a second. We did not study this problem further, but it seems that dlib and OpenCV implementations use some Intel-CPU-specific optimisations currently not available on ARM platforms. It is an interesting question for further development whether such optimisations could in principle be implemented.

A promising combination of small size and good performance for machine learning applications is provided by nVidia Jetson TX product line.[8] For example in terms of CUDA performance, GTX 1070 and Jetson TX2 are almost equivalent.[9] Due to having higher-end chips, being relatively new and lacking competition, Jetson development boards are priced higher than regular single board computers (depending on the configuration and deal around 300-500$). However, as a result of moving to mainline production and hopefully some healthy competition, the price will probably drop in the future, making this attack more accessible for a mid-resourced attacker.

GDPR compliance notice

All the participants in our facial recognition experiments were volunteers who signed an explicit consent form complying with the EU General Data Protection Regulation (GDPR). Information about the ongoing experiment was distributed through the mailing list and also made visible on the office wall.

X. Obtaining personalised facial image database

The exact characteristics of the required facial database depend on the attack scenario.

If the attacker is interested in coercing a specific group of voters (say, a factory director wanting to ensure the votes of all the employees during local municipal elections), he may directly gather the photographs. However, the extent of such an attack would be quite limited.

In order to implement a large scale vote buying attack, the attacker can set up an image submission service. The voter interested in selling her vote can submit there her photo together with, say, Bitcoin address for receiving the fee after voting for the coercer has been confirmed.

Of course, the attacker would need to advertise the service somehow (forums, social media, word-of-mouth), so he risks being noticed by the law enforcement, and the image submission service ceased, potentially compromising all the vote sellers. To counter this problem, the attacker does need to not store the actual images in the database, but rather the facial feature vectors extracted from them.

As facial recognition is not perfect and can give false positives, law enforcement will have hard time suing anyone based on his/her face giving a positive match to some feature vector in the database. On the other hand, as long as the percentage of false positives is sufficiently low, the attacker can be sure that most of his payments still go to the voters who voted the way he required them to.

Yet another attack scenario is a large scale political persecution. In this case the attacker can not assume cooperation from the coerced voters and needs to build a database without their consent.

One way to do this is to make use of vast amount of photographs available on Internet, say, social media sites. In 2011, a massive personalised facial database was created based on Facebook social network [52]. Since then, Facebook has improved its anti-scraping mechanisms, but the database of 2011 can still be found on Internet. There are a few private torrent trackers, links to which can be obtained in online forums. It took us about a day to locate a working link.

In 2016, a group of Russian developers set up a service called FindFace allowing to identify people by their VKontakte (large Russian social network) profile photo. However, the service was discontinued as of September 1st, 2018. The exact reasons for closing down have not been documented, but FindFace was criticised for privacy violations[10].

There exist stakeholders who may have relatively easy access to citizens' facial and name data. For example, large stores maintain customer loyalty programmes where people sign up to get discounts or other benefits. The stores also have surveillance cameras all around their facilities, allowing to collect facial data e.g. at the moment when the customer swipes her loyalty card at a cashier.

As a part of our study, we originally also planned to try out how good of a dataset we can scrape from the Internet ourselves. However, after having contacted our local Data Protection Authority, we decided to drop this experiment in view of potential legal actions against the authors of the paper. It is an open question how researchers working in the public domain can study certain attacker capabilities without being prosecuted.

XI. Countermeasures

Despite leading to rather high detection rates both in terms of ballot mark locations and facial identification, our attack relies on quite a number of assumptions, creating possibilities for countermeasures.

[7] https://www.hardkernel.com/main/products/prdt_info.php?g_code=G145457216438

[8] https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/

[9] https://developer.nvidia.com/cuda-gpus

[10] https://en.wikipedia.org/wiki/FindFace

First and foremost, the attacker needs to install custom hardware both in the voting booths and general areas of polling stations. Even though the mark location device is designed to work under the table, it can still be located when explicitly looked for. Even if the microphones can be concealed using table feet, our current set-up needs wires to connect the microphones.[11]

On the other hand, the device can only be found if someone is actively searching for it. We have conducted no study on polling station rule books, but we consider it unlikely that many of them include instructions to look for strange electronic devices installed under the tables. Thus, the main systematic countermeasure would be including such instructions into the rule books and training the polling station staff to follow them.

Some parts of the attack (setting up cameras, running an image collection server for vote buying) would need to happen in public space. Monitoring this space for suspicious activity is an important detection measure. This will be a part of the ever-ongoing race between the attackers and defenders of the voting environment. The main message of our research is that in the light of new technological attack vectors, the defence mechanisms of paper voting need constant review and updating as well.

In our experiments, we did not make an effort to distinguish the sound of ballot marking from other sounds, e.g. the noise a paper would make when moved around on the table. While the voter would not need to make any extra noises, she may choose to do it to improve her privacy. However, such a level of awareness can not be expected from masses of voters. Also, in the vote buying scenario, the voter is actively interested in correct recording of her vote, so she would make sure that the amount of noise is reduced.

XII. Conclusions and further work

In this paper, we have described a new side-channel attack, applicable against a wide range of paper ballot sheet designs. The average accuracy of detecting the correct 4×5 cm cell on the board reached over 90% in the best setup. This result greatly exceeded expectations of the authors, especially considering the low price tag (around 20-30$) of the device.

When combining this device with an appropriate solution for facial detection, the attacker can fully automate breaching paper voting privacy. Attack automation, in turn, is a necessary prerequisite for scalability.

Of course, it is not the only one. The attacker would still need to install the mark detection devices and cameras, but each setup will be able to reveal the preferences of hundreds, maybe thousands of voters. Assuming that the attacker will be able to deploy several installations of the devices, this gives rise to a privacy breach of much larger extent than at homes in case of remote Internet voting (at least for a mid-level attacker, unable of creating and distributing custom malware).

In fact, the most significant bottleneck of our attack is not the difficulty of installing apparatus in the polling stations, but the infrastructure and computing power required for facial recognition. Real-time on-site detection currently seems to assume hard- and software available mainly for law enforcement.

Using a commercial facial recognition service in the back-end is risky, so the primary option left for the attacker is to build such a back-end himself. This requires a non-trivial development effort together with a non-trivial monetary investment into hardware. The overall success of our attack will be determined by how well the attacker will be able to address these challenges.

At the same time, availability of new technologies (like off-the-shelf web frameworks and anonymous cryptocurrencies) has made other parts of the attack (like collecting a facial feature database and automatic anonymous transfer of vote buying fee) considerably more accessible for a mid-level attacker.

One may argue that integrity properties (e.g. correctness of the tally) are more important than vote privacy that can only be used indirectly in coercion scenarios. It will be an interesting direction for future work to try to develop proof-of-concept high-tech attacks against low-tech voting actions like, say, ballot box management or hand counting the paper ballots.

Our side channel attack works the best in case of extremely large ballots sheets. However, the reason for occurrence of such ballots is rather deep. In order for democracy to function properly, there need to be multiple candidates. In principle, every person eligible to vote should also be qualified to run for an elected position. From this viewpoint, having many candidates is good for the democratic political culture. On the other hand, presenting all the candidates to the voter is a challenge which may, as our research shows, introduce new kinds of privacy issues. Resolving this conflict of goals requires a larger discussion in the society, leading to an agreement on the new security measures to be deployed in the future.

Acknowledgements

The research leading to these results has received funding from the Estonian Research Council under Institutional Research Grant IUT27-1 and the European Regional Development Fund through the Estonian Centre of Excellence in ICT Research (EXCITE) and the grant number EU48684. The authors are also grateful to anonymous reviewers for fruitful comments and to Renee Undrits for his help in several stages of the research.

References

[1] E. Gerck, C. A. Neff, R. L. Rivest, A. D. Rubin, and M. Yung, "The Business of Electronic Voting," in *Financial Cryptography, 5th International Conference, FC 2001, Grand Cayman, British West Indies, February 19-22, 2002, Proceedings*, ser. Lecture Notes in Computer Science, P. F. Syverson, Ed., vol. 2339. Springer, 2001, pp. 234–259. [Online]. Available: https://doi.org/10.1007/3-540-46088-8_21

[11] During our research, we also considered using radio microphones, but due to the need to access the raw analog signal, this would require extra development to deal with modulation, increasing the complexity of the attack.

[2] L. J. Hoffman and L. F. Cranor, "Internet voting for public officials: introduction," *Commun. ACM*, vol. 44, no. 1, pp. 69–71, 2001. [Online]. Available: http://doi.acm.org/10.1145/357489.357510

[3] J. Mohen and J. Glidden, "The case for internet voting," *Commun. ACM*, vol. 44, no. 1, pp. 72–85, 2001. [Online]. Available: http://doi.acm.org/10.1145/357489.357511

[4] L. Mitrou, D. Gritzalis, and S. K. Katsikas, "Revisiting Legal and Regulatory Requirements for Secure E-Voting," in *Security in the Information Society: Visions and Perspectives, IFIP TC11 17th International Conference on Information Security (SEC2002), May 7-9, 2002, Cairo, Egypt*, ser. IFIP Conference Proceedings, A. Ghonaimy, M. T. El-Hadidi, and H. K. Aslan, Eds., vol. 214. Kluwer, 2002, pp. 469–480.

[5] D. R. Jefferson, A. D. Rubin, B. Simons, and D. A. Wagner, "Analyzing internet voting security," *Commun. ACM*, vol. 47, no. 10, pp. 59–64, 2004. [Online]. Available: http://doi.acm.org/10.1145/1022594.1022624

[6] R. Joaquim, C. Ribeiro, and P. Ferreira, "Improving Remote Voting Security with CodeVoting," in *Towards Trustworthy Elections, New Directions in Electronic Voting*, ser. Lecture Notes in Computer Science, D. Chaum, M. Jakobsson, R. L. Rivest, P. Y. A. Ryan, J. Benaloh, M. Kutylowski, and B. Adida, Eds., vol. 6000. Springer, 2010, pp. 310–329. [Online]. Available: https://doi.org/10.1007/978-3-642-12980-3_19

[7] M. Volkamer, J. Budurushi, and D. Demirel, "Vote casting device with VV-SV-PAT for elections with complicated ballot papers," in *2011 International Workshop on Requirements Engineering for Electronic Voting Systems, REVOTE 2011, Trento, Italy, August 29, 2011*. IEEE, 2011, pp. 1–8. [Online]. Available: https://doi.org/10.1109/REVOTE.2011.6045910

[8] P. S. Herrnson, M. J. Hanmer, and R. G. Niemi, "The impact of ballot type on voter errors," *American Journal of Political Science*, vol. 56, no. 3, pp. 716–730, 2012.

[9] M. D. Byrne, K. K. Greene, and S. P. Everett, "Usability of voting systems: baseline data for paper, punch cards, and lever machines," in *Proceedings of the 2007 Conference on Human Factors in Computing Systems, CHI 2007, San Jose, California, USA, April 28 - May 3, 2007*, M. B. Rosson and D. J. Gilmore, Eds. ACM, 2007, pp. 171–180. [Online]. Available: http://doi.acm.org/10.1145/1240624.1240653

[10] D. L. Dill, B. Schneier, and B. Simons, "Voting and technology: who gets to count your vote?" *Commun. ACM*, vol. 46, no. 8, pp. 29–31, 2003. [Online]. Available: http://doi.acm.org/10.1145/859670.859692

[11] J. Bannet, D. W. Price, A. Rudys, J. Singer, and D. S. Wallach, "Hack-a-vote: Security issues with electronic voting systems," *IEEE Security & Privacy*, vol. 2, no. 1, pp. 32–37, Jan 2004.

[12] A. J. Feldman, J. A. Halderman, and E. W. Felten, "Security Analysis of the Diebold AccuVote-TS Voting Machine," in *2007 USENIX/ACCURATE Electronic Voting Technology Workshop, EVT'07, Boston, MA, USA, August 6, 2007*, R. Martinez and D. A. Wagner, Eds. USENIX Association, 2007. [Online]. Available: https://www.usenix.org/conference/evt-07/security-analysis-diebold-accuvote-ts-voting-machine

[13] A. W. Appel, M. Ginsburg, H. Hursti, B. W. Kernighan, C. D. Richards, and G. Tan, "Insecurities and inaccuracies of the Sequoia AVC Advantage 9.00 H DRE voting machine," 2008.

[14] A. J. Aviv, P. Cerný, S. Clark, E. Cronin, G. Shah, M. Sherr, and M. Blaze, "Security Evaluation of ES&S Voting Machines and Election Management System," in *2008 USENIX/ACCURATE Electronic Voting Workshop, EVT 2008, July 28-29, 2008, San Jose, CA, USA, Proceedings*, D. L. Dill and T. Kohno, Eds. USENIX Association, 2008. [Online]. Available: http://www.usenix.org/events/evt08/tech/full_papers/aviv/aviv.pdf

[15] A. W. Appel, M. Ginsburg, H. Hursti, B. W. Kernighan, C. D. Richards, G. Tan, and P. Venetis, "The New Jersey Voting-machine Lawsuit and the AVC Advantage DRE Voting Machine," in *2009 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, EVT/WOTE '09, Montreal, Canada, August 10-11, 2009*, D. Jefferson, J. L. Hall, and T. Moran, Eds. USENIX Association, 2009. [Online]. Available: https://www.usenix.org/conference/evtwote-09/new-jersey-voting-machine-lawsuit-and-avc-advantage-dre-voting-machine

[16] S. Wolchok, E. Wustrow, J. A. Halderman, H. K. Prasad, A. Kankipati, S. K. Sakhamuri, V. Yagati, and R. Gonggrijp, "Security Analysis of India's Electronic Voting Machines," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, ser. CCS '10. New York, NY, USA: ACM, 2010, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1866307.1866309

[17] D. Castro, "Stop the presses: How paper trails fail to secure e-voting," 2007, http://itif.org/files/evoting.pdf.

[18] N. Cheeseman and B. Klaas, *How to Rig an Election*. Yale University Press, 2018.

[19] B. Harris, *Black Box Voting: Ballot Tampering in the 21st Century*. Talion Publishing, 2004.

[20] R. M. Alvarez and T. E. Hall, "Controlling democracy: the principal–agent problems in election administration," *Policy Studies Journal*, vol. 34, no. 4, pp. 491–510, 2006.

[21] A. Yakobson, "Secret ballot and its effects in the late Roman Republic," *Hermes*, vol. 123, no. H. 4, pp. 426–442, 1995.

[22] L. Loeber, "E-voting in the Netherlands; past, current, future," in *Proceedings of the 6th international conference on electronic voting (EVOTE). TUT Press, Tallinn*, 2014, pp. 43–46.

[23] R. Di Cosmo, "On privacy and anonymity in electronic and non electronic voting: the ballot-as-signature attack." Apr. 2007, https://hal.archives-ouvertes.fr/hal-00142440.

[24] J. Benaloh, "Ballot Casting Assurance via Voter-Initiated Poll Station Auditing," in *2007 USENIX/ACCURATE Electronic Voting Technology Workshop, EVT'07, Boston, MA, USA, August 6, 2007*, R. Martinez and D. A. Wagner, Eds. USENIX Association, 2007. [Online]. Available: https://www.usenix.org/conference/evt-07/ballot-casting-assurance-voter-initiated-poll-station-auditing

[25] R. Gonggrijp and W.-J. Hengeveld, "Studying the Nedap/Groenendaal ES3B Voting Computer: A Computer Security Perspective," in *Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology*. Berkeley, CA, USA: USENIX Association, 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1323111.1323112

[26] L. Loeber, "E-voting in the Netherlands; from general acceptance to general doubt in two years," in *3rd international Conference on Electronic Voting*, ser. GI-Edition Lecture Notes in Informatics, 2008, vol. 131, pp. 21–30.

[27] B. Jacobs and W. Pieters, "Electronic Voting in the Netherlands: from early Adoption to early Abolishment," in *Foundations of security analysis and design V*. Springer, 2009, pp. 121–144.

[28] J. A. Calandrino, W. Clarkson, and E. W. Felten, "Some Consequences of Paper Fingerprinting for Elections," in *EVT/WOTE*, 2009.

[29] E. Toreini, S. F. Shahandashti, and F. Hao, "Texture to the Rescue: Practical Paper Fingerprinting Based on Texture Patterns," *ACM Transactions on Privacy and Security*, vol. 20, no. 3, pp. 9:1–9:29, aug 2017.

[30] K. Krips, J. Willemson, and S. Värv, "Implementing an Audio Side Channel for Paper Voting," in *Proceedings of E-Vote-ID 2018*, ser. LNCS, R. Krimmer, M. Volkamer, V. Cortier, R. Goré, M. Hapsara, U. Serdült, and D. Duenas-Cid, Eds., vol. 11143. Springer, 2018, pp. 132–145.

[31] E. A. Barker and A. Lijphart, "A crucial test of alphabetic voting: the elections at the University of Leiden, 1973–1978," *British Journal of Political Science*, vol. 10, no. 4, pp. 521–525, 1980.

[32] R. Darcy, "Position effects with party column ballots," *Western Political Quarterly*, vol. 39, no. 4, pp. 648–662, 1986.

[33] ——, "Position effects in multimember districts: the New Hampshire House of Representatives, 1972–1994," *Polity*, vol. 30, no. 4, pp. 691–703, 1998.

[34] A. Reynolds and M. Steenbergen, "How the world votes: the political consequences of ballot design, innovation and manipulation," *Electoral Studies*, vol. 25, no. 3, pp. 570–598, 2006.

[35] S. J. Brams and P. C. Fishburn, "Approval voting," *American Political Science Review*, vol. 72, no. 3, pp. 831–847, 1978.

[36] A. Baujard and H. Igersheim, "Framed field experiments on approval voting: lessons from the 2002 and 2007 French presidential elections," in *Handbook on approval voting*. Springer, 2010, pp. 357–395.

[37] C. Alós-Ferrer and Ð.-G. Granić, "Two field experiments on approval voting in Germany," *Social Choice and Welfare*, vol. 39, no. 1, pp. 171–205, 2012.

[38] J. Toplak, "Preferential voting: definition and classification," *Lex Localis – Journal of Local Self-Government*, vol. 15, no. 4, pp. 737–761, 2017.

[39] Rovetta, D. and Sarti, A. and Tubaro, S. and Colombo, G., "Modelling elastic wave propagation in thin plates," in *Intelligent Production Machines and Systems*, D. Pham, E. Eldukhri, and A. Soroka, Eds. Elsevier Science Ltd, 2006, pp. 548–555. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780080451572500961

[40] P. Flach, *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, 2012.

[41] M. Breazeale and J. Ford, “Ultrasonic studies of the nonlinear behavior of solids,” *Journal of Applied Physics*, vol. 36, no. 11, pp. 3486–3490, 1965.

[42] J. F. Holland, “Sonic wood testing apparatus and method,” *The Journal of the Acoustical Society of America*, vol. 83, no. 4, pp. 1716–1716, 1988.

[43] J. Krautkrämer and H. Krautkrämer, *Ultrasonic testing of materials*. Springer Science & Business Media, 2013.

[44] C. Garvie, A. Bedoya, and J. Frankle, “The Perpetual Line-up: Unregulated Police Face Fecognition in America,” 2016, Georgetown Law: Center on Privacy and Technology. https://www.perpetuallineup.org/.

[45] “Human Rights in the Digital Era: An International Perspective on Australia,” 2018, AccessNow. https://www.accessnow.org/cms/assets/uploads/2018/07/Human-Rights-in-the-Digital-Era-an-international-perspective-on-Australia.pdf.

[46] S. Chen, “China to build giant facial recognition database to identify any citizen within seconds,” South China Morning Post. First lublished 12 October, 2017, updated 24 September, 2018, https://www.scmp.com/news/china/society/article/2115094/china-build-giant-facial-recognition-database-identify-any.

[47] A. Tekeoglu and A. S. Tosun, “Investigating Security and Privacy of a Cloud-Based Wireless IP Camera: NetCam,” in *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, Aug 2015.

[48] P. Wardle and C. Moore, “Optical Surgery: Implanting a Dropcam,” Aug 2014, Defcon 22 Hacking Conference, Synack Labs.

[49] L. Constantin, “Widely used wireless IP cameras open to hijacking over the Internet, researchers say,” Apr 2013, PCWorld, https://www.pcworld.com/article/2033821/security/widely-used-wireless-ip-cameras-open-to-hijacking-over-the-internet-researchers-say.html.

[50] D. Palmer, “Researchers find security flaws in popular smart cameras,” Mar 2018, ZDNet, https://www.zdnet.com/article/security-vulnerabilities-in-these-popular-smart-cameras-let-hackers-turn-them-into-surveillance/.

[51] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2014, pp. 1867–1874.

[52] A. Acquisti, R. Gross, and F. Stutzman, “Faces of Facebook: Or, How The Largest Real ID Database In The World Came To Be,” 2011, BlackHat USA, https://media.blackhat.com/bh-us-11/Acquisti/BH_US_11_Acquisti_Faces_of_Facebook_WP.pdf.