# Lecture 18
# November 5, 2025

ECS 235A, Computer and Information Security

# Administrative Stuff

- Answers to homework 1 and extra credit 1 are now posted
- The due date for Homework 3 is extended to Monday, November 10
  - Same for the extra credit
- No office hours today
  - Tomorrow's will be on Zoom at noon (the usual time, but *not* in person)

# Why Elections?

Topic of current interest

- Even though it is an off year, this election is important
- Most places will use some form of electronic voting (e-voting) systems
- Lots of FUD (fear, uncertainty, doubt) flying around
  - About election systems being inaccurate
  - About foreign actors rigging electronic voting machines
  - About requiring IDs to vote
  - …

# Key Questions

- Does using computers in an election process:
  - Introduce new ways for attackers to compromise the election, or prevent voters from voting?
  - Stop any of the previous ways for attackers to compromise the election, or provide new ways to enable voters to vote?

# Some Terms for E-Voting Systems

- BMD: Ballot Marking Device
  - Marks a paper ballot
- DRE: Direct Recording Electronic
  - Stores votes (ballots) electronically
- DRE + VVPAT: DRE + Voter Verified Paper Audit Trail
  - A DRE that also prints a paper record of the votes (ballots) cast on it
- PCOS: Precinct Count Optical Scanners
  - Used to count paper ballots at the precinct (polling station); these are stored electronically and the memory cards used to transfer results to central vote tabulator

# Some Terms for Elections

- Race
  - An element on a ballot that people vote on
- Overvote
  - More votes cast by a voter in a particular race than is allowed for a voter
- Undervote
  - Fewer votes cast by a voter in a particular race than is allowed for a voter
- Example
  - Race is 3 open seats for city council, 5 candidates for those seats
  - I vote for 2 of them, not 3: that's an undervote and it counts
  - I vote for 4 of them, not 3: that's an overvote and it doesn't count

# How an Election Works in Yolo County, CA

- Voters:
  - Go to polling station, give name, possibly proof of identity
  - Get ballot, enter booth, vote using marker to mark ballot
  - Put ballot in protective sleeve, leave booth
  - Drop ballot into ballot box
    - If provisional or conditional, put ballot and sleeve into envelope with voter's name, reason for the challenge (provisional) or condition (conditional) on the *outside*
- Vote-by-mail voters:
  - Fill in ballot
  - Put ballot into inner envelope
  - Put inner envelope into mailing envelope; sign the *outside* and mail it in

# End of the Day


- Election officials take ballot box to County seat
- Election officials remove ballots from envelopes
  - Provisional and conditional ballots handled separately
- Ballots counted, put into bags marked with precinct and count
- Ballots removed from bag, run through automatic counters
  - Humans intervene when problems arise
  - Intermediate tallies written onto flash cards
  - Every so often, cards removed, walked to tally computer, inserted, votes counted
- Reported tallies periodically updated, given for posting to web

# And Then . . .

- All places have provisional ballots
  - These are cast when it is unclear if the person is allowed to vote
  - In California, ***always*** on paper, never electronic
- California allows conditional ballots
  - These are cast by folks who register at the election (same day registration)
- Conditional and provisional ballots must be validated before being counted
- California also allows mail-in ballots arriving up to 3 days after Election Day to be counted

# The Canvass

Required by California law:

- Ballots for 1% of precincts counted by hand
  - Chosen with throw of dice; if some races not in precincts selected, add more in until all covered
  - Some counties have legal authority to use risk-limiting audit as well or instead
  - In California, you *must* use paper for this (hence, all DREs have VVPATs)
- Compared to tallies from election
  - If different, must be reconciled
- Certify final counts to Secretary of State
  - Has to be done within some number of days after election

# Some Election Requirements

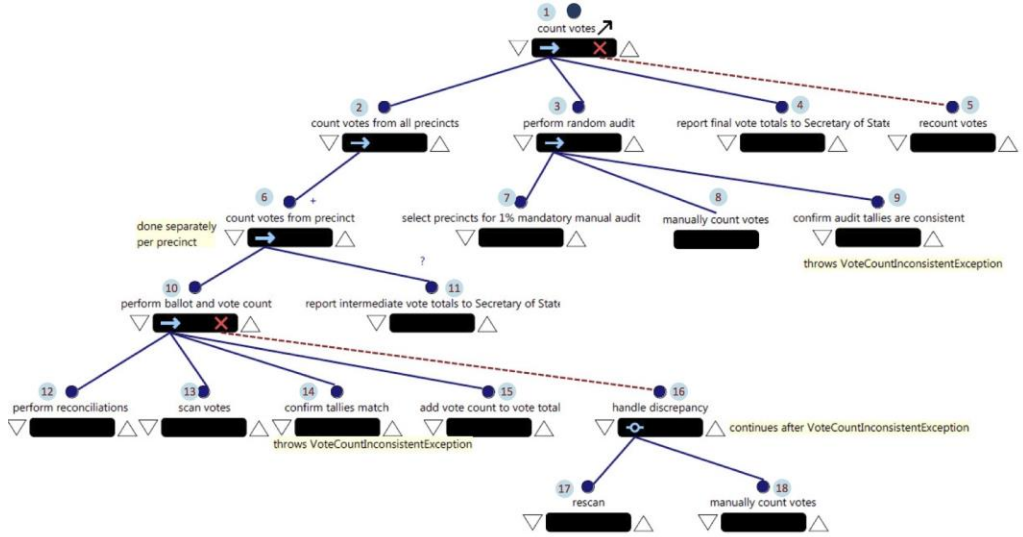
- Voter validation (authenticated, registered, has not yet voted)
- Ballot validation (voter uses right ballot, results of marking capture intent of voter as required by law)
- Voter privacy, secrecy (no association between voter, ballot; includes preventing voter showing others how he/she voted)
- Integrity (ballots unchanged, votes tallied accurately)

# Some Election Requirements

- Voting availability (voter must be able to vote, materials must be available)
- Voting reliability (voting mechanisms must work, even under adverse circumstances)
- Election manageability (process must be usable by those involved, including poll workers)
- Election transparency (audit election process, verify everything done right)

# What Should an E-Voting System Do?

- Replace manual activity, existing technology used in election process with better technology
  - Better in the sense of improving some aspect of the election process
- Examples
  - Easier to program ballots than print them
  - Can handle multiple languages easily
  - Easier to tally than hand counting



# Assurance

- Provide sufficient evidence of assurance to target audience that using e-voting systems makes elections at least as secure, accurate, etc. as elections without them (that is, using paper ballots)
- Who is “target audience”?
  - Computer scientists, election officials, politicians, *average person*

# Brief History

- Presidential election of 2000: massive confusion over ballots, and counting ballots, in Florida
  - Butterfly ballots did not align properly
  - Hanging chads made determining some votes difficult
- Help America Vote Act appropriated money to pay for electronic voting systems
- Federal standards developed by FEC
  - Voluntary Voting System Guidelines
  - NIST developing next generation

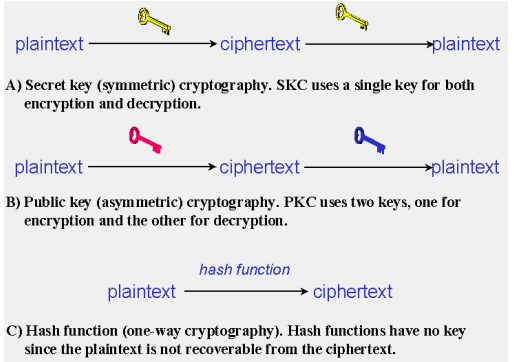
# Problems Developed in Testing

- 2003: Johns Hopkins people analyzed voting system program
- January 2004: RABA study of Diebold systems in Maryland
- April 2004: Diebold made available updates that were not certified
- Summer 2007: CA top-to-bottom review
  - Followed by EVEREST review in Ohio
- 2011: Washington DC internet voting test compromised
  - And the friendly attackers threw out the hostile ones
- 2014: Analysis of Estonia e-voting systems: many vulnerabilities found
- 2020: Voatz mobile voting app based on “blockchain technology”: many vulnerabilities found

# Problems Developed in Use

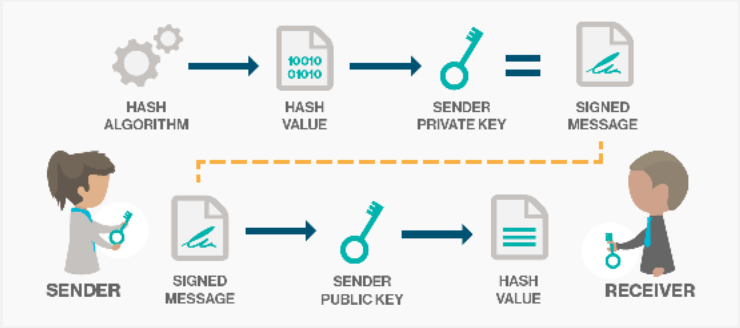
- Boone County, IN, 2003: 144,000 votes cast in a county with about 6,000 voters
- In 2006, polls opened late in several California (CA) counties (San Diego, Alameda, Plumas, Kern, Solano) due to system problems
- December 2006: Florida CD-13 post mortem of massive undervotes in a hotly contested race
- South Bronx, NY, 2010: a scanner miscounted 69/103 (70%) of ballots in Sep., then 156/289 (54%) in Nov.
- Los Angeles, CA, 2020: electronic poll books had connectivity problems, resulting in unacceptably long lines; BMDs failed, had paper jams

# Adding Cryptography




- RABA: Diebold's implementation of SSL protected confidentiality of precinct results, *but not integrity*
- Yolo County analysis: Hart used "random" access code on eSlates
  - Actually "pseudo-random", and it took looking at 20 such codes in sequence to regenerate all 10,000 possible codes (same for all systems)

# A Classic Example of Crypto



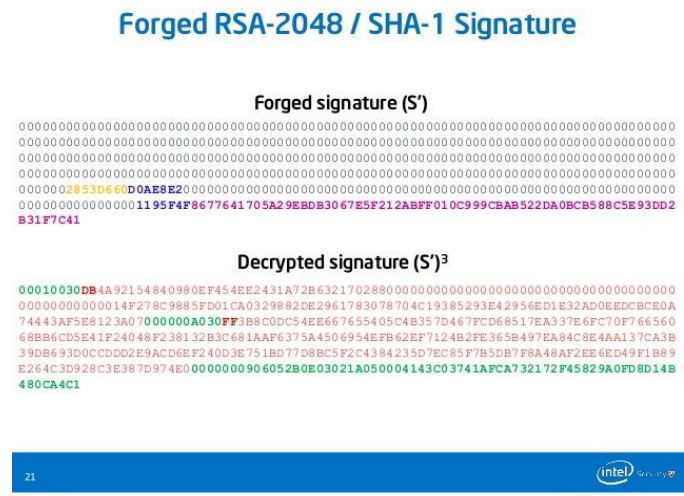

- Diebold added digital signatures to ballots in the version after the one California reviewed
  - Not examined in TTBR because it wasn't certified in California
- FSU SAIT: Alec Yasinsac and his team examined it
  - Signing technique was flawed, enabling forging of ballots

# Digitally Signing Ballots

- Signature is a SHA-1 160-bit digest signed using RSA:

  sign: write $M$; $S_{2048}$

  where $S_{2048} = RSA(privkey, 0_{1888} | SHA1(M)_{160})$



# Validating the Signed Ballot

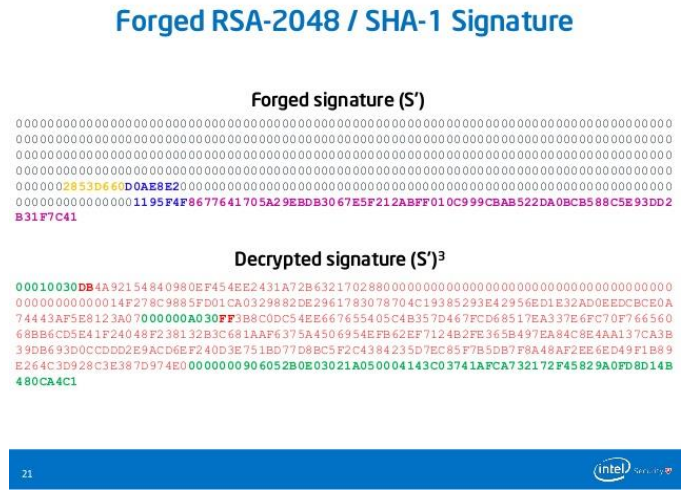
- Signature is a SHA-1 160-bit digest signed using RSA:

  sign: write $M$; $S_{2048}$

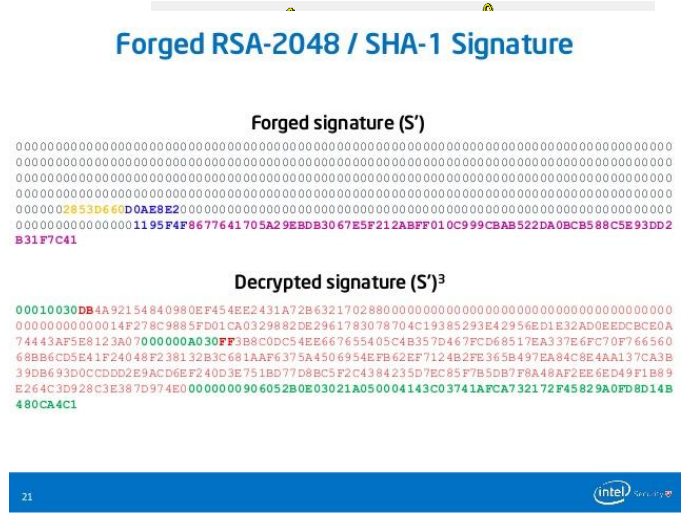
  where $S_{2048} = RSA(privkey, 0_{1888} | SHA1(M)_{160})$

  verify: read $M$; $S_{2048}$

  and if $RSA(pubkey, S_{2048})_{160} = SHA1(M)_{160}$, accept $M$



# Oops . . .




- Signature is a SHA-1 160-bit digest signed using RSA:

  sign: write $M$; $S_{2048}$

  where $S_{2048} = RSA(privkey; 0_{1888} | SHA1(M)_{160})$

  verify: read $M$; $S_{2048}$

  and if $RSA(pubkey; S_{2048})_{160} = SHA1(M)_{160}$, accept $M$
- But *privkey* is 3 and verify step above just checks low-order 160 bits!

Moral: Using cryptography doesn't make it secure; you have to use cryptography *correctly*

# When Random Isn't Random

- Hart Intercivic systems have 2 components
  - Hart e-voting system
  - Judge's Booth Controller
- JBC generates a "random" 4 digit number
- Voter goes to e-voting system, enters number, and then can vote
- But numbers are pseudorandom, *not* random
- Students generated 100 numbers, then wrote down the next 100 numbers
  - And verified they were correct

# How to Get There

- You need both standards and testing
- They must be independent of the developers of the systems
- They need to consider the users, operators, and maintainers of the systems
- Reports should show what tested, why, and how
- For e-voting systems, penetration testing is a *must*

# Add in the Internet

- It will enable authorized voters who cannot vote due to distance (or other factors) to do so
- It will increase authorized voter participation
- It will bring our elections into the modern, technological world
- It will be cheaper because we don't have to store the paper ballots

Problem:

- Election systems are now accessible to many more people than authorized voters!

# Where Would Attackers Strike?

- Probably not regular, individual electronic voting systems
- Attack the voter registration databases to disenfranchise voters
- Or attack the vendors and change the programs that run on those systems, or on the tallying systems

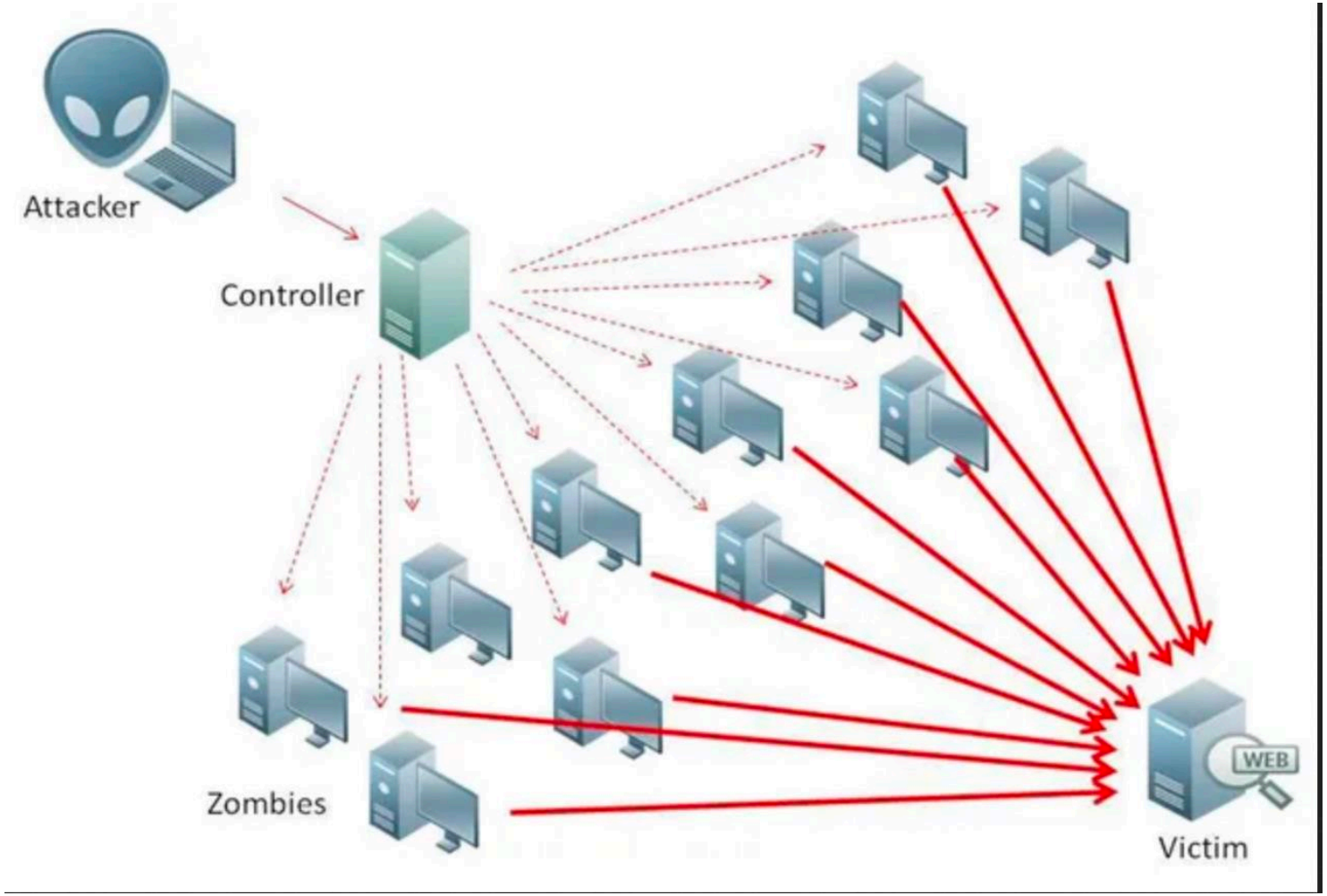
# Internet Elections

- If we can bank over the Internet, why don't we vote?
- Won’t it increase election turnout?
- Attack surface increases
- Election office resources won’t increase enough

# And If You Vote via Internet …

- Is your home PC/Mac secure?
- Is your smartphone secure?
- Are your router, ISP, . . secure?

# And If You Vote via Internet …



# Remote Voter Verification of Ballots

- Trick here is to protect against the validating mechanism being corrupted
- Example: we examined a system that enabled voters to check that their ballots were recorded correctly, and counted correctly, remotely
  - Used very neat cryptography, done by experts
  - We simply changed the web page on which the information that the user used to do the validation – no cryptography involved!

Moral: attackers don't have to rig or corrupt an election

They just have to make you *think* they did!

# Blockchains

- Background
  - Take ballot or chain of ballots and compute a hash from them
  - Encrypt this with a cryptographic key you keep secret (private key)
  - Publish the inverse cryptographic key (public key) so others can verify the small value was not changed
- For voting: many proposals for handling the chains

# Why Blockchains Fail for Elections

- Problem #1: denial of service (already discussed)
- Problem #2: how are those cryptographic keys generated?
  - A. Voter generates the pair (this is how it's usually done for other uses), and publishes the public key
  - A'. I vote multiple times, possibly under the name of a different voter each time. Prove I was the one who did this, and determine which votes are mine.
  - A''. I want to sell my vote. I give my private key to the purchaser. She can use the public key to verify that is my private key, and then see how I voted by finding the specific ballot added using that public key.
  - B. Election officials assign key. Now *they* can determine how I voted!

# How Not to Test Voting Over the Internet

- Occasional bills in various legislatures to do a "pilot study" using Internet voting in a real election
- A valid test requires knowing "ground truth", that is, what the results of the election should be
- How do you know this in a real election?


GROUNDTRUTH

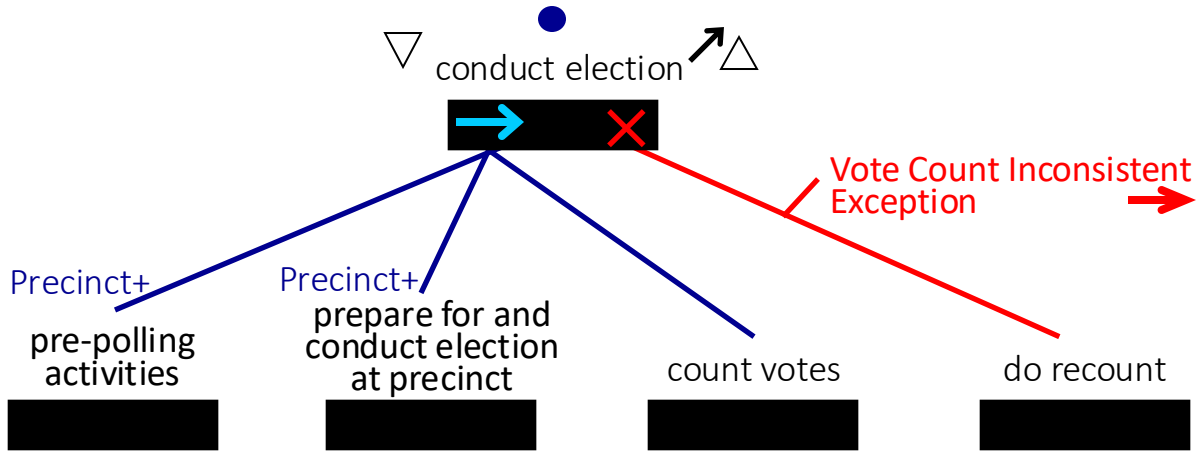
# The Take-Aways

- Know requirements of an election so you can *define* what you want
- Any computers used in an election process can be corrupted, so use good auditing techniques during the canvass
- And make sure the auditing techniques have good data!
  - Read: paper, as of now
- Given current election requirements, Internet voting poses great risks
  - The specific risks depend on how you do it

Remember, I don't have to rig an election to corrupt it; I just have to make you think I did!!!

# Election Process in Little-JIL

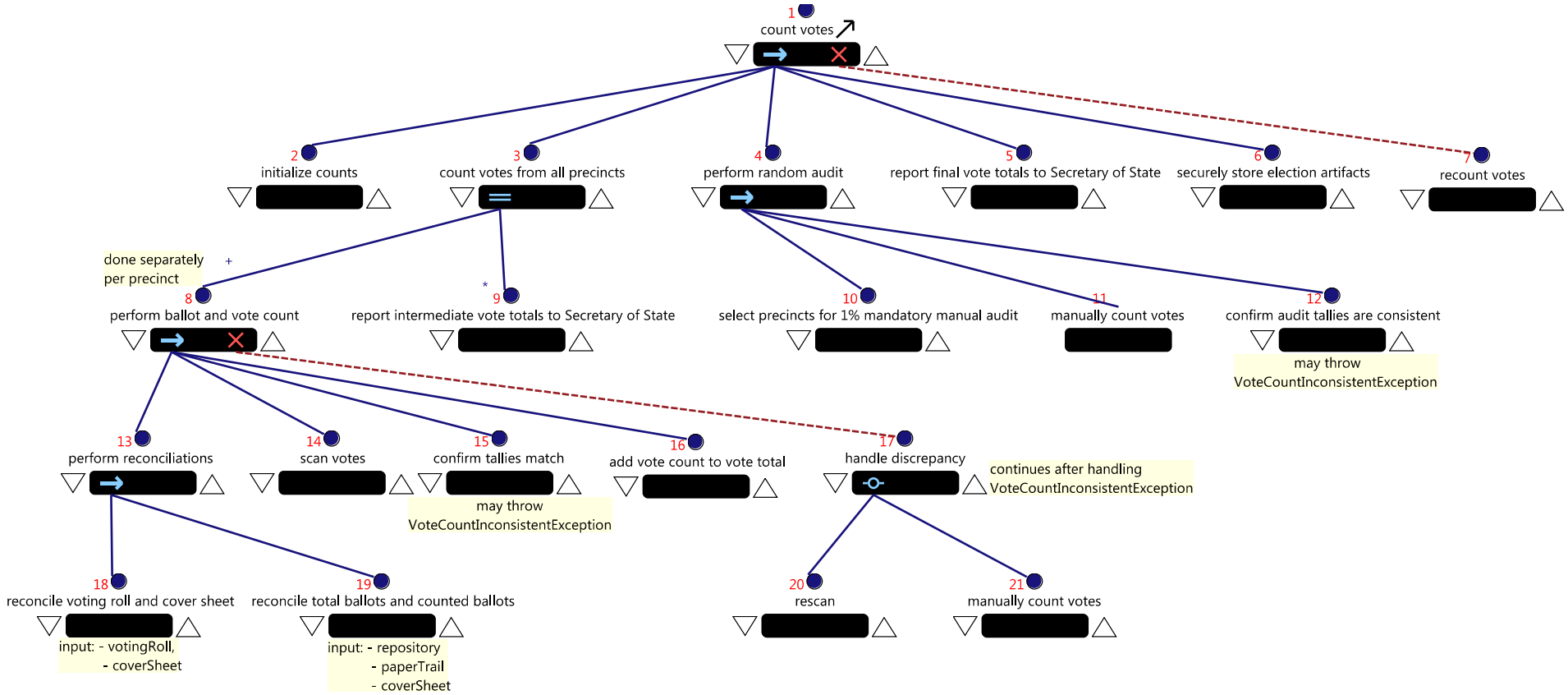
- Graphical process definition language with formal semantics; process represented as a hierarchical decomposition of steps



# Our Focus: Count Votes

1. Initialize counts
2. Count votes from all precincts
   - Count each precinct independently
3. Perform random audit
4. Report final vote totals to Secretary of State
5. Securely store election artifacts

# Subprocess “count votes”



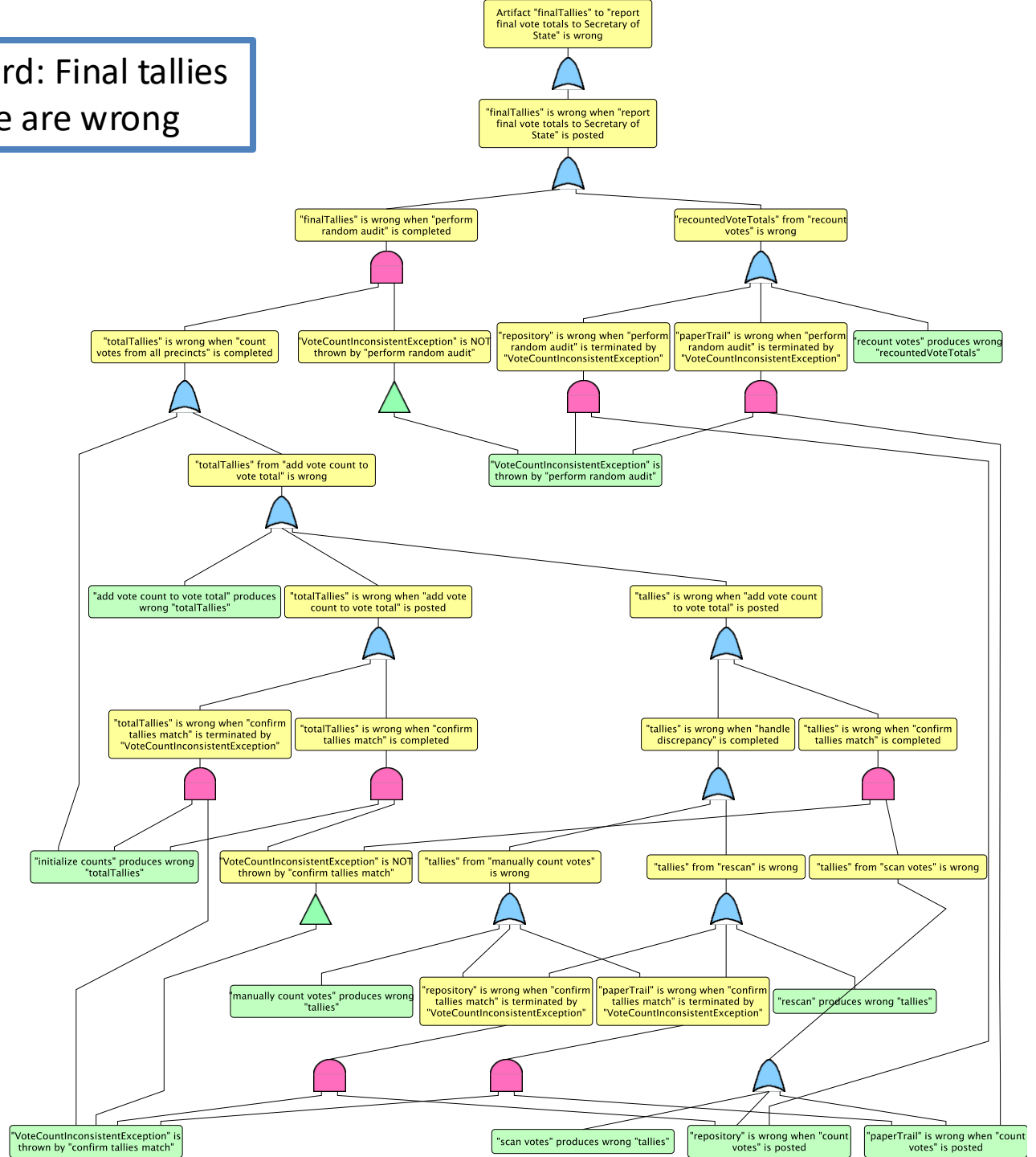
# Artifacts and Agents

| (ref #) step | Input artifacts | output artifacts | agent |
|---|---|---|---|
| (2) Initialize counts | | totalTallies | ElectionOfficial |
| (13) perform reconciliations | coverSheet; paperTrail; repository; votingRoll | | ElectionOfficial |
| (18) reconcile voting roll and cover sheet | coverSheet; votingRoll | | ElectionOfficial |
| (19) reconcile total ballots and counted ballots | coverSheet; paperTrail; repository | | ElectionOfficial |
| (39) check off voter as voted | votingRoll | timeStamp | ElectionOfficial |
| (44) put ballot in repository | repository | timeStamp | ElectionOfficial |

# Identifying Threats of Sabotage Attack

- Identify a *hazard* as the delivery of an incorrect artifact to a step in the process that delivers the artifact as a final process output
- From the process definition, automatically generate fault tree showing how hazard can occur

Selected hazard: Final tallies to Sec of State are wrong



# Example

- Hazard: wrong *finalTallies* delivered to the step *report final vote totals to Secretary of State*
  - Meaning the reported election results are wrong
- Automatically generate fault tree
- Use fault tree analysis tool to calculate minimal cut sets (MCSs)
  - Look for sets of activities where all agents are insiders and can modify final output (*finalTallies*) or artifact used to create final output

# 12 Possible Errors; Example Results

**1**

- Step *rescan* produces wrong artifact *tallies*
- Step *perform random audit* does not throw exception *VoteCountInconsistentException*

**2**

- Step *scan votes* produces wrong artifact *tallies*
- Step *confirm tallies match* does not throw exception *VoteCountInconsistentException*
- Step *perform random audit* does not throw exception *VoteCountInconsistentException*

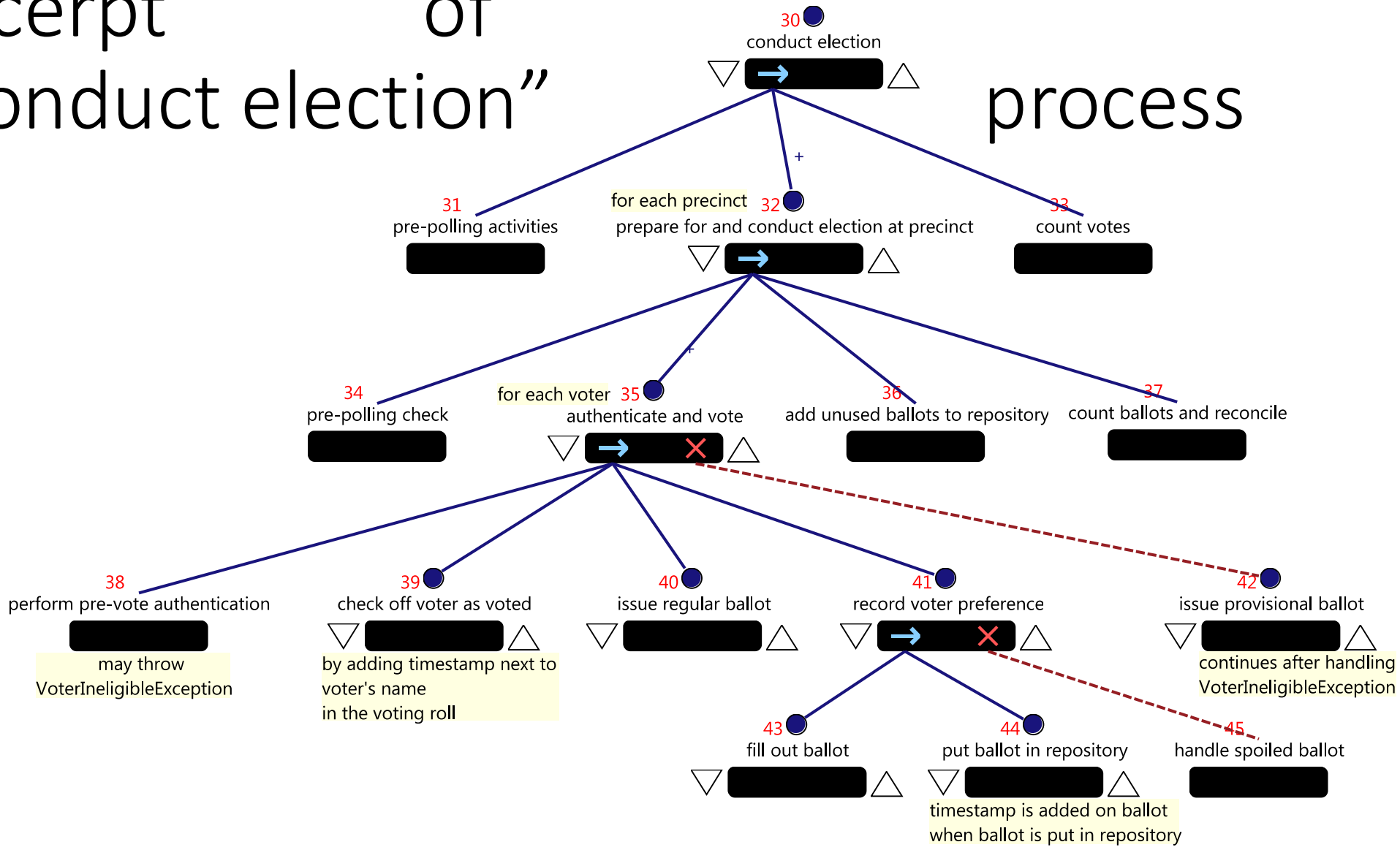
**3**

- Step *recount votes* produces wrong artifact *recountedVoteTotals*

# Data Exfiltration Attack

- In election context, associating a specific voter with a specific ballot
  - Done in Ohio, USA by correlating time-stamped ballots, poll books with times listed
- For expository purposes, voters vote on an electronic voting machine that time-stamps paper record of ballot
  - In Yolo, almost everyone uses paper, which is *never* time-stamped

# Excerpt of "conduct election" process



# Analysis

- If process executed as specified, only voter should know how she voted
- But . . .
  - Step 39: add timestamp next to name in roll
  - Step 44: add timestamp to ballot when placed in repository
- When can these be combined?
  - Artifacts are *votingRoll* (step 39), *repository* (step 44)
- Look in process model for a step, or sibling steps, using these artifacts
  - Steps 18, 19 here; parent is step 13, requiring both
  - *ElectionOfficial* is agent
  - So *ElectionOfficial* can exfiltrate data

# Evaluation

- Subjective
  - Process model validated by domain experts
  - Domain experts are better able to identify most worrisome types of insider attacks
- Objective
  - Focus on effectiveness, efficiency of process definition and analysis approaches
  - Little-JIL allows iterative process improvement based on feedback from domain experts

# Limitations

- Techniques are not always precise enough to fully describe the vulnerabilities and explain how they arise
- Analysis does not take into account full control and data dependencies of all steps
- Current agent descriptions are coarse
- Need to improve analysis of types of agents assigned to steps
- Use original analysis to suggest process modifications (automated or semi-automated)