

Computational Complexity and Information Asymmetry in Election Audits with Low-Entropy Randomness

Nadia Heninger

Princeton University

August 10, 2010

“Computational complexity and information asymmetry in financial products”

[Arora, Barak, Brunnermeier, Ge 10]

“On the security of election audits with low-entropy randomness”

[Rescorla 09]

Introduction: Auditing an election.

“Post-election vote tabulation audit”

1. Select a subset of $\left\{ \begin{array}{l} \text{ballots} \\ \text{voting machines} \\ \text{precincts} \\ \dots \end{array} \right.$ to audit.
2. Compare fully counted sample to preliminary election results.

Audited subset should be

- ▶ *statistically representative*
- ▶ *difficult to predict.*

Audit process should be *observable*.

Introduction: Auditing: A statistically ideal solution.

*Select audited subset uniformly at random,
after the election.*

- ▶ Statistics tells us size of set to ensure representative sample.
- ▶ Randomness ensures sample is difficult to predict.

Introduction: How to generate random numbers.

- ▶ Use a physical source.



flickr:darwinbell



flickr:diverkeith



flickr:jeremybrooks

- ▶ Use a physical source with processing.

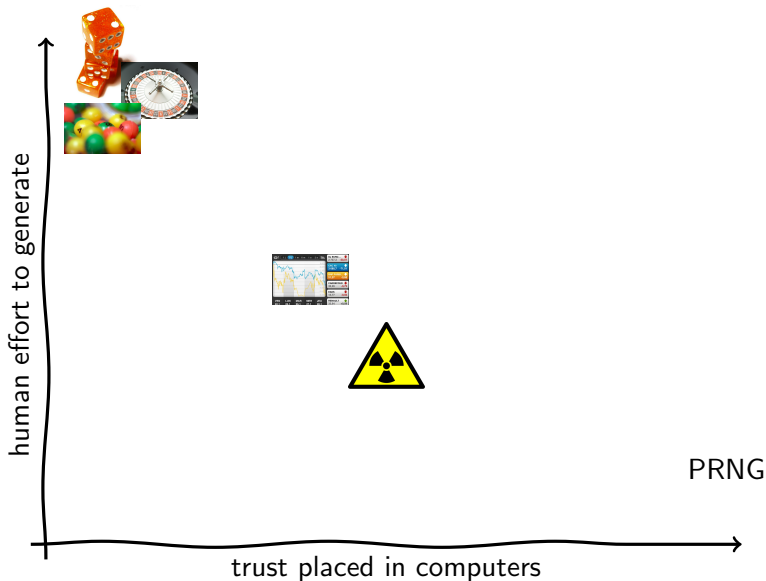


flickr:yahoo_presse



- ▶ Use a pseudorandom number generator with a random seed.

Introduction: Human vs. computer generated randomness



Introduction: Random tables: A low-tech compromise.

Proposal: [Cordero, Dill, Wagner 06] Combine

- ▶ a low-tech method of generating randomness (dice rolls) with
- ▶ a low-tech method of expanding randomness (random table).

Pro: Anyone can look at published table for problems.

Con: Is the audit really still reliable?

4

TABLE OF RANDOM DIGITS

00150	94015	46874	32444	48277	59820	96163	64654
00151	74108	88222	88570	74015	25704	91035	01755
00152	62880	87873	95160	59221	22304	90314	72877
00153	11748	12102	80580	41867	17710	59621	06554
00154	17944	05600	60478	03343	25852	58905	57216
00155	66067	42792	95043	52680	46780	56487	09971
00156	54244	91030	45547	70818	59849	96169	61459
00157	30945	57589	31732	57260	47670	07654	46376
00158	69170	37403	86995	90307	94304	71803	26825
00159	08345	88975	35841	85771	08105	59987	87112
00160	27767	43584	85301	88977	29490	69714	73035
00161	13025	14338	54066	15243	47724	66733	47431
00162	80217	36292	98525	24335	24432	24896	43277
00163	10875	62004	90391	61105	57411	06368	53856
00164	54127	57326	26629	19087	24472	88779	30540
00165	60311	42824	37301	42678	45990	43242	17374
00166	49739	71484	92003	98086	76668	73209	59202
00167	78626	51594	16453	94614	39014	97066	83012
00168	66692	13986	99837	00582	81232	44987	09504
00169	44071	28091	07362	97703	76447	42537	98524
00170	41468	85149	49554	17994	14924	39650	95294



```
int getRandomNumber()  
{  
    return 4; // chosen by fair dice roll.  
              // guaranteed to be random.  
}
```

<http://xkcd.com/221/>

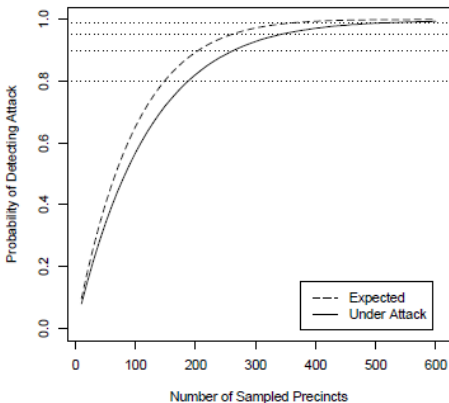
Introduction: Randomness Tables: Concerns

1. The audit is no longer random.
2. The audit is no longer representative.
3. Could this scheme enable new attacks on the audit system?

[Rescorla 09]: Attacks on low-entropy randomness.

An adversary can use a published table to lower chances of detection.

(Tactic: entries normally distributed; cheat in least common precincts.)



200,000 entries, 1000 precincts, 10 attacked

Results: Analyzing random number tables.

1. A truly random table can be used in a sound audit.

Tradeoff: For same statistical confidence, must audit more.

2. It is difficult for an attacker to use a table to optimize an attack on an election beyond known values.
3. It is possible to create a malicious table that is indistinguishable from random.

Preliminaries: Auditing procedure.

1. Roll some dice.
2. Dice rolls select a “page” in book.
3. Audit the elements listed on that page.

Simplifying assumptions: Any irregularity is detected by the audit.
Dice roll selects a page uniformly at random.

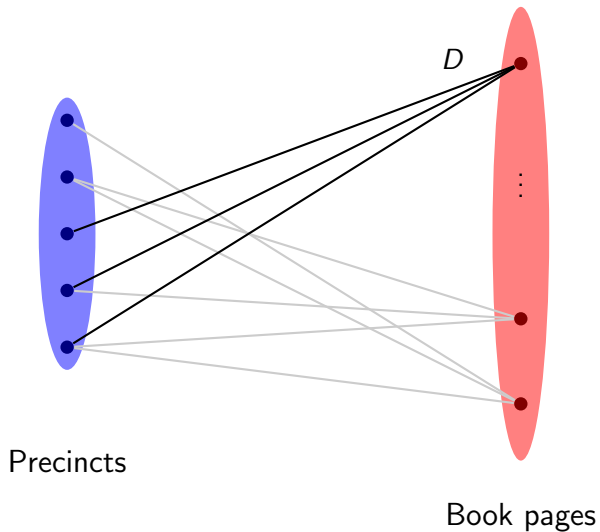
Auditor

wishes to maximize the chance of detection.

Adversary

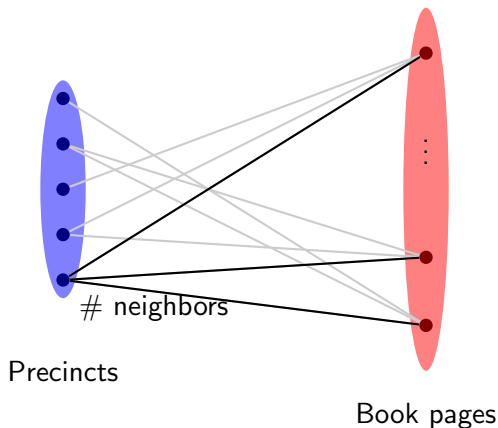
wishes to minimize the chances of detection.

The model: Auditing procedure viewed as a graph.



The model: Analyzing an audit using the graph.

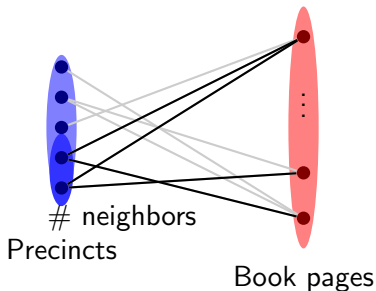
$$\Pr[\text{precinct } p \text{ audited}] = \frac{\# \text{neighbors}(p)}{\# \text{ pages in book}}$$



The model: Table determines probability of detection.

In order to detect a problem, must appear in audited set:

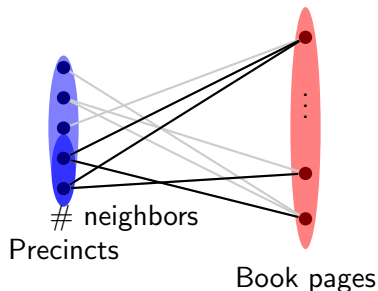
$$\Pr[\text{abnormality appears in audit set}] = \frac{\# \text{ neighbors of abnormal set}}{\# \text{ pages in book}}$$



The model: Table determines probability of detection.

In order to detect a problem, must appear in audited set:

$$\Pr[\text{abnormality appears in audit set}] \geq \min_{\substack{\{\text{sets}\} \\ a < |s| < b}} \frac{\# \text{ neighbors of set}}{\# \text{ pages in book}}$$



Related to *expansion* of graph.

The model: Facts about expanders

- ▶ Random graphs have good expansion properties.

Translation: A randomly generated table will give a good audit with high probability.

Caveat: We can calculate the *probability* that a random graph is good, but cannot certify a fixed graph. (More on this later.)

- ▶ The expansion is smaller than the average degree.

Translation: The confidence estimate will be smaller than the audit size suggests.

Thus we must audit more to maintain the same confidence level.

Example: Auditing an election with a table

Have 5000 precincts
wish to guarantee $< 5\%$ fraud with 80% confidence.

Truly random audit:

Need to audit 32 precincts and generate

$$\lg \binom{5000}{32} > 275 \text{ bits of randomness on the fly.}$$

Using a random table of size 10,000,000.

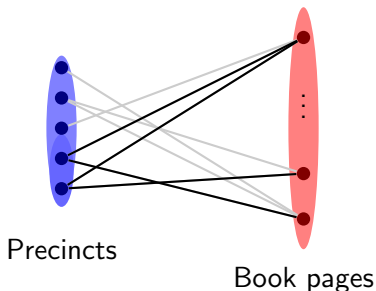
Need to audit 50 precincts, but only generate

$$\lg 200000 < 18 \text{ bits of randomness on the fly.}$$

Part 2: Using a table to optimize an attack.

Can an attacker use table to find optimal locations for fraud?

Problem: Given a bipartite graph, find set with smallest expansion.



Recently related to solving the unique games conjecture.

[Raghavendra Steurer 10]

Optimizing an attack: The counterpoint.

Attacker's goal: Find set with smallest expansion.

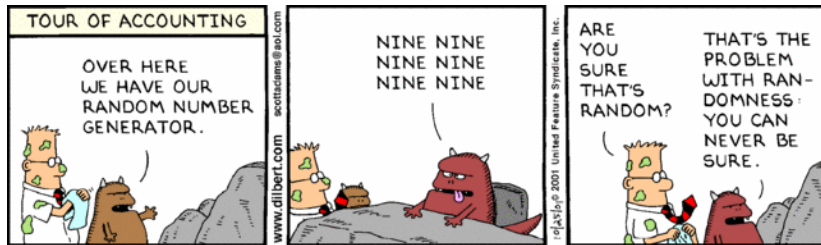
Auditor's goal: Ensure no set has small expansion.

Both seem to be hard.

New attack idea: Create a malicious table with a set that has small expansion.

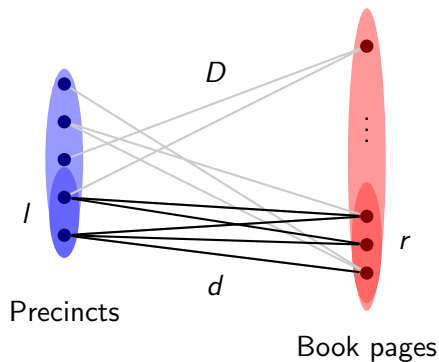
No auditor can distinguish such a malicious table from a truly random one.

Interlude: The problem with randomness.



<http://dilbert.com/strips/comic/2001-10-25>

Creating a malicious table: Planted dense subgraph.



Hardness of detecting planted dense subgraph used in

- ▶ Cryptosystem of [Appelbaum Barak Wigderson 10].
- ▶ Hardness of detecting tampering in financial derivatives [Arora Barak Brunnermeier Ge 10].

Example: The effects of a malicious table.

Ballot-based audit for 100 million voters,
“book” with 100 million entries,
2% fraud.
Audit size = 50.

In a truly random audit:

$$\Pr[\text{detect fraud}] \approx 63.2\%.$$

With an undetectably tampered book:

$$\Pr[\text{detect fraud}] \approx 2.2\%.$$

Conclusions

Lesson 1:

Randomness tables can expand expensive sources of randomness.

Can perform an effective audit in exchange for lower confidence or more work.

Lesson 2:

No computational method to verify that table has desired properties.

Such tables should be generated openly and verified before use.

Closing: The paradox of “observability”

Which is more transparent?

Let p, q be unequal primes congruent to 1 mod 4. Let i be an integers satisfying $i^2 \equiv -1 \pmod{q}$.

There are $8(p+1)$ solutions

$\alpha = (a_0, a_1, a_2, a_3)$ to

$a_0^2 + a_1^2 + a_2^2 + a_3^2 = p$. To each solution α associate the matrix $\tilde{\alpha}$ in $\text{PGL}(2, \mathbb{Z}/q\mathbb{Z})$.

$$\tilde{\alpha} = \begin{pmatrix} a_0 + ia_1 & a_2 + ia_3 \\ -a_2 + ia_3 & a_0 - ia_1 \end{pmatrix}$$

Form the Cayley graph of $\text{PGL}(2, \mathbb{Z}/p\mathbb{Z})$ relative to the above $p+1$ elements.

00150	94015	46874	32444	48277	59820	96163	64654
00151	74108	88222	88570	74015	25704	91035	01755
00152	62880	87873	95160	59221	22304	90314	72877
00153	11748	12102	80580	41867	17710	59621	06554
00154	17944	05600	60478	03343	25852	58905	57216
00155	66067	42792	95043	52680	46780	56487	09971
00156	54244	91030	45547	70818	59849	96169	61459
00157	30945	57589	31732	57260	47670	07654	46376
00158	69170	37403	86995	90307	94304	71803	26825
00159	08345	88975	35841	85771	08105	59987	87112
00160	27767	43584	85301	88977	29490	69714	73035
00161	13025	14338	54066	15243	47724	66733	47431
00162	80217	36292	98525	24335	24432	24896	43277
00163	10875	62004	90391	61105	57411	06368	53856
00164	54127	57326	26629	19087	24472	88779	30540
00165	60311	42824	37301	42678	45990	43242	17374
00166	49739	71484	92003	98086	76668	73209	59202
00167	78626	51594	16453	94614	39014	97066	83012
00168	66692	13986	99837	00582	81232	44987	09504
00169	44071	28091	07362	97703	76447	42537	98524
00170	41468	85149	49554	17994	14924	39650	95294