Secure and Observable Auditing of Electronic Voting Systems using Stock Indices

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

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

Carlisle Adams
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, K1N 6N5
Email: cadams@site.uottawa.ca

***Abstract*— In recent years, there has been a tremendous increase in the interest and discussion around the use of electronic voting in our society. Verifiable electronic voting systems rely on the ability to conduct random ballot audits to establish election integrity. However in order for the election results to be legitimate, not only must these audits be conducted in an unbiased manner, but the participants in the election (candidates and voters alike) need to be reasonably convinced thereof.**

In this paper we propose a procedure for fairly and transparently selecting ballots during an election audit using stock indices.

I. Introduction

Consider the old sleight-of-hand trick of the disappearing coin. The skilled magician can always make you believe it went into one palm, when in fact it is in the other. During the ballot auditing process however, there should never be a circumstance where certain ballots can escape the possibility of being audited. In the case of our magic trick, a fair audit would see the audience point to the hand holding the coin exactly half the time, regardless of the level of indirection on behalf of the magician. In order to fairly select which hand to audit, the audience might agree to flip another coin and ask to see the magician's left hand if it is heads, and the right hand if it is tails. What if the magician presented you a second coin, and then bet everyone in the audience $10 that they could not correctly select which hand the first coin was in? The audience might reasonably suspect this second coin was biased, and that they would lose money. If instead you were to pull the coin from your own pocket, you might expect the coin to be fair. However the other audience members might not be inclined to trust you. So what is a reasonable way for everyone in the audience to come to agree on a fair coin to use?

II. Random Audits

A. Audit Requirements in Electronic Voting Systems

It has been suggested by Cordero, Wagner and Dill [3], that any procedure for making a random selection for auditing in a voting system should satisfy four criteria:

1) **Simplicity**. It must be easy to understand why the procedure works and how to perform it.
2) **Verifiability**. There must be a method to verify the integrity of the procedure.
3) **Robustness**. It should be impossible for anyone, including the election officials, to predetermine which ballots will be audited.
4) **Efficiency**. The procedure should not require much time or many resources.

Cordero, et al., point out that using a black-box source of randomness fails the verifiability criterion because there is no clear way for an observer to determine that the outcome is completely unpredictable to everyone. Were it possible to rig the black-box, a malicious party may be able to affect the results of an election by escaping detection. The authors then propose and compare a number of sources of true randomness including lotteries, coin-flipping, shuffling cards, and rolling dice. Finally, they propose a system based on rolling a 10-sided die.

However pursuant to our example of the audience and the coin, the use of dice also suffers from issues of verifiability and efficiency. Verifying that a die is not loaded (biased to roll one number with higher probability than another) would require testing it over hundreds of rolls and calculating the probability of bias to a satisfactory margin of error. To be truly verifiable, the observers must personally monitor this testing procedure as well as the actual rolls used in making the random selection. The observability of dice is also physically bounded to being present in the room.

We propose using the random fluctuations of the stock market as a better source of randomness. Stock market prices are highly verifiable—closing prices of indices are published in nearly every major newspaper and they can be verified by anyone, anywhere. They are also available online in convenient digital formats, and so the random selection could be verified by downloading the stock data and running an open-source software tool to duplicate the results (or writing your own for the strongest verifiability). The number of stock indices available for sampling gives the potential for a large pool of randomness without the physical requirement of rolling a die. The remainder of this paper will be devoted to developing a specific procedure for use in a Punchscan election, and outlining why it satisfies the criteria above.

B. Auditing in a Punchscan Election

Punchscan [2], [7] is a open-source voting system originally proposed by David Chaum. The results of a Punchscan election are verifiable by voters through their participation in the audit process, a process that establishes a high statistical degree of confidence in the integrity of the outcome. When the election authority prints the ballots prior to an election, they *commit* to the unique information contained in each ballot by using a cryptographic one-way function. Though this commitment is made public, the actual information contained on the ballot remains sealed. Because the function is one-way, it is computationally infeasible to determine the information on the sealed ballot given only its publically posted commitment.

During the pre-election audit, *half* the ballots are selected to be examined. These ballots are unsealed and checked to ensure they are properly formed and that they match their commitments. If the election authority does not know *a priori* which ballots will be checked, it faces a high probability of getting caught if it publishes false commitments. Thus the ballots to be selected for the audit should not be guessable with any advantage at the time the commitments are published. Furthermore, it is required in a Punchscan election that any interested citizen or group be able to personally perform the audit to independently verify its findings. Therefore we must design a system to include a fifth criterion:

5) **Availability**. The audit must be able to be independently performed by any interested party on the platform of their choosing.

This represents an exciting new paradigm in electronic voting systems research—requiring that every step of the audit process to be transparent and repeatable. In the following section we will propose a random ballot selection method for a Punchscan pre-election audit.

III. Stock Indices as Sources of Randomness

A. Stock Index Prediction Assumption

This paper is premised on an assumption regarding the predictability of the future closing price of a given stock index. Broadly speaking, we define the Stock Index Prediction (SIP) assumption as follows: *an observer cannot precisely predict what the closing price of a stock index will be sufficiently far in the future*. For example, if a stock's closing price today is an odd number, one would not expect to be able to use this information to any advantage in predicting if it would also close odd on the following day. Thus we assume that the observer's advantage in predicting the parity of the future closing price is equivalent to a random guess.

B. The Audit Process

Putting this together, we define the steps of ballot selection for the purposes of a Punchscan pre-election audit as being:

1) The election authority and candidates agree on a portfolio of stock indices to sample, and set the audit date.
2) The election authority generates ballots and publishes their commitments.
3) When the stock market data becomes available on audit day, it is used to select half of the ballots to be audited.
4) These ballots are unsealed and made public by the election authority, and the pre-election audit is performed.

In the next sections we will discuss the process for creating the ballot selection (step 3), and offer some guidelines for establishing the portfolio and audit date (step 1).

C. The role of Pseudorandomness

The scheme presented in [3] uniquely chooses one ballot from the set of cast ballots by rolling a 10-sided die x times to generate an x-digit number of a ballot. This process is then repeated until enough ballots have been selected. A conventional audit may require the selection of only a few percent of the ballots cast. The voting age population of the United States was 215,694,000 in 2004 [1], and therefore would require 9 dice rolls to select one ballot. If we consider a scenario whereby 2% of the these ballots are selected to be audited, this would require over 38-million dice rolls. This is not practical and contravenes the efficiency criterion. A Punchscan audit has even more stringent requirements. As described in the previous section, the pre-election audit calls for 50% of the ballots to be unsealed, requiring the election authority to generate twice as many ballots as are intended to be cast. This represents a 50-fold increase in the number of ballots to be selected for audit under Punchscan.

Clearly even one roll of the die per ballot is not efficient, nor is sampling one stock index per ballot. What we propose as a reasonable alternative is to generate a truly random seed from a much smaller entropy pool and expand the sequence using a pseudorandom number generator (PRNG). Because the seed is not a secret, there is no need to use a cryptographically secure pseudorandom number generator (CSPRNG)—a class of PRNGs that have certain secure properties not required in this case, such as the infeasability of deriving the seed only from its output sequence. However one property that the PRNG must possess is the so-called strict avalanche criterion (SAC) [8], which states that if a single input bit is complemented, each of the output bits should change with a 50% probability. Of further importance is the requirement that the distribution of the output be statistically uniform. A PRNG possessing these two properties is suitable for our purpose (e.g. AES-256 in counter-mode).

D. System Definition

The approach in [3] for auditing is to randomly draw a ballot from the set of all ballots until the desired number of ballots have been drawn. If there are B ballots, this selection process in a Punchscan pre-election audit requires at least $B \cdot \lceil \log_2(B) \rceil$ bits of entropy. We propose an alternative in which a pseudorandom ballot selection sequence of length B is generated. If the i^{th} bit of this sequence is zero, the i^{th} ballot is audited; otherwise it is left sealed and can be used in the election. The advantage is a lower complexity, requiring one bit (as opposed to $\lceil \log_2(B) \rceil$ bits) to select a ballot. A

drawback is that it will not select an exact half of the ballots, but rather a probabilistic half.

The PRNG is seeded with a number derived from the stock market data. The precise amount of entropy in the price fluctuations of a single stock is unknown and estimating it is controversial. We will remain conservative, and sample 1 bit of entropy from each stock index. The sampling will be done across closing price and the closing volume of the index.

Algorithm 1: Select Ballots to Audit

```
foreach stock s ∈ P do
    p ← ClosingPrice(s)
    v ← ClosingVolume(s)
    k ← k||p||v
seed ← rightmost_N (Hash(k))
foreach ballot b ∈ B do
    Step PRNG(seed) to generate bit x
    if x=0 then
        Audit b
    else if x=1 then
        Do not audit b
```

The complete process is shown in Algorithm 1. For each stock, s, in the portfolio, $\mathbf{P}$, the closing price and volume is concatenated into a long integer k. This integer is then hashed and compressed to $N = |\mathbf{P}|$ bits and used to seed the PRNG. The purpose of the hash is to ensure that two values of k which differ by only one digit produce completely different seeds. The stock values are not secret and so like the PRNG, the hash does not have to be cryptographically secure. It merely needs to have good statistical properties and adhere to the strict avalanche criteria (e.g. SHA-256). In the next section, we will discuss how many stocks should be included in the portfolio relative to the number of ballots in the election.

IV. Establishing a Portfolio Size

Although it only takes a single stock for the output to be unpredictable, an attacker can work with less than complete unpredictability to compromise the random audit. Consider the situation where a single stock is used. The one bit of entropy produces one of two psuedorandom selection sequences: say, 100110... or 010010... From the SIP assumption, we can conclude the attacker will not be able to predict with non-negligible advantage which of the two streams will be produced. Observe however that the fifth bit in both selection sequences is a 1. This means the fifth ballot will not be audited regardless of which stream is generated. In this situation the ballot could be safely corrupted even though the attacker cannot predict the stock's future closing price (aka the seed).

We define a *corruptible ballot* (CB) as a ballot that will not be selected for auditing for all possible values of the seed. Finding corruptible ballots will be referred to as an *intersection attack*. This attack requires the adversary to generate the output stream for every possible initial seed—an N-bit seed produces $S = 2^N$ streams—and calculate the intersection (bitwise AND) of all the streams. For every 1 in this intersection, the corresponding ballot is corruptible. Assuming the output of the pseudorandom number generator is uniformly random and statistically independent for each different N-bit seed, the probability that a given ballot b is a corruptible ballot is:

$$\Pr[b \text{ is CB}] = \frac{1}{2^S} = \frac{1}{2^{2^N}} \tag{1}$$

We can take two approaches to defeating the intersection attack. The first is to ensure it is computationally infeasible to run every possible seed by making the seed space sufficiently large (e.g. to the order of 128 bits). A second option is to make S merely large enough that the probability of encountering *even one* corruptible ballot in a B-ballot election is less than a half (making the expectation essentially zero). This requires $N \geq \lceil \log_2 \lceil \log_2(2B) \rceil \rceil$. Recall the voting age population of the United States was 215,694,000 in 2004 and that B is twice this (because the half that are audited are thrown out). This would require a portfolio size of $N \geq 5$ to defeat the intersection attack. Using this minimal N, the expected number of corruptible ballots in an election of this size is 0.1.

However we should also consider ballots that have a low but non-zero risk of being audited. For example, a ballot may be selected for auditing in only 1 of the S streams for each possible seed. For a large S, the probability that the one seed that would select this ballot would be generated is quite small. We define *risk level*, R, to be the probability that a given ballot will be audited across all the possible seeds. To be clear, the risk is from the perspective of the attacker getting caught. A corruptible ballot has a risk level of 0, meaning the attacker has no risk of being caught. If a ballot will only be audited in, say, 1 out of 16 selection sequences, it has a risk of $R = 0.0625$. We define a *semi-corruptible ballot* (SCB) as a ballot that has a risk level R, for any $R < 0.5$. The probability that a ballot is a semi-corruptible ballot with risk level R (or less) is,

$$\Pr[b \text{ is SCB}_R] = \sum_{i=0}^{\lfloor 2^N \cdot R \rfloor} \binom{2^N}{i} \frac{1}{2^{2^N}}. \tag{2}$$

Table 1 shows the minimal portfolio size required for different sized elections to ensure the expected number of semi-corruptible ballots with a risk level of 45% will be zero. This is to say, the expectation is that *every* ballot would be audited with greater than 45% probability. For the purposes of the intersection attack, we regard this as suffiently high risk that an attacker could not change the outcome of an election without detection. Thus, for an American-sized election, the minimal portfolio size to avoid the intersection attack is 12 stock indices. Increasing the portfoilio size beyond this lower-bound serves to bring the probability of a ballot being selected for audit increasingly closer to the ideal 50%. In practice, the audit participants may be more comfortable using a much larger portfolio. An upper-bound on the portfolio size is only limited by the efficiency criterion.

Portfolio Size	Number of Ballots
8	1 – 8
9	9 – 41
10	42 – 783
11	784 – 172,382
12	172,383 – 6,262,358,931

TABLE I

V. Index Divergence between Commitment and Audit Events

The election authority publicly commits to using the closing prices of a set of stocks at some future time. For this number to be unpredictable, we need to allow the market adequate time to probabilistically diverge from its current price by at least a cent. There is limited consensus on the underlying statistical mechanics of the stock market, and so it is impossible to exactly determine the minimum amount of time required. However we will approach this problem by asserting that one trading-day is more than adequate and then verifying this assertion empirically.

The volatility of the market is typically calculated in terms of the rate of return,

$$r = \ln \frac{P_{t+1}}{P_t}. \tag{3}$$

P_t is the price at time t, while P_{t+1} is the price at the next time period. The return is logarithmic because stock investments represent continuously compounded interest. Volatility is typically calculated using the standard deviation of a stock's returns, σ_r, over a period of many years and then annualized,

$$\sigma = \sigma_r * T^{\frac{1}{\alpha}}. \tag{4}$$

T, the time division, is 12 or 252 if monthly or daily rates of return are used respectively in calculating σ_r. α is the divergence factor and a value of 2 is typically used, which assumes that rate of return follows a random walk. This assumption is widely held but controversial [5]. Some financial analysts believe it is less volatile [4], while other mathematicians such as Benoit Mandelbrot posit that the market is scale-invariant and thus advocate smaller alpha values [6]. The value of σ for a given stock is rarely published; instead volatility is given by β,

$$\beta = \frac{\mathrm{cov}(r_s, r_m)}{\mathrm{var}(r_m)} = \frac{\mathrm{cov}(r_s, r_m)}{\sigma_m^2}. \tag{5}$$

β measures the volatility of returns of a given stock (r_s) against that of the market (r_m). A $\beta > 1$ means the stock is more volatile than the market, while a $\beta < 1$ means it is less volatile. By convention, β uses the daily returns for the past 5 years, and the "market" is the performance of the S&P 500.

Although it is readily available, β it is not necessarily the best metric for our purposes. We are interested in simple stock movements, not returns (e.g., a stock moving up or down 10% is equivalent to us, whereas in terms of continuously compounded returns, the movement down is worst). We can define the movement of a stock, m, to be the absolute value of an arithmetic return on investment, $m = |(P_{t+1} - P_t)/P_t|$, and calculate it daily for the S&P 500 over the past 5 years (ending December 29, 2006). Using the actual market data for this period, we determined that the index had a mean value of \$1120.67, and a low/high of \$776.76 - 1427.08. The expected value of the daily movement was .73% or \$7.69 with a standard deviation of .71% or \$6.71.

This movement calculation suggests we should only choose stocks valued over \$10 to ensure a reasonable certainty that the stock will move by at least a cent, if it is as volatile as the S&P 500. To ensure that it is, this calculation could be performed on each individual candidate stock for five years of data. However this would break the simplicity criterion. Given that the β value is published and readily available, we will use it instead and recommend that stocks with a $\beta > 1$ are chosen. By ensuring all the stocks in the portfolio have these two properties, (i.e., a share price over \$10 and a $\beta > 1$) we can be assured that one day is adequate time for the future closing price to be unpredictable within the constraints of the SIP. Finally, to mitigate the possibility of certain members within any one exchange colluding to unduly affect closing prices and volumes, it would be advisable to assemble the portfolio across a diversity of exchanges.

VI. Conclusion

This paper presents a novel approach for election auditing using stock indices, which was designed to be simple, verifiable, robust, and efficient. Given the public and widely available nature of stock market data, this represents an audit process that can be implemented and performed by any interested party. Engaging voters more centrally in the election process (through auditing and observation), continues to be the focus of current voting systems research (such as in the Punchscan project). The ultimate goal of this effort is a process that preserves and enriches the quality of democracy in this and other countries around the world.

References

[1] *Estimates and Projections of the Voting-Age Population*. U.S. Census Bureau, Population Division, Education & Social Stratification Branch. http://www.census.gov/population/www/socdemo/voting.html

[2] David Chaum, Rick Carback, Jeremy Clark, Aleks Essex, Kevin Fisher, Ben Hosp, Stefan Popoveniuc, Jeremy Robin. "Punchscan and VoComp." Presented at the Rump Session of Crypto 06, 2006.

[3] Arel Cordero, David Wagner, and David Dill. The Role of Dice in Election Audits. *IAVoSS Workshop On Trustworthy Elections (WOTE 2006)*, 2006.

[4] Andrew W. Lo and A. Craig MacKinlay. *A Non-Random Walk down Wall Street*. Princeton University Press, 1999.

[5] Burton Malkiel. *A Random Walk Down Wall Street*. W. W. Norton & Company, 1973.

[6] Benoit B. Mandelbrot, Richard L. Hudson. *The (mis) Behaviour of Markets: A Fractal View of Risk, Ruin and Reward*. Basic Books, New York, 2004.

[7] Stefan Popoveniuc, Ben Hosp. An Introduction to Punchscan. *IAVoSS Workshop On Trustworthy Elections (WOTE 2006)*, 2006.

[8] A. F. Webster, Stafford E. Tavares: On the Design of S-Boxes. CRYPTO 1985: 523-534