CSE 484 / CSE M 584: Computer Security and Privacy

Spring 2016

Franziska (Franzi) Roesner
franzi@cs.washington.edu

Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Announcements

- TA office hours have been scheduled:
  - **Bo and Viktor:** Thursdays, 11:30am-12:30pm, CSE 021
  - **Thomas and Kevin:** Fridays, 2:00-3:00pm, CSE 021
- If you're enrolled, you should have received a test email on the mailing list.
- If you're not enrolled and haven't signed the overload form, see me after class.

Last Time

- Importance of the security mindset
  - (challenging design assumptions, thinking like an attacker)
- There's no such thing as perfect security
- Defining security per context: identify assets, adversaries, motivations, threats, vulnerabilities, risk, possible defenses

Security Reviews

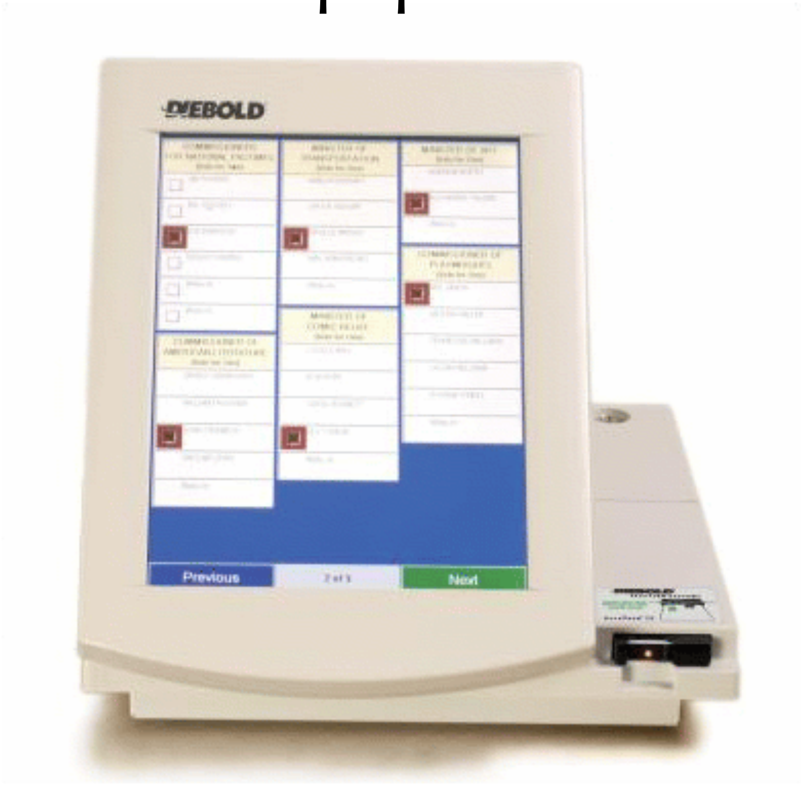
- Assets: What are we trying to protect? How valuable are those assets?
- Adversaries: Who might try to attack, and why?
- Vulnerabilities: How might the system be weak?
- Threats: What actions might an adversary take to exploit vulnerabilities?
- Risk: How important are assets? How likely is exploit?
- Possible Defenses

What Drives the Attackers?

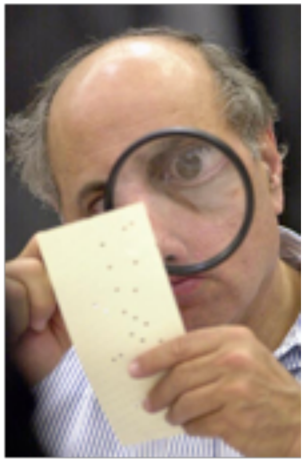
- Adversarial motivations:
  - Money, fame, malice, revenge, curiosity, politics, terror....
- Fake websites: identity theft, steal money
- Control victim's machine: send spam, capture passwords
- Industrial espionage and international politics
- Attack on website, extort money
- Wreak havoc, achieve fame and glory
- Access copy-protected movies and videos, entitlement or pleasure

Example: Electronic Voting

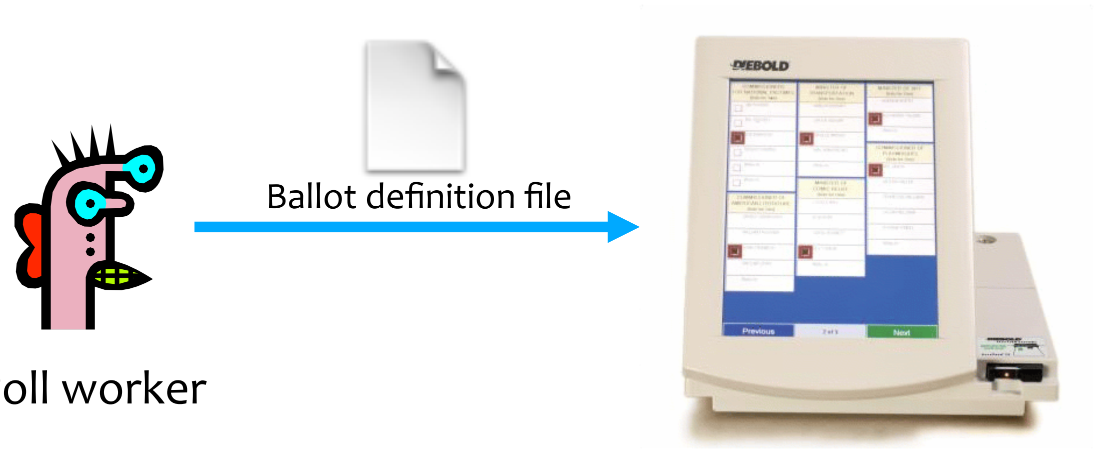
- Popular replacement to traditional paper ballots



AP



Pre-Election

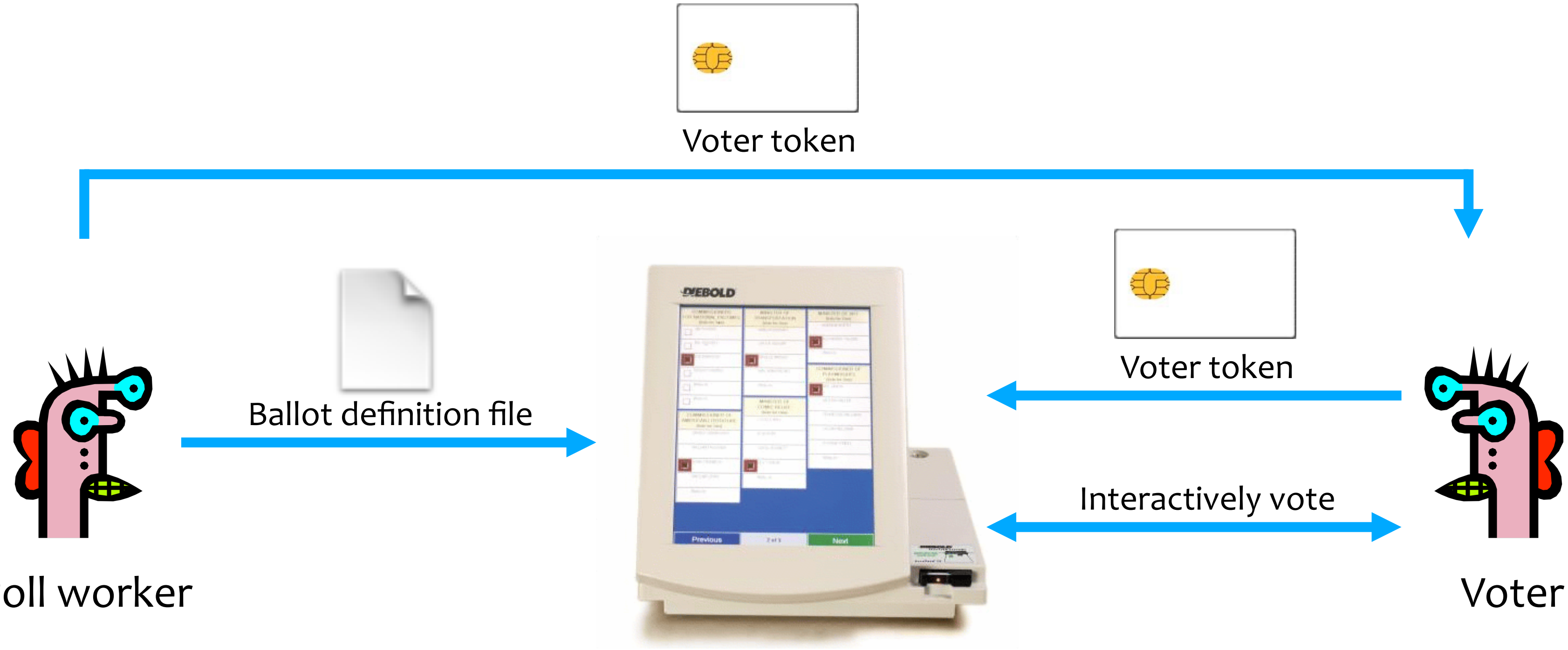


Ballot definition file

Poll worker

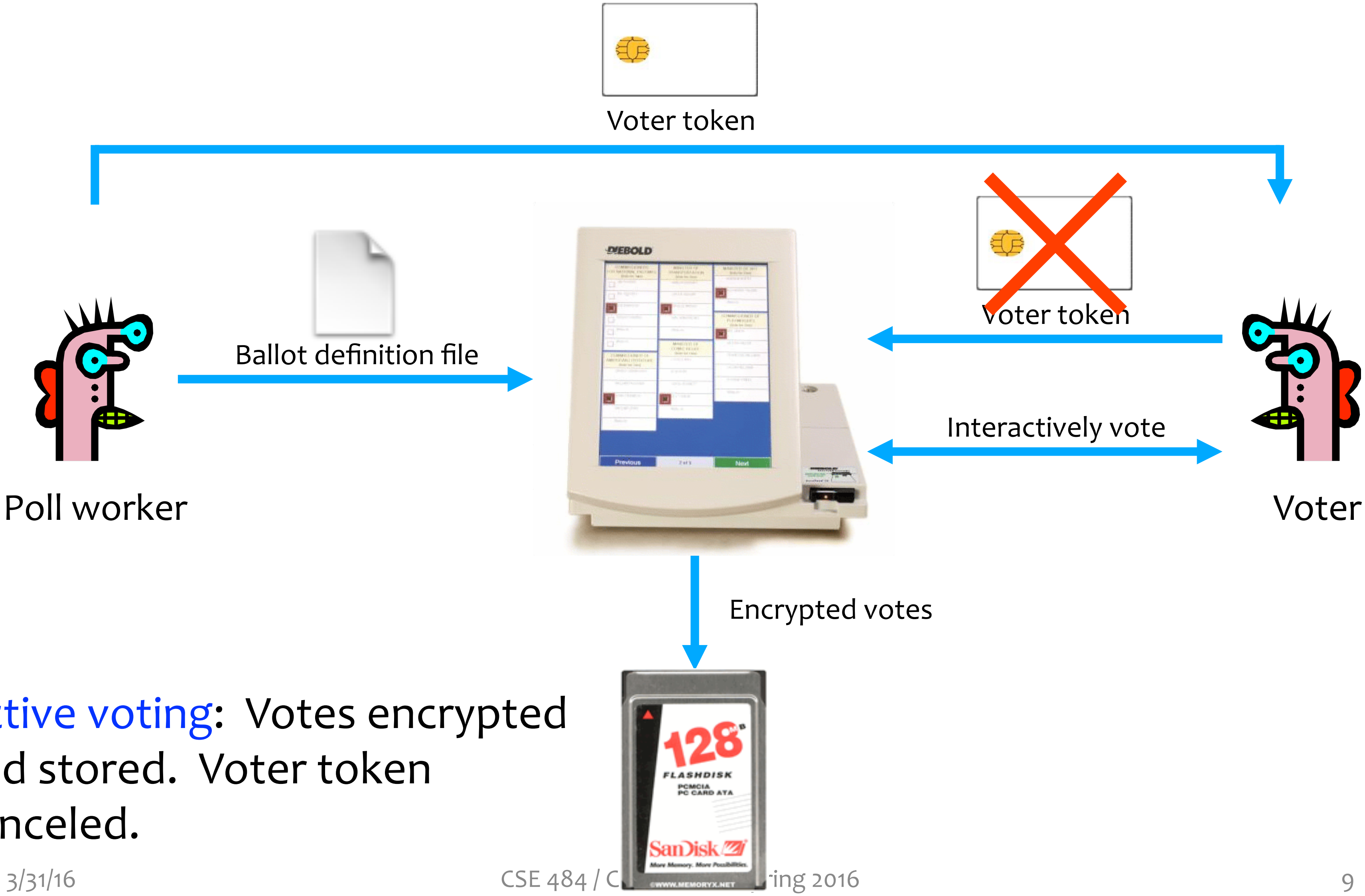
Pre-election: Poll workers load “ballot definition files” on voting machine.

Active Voting



Active voting: Voters obtain single-use tokens from poll workers. Voters use tokens to activate machines and vote.

Active Voting



Voter token

Voter token

Ballot definition file

Interactively vote

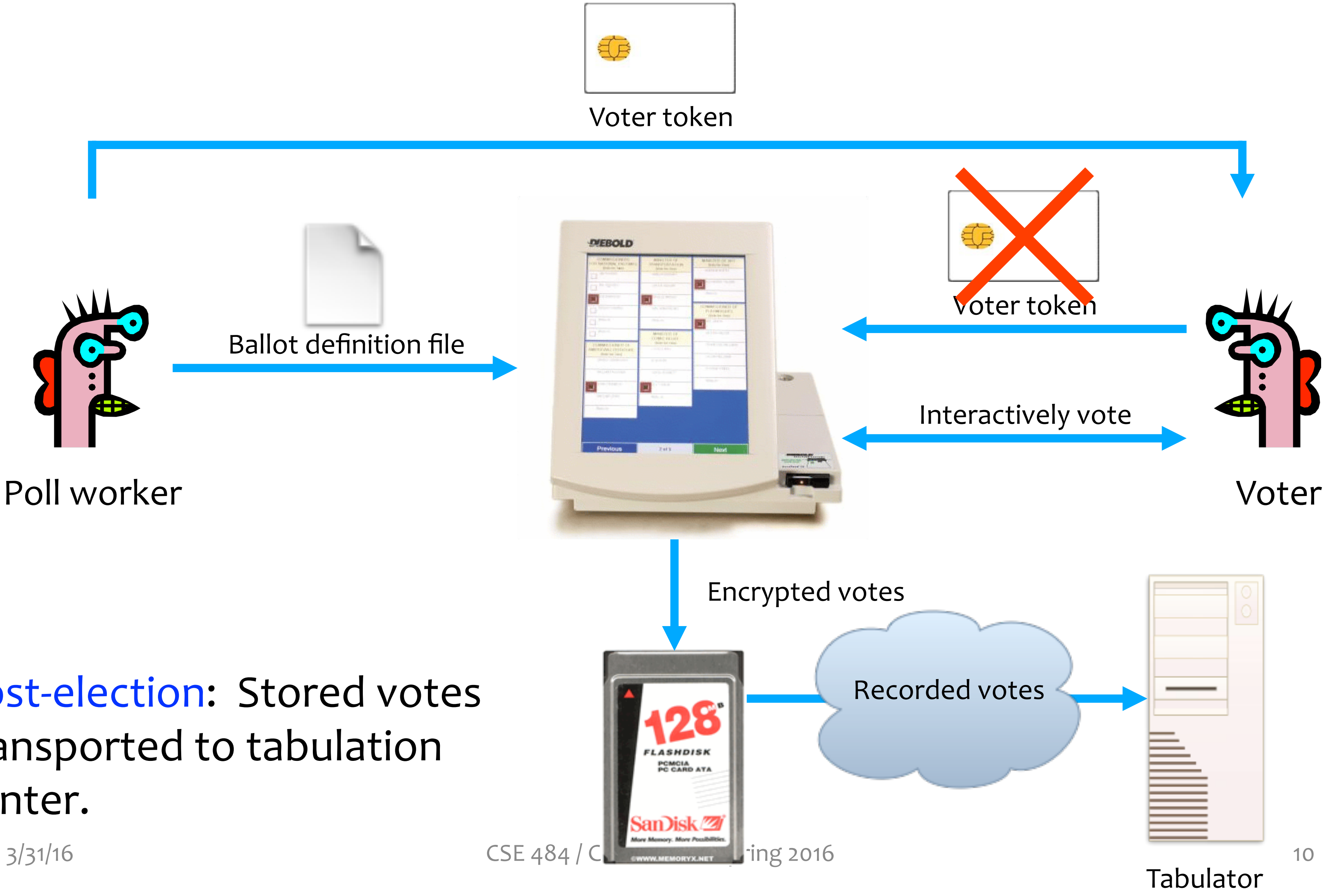
Poll worker

Voter

Encrypted votes

Active voting: Votes encrypted and stored. Voter token canceled.

Post-Election

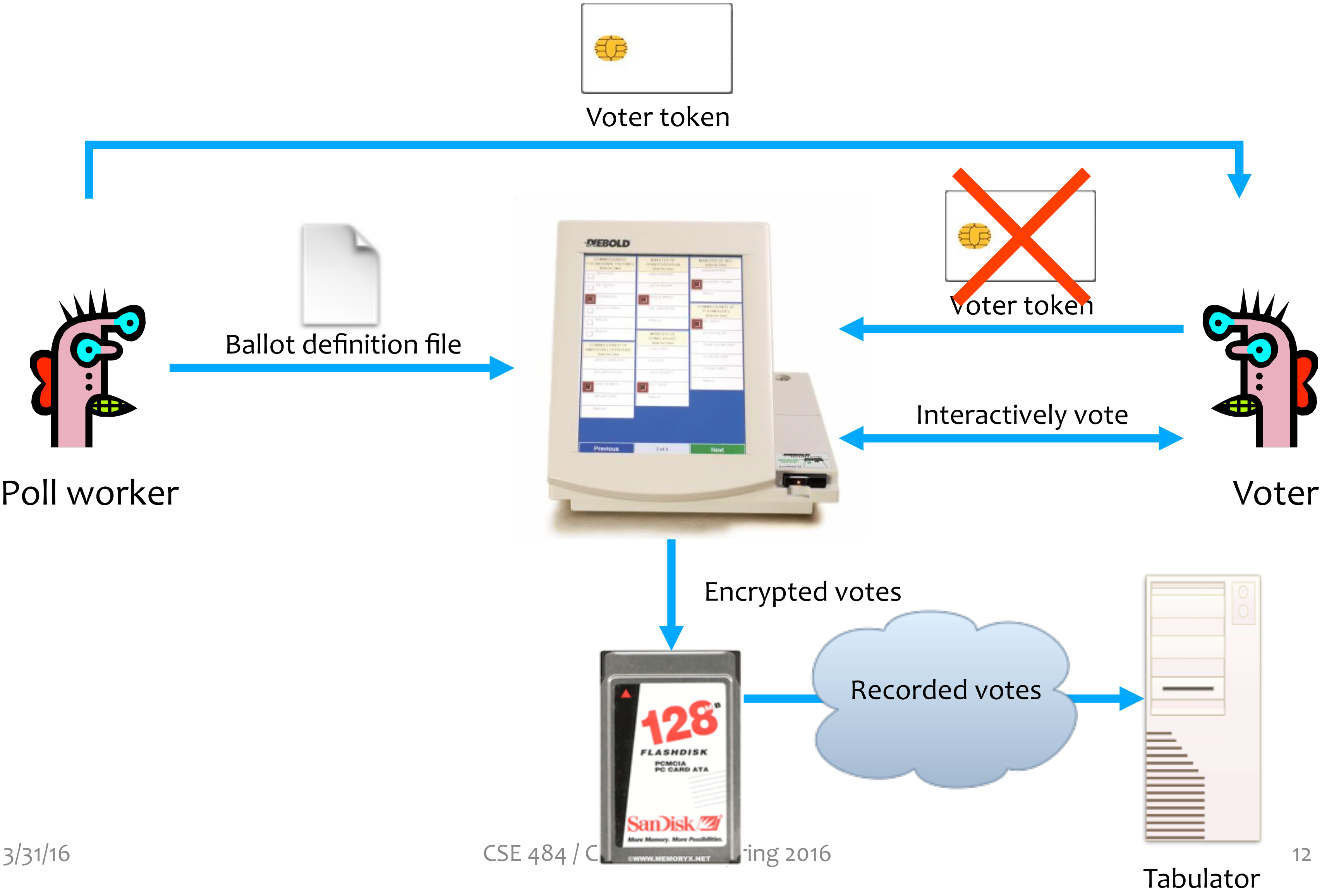


Post-election: Stored votes transported to tabulation center.

Security and E-Voting (Simplified)

- Functionality goals:
  - Easy to use
  - People should be able to cast votes easily, in their own language or with headphones for accessibility
- Security goals:
  - Adversary should not be able to tamper with the election outcome
    - By changing votes
    - By denying voters the right to vote
  - Adversary should not be able to figure out how voters vote

Can You Spot Any Potential Issues?



Voter token

Ballot definition file

Poll worker

Voter token

Interactively vote

Voter

Encrypted votes

Recorded votes

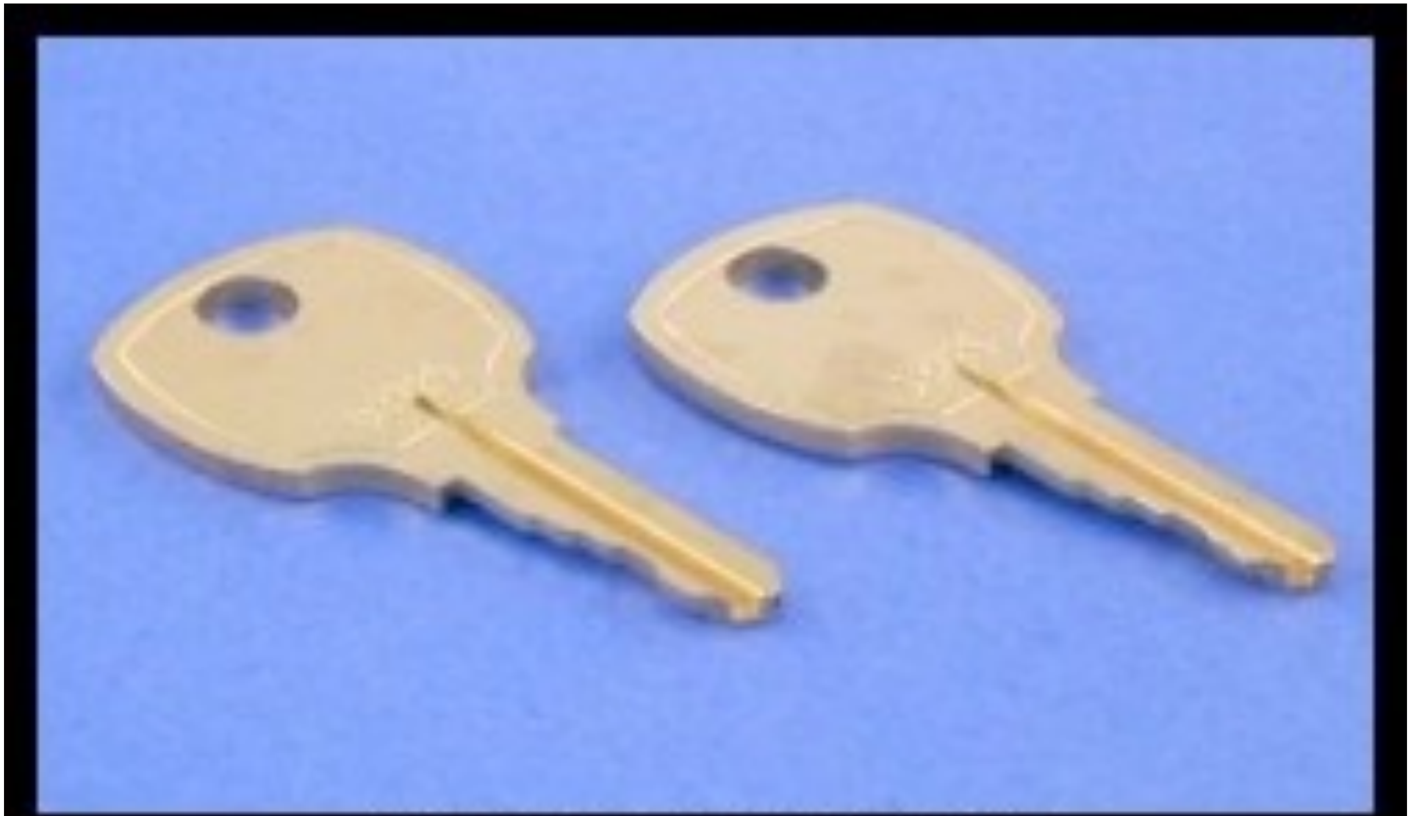
Tabulator

Potential Adversaries

- Voters
- Election officials
- Employees of voting machine manufacturer
  - Software/hardware engineers
  - Maintenance people
- Other engineers
  - Makers of hardware
  - Makers of underlying software or add-on components
  - Makers of compiler
- …
- Or any combination of the above

What Software is Running?

Problem: An adversary (e.g., a poll worker, software developer, or company representative) able to control the software or the underlying hardware could do whatever he or she wanted.

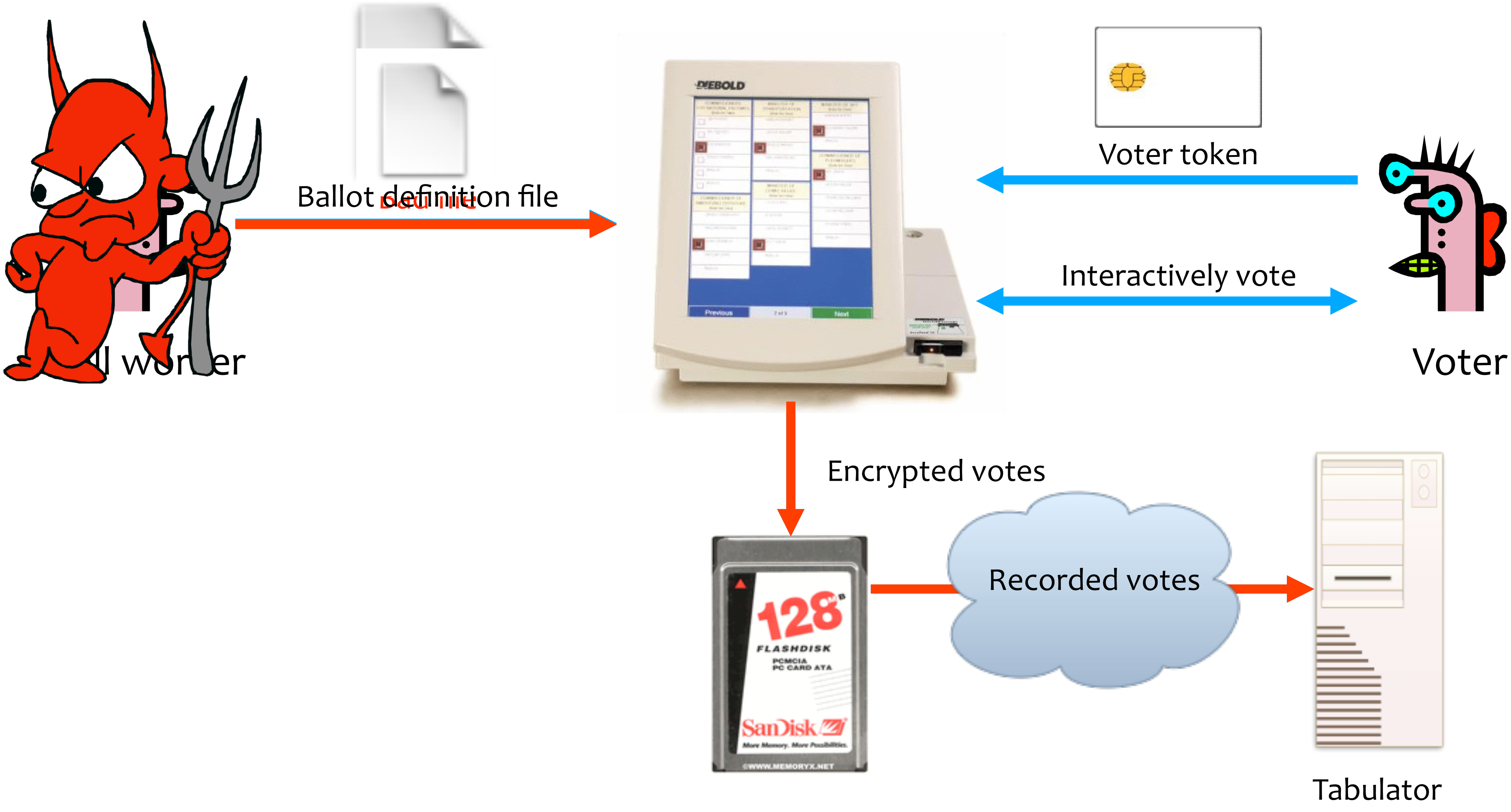


KEYS TO THE KINGDOM

Photo taken from Diebold's online store. The keys that open every Diebold touch-screen voting machine. Working copies have been made from the photo.

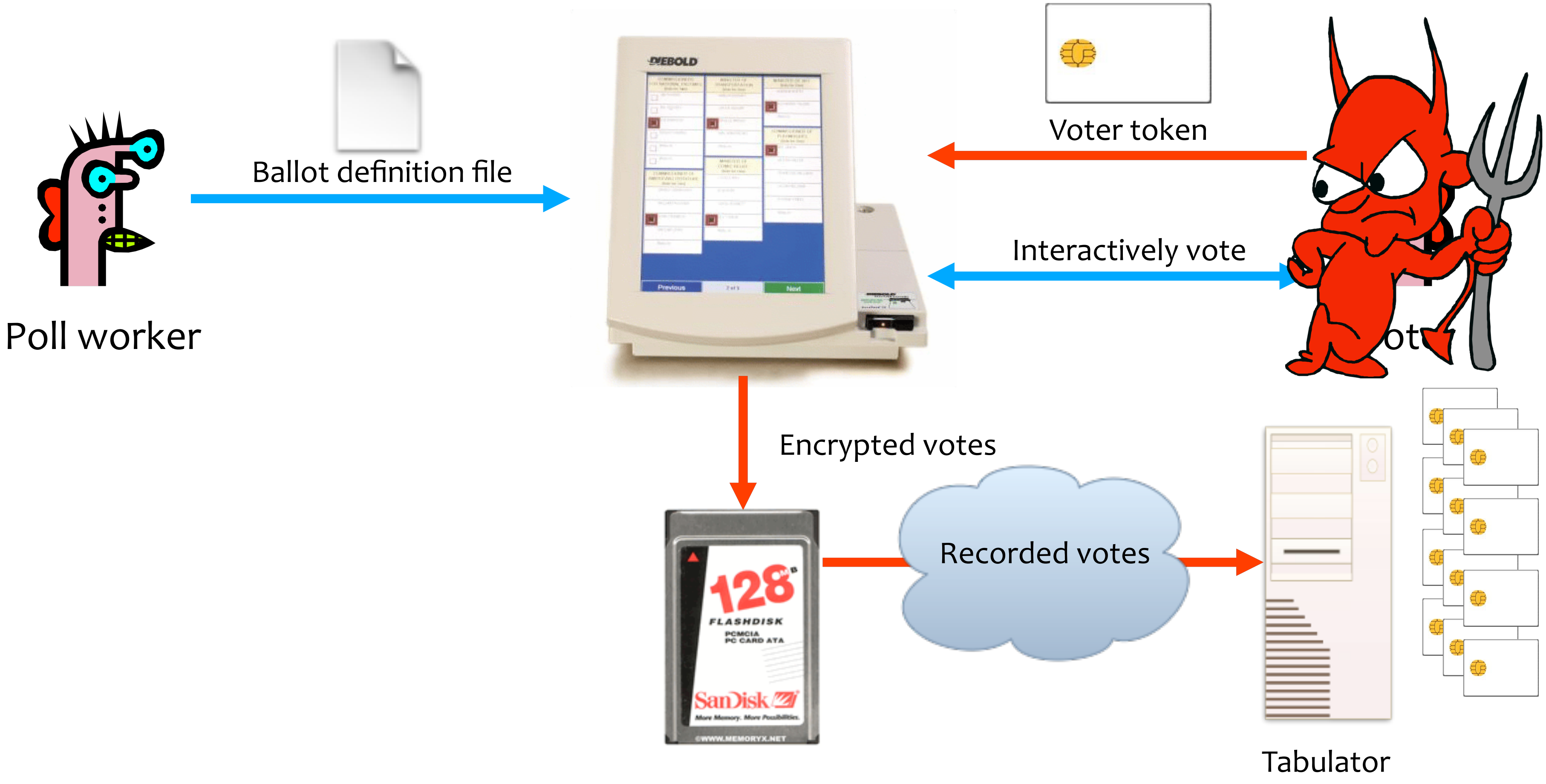
Problem: Ballot definition files are not authenticated.

Example attack: A malicious poll worker could modify ballot definition files so that votes cast for "Mickey Mouse" are recorded for "Donald Duck."



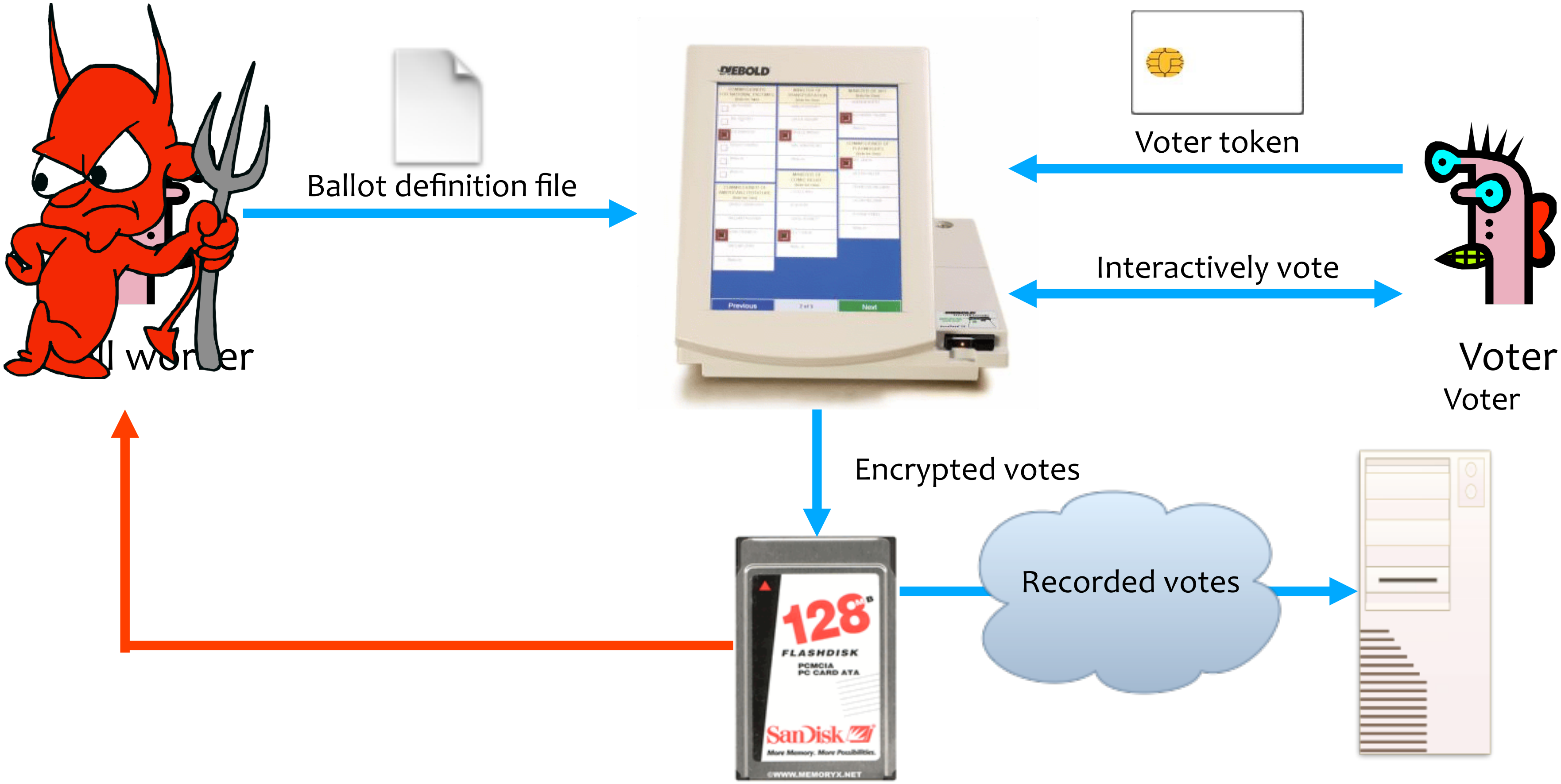
Problem: Smartcards can perform cryptographic operations. But there is no authentication from voter token to terminal.

Example attack: A regular voter could make his or her own voter token and vote multiple times.



Problem: Encryption key (“F2654hD4”) hard-coded into the software since (at least) 1998. Votes stored in the order cast.

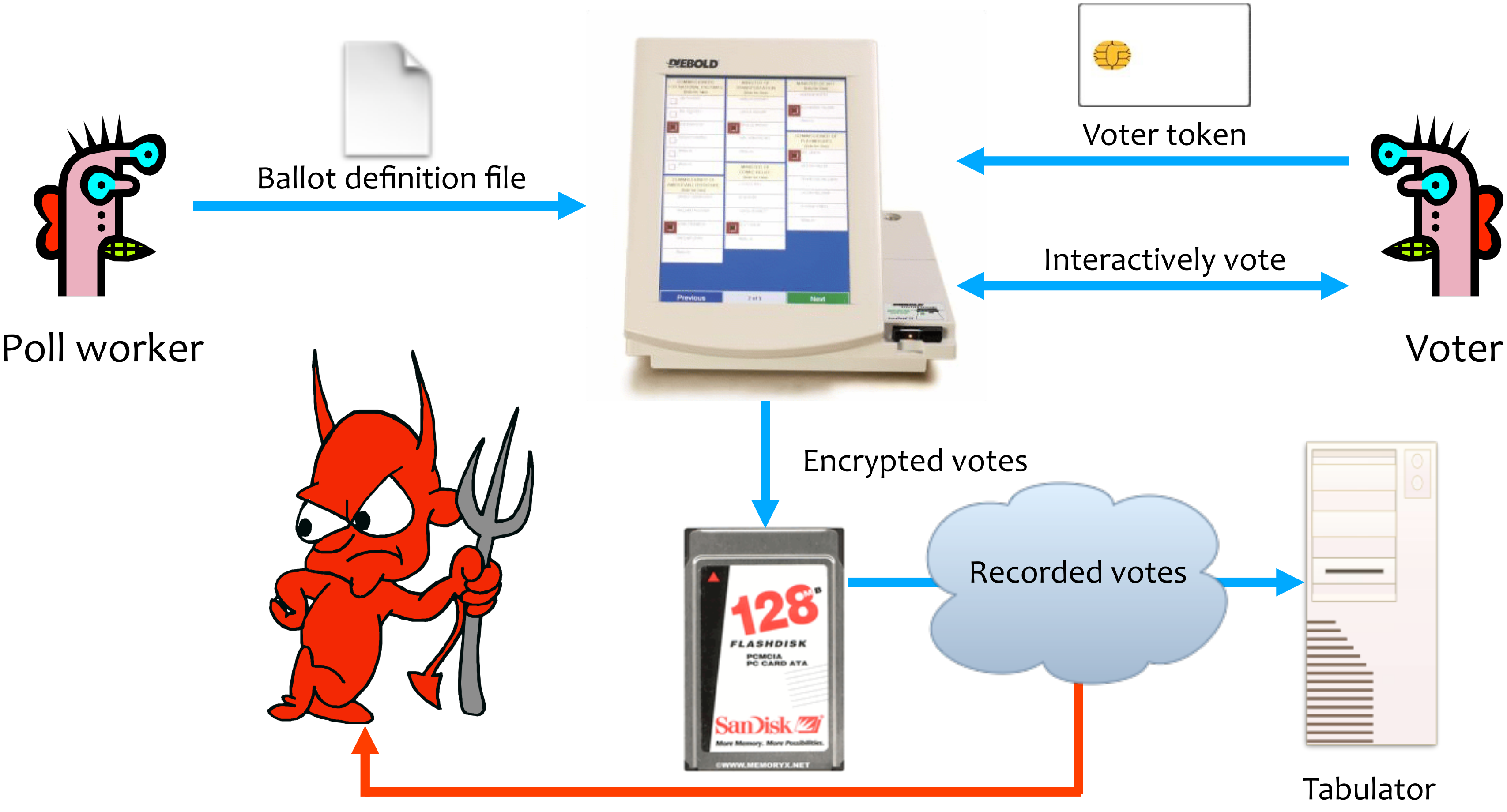
Example attack: A poll worker could determine how voters vote.



Tabulator

Problem: When votes transmitted to tabulator over the Internet or a dialup connection, they are decrypted first; the cleartext results are sent the the tabulator.

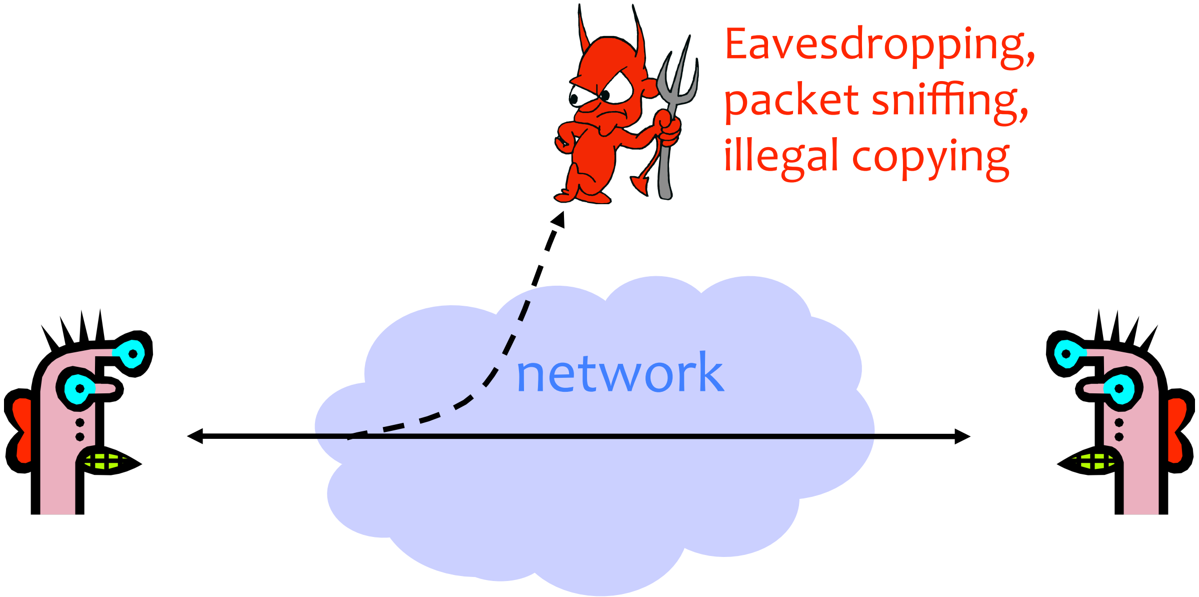
Example attack: A sophisticated outsider could determine how voters vote.



SECURITY GOALS (“CIA”)

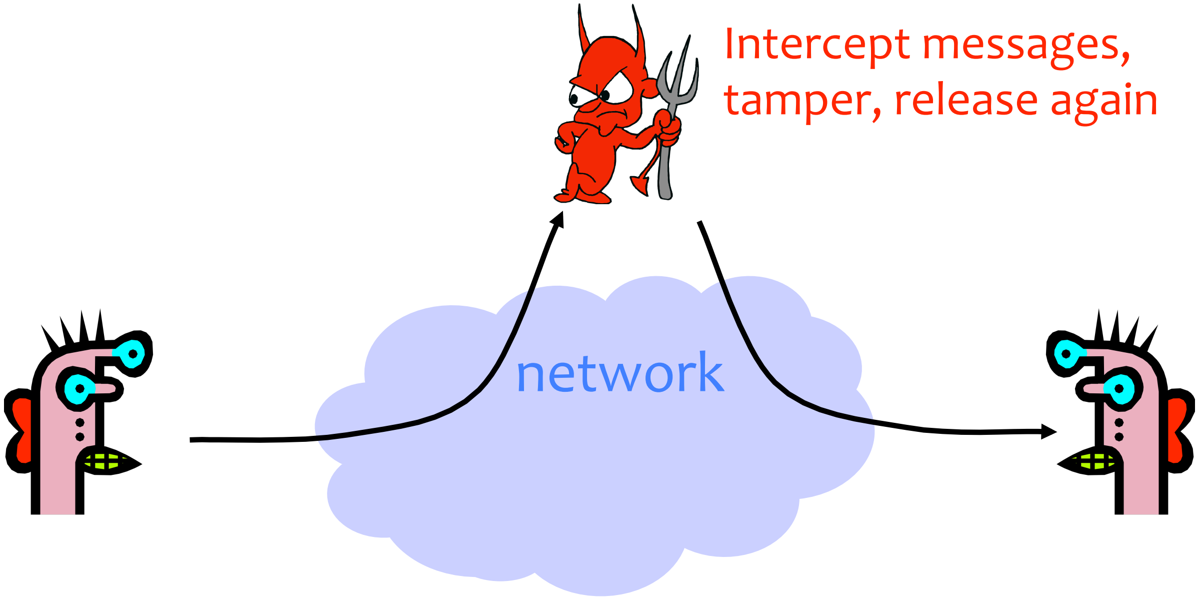
Confidentiality (Privacy)

- Confidentiality is concealment of information



Integrity / Authenticity (1)

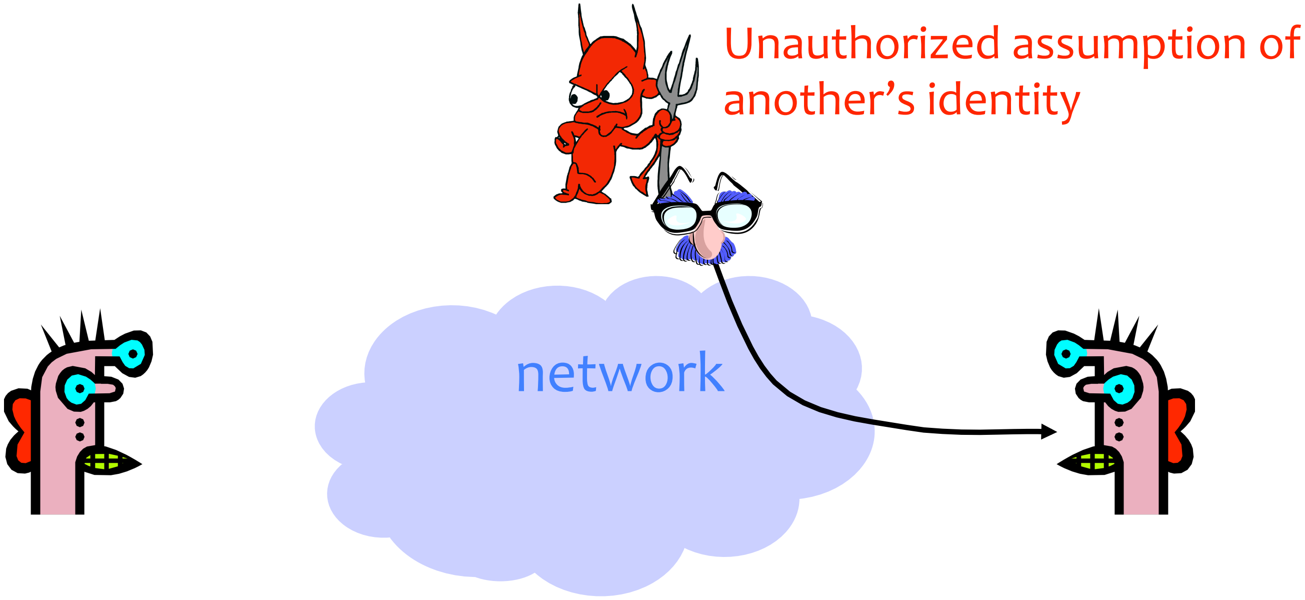
- Authenticity / integrity is prevention of unauthorized changes



Intercept messages, tamper, release again

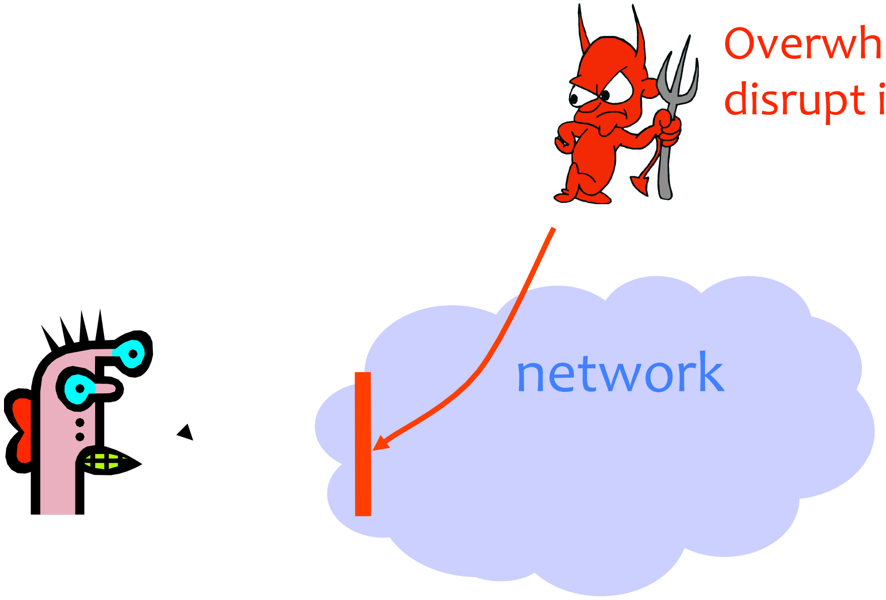
Integrity / Authenticity (2)

- Identification and assurance of origin of information



Availability

- Availability is ability to use information or resources desired



Overwhelm or crash servers, disrupt infrastructure

From Policy to Implementation

- After you've figured out what security means to your application, there are still challenges:
  - Requirements bugs
    - Incorrect or problematic goals
  - Design bugs
    - Poor use of cryptography
    - Poor sources of randomness
    - ...
  - Implementation bugs
    - Buffer overflow attacks
    - ...
  - Is the system **usable**?

Don't forget the users! They are a critical component!

Many Participants

- Many parties involved
  - System developers
  - Companies deploying the system
  - The end users
  - The adversaries (possibly one of the above)
- Different parties have different goals
  - System developers and companies may wish to optimize cost
  - End users may desire security, privacy, and usability
  - But the relationship between these goals is quite complex (will customers choose not to buy the product if it is not secure?)

Other (Mutually Related) Issues

- Do consumers actually care about security?
- Security is expensive to implement
- Plenty of legacy software
- Easier to write “insecure” code
- Some languages (like C) are unsafe

Approaches to Security

- Prevention
  - – Stop an attack
- Detection
  - – Detect an ongoing or past attack
- Response
  - – Respond to attacks

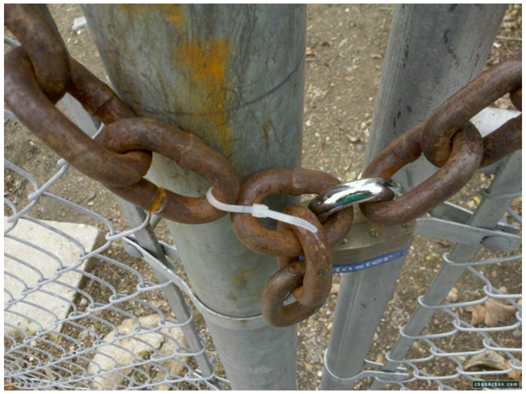
- The threat of a response may be enough to deter some attackers

Whole System is Critical

- Securing a system involves a whole-system view
  - Cryptography
  - Implementation
  - People
  - Physical security
  - Everything in between
- This is because "security is only as strong as the weakest link," and security can fail in many places
  - No reason to attack the strongest part of a system if you can walk right around it.
  - (Still important to strengthen more than the weakest link)

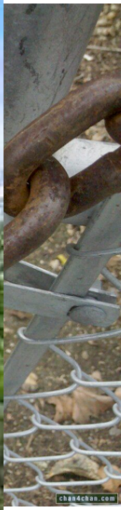
Whole System is Critical

- Securing a system involves a whole-system view
  - Cryptography
  - Implementation
  - People
  - Physical security
  - Everything in between



- This is because "security is only as strong as the weakest link," and security can fail in many places
  - No reason to attack the strongest part of a system if you can walk right around it.
  - (Still important to strengthen more than the weakest link)

Whole System is Critical



- Sec
  - C
  - I
  - P
  - P
  - E
- This
  wea s
  - N u can
    w
  - (nk)

Better News

- There are a lot of defense mechanisms
  - We'll study some, but by no means all, in this course
- It's important to understand their limitations
  - "If you think cryptography will solve your problem, then you don't understand cryptography… and you don't understand your problem" -- Bruce Schneier
  - Security is not a binary property
  - Many security holes are based on misunderstanding
- Security awareness and user "buy-in" help