# Not-So Hidden Information:
## Optimal Contracts for Undue Influence in E2E Voting Systems

Jeremy Clark, Urs Hengartner, and Kate Larson

Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada, N2L 3G1
{j5clark,uhengart,klarson}@cs.uwaterloo.ca

**Abstract.** This paper considers coercion contracts in voting systems with end-to-end (E2E) verifiability. Contracts are a set of instructions that an adversary can dictate to a voter, either through duress or by offering payment, that increase the probability of a compliant voter constructing a vote for the adversary's preferred candidate. Using a representative E2E system, we place the attacks in game-theoretic terms and study the effectiveness of three proposed contracts from the literature. We offer a definition of optimality for contracts, provide an algorithm for generating optimal contracts, and show that as the number of candidates increases, the adversary's advantage through the use of contracts decreases. We also consider the use of contracts in a heterogeneous population of voters and for financially constrained adversaries.

## 1 Introduction

End-to-end verifiable voting systems (E2E systems) allow voters to independently verify the correctness of the final tally, without needing to trust the chain-of-custody over the ballots after the election in paper voting settings, nor any software or hardware used for vote capture and tallying in electronic and remote voting settings. E2E systems often use cryptographic primitives to achieve these properties while maintaining the secrecy of every cast ballot. A sample of recently proposed E2E systems include VoteHere [20], "Votegrity" [12], Prêt à Voter [14], "Benaloh-06" [7], Scratch and Vote [3], Punchscan [15,23], ThreeBallot [24], Scantegrity [10,11], Civitas [19], VoteBox [25] and Helios [1]. A common element of these systems is the production of some kind of obfuscation of each vote, which voters can retain, digitally or physically, as a privacy-preserving receipt of their vote. Since the receipt does not reveal which candidate the voter selected, it ostensibly cannot be used effectively in a scheme to buy votes or coerce voters into voting for a particular candidate. However this is not the case: even if votes are correctly obfuscated, undue influence can still be accomplished by paying or forcing voters to follow certain procedures in the construction of their receipts, such that the receipts become probabilistically biased toward a chosen candidate. We call these procedures, and consequences for not following

P.Y.A. Ryan and B. Schoenmakers (Eds.): VOTE-ID 2009, LNCS 5767, pp. 1–17, 2009.

them, a contract. In this paper, we argue that contracts are persistent enough in E2E systems to warrant further study and, in response, we conduct a detailed analysis in a representative E2E system—Punchscan.

Our contributions can be summarized as

- a new analysis of the effectiveness of three existing attacks [9,17,18] using coercion contracts in Punchscan with two candidates,
- a definition of optimality for contracts and a linear-time algorithm for generating optimal contracts,
- an analysis of multiple-candidate contracts showing that their effectiveness decreases with the number of candidates,
- an analysis of contracts in the setting where some voters have intentions other than accepting the highest payment available to them and hide their real intentions from the adversary, and
- an analysis of contracts in the setting where the adversary is financially constrained showing that the adversary must value the vote by, approximately, an order of magnitude more than the voter selling the vote.

## 2 Preliminaries

### 2.1 End-to-End Verifiability

Voting systems that offer end-to-end verifiability often use a variety of cryptographic techniques to simultaneously achieve ballot secrecy and tally correctness. One common construction includes, abstractly, these three critical steps:

i. The voter produces and retains an obfuscation of her vote, such that given only the obfuscated vote, it is not possible to determine the vote.
ii. Obfuscated votes are collected by the election authority, published publicly, and voters check that the obfuscation of their vote is included and correct in this collection.
iii. Obfuscated votes are collectively deobfuscated to produce a tally in a way that is verifiably correct and does not reveal the link between any obfuscated and deobfuscated votes.

While there is little room for variation within (ii), a variety of approaches to (i) and (iii) have been presented in the literature. The integrity of (i) is sometimes referred to as *ballot casting assurance* [2] or *voter initiated auditing* [5], while privacy is called *coercion resistance* [16] or *receipt freeness* [8]. The dominant mechanism for achieving obfuscation in (i) is encryption, but more recent literature includes use of permutations, code substitutions, information splitting, and vote swapping. When the obfuscation technique is encryption, the deobfuscation in (iii) is typically achieved through a mix network [13,22] or additive homomorphic encryption [6].

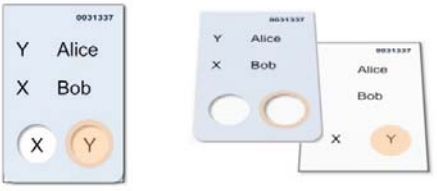



**Fig. 1.** A marked Punchscan ballot, showing top and bottom layers. In our notation, this ballot is of type {YX,XY} and the position marked is R. It represents a vote for Alice.

### 2.2 Undue Influence

The subject of this paper pertains to the privacy property in (i). We are interested in cases where given only an obfuscated vote, the voter's selection remains hidden; yet if certain decisions in the construction and verification of the obfuscated vote are dictated to the voter by an adversary, the voter's compliance results in a non-negligible probability that the voter selected the adversary's preferred candidate.[1] We call such a set of instructions a *contract* and this class of attack *contract-based attacks*.

Contract-based attacks have been proposed for a variety of E2E systems. In the experience of the first authors, they have also proven non-trivial to avoid in the design of Scantegrity, which has been specifically hardened against them. It is our belief that this category of attack is sufficiently wide-spread that a detailed analysis of contracts can provide value to voting system designers in understanding the mechanisms at play and the effectiveness of these attacks in a realistic setting. Instead of a light-touch on a range of systems, we have undertaken a very detailed analysis of contract-based attacks in one representative system—Punchscan [23]—which has been found to be vulnerable in this regard [9,17,18].

### 2.3 The Punchscan Voting System

In a Punchscan election, two-layer paper ballots are used (See Figure 1). Both layers have a serial number and a list of candidates. Additionally, a column of symbols is printed on the top layer beside the candidates' names and a row of symbols is printed on the bottom layer underneath the candidates. These bottom layer symbols are visible through circular holes in the top layer. A voter marks a ballot by finding the symbol in the bottom row that corresponds to the symbol

[1] Other types of manipulation may include forcing the voter to cast a random vote [16] or to vote *against* a particular candidate instead of *for* one. This latter distinction is called destructive manipulation, as opposed to constructive, and can be accomplished through a combination of constructive manipulations. Forming a strategy of constructive manipulations can be intractable in the worse-case for some scoring protocols but it is trivial for plurality voting [4].

beside their preferred candidate and daubs this position with a suitably-sized Bingo dauber such that the ink is clearly visible on both layers of the paper.

After marking the ballot, the voter separates the layers of paper and is allowed to keep either layer of paper as a receipt.[2] The other layer is shredded without anyone except the voter having seen its contents. The receipt is scanned and then retained by the voter, who can use it to perform steps (ii) and (iii) in the E2E construction from Section 2.1.

Both sets of symbols—the column on the top layer and the row on the bottom layer—are randomly ordered on a per ballot basis. In other words, the top symbols on a ballot could be X beside Alice and Y beside Bob or vice versa as in the ballot in the figure; similarly with the bottom symbols. Thus if shown the top layer in the figure, it is not possible to identify whether the symbols on the bottom layer were ordered XY (resulting in a vote for Bob) or YX (a vote for Alice).[3] The same property holds when shown only the bottom layer. For this reason, the voter can ostensibly show her receipt to anyone without violating her privacy. Furthermore, unlike in conventional optical scan voting systems, the scanner does not know which candidate the voter voted for (nor would anyone who hacks into the scanner).

## 3 Extensive Form of the Ballot Casting Process

To analyse the effective privacy of Punchscan ballot receipts, we will formalize the ballot casting process using game-theoretic conventions.[4] Contract-based attacks will ultimately involve three players—nature ($N$),[5] the voter ($V$), and the adversary or influencer ($I$), whose role will be outlined in the next section. For now, we consider the initial interaction between $N$ and $V$ in ballot casting. The extensive form of this interaction is shown in Figure 2.[6] Nature's first two moves are randomly drawn with equal probability from the action sets $A_{N_1} = \{$Top: XY, Top: YX$\}$ and $A_{N_2} = \{$Bottom: XY, Bottom: YX$\}$ and will define the layout of the ballot given to the voter.

Upon observing $N$'s moves, $V$ chooses a position to mark, left or right, from action set $A_{V_1} = \{$L, R$\}$. In particular, $V$ will choose an action such that a particular candidate will be voted for. $V$ then chooses to keep the top sheet

[2] It is important that either layer could be potentially kept. This is for security reasons that we are deliberately omitting, as they are not essential to the results of this paper. For full details, see [15,23].

[3] More properly, it is not possible without knowledge of a secret cryptographic key held by a committee of election trustees, which ties the serial number to the information needed to deobfuscate the vote and produce a verifiable tally.

[4] All game theoretic conventions employed in this paper can be found in most introductory textbooks on the subject (*e.g.,* [21]). Future footnotes will provide additional background on game theoretic concepts as they are used.

[5] When a game incorporates randomness, a special player called nature chooses random actions from a known distribution as needed.

[6] An extensive form diagram is a tree, with the root node defined as the first player to move and a vertex defined for each action the player can take for this move.

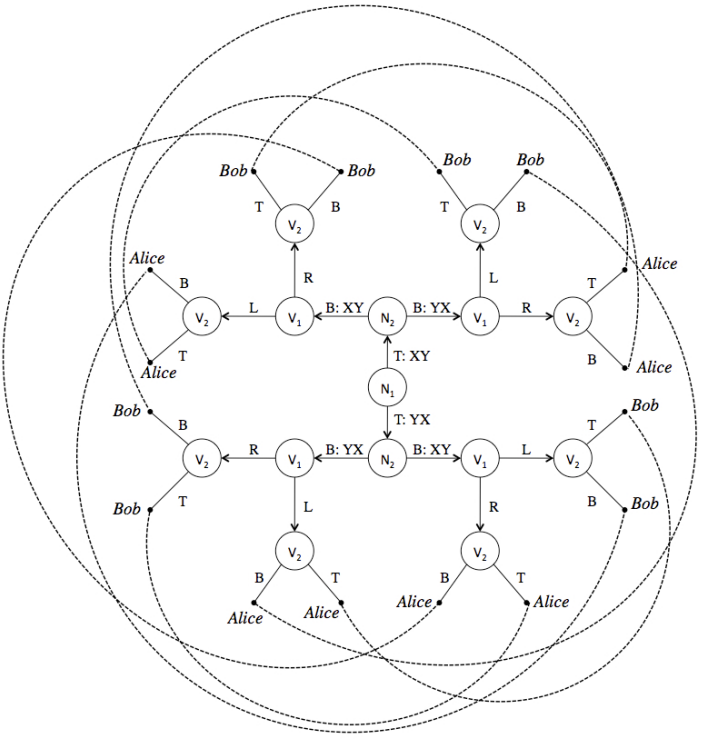



**Fig. 2.** Extensive form of the ballot casting process, involving the voter and nature, in a Punchscan election. The ballot casting process begins at $N_1$ and ends at a candidate. The dotted lines represent hidden information.

or bottom sheet: $A_{V_2} = \{\text{T, B}\}$. This decision does not influence, of course, which candidate was voted for, however the three previous moves all influence the outcome. This will become important in section 5.2.

The privacy of the receipt comes from the fact that this model contains hidden information. Depending on how $V$ moves, either $A_{N_1}$ or $A_{N_2}$ will be hidden from any observer of the receipt. If one were to only observe $V$'s receipt and not both moves by $N$, they could only determine the outcome to be in a set of outcomes joined with a dotted line in Figure 2 but not know which outcome. For all outcome sets, the state of the world could be a vote for Alice or Bob with equal probability. For this reason, the privacy of a Punchscan receipt appears very strong, however this does not imply coercion resistance.

## 4 Contract-Based Attacks

Despite appearances to the contrary, Punchscan receipts can be exploited to bias a voter's choice. This is accomplished through a contract, which is presented

to $V$ by the adversary. A contract specifies, for each possible receipt a voter can construct, a payoff the voter will receive for that receipt. Assuming $V$ is utility-maximizing, $V$ will construct her receipt in a way that maximizes her payoff. Using this property, the adversary seeks to offer a contract that will result, on balance, in more votes for his preferred candidate than the other candidates. We study three proposed contracts that accomplish this for two-candidate races, named for their authors: MN [18], BMR [9], and KRMC [17]. These three contracts are not central to their respective works, and thus certain subtleties are glossed over by the authors which we will fill in.

An alternative to contracts suggested in the same literature are scratch-off cards. A scratch-off card, in a race between Alice and Bob, would be a $2 \times 2$ matrix, with the rows marked X and Y, the columns L and R, and each cell would contain a random T or B underneath a scratch-off layer. The voter is given a new card and instructed to vote for Alice, scratch off the cell that corresponds to the letter beside Alice's name and the position where this letter appears on the ballot received by the voter. The voter then retains the top or bottom layer, as revealed. Both the receipt and the card must be returned to the adversary, who checks that they are consistent. If the voter does not vote for Alice, the voter must scratch off a cell that does not correspond to either Alice's symbol or the position of the asserted symbol on the bottom layer of the ballot. In both cases, the voter will be caught with probability 0.5—if the scratch reveals T in the former case or B in the latter.

Scratch-off cards are attractive since they fix the adversary's ability to gain votes for Alice, while we will show in Section 5.1 that contracts perform worse as the number of candidates increases. By contrast, contracts are attractive because they are informational and can be memorized by voters (especially in the case of voting buying, where the voter has such an incentive). This eliminates the risk of being caught using a scratch-off card or even giving the voter incriminating evidence of the undue influence. In addition, contracts do not need to be secure against physical tampering (scratch-off surfaces can be removed and reapplied). Finally, contracts do not necessarily require the voter to rendezvous with the adversary after the attack. The voter can simply report their serial number and the adversary can retrieve the information from the public record (this assumes the voter does not collude with other voters to misreport their serial number as the serial number of another voter's receipt that coincidently meets the conditions of the contract; a difficult task even if allowed). Our purpose is not to argue that contracts are better than scratch-off cards, merely that contracts have enough interesting advantages to warrant their own thorough study.

### 4.1 Voter Coercion and Vote-Buying

It is useful to distinguish between voter coercion and vote-buying. As mentioned, the contract will offer payoffs in the form of utility. These utilities are in either two or three amounts with strict ordering: $\{u_0, u_1, u_2 \mid u_2 > u_1 > u_0\}$. Generally, a vote-buying contract will promise positive utilities, such as $u_0 = \$0$, $u_1 = \$5$, and $u_2 = \$10$, while a coercive contract will threaten negative utilities, such as

$u_0$ as arson against a home, $u_1$ as slashed tires, and $u_2$ as nothing happening. Generally participation in vote-buying is voluntary, while coercion is involuntary as no rational voters would opt into a negative utility. We use the term **vote-buying** to refer to a voluntary contract with positive utilities and **coercion** to refer to an involuntary contract with at least one negative utility.

### 4.2 The MN Contract

The first contract we consider is due to Moran and Naor [18]. It is presented by the authors as a vote-buying contract and is w.l.o.g. biased toward Alice.[7] It is as follows:

$$\text{Contract}_{MN} = \begin{cases} u_1 = \pi_V(L) \\ u_1 = \pi_V(R, T \mid \{XY, \_\_\}) \\ u_1 = \pi_V(R, B \mid \{\_\_, XY\}) \\ u_0 \text{ otherwise} \end{cases}$$

In our notation, this means that $V$ is given a payoff ($\pi_V$) equal to $u_1$ for any receipt where the left position is marked, or a top sheet with symbols XY and the right position marked, or a bottom sheet with symbol order XY and the right position marked. Any other receipt is given $u_0$. The underscores denote information that is hidden due to the choice of T or B.

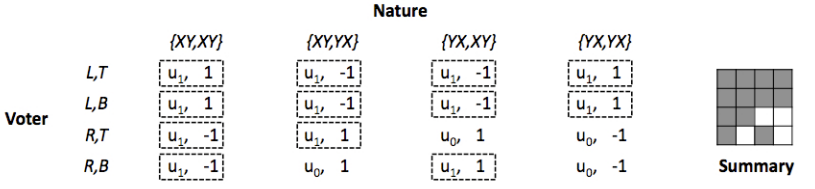
The normal form of the contract is shown in Figure 3(a).[8] Since $V$, the row player, only moves after observing the move made by $N$, we consider $V$'s best response to each of $N$'s actions separately, which is the highest payoff to $V$ (the first number in the pair of payoffs) in each column. This assumes the voter is utility-maximizing and is only interested in the highest payoff, a simplifying assumption that we will reconsider in Section 5.3.

The second payoff in the pair, with a slight abuse of notation, is to the influencer $I$ and not to the column player $N$. $I$ receives +1 when $V$ votes for Alice and -1 when she votes for Bob. Since $I$'s payoffs are not a function of how much money he is paying to $V$, this implicitly assumes that money is no object. This is a simplification that we will rectify in Section 5.4.

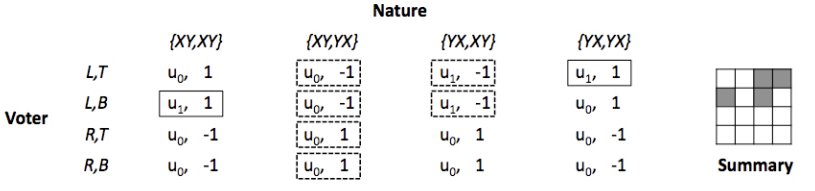
Recall that $N$ chooses each column with equal probability: 0.25. If the first column is selected, the voter will receive $u_1$ in any case. It is difficult to interpret what these weakly dominant responses mean to a utility-maximizing voter but

[7] All the contracts considered in this paper will be presented in their pro-Alice form for consistency and easy comparison. Due to the symmetric nature of the ballot casting process, any contract can be adopted for Bob instead.

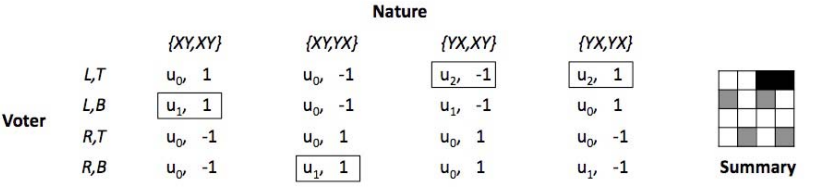
[8] The normal form of a game is a matrix with player 1's action set as the rows and player 2's action set as the columns. The elements contain a tuple: the payoff to player 1 and player 2 respectively for the selection of these actions (however note the deviation from convention in this case). A player's dominant strategy, if one exists, is the selection of an action that will always yield a higher payoff than any other action, and a weakly dominant strategy is the selection of an action that will yield at least as high of a payoff as any other action.




(a) MN




(b) BMR




(c) KRMC

**Fig. 3.** Three pro-Alice contracts in normal form. Lined boxes are dominant best responses, while dotted boxes are weakly dominant best responses. The underlying game is sequential with the column player moving first; thus each column is a subgame. *Notational abuse:* the first element of the payoff is to the row player, $V$, while the second element is to the influencer $I$; not the column player $N$. This latter utility distinguishes votes for Alice (+1) and for Bob (-1). The summary captures the payoff to the row player, visualizing higher utilities as darker squares.

let us assume the voter will choose randomly between them. This is more problematic in the second column, where the voter has three options: two of which result in a vote for Bob and one for Alice. We could assume the voter, caring only for the payoff, (i) chooses randomly between the two candidates or (ii) chooses randomly between the three options. This has an effect on $I$'s expected payoff. The third column is much like the second, while the final column is the interesting one: both options produce a vote for Alice. Thus $I$ is guaranteed a vote for Alice whenever this column is chosen by $N$.

We can calculate the probability of this contract resulting in a vote for Alice to be 0.625 under interpretation (i) and 0.54 under interpretation (ii). For all outcomes, $I$ will incur $u_1$, thus the purchase of a full vote for Alice requires manipulating 1.6 voters for a cost of $(1.6)u_1$ per vote under (i) and 1.85 voters

under (ii) for a per vote cost of $(1.85)u_1$. Although proposed as a vote-buying contract, it also works for coercion.

### 4.3 The BMR Contract

The second contract is due to Bohli, Müller-Quade, and Röhrich[9], and is presented by the authors as a vote-buying contract:

$$\text{Contract}_{BMR} = \begin{cases} u_1 = \pi_V(L, T \mid \{YX, \_\_\}) \\ u_1 = \pi_V(L, B \mid \{\_\_, XY\}) \\ u_0 \text{ otherwise} \end{cases}$$

The normal form of the contract is shown in Figure 3(b). A curiosity here is the second column, which yields $u_0$ regardless. If used coercively, with probability 0.25, the voter cannot escape punishment: there is no way, given a ballot like this from $N$, to please $I$. For this reason, we rule out the BMR contract as viable for coercion. The probability of the contract resulting in a vote for Alice is 0.625. This outcome will cost $I$ $(0.75)u_1$ (assuming $u_0$ is zero). The purchase of a full vote for Alice requires manipulating 1.6 voters for a cost of $(1.2)u_1$ per vote. Thus this contract is better than the MN contract for vote-buying.

### 4.4 The KRMC Contract

The final contract is due to Kelsey, Regenscheid, Moran, and Chaum [17], and is presented by the authors as a vote-buying contract:

$$\text{Contract}_{KRMC} = \begin{cases} u_2 = \pi_V(L, T \mid \{YX, \_\}) \\ u_1 = \pi_V(L, B \mid \{\_\_, XY\}) \\ u_1 = \pi_V(R, B \mid \{\_\_, YX\}) \\ u_0 \text{ otherwise} \end{cases}$$

The normal form of the contract is shown in Figure 3(c). The contract is similar to BMR, only it uses graduated payoffs and includes an additional clause to resolve the ambiguity in the second column of BMR. Every column contains a strongly dominant response, leaving no ambiguity to a utility-maximizing voter. It works for coercion, as well as vote-buying. The probability of the contract resulting in a vote for Alice is 0.75. This outcome will cost the influencer $(0.5)(u_1 + u_2)$. The purchase of a full vote for Alice requires manipulating 1.3 voters for a cost of $(0.5)(u_1 + u_2)$. Since $u_2$ needs to be only epsilon greater than $u_1$, this contract is more effective than BMR and MN; more applicable than BMR; and has less ambiguity than BMR and MN.

### 4.5 The Optimal Contract

We have seen three contracts with different properties and expected votes for Alice. KRMC is the best contract, and we seek to prove that it is optimal for the two-candidate case. We also demonstrate that using more than three levels

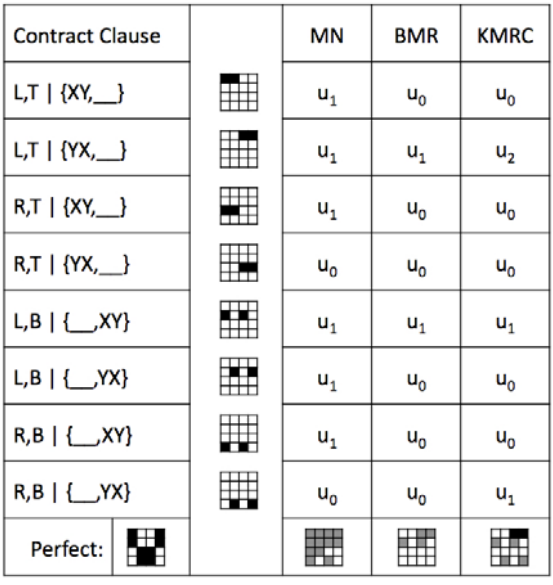



**Fig. 4.** A summary of the three contracts, delimitated by possible clauses

of utility does not increase the expected votes, independent of the number of candidates. Consider Figure 4. The leftmost column shows every possible (most specified) clause that could appear in a contract. While clauses do not have to be fully specified in each variable, such as the first clause in MN, such general clauses are some combination of the most specified clauses: the combination of the first, second, fifth, and sixth clauses in this case.

For each clause, the second column of the figure contains a small grid. This grid is intended to be a visualization of the payoff matrix, like the summaries in Figure 3. Let $C$ be the number of candidates. The rows of the grid represent the voter's binary choice between the top or bottom layer as well as the $C$-way choice of which position to mark; hence, $2C$ rows. The columns represent the order of the symbols on the ballots. These orderings are random rotations, not full permutations which simplifies the tallying process of Punchscan. There are $C^2$ possible orderings, not $C!$, and hence $C^2$ columns. Black cells represent the positions in the payoff matrix that will be affected by adding the clause. For example, if a contract offers a payoff of $u_1$ for receipts matching the first clause in the figure, $u_1$ will be added to the two indicated cells in the payoff matrix for the contract: cells (1,1) and (1,2). MN in Figure 3(a) is an example of contract that includes such a clause.

The next three columns summarize the three contracts in the literature and the payoffs they award for each clause. This information can be combined into a concise visualization of the contract by layering the grids associated with each clause on top of each other, where the darker squares represent a higher payoff to $V$ for that outcome. The concise form is shown in the bottom row of each contract.

As a reminder of which outcomes result in a vote for Alice, these outcomes are marked with black cells in the perfect contract in the bottom-left of the figure (*i.e.,* the elements in Figure 3 with payoffs of 1 to $I$). We refer to these cells as the Alice region of the grid (and the inverse set of cells as the Bob region). The perfect contract is not possible to achieve with the available clauses; however an optimal contract will resemble it as closely as possible.

Continue to consider a contract as a grid, with rows $0 \leq j \leq 2C-1$ and columns $0 \leq k \leq C^2-1$. Each element contains $u_i$ with $i \geq 0$. We note three properties:

**P1**: For each clause with utility $u_i$ in the contract, a column $\hat{k}$ has $u_i$ added to it in the Alice region.
**P2**: In P1, $u_i$ is always added to the region of each additional candidate in the same row and some column other than $\hat{k}$.
**P3**: In P2, the (set of) column(s) is either $\{k|\lfloor\frac{k}{C^2}\rfloor = \lfloor\frac{\hat{k}}{C^2}\rfloor\}$ or $\{k|k \equiv \hat{k} \mod C\}$.

Most specified clauses include a top or bottom layer, $T$ or $B$, and a marked position that we will now call $P_m$, where $0 \leq m \leq C-1$, instead of the two-candidate specific terms left and right. Clauses also include an ordering of symbols to appear on the receipt. Consider an arbitrary ordering to be the canonical ordering $\hat{o}$. The other possible orderings are generated by rotating this ordering right or left, which we denote with functions `ror`() or `rol`(). For example, if $\hat{o} = XY$ then $\texttt{ror}(\hat{o}) = YX$. This set has closure, such that $\texttt{ror}^C(\hat{o}) = \hat{o}$ (*i.e.,* `ror` applied C times to an ordering is the same ordering).

With these notational conventions, we construct a simple, $\mathcal{O}(C)$ greedy algorithm to select an optimal contract. An optimal contract should have three properties: (**O1**) the highest expected votes for Alice from utility-maximizing voters, (**O2**) no ambiguity (unlike MN), and (**O3**) no columns with all $u_0$ (unlike BMR). The algorithm selects a contract, w.l.o.g., for the first listed candidate (*i.e.,* a pro-Alice contract). There are many contracts satisfying the properties for optimality—this algorithm finds one instance. It is given in Algorithm 1.

For analysis of the algorithm, we explain each line in terms of the visualization of a contract as a grid. Line 1 of the algorithm adds $u_1$ to the Alice region in column 1 of the contract's grid. All clauses will add $u_1$ to the Alice region somewhere (**P1**), thus this clause is no worse than any other clause with respect to **O1**, and it is strictly better than adding no clauses with respect to **O3**.

**Algorithm 1.** Optimal Contract Generation

1 Add to contract: $u_1 = \pi_V(P_0, B|\{\_\_, \hat{o}\})$
2 **for** *m from* 1 *to* $C-1$ **do**
3 ⌊ Add to contract: $u_1 = \pi_V(P_m, B|\{\_\_, \texttt{ror}^m(\hat{o})\})$
4 **for** *m from* 1 *to* $C-1$ **do**
5 ⌊ Add to contract: $u_2 = \pi_V(P_m, T|\{\texttt{rol}^m(\hat{o}), \_\_\})$
6 Add to contract: $u_0 =$ otherwise

Lines 2-3 add $u_1$, on each iteration, to the Alice region in a new column, which is strictly better than not adding additional clauses with respect to **O2**. Each clause never adds $u_1$ to more than one column, whether in the Alice region or not (following $\{k|k \equiv \hat{k} \mod C\}$ in **P3**), thus it is no worse than any other group of clauses that could be provided. After completion of the loop, every column contains exactly one $u_1$, satisfying **O3**.

Lines 4-5 add additional clauses. To ensure **O2**, additional clauses have payoff $u_2$ and are non-overlapping in the columns that they affect. On each iteration, a clause adds $u_2$ to the Alice region, making it the best response over the $u_1$ already present in the column. Furthermore, the distribution of the added clauses follows $\{k|\lfloor \frac{k}{C^2} \rfloor = \lfloor \frac{\hat{k}}{C^2} \rfloor\}$ in **P3**, which ensures that the set of each $u_2$ added to another candidate's region (**P2**) is disjoint from the set of columns where Alice is the best response. Taken together, the addition of these clauses is strictly better than not adding the clause with respect to **O1**.

Line 6 suggests that no more clauses can be added that would improve the contract with respect to **O1** and the contract should be closed. Consider the addition of a clause with $u_i$. By **P1**, this would add $u_i$ to the Alice region in some column. For it to improve the contract with respect to **O1**, the Alice region of this column must not already contain the highest utility and $i$ must be greater than the highest utility present. All such columns contain $u_2$; thus all candidate clauses must be $u_3$. However by **P2**, this would also add $u_3$ to the regions of the other candidates in other columns. Given the distribution of this addition by either type in **P3**, this would add $u_3$ to a column that contains $u_2$ in the Alice region, making Alice no longer a best response for that column. Thus additional clauses cannot improve the contract with respect to **O1**.

Recall that Algorithm 1 finds one instance of the many contracts satisfying the properties for optimality. Running this algorithm for the two-candidate case produces the following contract:

$$\text{Contract}_{opt} = \begin{cases} u_2 = \pi_V(R, T \mid \{YX, \_\}) \\ u_1 = \pi_V(L, B \mid \{\_, XY\}) \\ u_1 = \pi_V(R, B \mid \{\_, YX\}) \\ u_0 \text{ otherwise} \end{cases}$$

This contract is equivalent to KRMC with respect to **O1**, **O2**, and **O3**. Therefore since the output contract is optimal, KRMC is as well.

## 5 Extending the Base Model

### 5.1 Multiple-Candidate Contracts

For our first extension to the basic contract, we consider the optimality of a contract as the number of candidates is increased from two to an arbitrary number, $C$, of candidates. **P1**, **P2**, and **P3** still hold. For an arbitrary $C$, an optimal contract for Alice can be constructed from the union of the first $C$ columns with payoff $u_1$ and every additional $C^{\text{th}}$ column with payoff $u_2$.

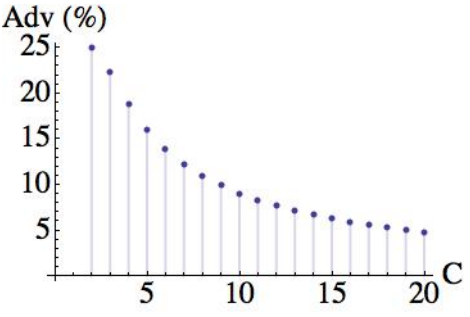



**Fig. 5.** The advantage of an optimal contract over a random selection as $C$, the number of candidates, grows. As seen, the advantage appears asymptotic to 0 in the number of candidates.

Figure 5 shows that the advantage an optimal contract offers over forcing a utility-maximizing voter to vote for a random candidate. The two-candidate case had a probability of 75% of resulting in a vote for Alice, which is a 25% advantage over a random choice between two candidates. The probability of a vote for Alice in the three-candidate case is 56%, which is only a 22% advantage (over a random choice between three candidates). Likewise, as $C$ increases, the adversary's advantage decreases.

### 5.2 Reordering the Game

We now consider the order of play. If $V$ were to choose either R or L prior to $N$ choosing the ballot layout, the candidate voted for would be random. Enforcing this in a contract, without any additional clauses, could be an effective denial of service attack—but it is no better than simply paying the voter to not vote at all. Thus if $V$ is to vote with intention, she can only choose between R and L after observing the moves by $N$. However the outcome of the game is invariant to whether $V$ chooses T or B, therefore this move could be safely relocated in the sequence of events. If it was chosen by $V$ prior to observing the moves by $N$, then no contract can be formed that would favour Alice or another candidate. In other words, this simple change solves the problem.

To see why, consider again the properties in Section 4.5. In this new extensive form, **P1** and **P2** still hold. However **P3** does not. If the top sheet is selected, then the set of columns in **P3** will only be $\{k|\lfloor\frac{k}{C^2}\rfloor = \lfloor\frac{\hat{k}}{C^2}\rfloor\}$. Likewise, if the bottom sheet is selected, the set of columns will only include $\{k|k \equiv \hat{k} \mod C\}$. With this symmetric pairing of columns, there is no way to asymmetrically win a column for Alice without losing one to another candidate.

Requiring the voter to select the top or bottom layer before seeing the ballot is a known solution, and the Punchscan procedure has been subsequently modified to reflect this change. However this change does cause the privacy of the system to be contingent on poll worker procedure, which is a weak foundation for something as critical as ballot secrecy. From the first author's experience, poll workers may

not follow procedures exactly, especially when deviation does not affect the voters ability to cast their ballot and the poll-workers do not have a solid mental model of why a procedure is important.

In general, removing decisions that a voter must make during the voting process, especially arbitrary choices made after observing the actions selected by nature can help resolve issues of coercion. However one choice can never be eliminated: selecting a candidate to vote for. For some obfuscation mechanisms in E2E systems, this decision alone may be exploitable. Thus, there is no simple trick—vote casting procedures must always be carefully examined.

### 5.3 Voter Types

So far, we have considered $V$ to be utility-maximizing and thus follows the contract fully. However, this is not necessarily the case, especially for vote-buying. There may be some voters who will forgo payment and always vote for Alice or Bob. In this case, they are still utility-maximizing: they receive utility that is external to the contract from their political convictions or expected benefits from an elected candidate. Our use of the term utility-maximizing should be interpreted as maximizing only the utility internal to the contract. There may be other "vengeful" voters who would punish the adversary whenever possible: for example, by always choosing to vote contrary to the adversary when given the choice. In the next section, we also consider opportunistic voters who will sell their vote if they are already intending to vote for the adversary's candidate. In all of these cases, the true type of the voter is hidden from the influencer.

Consider a simple split between the fraction of utility-maximizing voters ($\alpha$) and vengeful voters ($1-\alpha$) in the coercion model. The adversary will accept any payoff in the set of best responses. In MN, the expected votes for Alice from a vengeful voter is 0.25. Thus to make ground for Alice, the following expression should hold: $(0.54)\alpha + (0.25)(1 - \alpha) > 0.5$. The means $\alpha > 0.86$ or at least 86% of the voters need to be utility-maximizing for the attack to work. Using the same analysis for BMR, recall that in BMR it is possible for $V$ to obtain a layout from $N$ that has $u_0$ as a payoff for all moves by $V$. As a result, a vengeful $V$ can always vote for Bob, even if the payoff is $u_0$ since $V$ could have plausibly received a bad ballot type. As a result, $\alpha > 0.769$ which means BMR can tolerate a higher proportion of vengeful voters than MN while maintaining profitability. For KRMC, a vengeful voter can at best vote for Bob on half of the columns (by receiving $u_1$ on the fourth column). Thus any $\alpha > 0$ will produce profitability for KRMC, making it the most resilient of the three.

### 5.4 Money Is an Object

Consider the vote-buying model. In this case, voters could simply choose to reject the contract, receive $u_0$, and vote for either Alice or Bob. However, occasionally such strategies will coincidentally allow them to meet the terms of the contract and be paid. Say that the fraction of voters rejecting the contract and voting for Alice is $p_a$ and for Bob is $p_b$. The fraction that are utility-maximizing and opt

into the contract is, as before, $\alpha$, and the vengeful voters make up the remainder. Unlike in the coercive case, vengeful voters will accept $u_0$ and thus act like voters in $p_b$. For KRMC, the expected amount of money paid by the adversary to a voter of a hidden type is,

$$0.5(u_1)(p_a) + 0.25(u_2)(p_a) + 0.25(u_1)(p_b) + 0.25(u_2)(p_b) + 0.5(u_1)(\alpha) \\ +0.5(u_2)(\alpha) + 0.5(u_1)(1 - p_a - p_b - \alpha) + 0.25(u_2)(1 - p_a - p_b - \alpha)$$

Before we assumed that money was no object, so the value of this expression is irrelevant. However if money does matter, then the adversary must ensure that he is not paying more for a vote than it is worth to him. The difference between a voter in $p_a$ and $\alpha$ can be rephrased: the latter place less value on their vote and thus will choose to accept a payoff that is higher than the amount of value they place on their vote. Let $U_v$ be the value of $u_1$ such that $\alpha$ voters will accept the contract; in other words, the maximum value a voter in $\alpha$ places on their vote. Since $u_2$ only needs to be marginally greater than $u_1$ (*i.e.,* $u_2 = u_1 + \epsilon$), we can assume for simplicity that they are equivalent. Furthermore, assume that $p_a$ is the same as $p_b$, since close elections will more plausibly have attempts at undue influence. This reduces the equation above to

$$U_v(0.75 - 0.25(p_a) + 0.25(\alpha)).$$

For it to be profitable for the vote buyer, he expects to influence a share of $0.75(\alpha)$ votes in favour of Alice, and if a vote is worth on average $U_b$ to the buyer then

$$U_v(0.75 - 0.25(p_a) + 0.25(\alpha)) < U_b(0.75(\alpha)).$$

This expression forms a ratio between how much a vote is worth to the coercer and how much it is worth to the voter, and how large the ratio must be for KRMC to be profitable. For example, if $p_a = p_b = 0.45$ and $\alpha = 0.10$, the ratio is 8.17. This means that if 10% of voters value their vote at less than, say, \$10, the buyer should only exploit this opportunity if a vote gained is worth at least \$82 to him. For three candidates, $p_a = p_b = p_c = 0.30$ and $\alpha = 0.10$, the vote should be worth at least \$96 to him.

## 6 Future Work and Concluding Remarks

We have shown how game theoretic-models can be applied to analysing coercion contracts in E2E voting systems. We developed an algorithm for devising optimal contracts, proved that KRMC is optimal in the two candidate case, and found that the effectiveness of contracts decrease with the number of candidates are added. We also show that no more than two levels of utility are needed and that contracts are costly for the adversary.

We conclude with a few avenues for future work. In paper-based elections, where unrecoverable errors are possible, voters are typically given the option to

spoil a ballot and receive a new one. Future work could examine the impact of spoiling on coercion contracts in realistic scenarios, like being allowed up to two spoiled ballots: some voters will spoil to try and receive higher payoffs, others may spoil to avoid meeting the adversary's demands. Voters must strategize whether spoiling is likely to increase or decrease their fortunes when the payoffs are ternary or when there are multiple contests on the ballot, each with its own payoff. It may also be plausible for the adversary to observe when the voter spoils a ballot, and he may adjust his own strategies accordingly.

Our definition of optimality assumes voters are utility-maximizing, and we later study the performance of these contracts in a setting for which they were not optimized: voters with hidden types. We conjecture that reoptimizing the contracts for this setting would not change the contract; however, we leave proof of this for future work. A final topic for further exploration is the potential for adversaries to employ screening techniques to differentiate between voters with hidden types. For example, the payoff could include a contribution to one candidate's campaign to prevent supporters of another candidate from accepting the contract if their receipt coincidentally meets its conditions.

We hope our study of contracts in Punchscan, and the tools we have used in our analysis, is of assistance to the designers of E2E systems. The more we understand these attacks, the easier it will be to design against them.

**Acknowledgements.** The authors acknowledge the support of this research by the Natural Sciences and Engineering Research Council of Canada (NSERC)—the first author through a Canada Graduate Scholarship, and the second and third through Discovery Grants.

## References

1. Adida, B.: Helios: web-based open-audit voting. In: USENIX Security Symposium (2008)
2. Adida, B., Neff, C.A.: Ballot Casting Assurance. In: Electronic Voting Technology Workshop, EVT (2006)
3. Adida, B., Rivest, R.L.: Scratch & vote: self-contained paper-based cryptographic voting. In: Workshop on Privacy in Electronic Society (2006)
4. Bartholdi, J., Orlin, J.: Single transferable vote resists strategic voting. Social Choice and Welfare 8(4) (1991)
5. Benaloh, J.: Ballot Casting Assurance via Voter-Initiated Poll Station Auditing. In: Electronic Voting Technology Workshop, EVT (2007)
6. Benaloh, J.: Secret sharing homomorphisms: Keeping a secret secret. In: Eurocrypt 1986 (1986)
7. Benaloh, J.: Simple veriable elections. In: Electronic Voting Technology Workshop, EVT (2006)
8. Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections. In: ACM Symposium on Theory of Computing, STOC (1994)
9. Bohli, J., Müller-Quade, J., Röhrich, S.: Bingo Voting: Secure and Coercion-Free Voting Using a Trusted Random Number Generator. In: Alkassar, A., Volkamer, M. (eds.) VOTE-ID 2007. LNCS, vol. 4896, pp. 111–124. Springer, Heidelberg (2007)

10. Chaum, D., Carback, R., Clark, J., Essex, A., Popoveniuc, S., Rivest, R., Ryan, P., Shen, E., Sherman, A.: Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems using Invisible Ink Confirmation Codes. In: Electronic Voting Technology Workshop, EVT (2008)
11. Chaum, D., Essex, A., Carback, R., Clark, J., Popoveniuc, S., Sherman, A., Vora, P.: Scantegrity: End-to-End Voter Verifiable Optical-Scan Voting. IEEE Security & Privacy 6(3) (2008)
12. Chaum, D.: Secret-Ballot Receipts: True Voter-Verifiable Elections. IEEE Security & Privacy 2(1) (2004)
13. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM (1981)
14. Chaum, D., Ryan, P.Y.A., Schneider, S.: A practical, voter-verifiable election scheme. Technical Report CS-TR-880, University of Newcastle upon Tyne (2004)
15. Fisher, K., Carback, R., Sherman, A.T.: Punchscan: introduction and system definition of a high-integrity election system. In: Workshop on Trustworthy Elections, WOTE (2006)
16. Juels, A., Catalano, D., Jakobsson, M.: Coercion-Resistant Electronic Elections. In: Workshop on Privacy in Electronic Society (2005)
17. Kelsey, J., Regenscheid, A., Moran, T., Chaum, D.: Hacking Paper: Some Random Attacks on Paper-Based E2E Systems. Frontiers of Electronic Voting (2007)
18. Moran, T., Naor, M.: Split-Ballot Voting: Everlasting Privacy With Distributed Trust. In: ACM Conference on Computer and Communications Security, CCS (2007)
19. Myers, A.C., Clarkson, M., Chong, S.: Civitas: Toward a secure voting system. In: IEEE Symposium on Security and Privacy (2008)
20. Neff, C.A.: A Veriable Secret Shufe and its Application to E-Voting. In: ACM Conference on Computer and Communications Security, CCS (2001)
21. Osbourne, M.J.: An introduction to game theory. Oxford University Press, Oxford (2003)
22. Park, C., Itoh, K., Kurosawa, K.: All/Nothing Election Scheme and Anonymous Channel. In: Eurocrypt 1993 (1993)
23. Popoveniuc, S., Hosp, B.: An introduction to Punchscan. In: Workshop on Trustworthy Elections, WOTE (2007)
24. Rivest, R., Smith, W.D.: Three Voting Protocols: ThreeBallot, VAV, and Twin. In: Electronic Voting Technology Workshop, EVT (2007)
25. Sandler, D.R., Derr, K., Wallach, D.S.: VoteBox: a tamper-evident, veriable electronic voting system. In: USENIX Security Symposium (2008)