eVoting after Nedap and Digital Pen

Why cryptography does not fix the transparency issues

Ulrich Wiesner

25C3, Berlin, 29th December 2008

Agenda

- Why is eVoting an issue?
  - Physical copies, paper trail?
- Cryptographic Solutions?
  - Three Ballot
  - Punchscan
  - Bingo Voting
- Conclusions

Motivation

- Strong community believing
  “The eVoting issues are fixable – it just needs to be done properly”
- Media hype (confined to Germany) after German IT Security Award 2008 for BingoVoting.
- I don’t think it is that easy

Thank you!

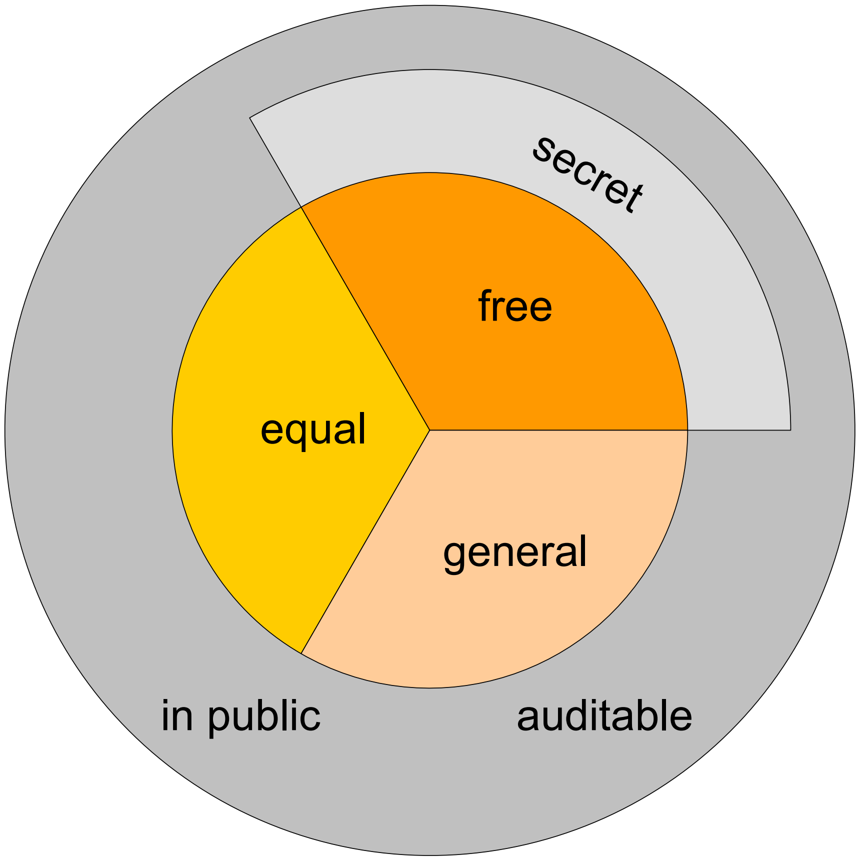
Relevance

- Voting Computers in polling stations
  - Netherlands – almost 100% coverage, discontinued
  - Ireland – 100 % coverage, never used
  - Belgium – 40% coverage, discontinued
  - France – 5% coverage, growing
  - Germany – 5% coverage, Federal Constitutional Court to decide on future use during next sweeks
- Voting via Internet
  - Estonia – since 2006,
    now even looking into voting via Mobile Phone
  - Switzerland – in some cantons
- Discussions and trials
  - UK, Austria, Norway, Russia

Why is eVoting an issue?

Election Principles

- **Verifiability**, **transparency** and **secrecy** **(procedure)** ensure that elections are **free**, **fair** and **general** **(values)**



Procedural Principles

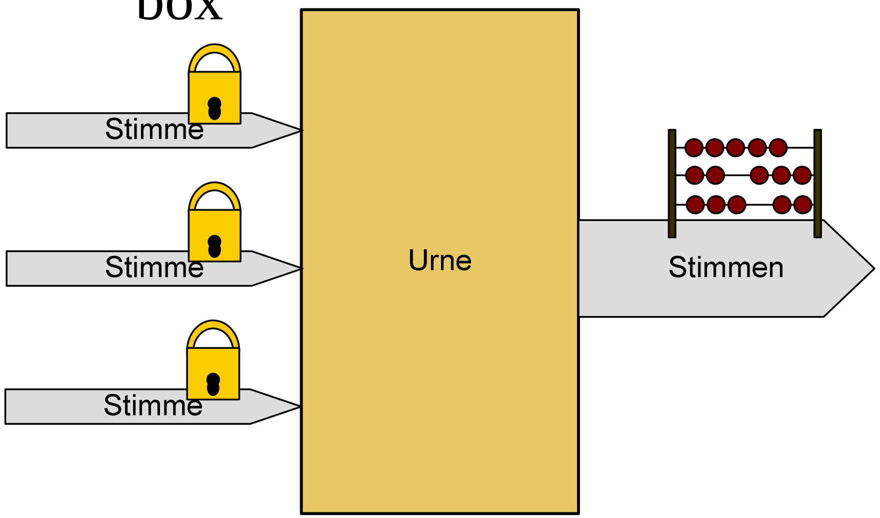
- Secrecy
  - protects free elections
  - Choice has no personal consequences
  - Vote can not be sold
- Auditability
  - Measure of Quality Assurance: identify and correct errors
  - Typically conducted by authorities (e.g. re-counts)
  - Auditability can never replace Transparency
- Transparency
  - Ensures that election is conducted according to regulations and principles – and that everybody can verify this
  - Creates trust: contributes to Legitimacy of the elected body
  - Prevents denunciation of election result
  - Transparency can not be delegated to authorities

Implementation of Transparency

- Transparency of elections is mandatory for all OSCE member states
  - (Copenhagen declaration 1990)
- Different approaches in different countries
  - Germany
    - Anybody can observe election and counting
    - Access to polling stations only restricted by means of safety and public order
  - Austria
    - Participating parties can nominate two election witnesses per polling station
  - UK
    - Participating parties can nominate election witnesses
    - Organisations and individuals can register for observation

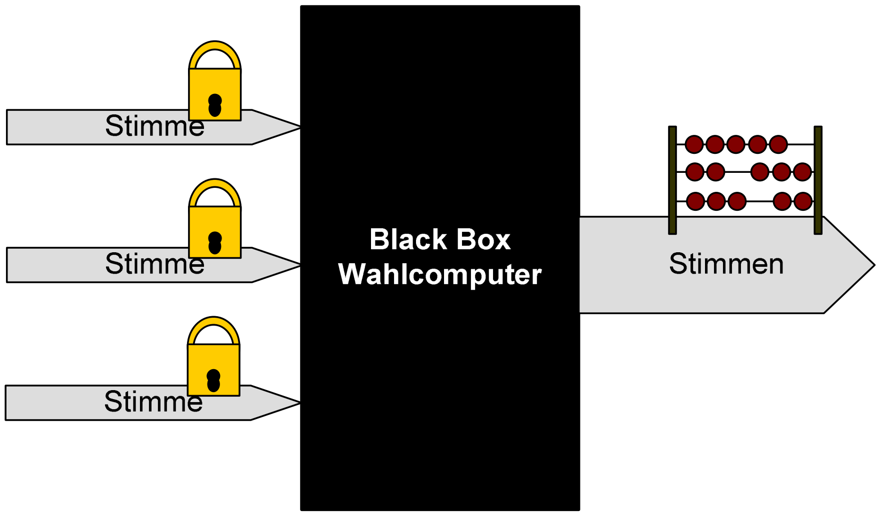
e-Voting: what is the issue?

- Paper based election: white box



- Ballot box is passive device
- No processing: Output is input
- Manipulations need to be conducted under the public's eyes

- eVoting: black box



- Voting computer is active device
- Output might be input
- Processing not observable

Why eVoting?

Inappropriate reasons

- Because it’s cheaper
- Because we’ve already spent the money on the equipment
- Because it saves 1 hr of counting

Why eVoting?

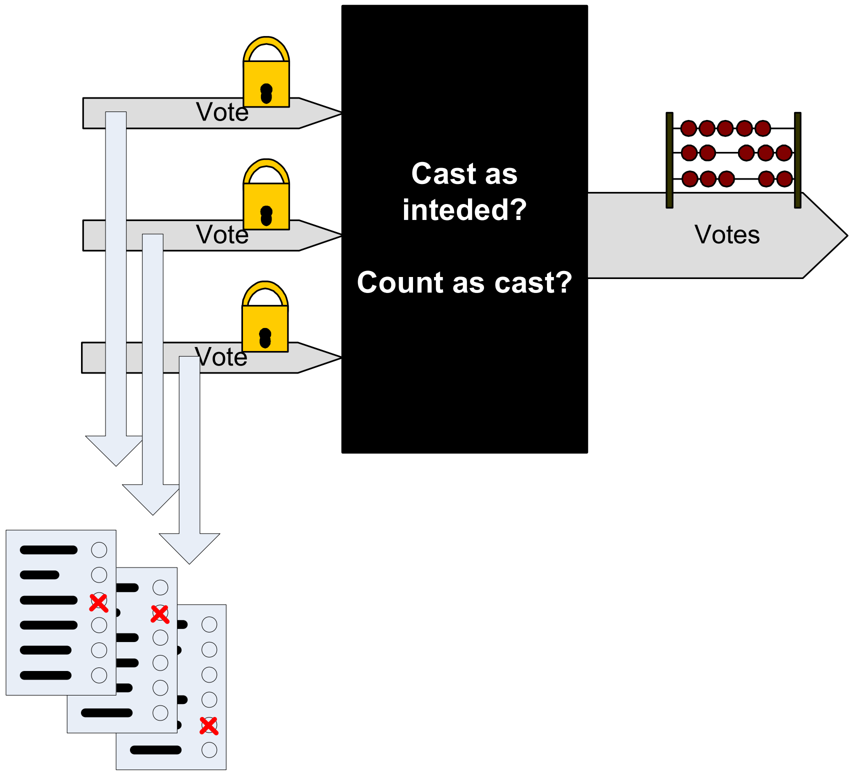
Better reasons

- Multi-vote elections (cumulative voting)
  - E.g. Hesse, Bavaria, Baden-Württemberg, Rhineland-Palatinate
    - Voter has one vote per city council member
    - 50+ votes for bigger cities.
  - E.g. Hamburg, Brandenburg
    - Voter has 3-5 votes
    - Can be distributed on candidates from various parties
    - Can be accumulated on same candidate
- Preferential systems
    - Single Transferrable Vote
    - If Candidate A is not successful, my second priority is B
- Manual counting can be prohibitive

Keep Physical Copies?

Keep Physical Copies?

- Paper Trail, Digital Pen
- Allows validation of result independent of voting device
- However:
  - What triggers re-count?
  - Which polling stations get audited? Who decides?
  - When and where is the re-count conducted?
  - Who has control over the physical copies until re-count?



Keep Physical Copies?

- Paper trail can fix the auditability issue, but will typically not fix transparency
- Transparency would require
  - Recount immediately after election
  - In the polling station
  - Kills business case: why not using paper ballots in the first place

Keep Physical Copies?

- And if recount is restricted to a sample?
  - City of Hamburg suggested re-count for 1.5% of polling stations in first election, to proof correctness once and forever.
- Sample needs to be truly random
  - Prevent fraud in not audited polling stations
- Sample size needs to be dependent on outcome
  - Tight results require few votes flipped to change outcome
- Which sample size ensures high probability to detect fraud?
  - Easy in a two candidate race like US president elections
    - Look at number of votes that need to flip.
  - difficult in a multi party / multi coalition scenario
    - Germany: 5% threshold for party to join elected body
    - State of Hesse 2008:
      - Die Linke passes threshold by 3621 votes (approx. 1 vote per polling station)

Keep Physical Copies?

- Sample Size… State of Hesse 2008:
  - Normally: Approx 25,000 votes to flip a seat
  - CDU/FDP is lacking 75,000 votes to win election
  - But: 3621 votes less would kick *Die Linke* out of the parliament
    - 6 seats distributed to other parties, CDU/FDP wins

	Reality	
	Votes	**Seats**
CDU	1,009,775	42
FDP	258,550	11
	1,268,325	**53**
SPD	1,006,264	42
Grüne	206,610	9
Linke	140,769	6
	1,353,643	**57**
Total	2,621,968	110

- 3621

Scenario	
Votes	**Seats**
1,009,775	45
258,550	11
1,268,325	**56**
1,006,264	45
206,610	9
137,147	0
1,350,021	**54**
	110

Keep Physical Copies?

- Other issues
  - What if the electronic and audit result do not match?
    - Which result is used?
      - City of Hamburg suggested that electronic result should be binding
    - Do you have to increase the sample size?
  - TEMPEST proof printers?
    - difficult to protect the secrecy of the vote.
  - Printers fail or create paper jam
    - Mainly a concern of vendors who don’t want a paper trail

Transparency through cryptography?

Transparency through cryptography?

- Idea:
  - Use cryptography to ensure election integrity
    - Provide the voter with an encrypted receipt
    - Allow voter to verify that his vote is
      - cast as intended
      - counted as cast.
  - Cryptography prevents that voter can proove how he voted
    - Protects secrecy and free election
    - Prevents vote selling and coercion *(Nötigung)*

Transparency through cryptography?

- Proposals:
  - Prêt-à-Voter (P A Ryan, D Chaum, S A Schneider, 2005)
  - ThreeBallot (R L Rivest, 2006)
  - Scratch & Vote (B Adida, R Rivest, 2006)
  - Punchscan (D Chaum, 2006)
  - Scantegrity (D Chaum, 2007)
  - Bingo-Voting (J M Bohli, J Müller-Quade, S Röhrich, 2007)
  - VoteBox (D Wallach et al, 2007)

Approach

- What all proposals have in common:
  - Ballots have a unique id (random/serial number)
  - Voter receives a receipt which contains his vote in an encrypted form
  - All encrypted votes are published
  - Voter can verify that his vote is on the list

Immediate issues

- Can verification that **my** vote is counted as cast replace verification of entire election?
  - Does not protect against ballot stuffing
  - Does not allow external observers
  - How many voters need to cooperate to unveil fraud? Can cooperation be sabotaged?
  - If I know someone will not check, can I flip his vote?
    - Waste bin attack
    - Collect receipts through vote checking organisation

Immediate issues

- Who protects encrypted votes from decryption?
  - Is my vote really secret?
  - Who controls/protects the encryption keys?
  - Do serial/”random” numbers contain information about voter’s identity or on vote casted?
- Coercion might not require breach of secret, doubt in secrecy might be sufficient

Immediate issues

- Who ensures that each receipt is issued to a single voter only?
  - Give same serial number to multiple voters with same choice
  - Use serial numbers freed up to change the outcome

ThreeBallot

Ronald Rivest, 2005

ThreeBallot

- Ballot paper has three columns (“ballots”)
  - Chosen candidates are marked twice
  - Other candidates are marked once

Race 1			
Candidate A	❑	❑	❑
Candidate B	❑	❑	❑
Candidate C	❑	❑	❑
Race 2			
Candidate E	❑	❑	❑
Candidate F	❑	❑	❑
	154685	487762	019746

ThreeBallot

- Step 1: Mark every row once randomly

Race 1			
Candidate A	☐	☒	☐
Candidate B	☐	☐	☒
Candidate C	☐	☐	☒
Race 2			
Candidate E	☐	☒	☐
Candidate F	☒	☐	☐
	154685	487762	019746

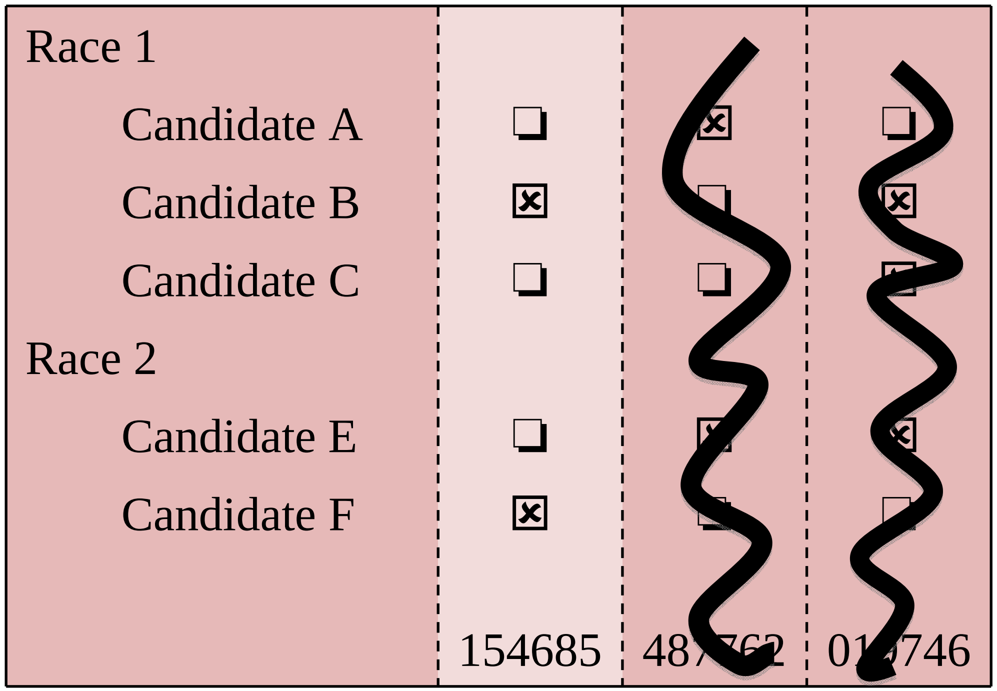
ThreeBallot

- Step 1: Mark every row once randomly
- Step 2: Mark your choice twice
- Step 3: A trusted “checker machine” ensures that the voter has submitted a valid ballot.

Race 1			
Candidate A	☐	☒	☐
Candidate B	☒	☐	☒
Candidate C	☐	☐	☒
Race 2			
Candidate E	☐	☒	☒
Candidate F	☒	☐	☐
	154685	487762	019746

ThreeBallot

- Step 4: Voter secretly and randomly chooses one of the three ballots for which he receives a carbon copy.
- Step 5: Voter compares original ballot and carbon copy
- Step 6: The three ballots are separated and cast.

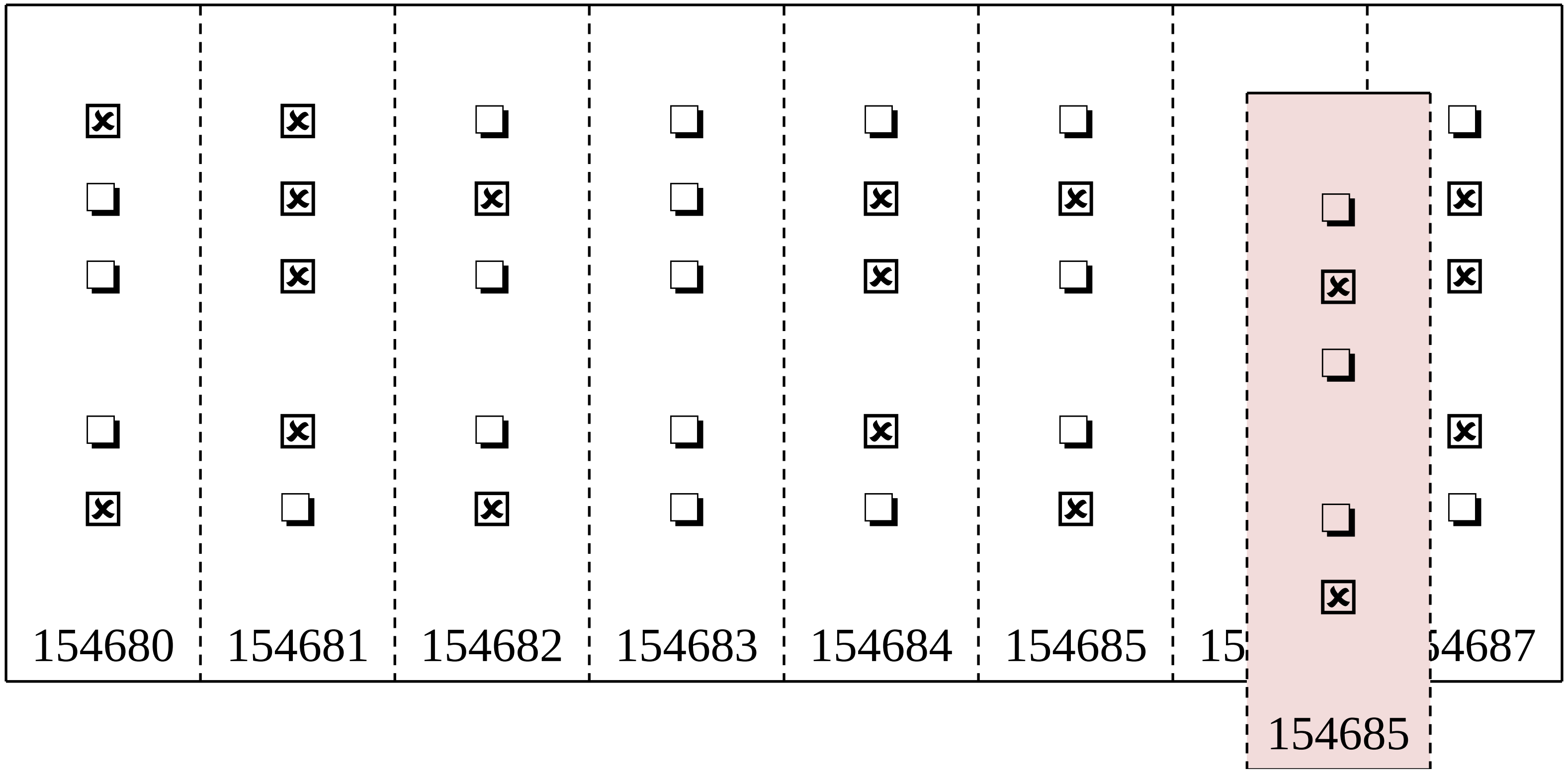


ThreeBallot

- Step 7:
  - Votes are counted as usual
  - With n participating voters, 3n votes are cast
  - If m voters select a candidate, he receives m+3n votes
- Step 8:
  - All Ballots get published on a bulletin board

ThreeBallot

- Step 8: Compare receipt with published ballots
- Receipt allows to verify that the ballot has been counted as cast, but does not unveil the choice of the voter



ThreeBallot

- Rivest: “Three Ballot is not a cryptographic voting protocol”
  - However, vote is pseudo-encrypted with voter generated random key
- Can be implemented for paper based and electronic elections
- ThreeBallot is intended as an academic discussion paper rather than a serious proposal for use in elections

ThreeBallot

- Not Coercion Free
  - Vote buyer can request certain pattern and check pattern appear under published ballots
  - E.g. election with two races and 10 candidates/parties per race (typical Bundestag election)
    - 20 rows, 22 votes (approx 7 per column)
      - 240k different possibilities to place 6, 7 or 8 votes into one column
      - 20^3 = 3G random patterns (minus permutations of the three ballots)
    - In a polling station with approx 1000 voters, it is extremely unlikely that all 3 requested ballots appear by accident

ThreeBallot

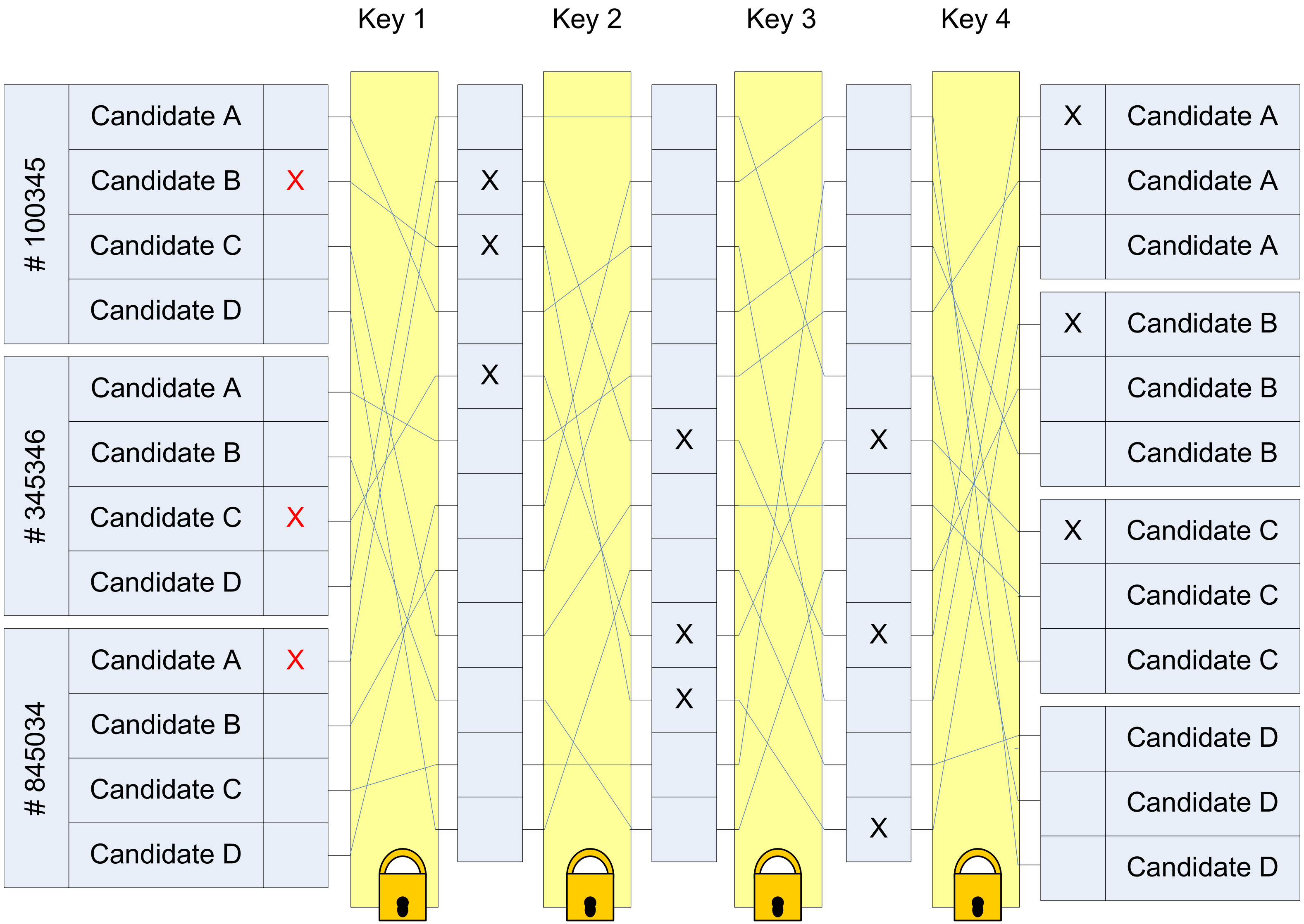
- More issues
  - Requires trust in serial numbers being secret and truly random
    - Puts secrecy of election at risk
  - Requires trust in checker/carbon copy algorithm
  - If voting organisation knows which ballot is chosen for copying, the two other ballots can be tempered
  - Extremely user un-friendly approach

ThreeBallot

- Might enhance auditablility
  - If nobody complains, voting organisation can be confident that everything went ok
- Does not enhance transparency
  - Requires trust in checker/copier
  - A evil checker can break secrecy of vote
  - Integrity of two ballots not copied is at risk
  - Why not trust counting in the first place

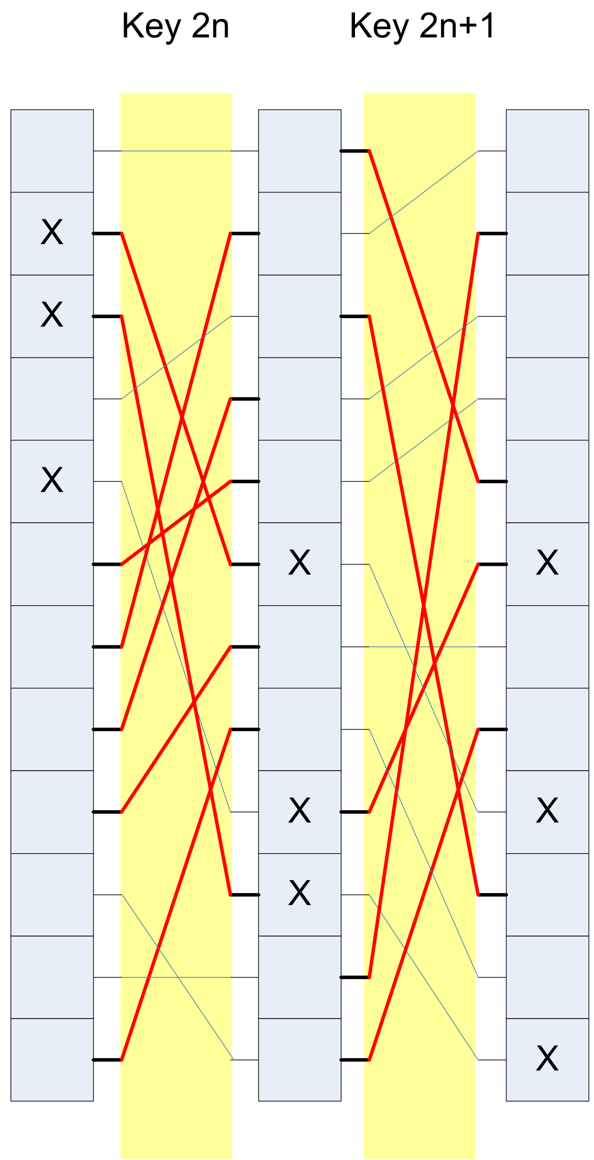
Some Fundamental Concepts

Mix Nets – D Chaum 1981



Randomized Partial Checking

- M Jacobsson A Juels, R L Rivest, 2002
- Audit pairs of keys/connections/servers
- Uncover 50% of all connections
- For each middle bit, either uncover inbound or outbound connection
- For every flipped vote, 50% chance to find in audit
- Chance to get away with n flipped votes is 2^{-n}
- Maintains vote secret depite of audit



Some Math: a^i mod p

- For any Integer a, Prime p
  - **c = g^i mod p** with $i \in [0, p-2]$ creates a sequence of numbers between [1, p-1]
  - Example: g = 3, p = 7

i	0	1	2	3	4	5
3^i	1	3	9	27	81	243
c =3^i mod 7	1	3	2	6	4	5

  - Creates pseudo random permutation of sequence 1, 2, ..p-1
  - For large p, difficult to solve for i with given c, g

Committments

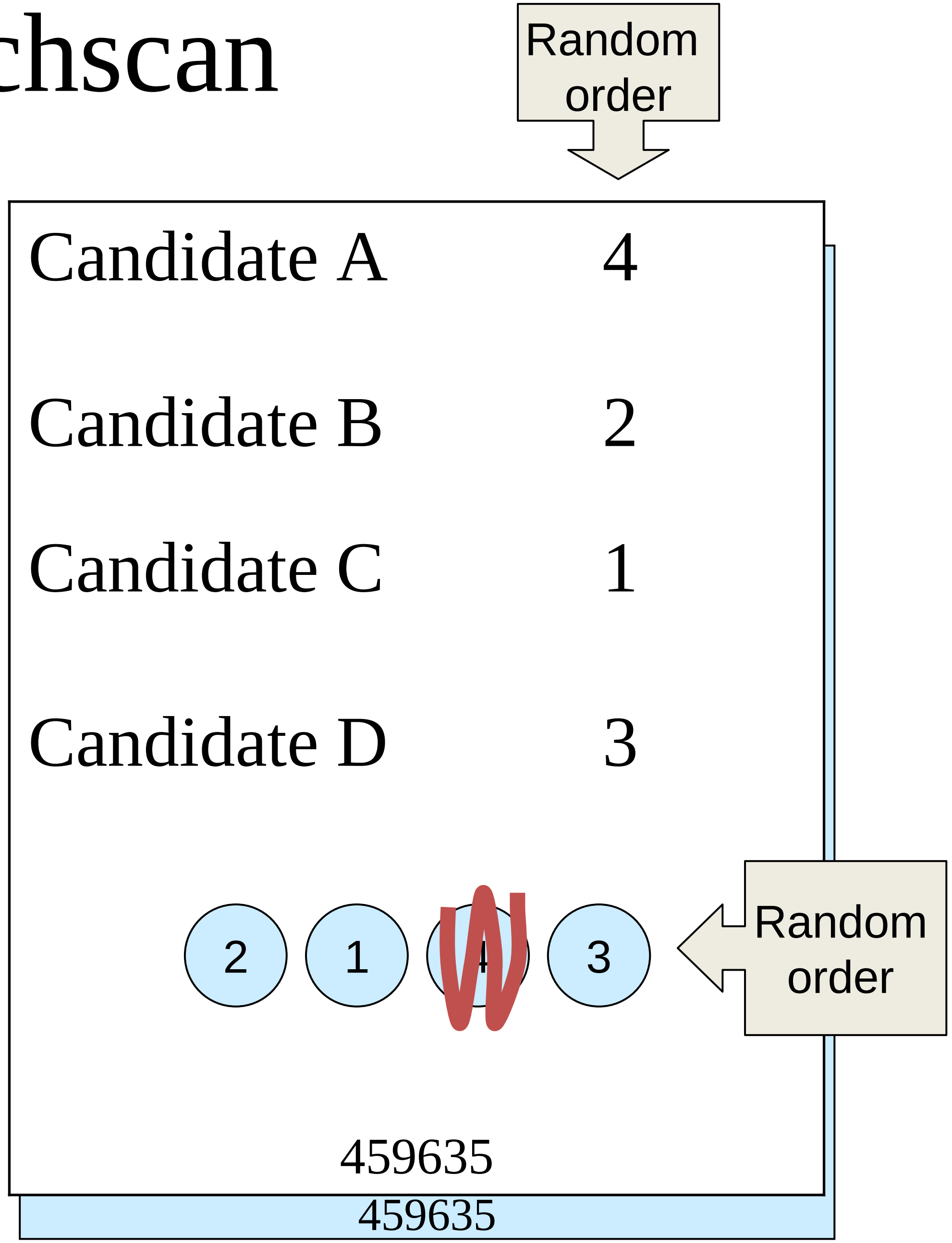
- E.g. Petersen Commitments
  - Large primes p, q and q devides p-1
  - Private key a
  - Public key $h = g^a \bmod p$
  - Commit to a secret x:
    Choose random r, Publish $c = g^{x+ar} \bmod p$
  - Reveal r, x
    Receiver verifies $c = g^x h^r \bmod p$

Punchscan

David Chaum, 2006

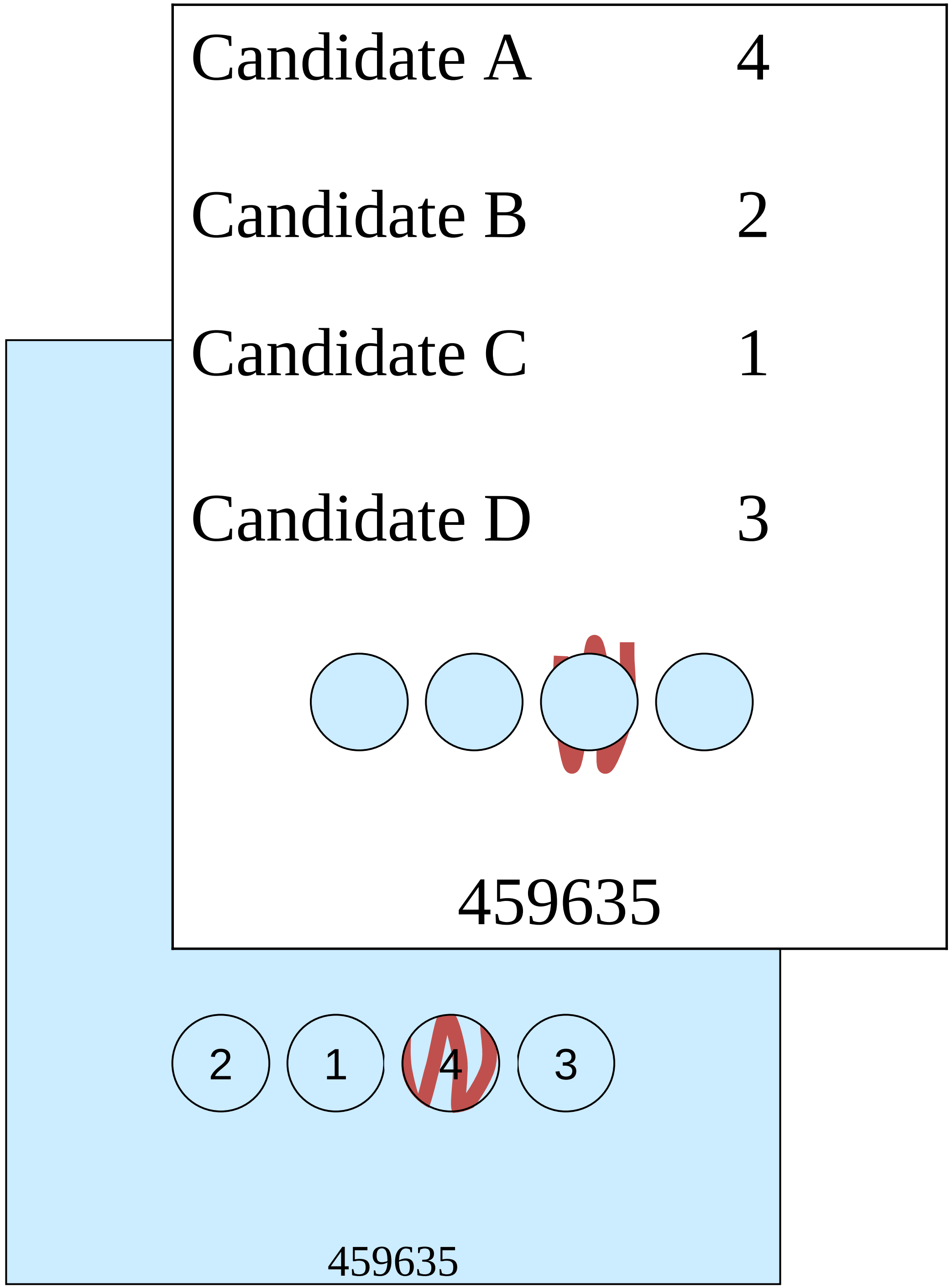
Punchscan

- Two superimposed sheets
- Voters receive individual sheets with codes next to each candidate.
- Candidate codes on bottom sheets are visible through holes on top sheet
- Voter marks selected candidate on both top and bottom sheet



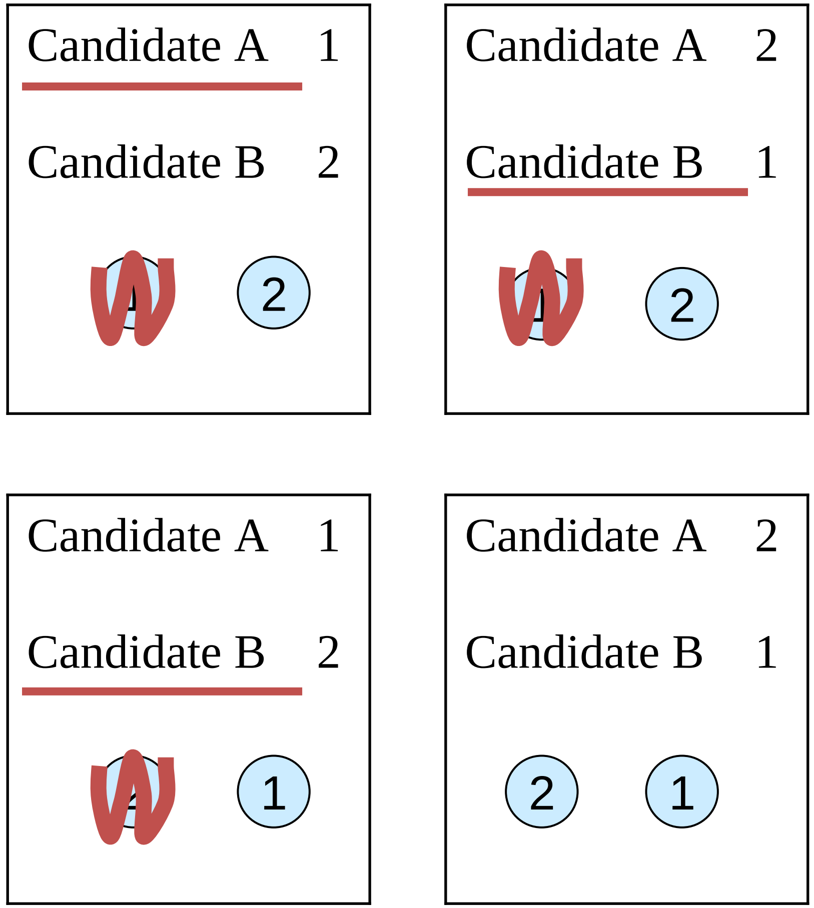
Punchscan

- Separate sheets
- Voter selects one sheet as receipt
- Receipt is scanned, other half is destroyed.
- All receipts are published on a bulletin board
- Permutations are validated through Mix Net / Randomized Partial Checking



Punchscan

- Protection against coercion dependent on sequence of events:
  - Voter needs to select top or bottom sheet as receipt before the ballot is presented
  - Had been overlooked by authors in earlier versions
  - Coercion attack:
    - Bring top layer with “1” assigned to Candidate A and left hole marked, or
    - Bring bottom layer where “1” appears left and is marked
    - Prefers Candidate B at 2:1



Scantegrety

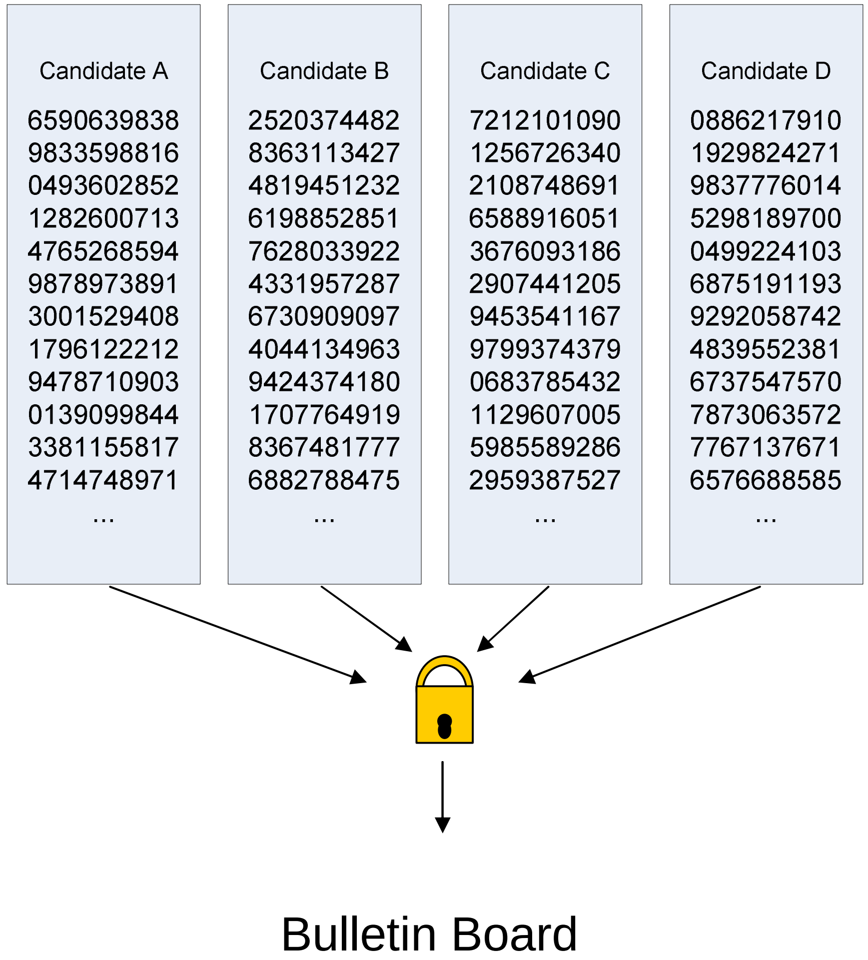
- Is a successor of Punchscan
- Similar concept, but all on one sheet
  - Random codes next to candidate names
  - Ballot paper is scanned
  - Codes related to chosen candidates are published
- Scantegrity 2
  - Only uncovers random codes of chosen candidates
  - Easier complaint validation

Bingo Voting

Jens-Matthias Bohli,
Jörn Müller-Quade,
Stefan Röhrich, 2007

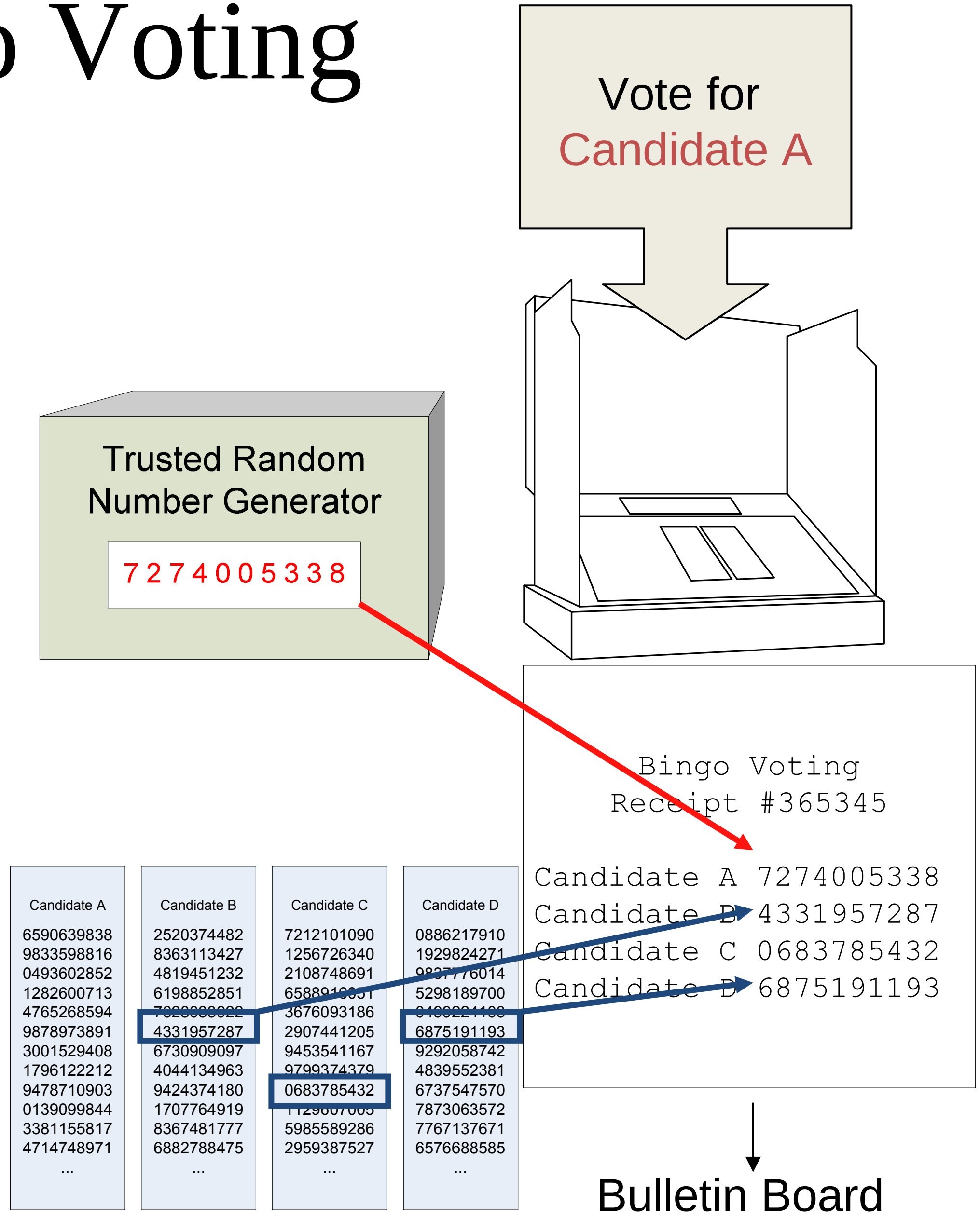
Bingo Voting

- Preparation Phase
  - For each voter, prepare a random number for every candidate (“*dummy votes*”)
  - Commit to candidate/number pairs
  - Commitments are shuffled and published on bulletin board



Bingo Voting

- Voting Phase
  - Voter selects candidate
  - Fresh random number is generated (“Bingo”) and presented to voter
  - Machine will print receipt with
    - fresh random number next to chosen candidate
    - Dummy votes next to other candidates
  - Voter verifies that fresh random number is next to the chosen candidate
    - Voter takes receipt home for later verification
    - Receipt does not allow the voter to proof his vote



Bingo Voting

- With his vote for Candidate A, the voter reduces the number of remaining dummy votes for all other voters by 1
- At the end of the election, the result can be determined (and verified) by counting the un-used dummy votes.

Candidate A	Candidate B	Candidate C	Candidate D
6590639838	2520374482	7212101090	0886217910
9833598816	8363113427	1256726340	1929824271
0493602852	4819451232	2108748691	9837776014
1282600713	6198852851	6588916051	5298189700
4765268594	7628033922	3676093186	0499224103
9878973891	~~4331957287~~	2907441205	~~6875191193~~
3001529408	6730909097	9453541167	9292058742
1796122212	4044134963	9799374379	4839552381
9478710903	9424374180	~~0663765432~~	6737547570
0139099844	1707764919	1129607005	7873063572
3381155817	8367481777	5985589286	7767137671
4714748971	6882788475	2959387527	6576688585
...	...	...	...

Bingo Voting

- Post Voting Phase
  - Publish results
  - Publish all receipts
  - List all unused dummy votes and corresponding commitments
  - Prove that every unopened commitment was used on one receipt
    - Makes use of Randomized Partial Checking

Bingo Voting

- Real World Implementation
  - Student council elections, Karlsruhe University
  - Java code published: `iaks-www.ira.uka.de/wahl`
    - But code does not compile due to missing object `de.uka.iaks.preelection.KonstantCollection`
    - Code comes with no documentation and does not use Javadoc tags

Bingo Voting

- If random number is not random, votes can be stolen
  - Dummy votes A_i, B_i, C_i, D_i
  - Voter 1 votes for Candidate A
    - Random number R_1
    - Receipt contains R_1, B_1, C_1, D_1
  - Voter 2 votes for Candidate B
    - Random number R_2
    - Receipt contains A_2, R_2, C_2, D_2
  - Voter 3 votes for Candidate A
    - Present R_1 to voter instead of Random Number R_1
    - Paper Receipt contains R_1, B_1, C_1, D_1 (same as for Voter 1)
    - Publish Receipt A_3, B_3, R_3, D_3
    - Vote has flipped to C, voter will still find “his” receipt published
- Transformation of problem:
  - Trust in random number generation rather than trust in voting computer

Bingo Voting

- Real world hassle
  - Commitments are only binding if shared
    - Publish commitments separately for every polling station (80k in Germany)
    - Where commitments are not downloaded before the end of the election, votes can be flipped and commitments can be re-issued.

General Issues

Concept vs. Implementation

- Secure Concept does not ensure Secure Implementation
  - E.g. Randomness
    - Random nature of pretended random values can never be verified by observer
  - E.g. Debian OpenSSH implementation
    - Until May 2008, Debian implementation of OpenSSH only created 32,767 different keys
  - What if we find out later that concept or implementation was not secure
    - Can not un-publish bulletin board

User vs. Administrator

- Even if concept is secure and code is shared
  - Fact that production system runs the same code is typically not verifiable by user
  - You need to be an administrator or rely on trust
- Are there *evil* implementations of the Secure Concept that (from user's perspective) behave similar to an *honest* one?
- Can I fool inexperienced users,
  e.g. by swapping the sequence of user interactions?
    - Who commits first, user or machine?

Denunciation Attack

- If you don't like the outcome of an election, denounce it:
  - manipulate data on bulletin board (e.g. receipts published)
  - (Some) voters checking their receipts will find mismatch between receipt on paper and published
  - "Evidence" that the unwanted outcome is a result of tampering
- Works for all protocols where receipts are published

Alice & Bob vs. Reality

- Werder (Havel) – State of Brandenburg
  - 35 km from Berlin, population 23’000
  - City council election 2008
    - 29 city council members
    - 8 parties, 109 candidates
    - 3 votes per voter , Cumulative voting – can all go to same candidate
- Frankfurt am Main – State of Hesse
  - City Council election 2006
    - 93 city council members
    - 11 parties, 643 candidates,
    - 93 votes per voter – cumulative voting, max 3 per candidate

Usability

- Werder (Havel), 2008 City Council election
  - 3 votes, 109 candidates
  - ThreeBallot
    - Mark 324 rows once, mark 3 rows twice
  - Punchscan
    - 327 holes (at best: 109 groups of 3)
    - Random order – good luck with finding your candidate
  - BingoVoting
    - Receipt will contain 327 random numbers
    - Check 3 of 327 numbers for correctness

Usability

- Frankfurt am Main, 2006 City Council election:
    - 93 votes (max 3 per candidate), 643 candidates
  - ThreeBallot
    - Mark 1836 rows once, mark 93 rows twice
  - Punchscan
    - 1929 holes (at best: 643 groups of 3)
    - Random order – marking your 93 choices becomes serious work
  - Bingovoting
    - Receipt will contain 1929 random numbers
    - Check 93 of 1929 numbers for correctness

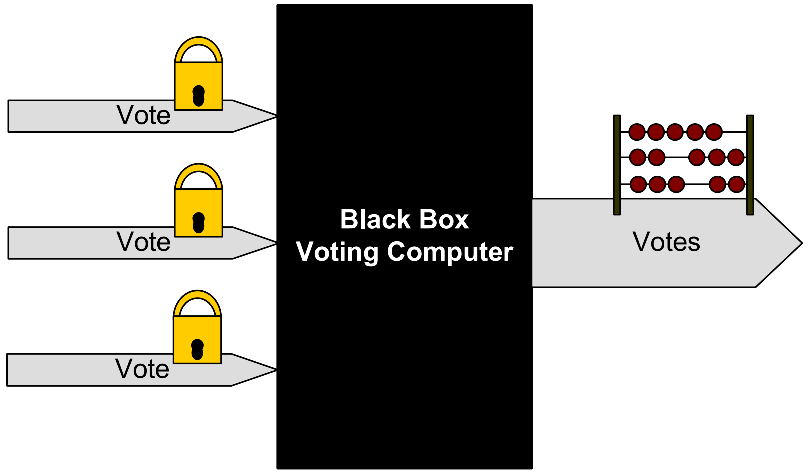
Scrutiny

- In case of dispute
  - Who can evaluate/understand integrity of election?
  - Who can understand/evaluate/challenge if the cryptographic method really insures integrity?
- Scrutiny process would become a battle between experts
  - Not longer resolvable by scrutiny committees or judges

Conclusions

Conclusions

- Core Issue is combination of secret input (votes) and black box process
  - Every attempt to fix auditability and transparency will put secrecy of vote at risk
- Can Cryptography fix it?
  - Interesting academic problem
  - Academic word is where this topic should remain



Conclusions

- Usability of described cryptographic methods collapses where eVoting has its biggest strengths (many votes, cumulative voting)
  - For simpler election systems, the added level of complexity is disproportional to the benefits of eVoting

Conclusions

- Even if cryptography fixed auditability:
  - Transparency remains issue because methods are too complex
  - Purpose of transparency is that voters have no doubt in the integrity of the election
  - This goal can not be achieved with methods that Alice and Bob do not understand

Discussion

www.ulrichwiesner.de

wahlcomputer at ulrichwiesner de