Strategies for Voter-Initiated Election Audits

Chris Culnane(✉) and Vanessa Teague

Department of Computing and Information Systems,
Melbourne School of Engineering, University of Melbourne,
Parkville, VIC 3010, Australia
{christopher.culnane,vjteague}@unimelb.edu.au

Abstract. Many verifiable electronic voting systems are dependent on voter-initiated auditing. This auditing allows the voter to check the construction of their cryptographic ballot, and is essential in both gaining assurance in the honesty of the constructing device, and ensuring the integrity of the election as a whole. A popular audit approach is the Benaloh Challenge [5], which involves first constructing the complete ballot, before asking the voter whether they wish to cast or audit it.

In this paper we model the Benaloh Challenge as an inspection game, and evaluate various voter strategies for deciding whether to cast or audit their ballot. We shall show that the natural strategies for voter-initiated auditing do not form Nash equilibria, assuming a payoff matrix that describes remote voting. This prevents authorities from providing voters with a sensible auditing strategy. We will also show that when the constructing device has prior knowledge of how a voter might vote, it critically undermines the effectiveness of the auditing. This is particularly relevant to internet voting systems, some of which also rely on Benaloh Challenges for their auditing step.

A parallel version, in which the voter constructs multiple ballots and then chooses which one to vote with, can form Nash equilibria. It still relies on some uncertainty about which one the voter will choose.

1 Introduction

Verifiable electronic voting systems aim to provide strong integrity guarantees and protection from tampering. In order to deliver this, they provide a number of verifiability properties, namely, cast-as-intended and counted-as-cast. Cast-as-intended means that the cast ballot accurately reflects the intentions of the voter, it is verifiable if the voter has the opportunity to gain assurance that the vote was cast in keeping with their intentions. Counted-as-cast means that the cast ballots are correctly counted.

In this paper we are only interested in the first of these properties, cast-as-intended. A popular technique for providing cast-as-intended verifiability is to

C. Culnane—Partially funded by Australian Research Council Discovery project DP140101119 'More information for better utility; less information for better privacy'.

Q. Zhu et al. (Eds.): GameSec 2016, LNCS 9996, pp. 235–247, 2016.
DOI: 10.1007/978-3-319-47413-7_14

provide an auditing step of ballots. Such a step aims to assure the voter that ballot is correctly constructed, and will therefore accurately reflect their vote. The exact methodology of the audit is dependent on the system, but broadly falls into two categories, cut-and-choose [2] and the Benaloh Challenge [5]. The cut-and-choose approach is applicable to systems that pre-construct ballot papers. Such systems allow a voter to choose whether to audit or vote with the ballot they are given. If they choose to audit, the ballot is opened and the voter may check the construction of any cryptographic values. However, once a ballot has been audited it cannot be used for voting, since this would break the privacy of the voter and the secrecy of the ballot. Provided a sufficient number of audits are performed, and assuming none fail and that the constructing device did not know whether a particular ballot would be audited, there is a very high probability that the ballots were honestly constructed. Crucially, the audit takes place prior to the voter expressing any preferences. Such an approach is used in Prêt à Voter [13] and Scantegrity [8].

The Benaloh Challenge is similar, except it is used in systems where ballots are not pre-constructed. In such systems, a voter first enters their preferences and constructs their encrypted ballot on a voting device, they are then given the choice of whether they wish to vote or audit the constructed ballot. If they choose to audit it, the ballot is opened, allowing the voter to check the cryptographic construction of the ballot. Crucially, the audit takes place after the voter has expressed their preferences. Like the cut-and-choose approach, once a ballot has been opened it cannot be used for voting, and therefore the voter must construct a new ballot to vote with. Such an approach is used in Helios [1], Wombat [4] and Star-Vote [3]. Both approaches allow voters to repeat the audit step as many times as they like—the protocol ends when the voter decides to cast their ballot. As such, their final cast ballot will not be audited, and their assurance that it has been cast-as-intended is based on having run a number of successful rounds of auditing previously, or in the general case, that enough other people have run successful rounds of auditing.

In this paper, we will evaluate Benaloh Challenges from a game theoretic point of view using a game that describes the payoffs of a remote voting setting. We analyze the effectiveness of various voter strategies when choosing whether to cast or audit a constructed ballot, and the corresponding device strategies for constructing a dishonest ballot. We will show that none of the natural strategies for voter-initiated auditing, using Benaloh Challenges, form Nash equilibria. This presents a particular problem for voting systems relying on such auditing, since it precludes providing the voter with instructions on a sensible auditing strategy. The provision of such advice, when it does not form a Nash equilibria, can do more harm than good, creating a potential advantage for a cheating device. This calls into question the validity of the cast-as-intended auditing in verifiable remote electronic voting systems that utilise Benaloh Challenges. Modelling an attendance voting setting, in which there is a higher penalty for device misbehaviour, is important future work.

A simple parallel variant, in which voters are instructed to make multiple ciphertexts and then choose one to vote with, can form a Nash equilibrium. However, this too needs careful analysis of the cheating machine's ability to guess which vote will be cast. The estimate must be correct or what seems to be a Nash equilibrium might not be.

2 Voter-Initiated Auditing

We are primarily interested in voter-initiated auditing used in schemes that construct encrypted ballots. As such, we shall focus on Benaloh Challenges [5], which have been widely adopted as the auditing technique for such schemes.

2.1 Purpose of Auditing

Arguably the purpose of audits is not just to detect cheating, but to provide an evidence trail after the fact to support an announced election outcome. For example, Risk Limiting Audits [12] of a voter-verifiable paper trail provide a statistical bound on the likelihood that an undetected error might have changed the election outcome. We might hope to conduct a statistical assessment of the transcript of a voter-initiated electronic auditing procedure, in order to produce the same sort of guarantee. However, this work shows that such an assessment would be very difficult to perform. In particular, a naive computation of the probability of detection given the rate of auditing would give an incorrectly high degree of confidence.

2.2 Origins of Benaloh Challenges

Benaloh Challenges were first introduced in [5], and later refined in [6]. Benaloh Challenges are an auditing technique that can be used by voting systems that construct encrypted ballots. They are commonly referred to as "cast-or-audit", on account of the technique involving first constructing an encrypted ballot, followed by asking the voter whether they wish to cast or audit it. If the voter chooses to cast the ballot it will be signed, or otherwise marked for voting, and included in the tally. If the voter chooses to audit the ballot, the encryptions are opened to allow the voter to verify that the ballot was correctly constructed from their preferences. The Benaloh style of auditing has been widely adopted in the verifiable voting field, in both theory and practice, including in Helios [1], VoteBox [15], Wombat [4], and StarVote [3]. Of particular interest is Helios [1], which is a web-based open-audit voting system, which has been used in binding elections, notably, the International Association for Cryptologic Research (IACR) elections.

2.3 Making Audit Data Public

Benaloh, in [6], makes no mention of whether the audit information is made public. However, there is a discussion on providing assurance of integrity to a wider population from a smaller random sample of audits. This would seem to indicate the necessity that the auditing is made public, so as to enable that wider population to inspect it. The original version of Helios [1] did not mention the posting of audit data, however, in Helios V3 there is a provision for posting audit data to the public bulletin board [10]. In Wombat [4] the audited ballot must be shown to an election worker, and taken home to be checked, all of which threatens the secrecy of the vote.

2.4 Revealing Voter Intent via an Audit

The auditing step, by its very nature, reveals a set of preferences and the corresponding ciphertext construction. If those preferences are a true reflection of the voters intent, the audit will reveal the voters intent, and thus break ballot secrecy. This is equally problematic whether the information is posted publicly, or shown to an election official for checking.

If the voter is deciding after construction whether to vote or audit, as described in [6], the voter will be obliged to always construct a ballot with their true preferences, and as a result, any audit will break ballot secrecy. A simple counter strategy is for the voter to construct a ballot with fake preferences to audit. Crucially, this requires the voter to decide whether to cast or audit prior to ballot construction. It is critical that the machine cannot distinguish between a voter creating a genuine ballot and a voter constructing an audit ballot.

2.5 Indistinguishability of Real and Fake Ballots

The requirement for indistinguishability between a ballot that will be audited and one that will be voted with is implicitly covered by an assumption in [6], which states that it is crucial that the ballot encryption device does not receive any information that may indicate the likelihood of a ballot being audited. However, realising this assumption presents a significant challenge, even in a secure polling place. Whilst it seems possible that the voters identity could be hidden from the machine, it seems impossible to exclude global information from being used to indicate whether a ballot will be cast or audited. Such information could include voting preference patterns, geographical voting patterns and election wide voter guidelines, all of which could be used to infer information about whether a ballot is being constructed for voting or audit.

For example, it is easy to see how voters could easily fall into a pattern of auditing one ballot, and if that succeeds, voting with the next. Such a pattern has been seen in real-world elections using Helios, in [11] the authors analyse the first IACR election, showing a clear pattern for performing zero or one audit, but very rarely anymore.

2.6 Benaloh Challenges in Remote Voting

Benaloh Challenges were proposed for the supervised voting setting, and not for remote voting. Supervised voting refers to voting that takes place in a controlled environment, for example, at a polling location on election day. Remote voting is considered to be any voting that takes places outside of a controlled environment, for example, voting from home over the internet. Helios [1] is a remote voting system which constructs ballots on a voter's own device. Such a device is likely to be able to infer significant information about the behaviour, and therefore voting preferences, of the user. In particular, since Helios was first designed in 2008, there has been a great increase in the intrusiveness of privacy invasion via identifying individuals' online behaviour [9]. It is clear that in the remote setting it is feasible for the device to be able to predict the voting intention of the voter. In the supervised setting, identifying an individual voter is unlikely, however, identifying groups of voters, or general patterns, is still feasible.

The payoffs for cheating and the penalties for failure are also different in the remote vs attendance setting. In the remote setting, typically only one or two voters use the same device, and there is no independent check when cheating is detected; in the attendance setting, a successfully cheating device could take hundreds or thousands of votes, and the penalties for cheating could be severe. For the rest of the paper, we consider only the remote setting, leaving the attendance game for future work.

3 The Game Theory Model - Inspection Game

We model the interaction as an inspection game in which the voter is the inspector and the device wins only if it cheats and is not inspected. Voters incur a small cost for inspecting, a benefit from successfully casting the vote of their choice, and a large cost for having their vote inaccurately recorded. The device (which aims to cheat) benefits from getting away with casting a vote other than the voter's intention.

The voter begins with a true vote v_t chosen from a publicly-known probability distribution Π.

In the first step, the voter (V) chooses a vote from the range of Π and sends it to the device (D). The device then chooses whether to encode it truthfully (T) or falsely (F), but this choice cannot be observed by V. Next, V may cast the vote (C), in which case the game ends without revealing the device's choice, or she may audit the vote (A), so the device's choice is revealed. If she audits a truthfully-encoded vote, the process begins again. Otherwise, the game ends. Payoffs for one step of the game are shown in Fig. 1. G_V is a positive payoff reflecting the voter's successful casting of her intended ballot; $-B_V$ is the negative payoff when she is tricked into casting a vote other than v_t. For the device, G_D is the positive payoff associated with successfully casting a vote other than the voter's intention; $-B_D$ is the negative payoff associated with being caught cheating. The voter incurs a small cost $-c_{audit}$ for each audit.

		Voter(*V*)	
		Cast(*C*)	*Audit*(*A*)
Device (D)	*Truthful*(*T*)	$(0, G_V)$	$Add(0, -c_{audit})$; *repeat.*
	False(*F*)	$(G_D, -B_V)$	$(-B_D, -c_{audit})$

Fig. 1. Payoffs for one step of the game. If the device is truthful and the voter audits, the game repeats.

	Voter Payoff	**Device Payoff**	**Description**
$n_{cast} > n_{false}$	$-n_{false}c_{audit}$	$-B_D$	Voter catches cheating device.
$n_{cast} = n_{false}$	$-(n_{cast} - 1)c_{audit} - B_V$	G_D	Device successfully cheats.
$n_{cast} < n_{false}$	$-(n_{cast} - 1)c_{audit} + G_V$	0	Voter votes as intended.

Fig. 2. Payoffs for the extended game. The voter casts at step n_{cast}; the device encodes falsely (for the first and only time) at step n_{false}

In order to model the repeated nature of the game, the voter's strategy is a sequence of n votes, followed by n choices to audit, then a final $n+1$-th vote that is cast. The device's strategy is a sequence of choices to encode truthfully or falsely, which may be random or may depend on what the voter has chosen.

Assumptions.

1. that $c_{audit} < B_V$,
2. that (it's common knowledge that) the voter never casts a vote other than v_t,

Whatever the voter's strategy, the first false encoding by the device ends the game. We can therefore describe D's strategy completely as a choice of n_{false}, the first step at which D encodes falsely, preceded by truthful rounds. Of course this can be random, or can depend on the votes that the voter has requested before then. The game's outcome depends on whether n_{false} is earlier, later, or exactly equal to the round n_{cast} in which the voter chooses to cast. This gives us the payoff table, shown in Fig. 2, for the extended game.

3.1 Negative Results: Simple Sequential Strategies Do Not Form Nash Equilibria

As expected in an inspection game, it is immediately clear that there is no pure strategy equilibrium. Indeed, there is no equilibrium with a fixed value of n.

Lemma 1. *If $c_{audit} < B_V$, there is no Nash equilibrium in which the voter's number of audits is fixed.*

Proof. Suppose V always audits $n_{cast} - 1$ times, and suppose this is a Nash equilibrium with some strategy S_D by the device D. Then S_D must specify encoding untruthfully in the n_{cast}-th round—otherwise there would be a strict unilateral improvement by doing so. But this gives V a payoff of $nc_{audit} - B_V$, which is bad. This could be improved to $(n-1) * c_{audit}$ by auditing at round n_{cast}, which is strictly better assuming that $c_{audit} < B_V$. □

Also backward induction applies:

Lemma 2. *Suppose there is a common-knowledge upper bound $n_{\max}$ on n_{cast}. If $c_{audit} < B_V$, then there is no Nash equilibrium in which the voter votes as intended.*

Proof. Backward induction. The device's best response is to cheat at round $n_{\max}$, whenever the game gets that far, thus giving V the worst possible payoff. But then V improves her payoff by never waiting until $n_{\max}$, and instead casting at round $n_{\max} - 1$. The induction step is similar: if V is guaranteed not to audit at round n_i, then D should cheat at round n_i, and V would improve her payoff by casting at round $n_i - 1$. □

Lemma 3. *There is no Nash equilibrium in which, for any n, the probability that D encodes falsely at round n is zero.*

Proof. V should always cast then, so D should always cheat then. □

Now we can address our main question: whether information about the true vote can influence the device's success in getting away with cheating (and hence both parties' payoffs in the game).

Lemma 4. *If $-B_D < 0$, there is no Nash Equilibrium in which, with nonzero probability, D falsely encodes a vote outside the support of Π.*

Proof. Suppose S_D is a device strategy and let n be the first round at which, with nonzero probability, V chooses and D falsely encodes a vote outside the support of Π. Then V will certainly audit this vote (by Assumption 2), leading the device to a payoff of $-B_D$, the worst possible. If D was always truthful, it could guarantee a payoff of 0. □

Lemma 5. *If $-B_D < 0$, then every device strategy in which, with nonzero probability, D falsely encodes a vote outside the support of Π, is weakly dominated.*

Proof. Similar. Weakly dominated by the always-truthful strategy. □

So whether we take the solution concept to be Nash Equilibrium or survival of iterated deletion of weakly dominated strategies, we see that there is no solution in which the device falsely encodes a vote that the voter could not possibly intend to cast. This has important implications, particularly for voting from home, where the device may have very accurate information about the voter's intentions. In many practical scenarios, Π is a point function—the device knows exactly how the person will vote.

Note that the argument does not hold if $B_D = 0$, meaning that there is no downside to being caught cheating.

The strategy most commonly recommended to voters is to toss a coin at each stage and then, based on the outcome, to either cast their true vote v_t or choose some vote and audit it. We distinguish between a few such strategies:

TruthAndCoinToss. Always request v_t; toss a coin to decide whether to cast or audit.

PiAndCoinToss. Toss a coin to decide whether to cast or audit; in the case of audit, choose a vote at random according to Π.

These two are the same if Π has only one nonzero probability.

On the device side, recall that the strategy is determined entirely by the choice of which round to encode untruthfully in. We have already argued that there is no Nash equilibrium in which there is an upper bound on n_{false} (Lemma 3). We first examine the equilibria in which the voter plays TruthAndCoinToss. There is no new information communicated to the device as this strategy plays out: S_D consists entirely of a (static) probability distribution for choosing n_{false}.

We begin with the surprising result that there is no Nash equilibrium in which V plays TruthAndCoinToss—the probability of detecting cheating falls off too quickly.

Theorem 1. *There is no Nash equilibrium in which V plays* TruthAndCoinToss*.*

Proof. We're assuming that $Pr(n_{cast} = i) = 1/2^i$. Crucially, the game tree has only one branch, the one in which the voter always requests the same vote. The device's strategy is therefore simply a probability distribution P_D for n_{false}. Its expected payoff is

$$\begin{aligned}\mathbb{E}(D\text{'s } payoff) &= \sum_{i=1}^{\infty} \big(G_D Pr(n_{cast} = i) Pr_D(n_{false} = i) - B_D Pr(n_{cast} > i) Pr_D(n_{false} = i)\big) \\ &= (G_D - B_D) \sum_{i=1}^{\infty} Pr_D(n_{false} = i)/2^i\end{aligned}$$

(Note that the case in which $n_{cast} > i$ and $n_{false} = i$ gives D a zero payoff.)

This is strictly maximised by setting $Pr_D(n_{false} = 1) = 1$, that is, falsely encoding always on the first round. But then V could improve her payoff by always auditing in round 1 (by Assumption 1, $c_{audit} < B_V$). □

The following corollary shows that, if the device knows exactly how the voter is going to vote, the coin-tossing advice doesn't produce a Nash equilibrium.

Corollary 1. *If Π is a point function, then there is no Nash equilibrium in which V plays* PiAndCoinToss*.*

Proof. Immediate from Theorem 1

This easily generalises to any exponentially-decreasing auditing strategy with any coefficient. Suppose the voter, at round i, chooses to audit the vote with probability r, and otherwise to cast. The generalised strategies are

TruthAndRandomChoice(r). Always request v_t; audit with probability r, otherwise cast.

PiAndRandomChoice(r). Audit with probability r, otherwise cast. In the case of audit, choose a vote at random according to Π.

Again these are not part of any Nash equilibrium.

Lemma 6. *There is no Nash equilibrium in which V plays* TruthAndRandomChoice(r) *for any $r \in (0, 1)$.*

Proof. First compute the probabilities of casting at or after round i:

$$Pr(n_{cast} = i) = r^{i-1}(1 - r).$$

$$Pr(n_{cast} > i) = r^i.$$

So we can recompute D's expected payoff as

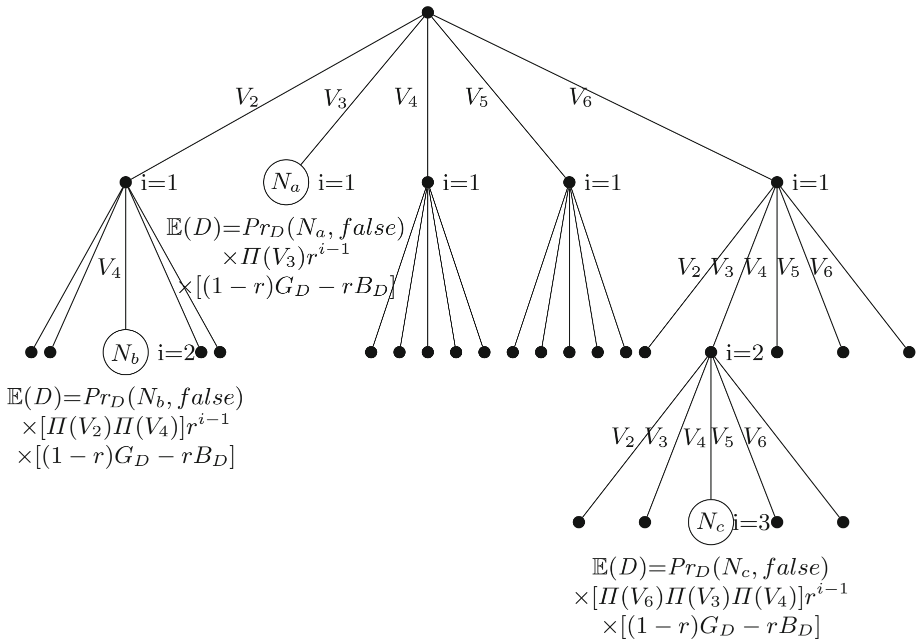
$$\begin{aligned}
\mathbb{E}(D\text{'s payoff}) &= \sum_{i=1}^{\infty} \left(G_D r^{i-1}(1-r) Pr_D(n_{false} = i) - B_D r^i Pr_D(n_{false} = i) \right) \\
&= [(1-r)G_D - rB_D] \sum_{i=1}^{\infty} Pr_D(n_{false} = i) r^{i-1} \\
&= [(1-r)G_D - rB_D] \left(Pr_D(n_{false} = 1) + r \sum_{i=2}^{\infty} Pr_D(n_{false} = i) r^{i-2} \right) \\
&\leq [(1-r)G_D - rB_D] \left(Pr_D(n_{false} = 1) + r(1 - Pr_D(n_{false} = 1)) \right) \\
&\leq [(1-r)G_D - rB_D] Pr_D(n_{false} = 1) \text{ because } r < 1.
\end{aligned}$$

Again, this is strictly maximised, to $[(1-r)G_D - rB_D]$, when $Pr_D(n_{false} = 1) = 1$. In other words, the device always cheats in the first round. This is clearly not part of any Nash equilibrium in which the voter does not always audit. □

This result generalises to Π being a more general distribution over possible votes. Suppose the Voter's strategy is PiAndRandomChoice(r). Suppose also that the voter's true vote v_t is chosen according to Π. One way to think of it is that Π represents the voter's guess about what the machine guesses V's distribution to be. In equilibrium, they should match.

Theorem 2. *There is no Nash equilibrium in which V plays* PiAndRandomChoice(r) *for any $r \in (0, 1)$ or any probability distribution Π, assuming that the true vote v_t is also chosen according to Π.*

Proof. Think about the tree of all possible sequences of vote requests, shown in Fig. 3. The device's strategy is described by a probability P_D that takes a node N in the tree and outputs a probability of playing F for the first time at N.



V_2
V_3
V_4
V_5
V_6
i=1
N_a
i=1
i=1
i=1
i=1
$\mathbb{E}(D)$=$Pr_D(N_a, false)$
$\times \Pi(V_3)r^{i-1}$
$\times[(1-r)G_D - rB_D]$
V_4
N_b
i=2
$\mathbb{E}(D)$=$Pr_D(N_b, false)$
$\times[\Pi(V_2)\Pi(V_4)]r^{i-1}$
$\times[(1-r)G_D - rB_D]$
V_2 V_3 V_4 V_5 V_6
i=2
V_2 V_3 V_4 V_5 V_6
N_c
i=3
$\mathbb{E}(D)$=$Pr_D(N_c, false)$
$\times[\Pi(V_6)\Pi(V_3)\Pi(V_4)]r^{i-1}$
$\times[(1-r)G_D - rB_D]$

Fig. 3. Game tree

To be a meaningful probability distribution, we require that, along any (possibly infinite) path p down the tree, $\sum_{N \in p} P_D(N, \mathit{false}) \leq 1$.

The probability of reaching node N at all, assuming that D is truthful until then, is determined by V's strategy S_V. The probability that a particular node N is reached is simply the product of all the vote choices along that path, times r^{i-1}, where i is its depth in the tree (starting at 1).

Since a false encoding ends the game (one way or another), we can attach an expected payoff to each node, representing D's expected payoff from the game ending at that node. Remember that when D is truthful it derives no payoff. For example, in Fig. 3, the probability of reaching node N_b is $[\Pi(V_2)\Pi(V_4)]r$ and the probability the device plays false at that node is $Pr_D(N_b, false)$. In general:

$$\mathbb{E}(\mathit{D\text{'}s\ payoff\ from\ node\ N}) = [(1-r)G_D - rB_D]Pr_D(N, \mathit{false})Pr_{S_V}(\mathit{N\ is\ reached})$$

We claim that D's best response to PiAndRandomChoice(r) is to play *false* always at $i = 1$. In other words, to cheat at the first opportunity. To see why, suppose instead that there is some best response P_{D-best}, in which there is some (parent) node N_p at level $i \geq 1$ such that

$$\sum_{N_c\, a\, child\, of\, N_p} P_{D-best}(N_c, \mathit{false}) > 0.$$

But now D's payoff can be strictly improved by shifting to strategy P'_{D-best} in which all the probabilities in N_p's children are shifted up to N_p. Let $\alpha = \sum_{N_c\, a\, child\, of\, N_p} P_{D-best}(N_c, false)$. The improved strategy is:

$$P'_{D-best}(N, \mathit{false}) = \begin{cases} P_{D-best}(N, \mathit{false}) + \alpha, & \text{when } N = N_p; \\ 0 & \text{when } N \text{ is a child of } N_p; \\ P_{D-best}(N, \mathit{false}) & \text{otherwise.} \end{cases}$$

This is strictly better than P_{D-best} because the total probability of reaching any of N_p's children is at most r (conditioned on having reached N_p), which is less than 1. The expected payoff is increased by at least $(1-r)\alpha[(1-r)G_D - rB_D]$.

Hence there is no best response to PiAndRandomChoice(r) other than always playing *false* at the first opportunity. Hence PiAndRandomChoice(r) is not part of any Nash equilibrium. □

3.2 Positive Results: Parallel Strategies Can Form Nash Equilibria, but only if the Device's Probability of Guessing the Voter's Choice Is Small Enough

Now consider an apparently-slight variation: instead of auditing sequentially, the voter makes some fixed number (k) of ciphertexts, chooses one at random to cast, then audits the other $k - 1$. Again, if they're all the same, the device has no information about which one will be cast, but privacy is compromised; if they're not all the same then the voter has to simulate some distribution for the $k - 1$ that are audited. In either case, if the device's probability of guessing correctly which vote will be cast is α, its expected payoff for cheating is

$$\mathbb{E}(\mathit{D\text{'}s\ payoff\ from\ cheating}) = [\alpha G_D - (1 - \alpha)B_D]$$

If all the votes are identical, or if the device has no information about how V will vote, then $\alpha = (k - 1)/k$. Depending on whether its expected payoff for cheating is positive or negative, it will be a Nash equilibrium either to cheat on the most-likely-voted ciphertext, or not to cheat, and for the voter to audit as instructed.

4 Conclusion

We have shown that none of the natural sequential strategies for voter-initiated auditing form Nash equilibria in a game that captures remote (Internet) voting.

This is significant because voter-initiated auditing is probably the most promising of strategies for verifiable Internet voting. The only alternatives are codes [7,14], which require a secrecy assumption and hence a threshold trust assumption on authorities, and which anyway don't work for complex ballots. Preprinted auditable ballots [8,13] only work in polling places. We have shown that voter-initiated auditing must be conducted with multiple parallel ballots, rather than sequential challenges.

The next step is to repeat the analysis for a game that captures the payoffs for polling-place voter-initiated auditing. This setting has a significantly higher cost to the device for cheating, so probably has very different equilibria.

Acknowledgments. Thanks to Wojtek Jamroga, Ron Rivest, and Josh Benaloh for interesting conversations about this work.

References

1. Adida, B.: Helios: web-based open-audit voting. USENIX Secur. Symp. **17**, 335–348 (2008)
2. Adida, B., Rivest, R.L.: Scratch & vote: self-contained paper-based cryptographic voting. In: Proceedings of the 5th ACM Workshop on Privacy in Electronic Society, pp. 29–40. ACM (2006)
3. Bell, S., Benaloh, J., Byrne, M.D., Debeauvoir, D., Eakin, B., Kortum, P., McBurnett, N., Pereira, O., Stark, P.B., Wallach, D.S., Fisher, G., Montoya, J., Parker, M., Winn, M.: Star-vote: a secure, transparent, auditable, and reliable voting system. In: 2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 2013). USENIX Association, Washington, D.C. https://www.usenix.org/conference/evtwote13/workshop-program/presentation/bell
4. Ben-Nun, J., Fahri, N., Llewellyn, M., Riva, B., Rosen, A., Ta-Shma, A., Wikström, D.: A new implementation of a dual (paper and cryptographic) voting system. In: Electronic Voting, pp. 315–329 (2012)
5. Benaloh, J.: Simple verifiable elections. EVT **6**, 5 (2006)
6. Benaloh, J.: Ballot casting assurance via voter-initiated poll station auditing. EVT **7**, 14 (2007)
7. Chaum, D.: Surevote: technical overview. In: Proceedings of the workshop on trustworthy elections (WOTE 2001) (2001)
8. Chaum, D., Carback, R., Clark, J., Essex, A., Popoveniuc, S., Rivest, R.L., Ryan, P.Y., Shen, E., Sherman, A.T.: Scantegrity II: end-to-end verifiability for optical scan election systems using invisible ink confirmation codes. EVT **8**, 1–13 (2008)
9. Eckersley, P.: How unique is your web browser? In: Atallah, M.J., Hopper, N.J. (eds.) PETS 2010. LNCS, vol. 6205, pp. 1–18. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14527-8_1
10. Karayumak, F., Olembo, M.M., Kauer, M., Volkamer, M.: Usability analysis of helios-an open source verifiable remote electronic voting system. In: EVT/WOTE 2011 (2011)
11. Kiayias, A., Zacharias, T., Zhang, B.: Ceremonies for end-to-end verifiable elections (2015)
12. Lindeman, M., Stark, P.B.: A gentle introduction to risk-limiting audits. IEEE Secur. Priv. **5**, 42–49 (2012)
13. Ryan, P.Y., Bismark, D., Heather, J., Schneider, S., Xia, Z.: Prêt à voter: a voter-verifiable voting system. IEEE Trans. Inf. Forensics Secur. **4**(4), 662–673 (2009)

14. Ryan, P.Y.A., Teague, V.: Pretty good democracy. In: Christianson, B., Malcolm, J.A., Matyáš, V., Roe, M. (eds.) Security Protocols 2009. LNCS, vol. 7028, pp. 111–130. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36213-2_15
15. Sandler, D., Derr, K., Wallach, D.S.: Votebox: a tamper-evident, verifiable electronic voting system. In: USENIX Security Symposium, vol. 4, p. 87 (2008)