Clients that purchase the systems can't tell if a system is really secure or not. The vendors can claim all they want. Vendors just build what the market demanded. It's the whole economic structure.

What's hard is the secret ballot issue:

- Prevent the voter from selling their vote
  - Cannot prove the vote, so even if he sells it, he can't prove it to the candidates
- Should be able to keep the vote secret

Defenses against vulnerabilities for the voting systems:
Technical:

- A lot of problems came from inputs other than the voting interface (i.e. memory cards), thus you can network the systems together with a central poll management machine.
  - Problems:
    - Adding another level of complexity
    - you can plug some device into the network, then you can tamper with the communications. Thus you need some sort of authentication
    - Anytime you network something together, if you compromise one, you compromise them all. (Or the central one)
    - If the central machine goes down, then all goes down. (Single point of failure)
    -
- None dynamic machines
  - Load the software and vote environment at the warehouse before it's shipped out, then you can manually change the voting machine.
  - Now, some of the warehouse seals the memory cards in a place holder and puts a seal on it to ensure no one tampers with it.
    - This will introduce denial of service, because if someone just tampers with the seal or by accident, then it renders the machine bad.

- Voter verification:
    - It's hard because of secret ballot
    - It's also hard because it's hard to count the votes if you try to encrypt the output of voter results.
    - Encryption mechanism:
        - Several electoral authority who run the vote(X number of them), choose them so they won't all work together to scam the vote
        - Single public key. There is a magic encryption scheme that if all the authorities come together, they can decrypt the key.
        - Names on ballot will be randomized for each ballot. Each one has two sheets of paper, one underneath, so filling in one bubble also fills in the other one. The second sheet doesn't have candidate names. The first paper goes in to vote, the second paper you can take home.
        - The scanning machine takes a picture of the bubble on the top ballot sheet, and then publishes the picture of the bubble somewhere, and can be used to verify the vote.
        - There's a bar code on the bubble side which contains the order of the candidate names. The bar code is encrypted under the public key.
        - The vote is tallied according to the public bulletin board. If the electoral authorities are trusted, they have enough information to talley the vote.
        - 2 problems:
            - They can know how I voted
            - I can't verify that they tallied the vote corrected.
        - Properties needed to tabulate:
            - First, you need to tabulate obliviously, they can only know the voter if all the electoral authorities collaborated together.
            - 2nd, they will provide a proof that they tabulated correctly

        - So you have a Mixed net.
            - The mixed net shuffles the net, a random permutation, and they won't tell anyone what shuffle they use. Each electoral authority gets a shuffle, so they unless all authorities get together, the privacy is ensured.
            - After the shuffle, you pickup to half the outputs, and the shuffler reveals where the outputs came from. Then the shuffler does a proof that the encryption corresponds to the same message.
            - There is a small defect, that the chance for electoral authorities of getting caught decreases exponentially with the number of votes they change.
            - There's also a privacy issue, where half the voters are revealed, so you do this a bunch of times.
            - Complex.
- Paper trail:
    - If the seal is in tact, then you don't' need to see the paper trail, but if the seal is broken, then you check the paper trail.
- Independent dual verification:
    - Have one screen connect to two independent system. Then you can verify the inputs that both machines match up.
    - If 2 systems have the same software, then one vulnerability from one machine, then both will be vulnerable.
    - But if you only have 2 systems, then one system can fail on purpose and create denial of service attack, so maybe we need 3.
- Fake voters:
    - Need to ask poll workers to keep track of how many humans voted.
- One proposal is that you use the computer with all the UI and voice, and then print out a paper ballot. Then the paper ballot is the one that is counted
    - Blind people still can't verify their own ballot

California currently:

- You stick the paper ballot into an optical scan machine
- To cross check and verify the machines, is they do a random audit of the paper ballots.
- After the election, they publish the results of the vote. Then they randomly select a handful of precincts, and then they get the paper ballots, and have someone manually count the ballots.