

# CSE 484 / CSE M 584: Computer Security and Privacy

Spring 2021

Tadayoshi Kohno  
yoshi@cs

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

# Announcements

- Things Due:
  - Ethics form: Due next Monday (4/5)
  - Homework #1: Due next Thursday (4/8)
- Questions from last time:
  - Groups can span CSE 484 and CSE M 584
  - In-class assignments can be done up until the start of the next class (e.g., if you are in different time zone, participating remotely and watching Zoom recording)
- DRS:
  - Disability Resources for Students is looking for at least two notetakers to assist a student with a documented disability. Please consider notetaking as it will help your peer gain access to material from class. Notetakers will receive a letter of recommendation for their service from our office when the quarter is complete.
  - Email [drsnotes@uw.edu](mailto:drsnotes@uw.edu) if you are interested
- Any logistics questions at this point?

# THREAT MODELING

# Threat Modeling (Security Reviews)

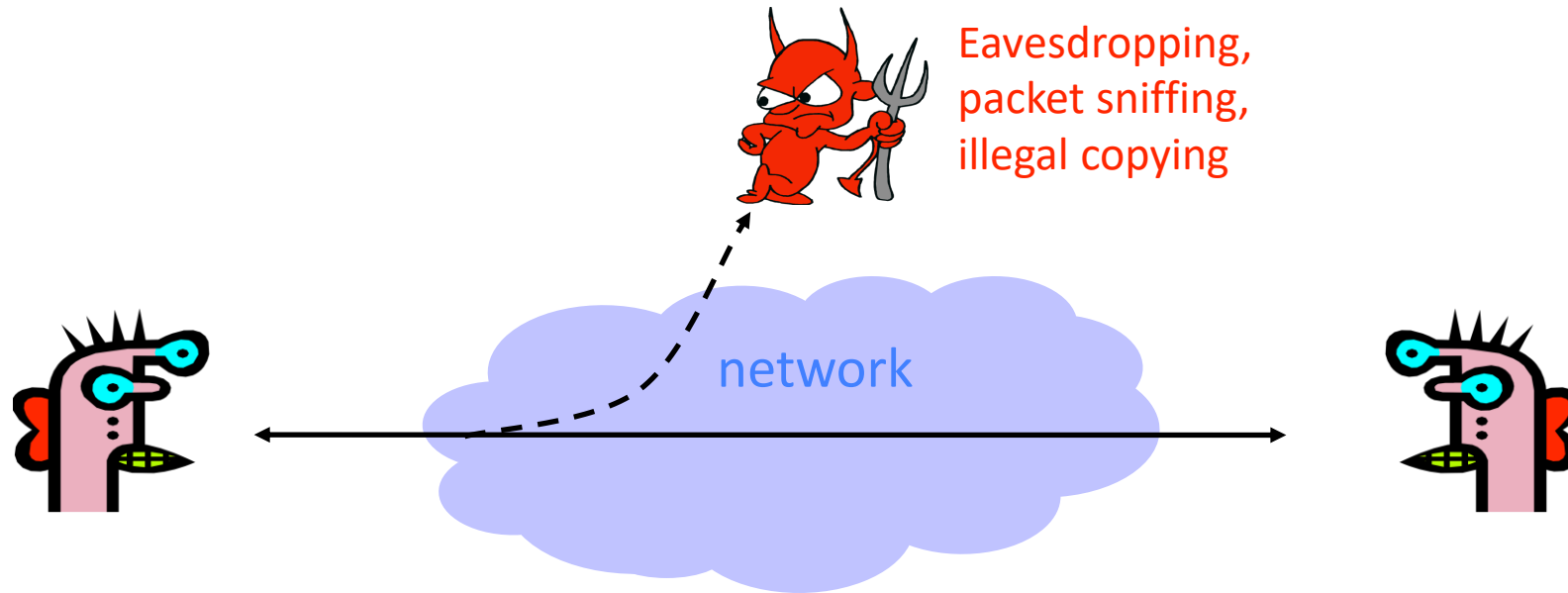
- **Assets**: What are we trying to protect? How valuable are those assets?
- **Adversaries**: Who might try to attack, and why?
- **Vulnerabilities**: How might the system be weak?
- **Threats**: What actions might an adversary take to exploit vulnerabilities?
- **Risk**: How important are assets? How likely is exploit?
- **Possible Defenses**
- Not “traditional” threat modeling, but important (both in general, and to help better understand the system prior to threat modeling):
  - **Benefits**: Who might the system benefit, and how?
  - **Harms**: Who might the system harm, and how?

# What's *Security*, Anyway?

- Common general security goals: “CIA”
  - Confidentiality
  - Integrity
  - Availability
- Or the extension: CPIAAU (Parkerian Hexad)
  - Control
  - Authenticity
  - Utility

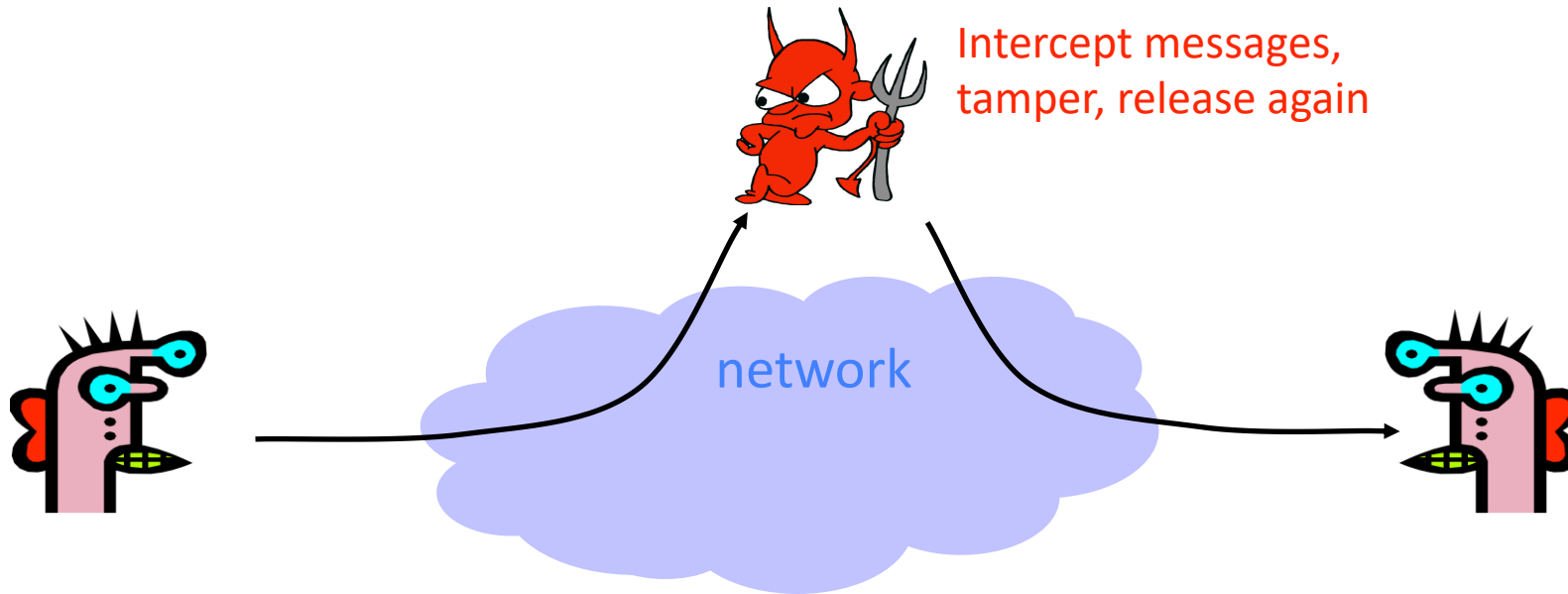
# Confidentiality (Privacy)

- Confidentiality is **concealment of information**.



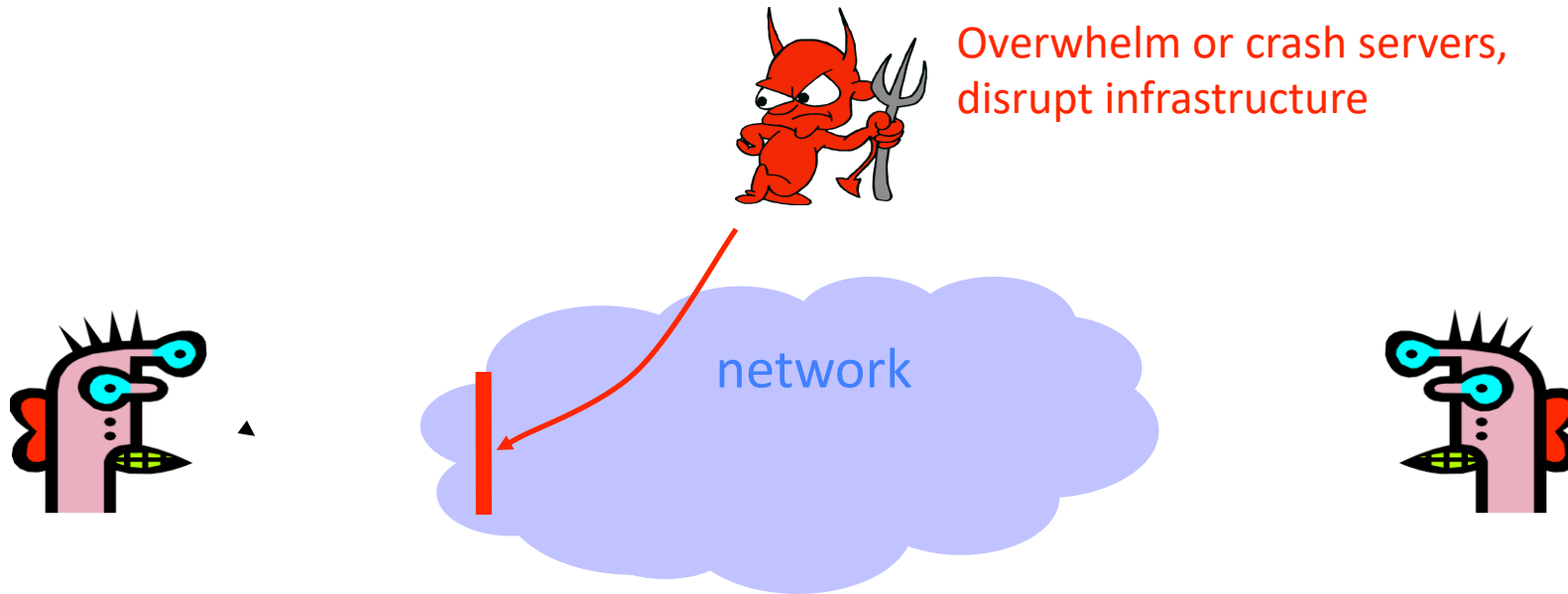
# Integrity

- Integrity is prevention of unauthorized changes.



# Availability

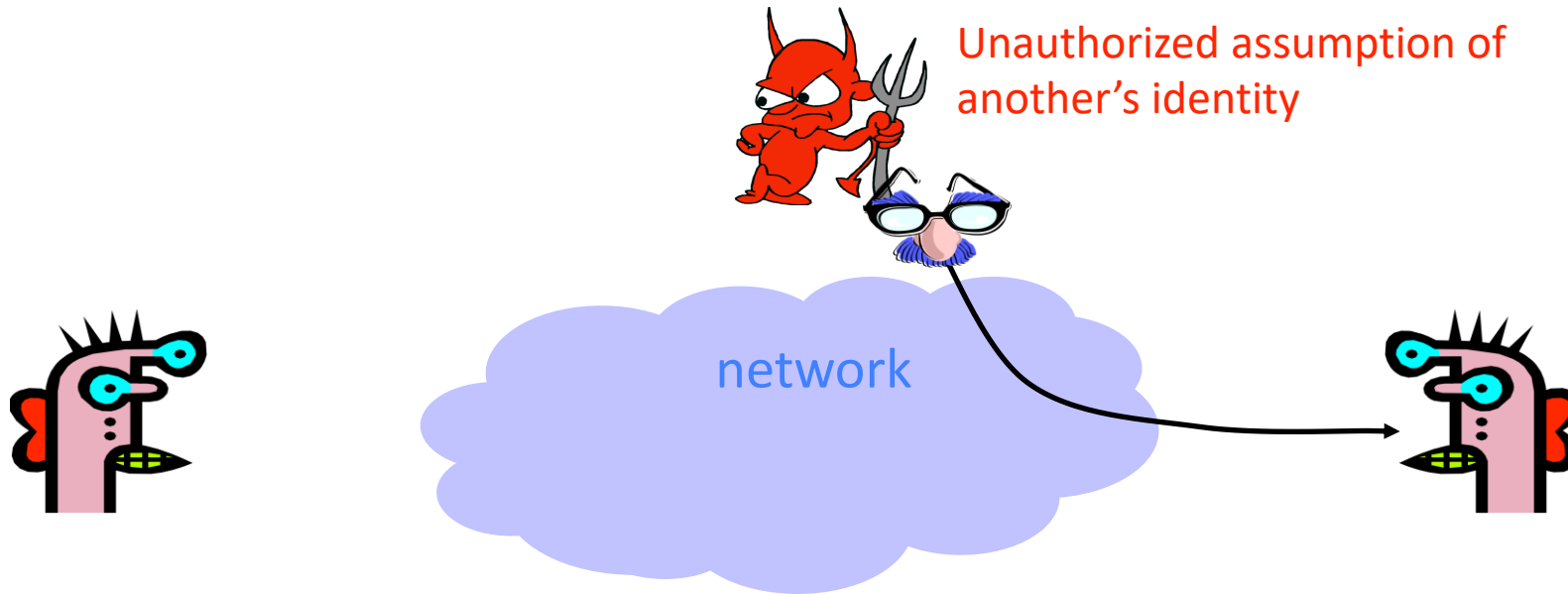
- Availability is ability to use information or resources.





# Authenticity

- Authenticity is **knowing who you're talking to.**

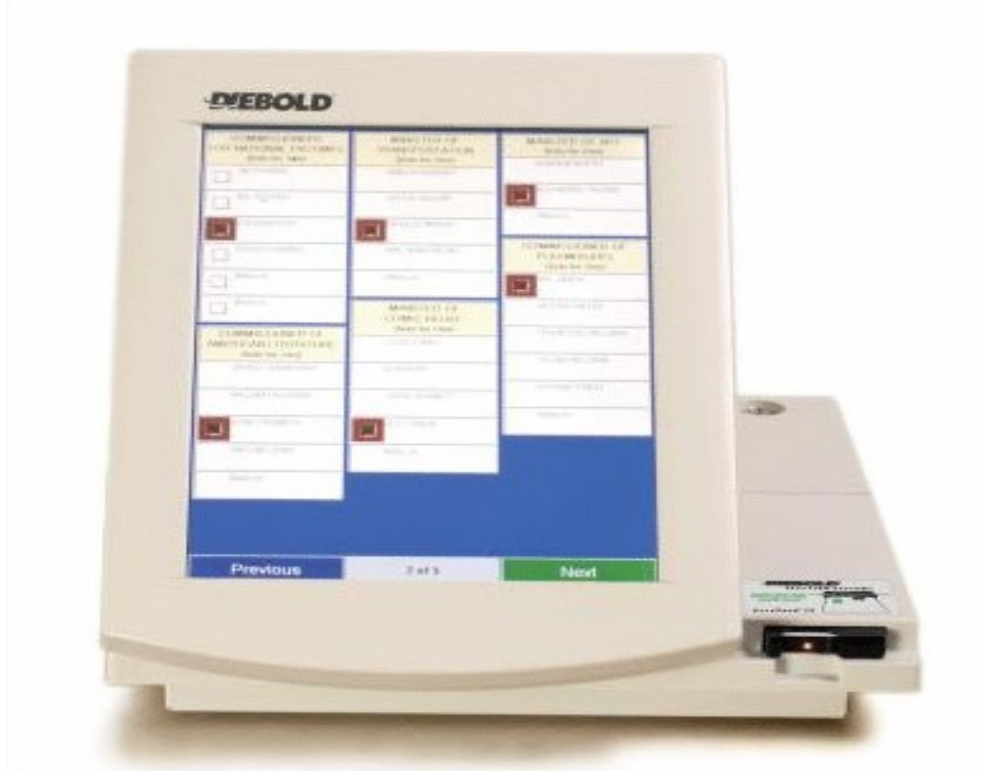


# Threat Modeling

- There's no such thing as perfect security
  - But, attackers have limited resources
  - **Make them pay unacceptable costs / take on unacceptable risks to succeed!**
- Defining security per context: identify assets, adversaries, motivations, threats, vulnerabilities, risk, possible defenses

# Threat Modeling Example: Electronic Voting

- Popular replacement to traditional paper ballots



# Pre-Election



Poll worker

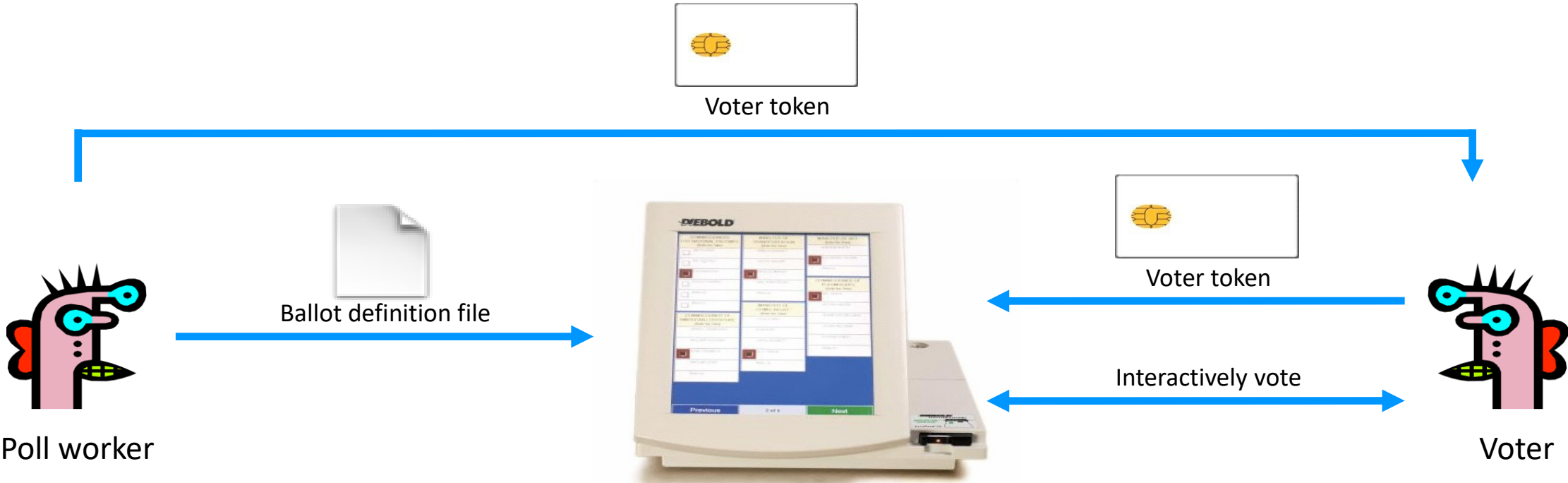


Ballot definition file



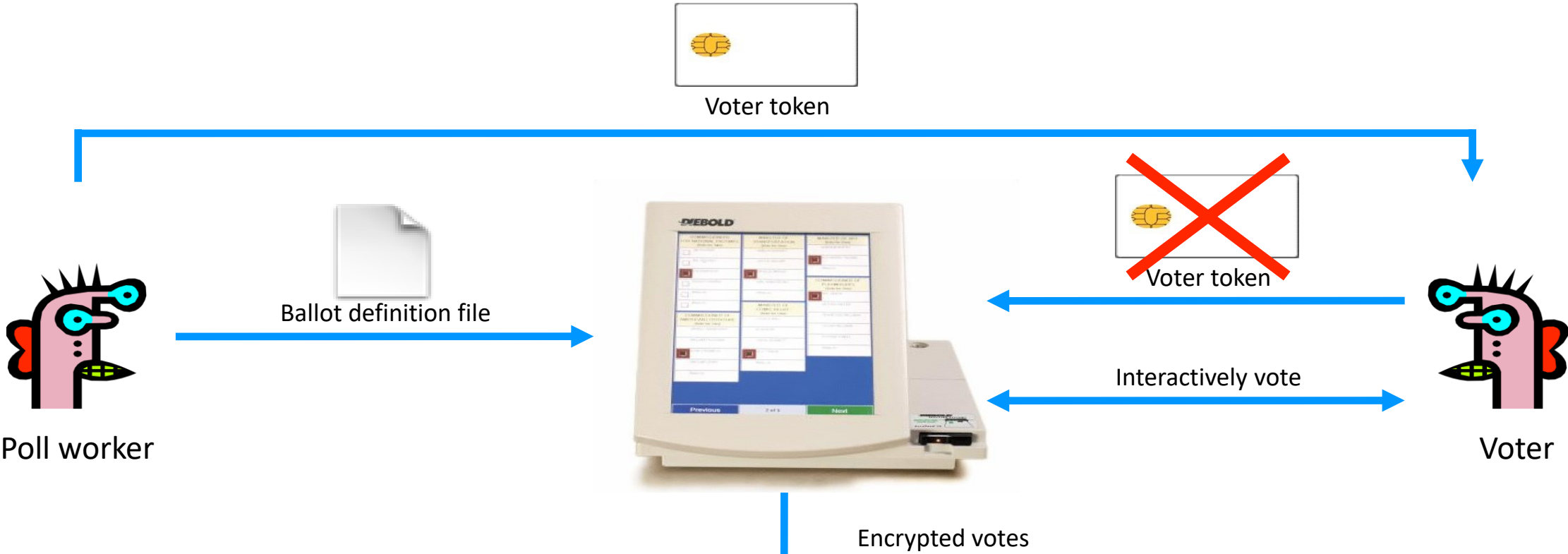
**Pre-election:** Poll workers load “ballot definition files” on voting machine.

# Active Voting



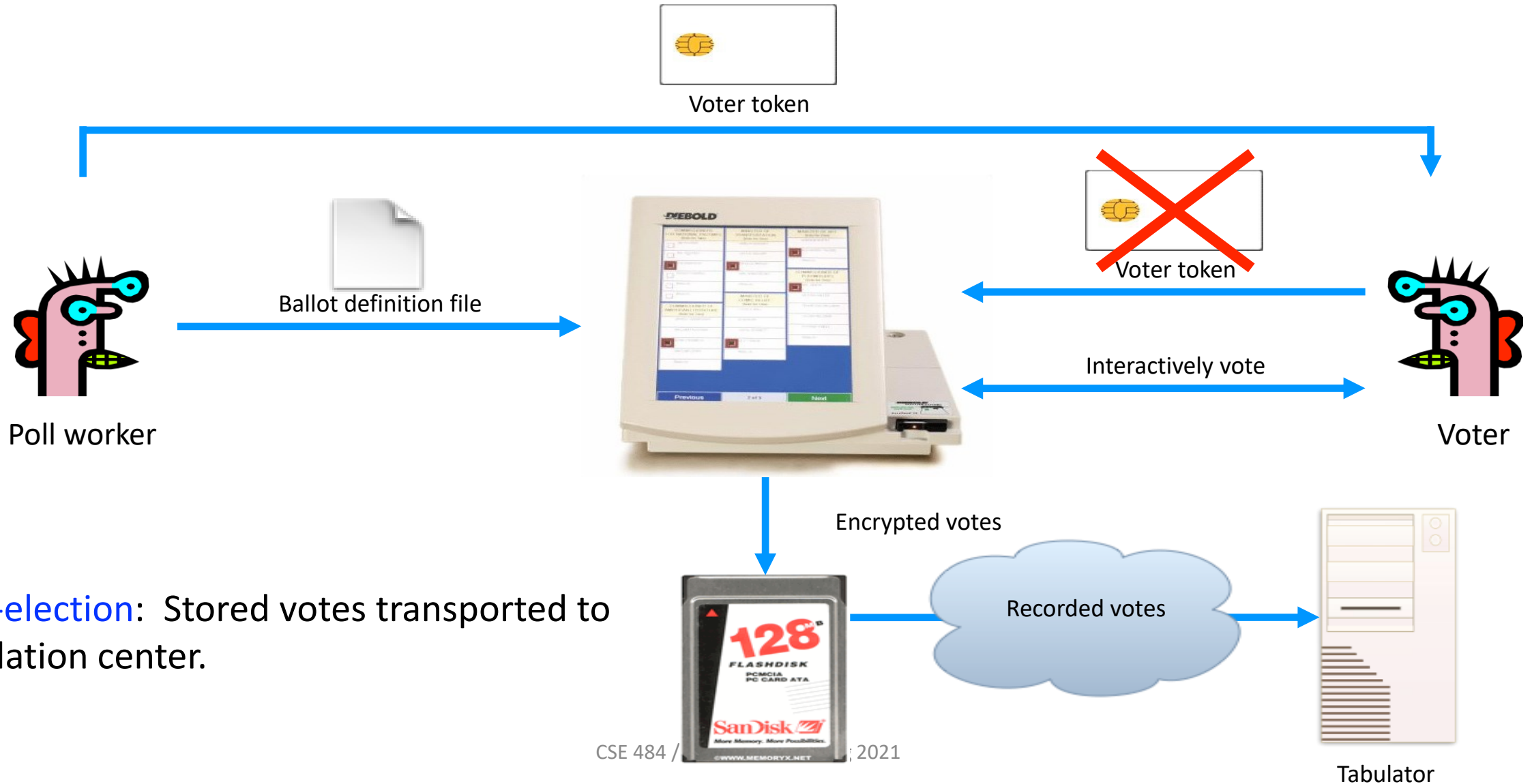
Active voting: Voters obtain **single-use** tokens from poll workers. Voters use tokens to **activate machines** and vote.

# Active Voting



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

# Post-Election



**Post-election:** Stored votes transported to tabulation center.

# In-Class “Worksheet” Experiment

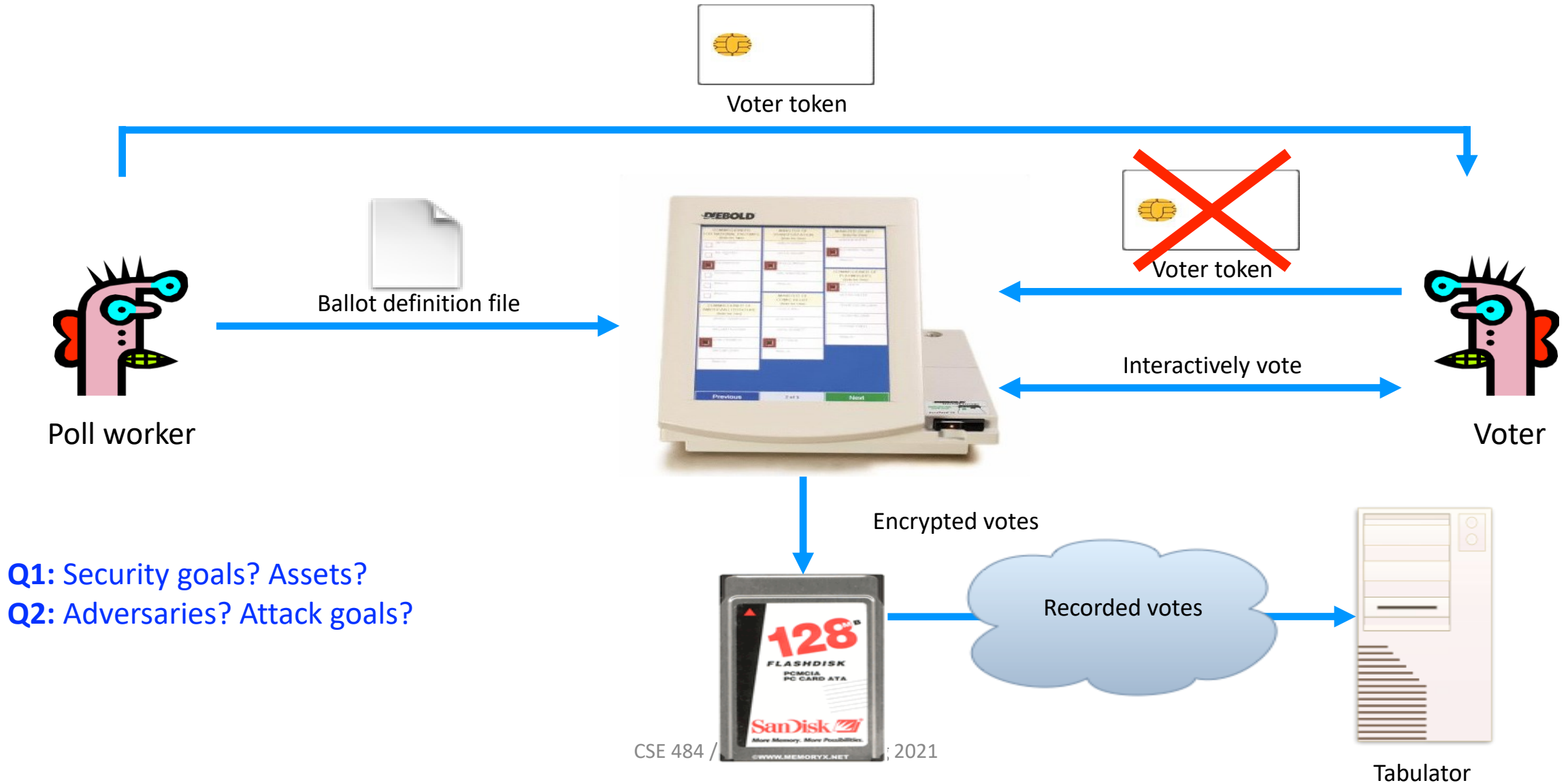
- Go to Canvas -> Quizzes -> “In-Class Activity – March 31”
- Fill out the questions while discussing with your breakout group
  - Everyone should submit their own
  - **No need for polish or complete sentences** – jot things down as you would on a piece of paper while chatting in class



# Security and E-Voting (Simplified)

- Functionality goals:
  - Easy to use, reduce mistakes/confusion, make voting more accessible
- Security goals:

# Can You Spot Any Potential Issues?



provent cheat  
protect ballots voter confidence  
toekns who they are voting for  
identity concealment single voting  
**one token per voter** vote  
voter authentication  
**voter privacy** voter's information  
reduce voter fraud validity of votes  
Fairness  
accuracy **integrity of the votes**  
**protect voter identity**  
privacy vote count protect vote  
fair election **votes** voting machine  
results tokens  
availability  
**confidentiality** election integrity  
authenticity **integrity** anonymity  
only vote once accurate votes  
**integrity of vote count**  
token forgery **voter identity** **Voter token**  
availability of voting  
**integrity of votes** voter authenticity  
no changing of cast votes no repeated vote  
confidentiality of votes one person one vote  
the data for voters might  
personal informations

intercept votes to change  
prevent people voting  
foreign governments organized crime  
foreign countries troublemakers  
political groups voter fraud politicians  
sway votes Russia undermine legitimacy  
opposite party  
voter token authenticity  
trolls foreign states trump supporters  
change vote candidates hackers  
**Foreign nations**  
political parties corporations  
Lobbyists political party  
parties  
change outcome physical security  
poll worker  
nation state terrorists Poll workers  
other countries rigged election  
change votes political affiliate party  
Foreign government  
change election outcome

# 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 they wanted.

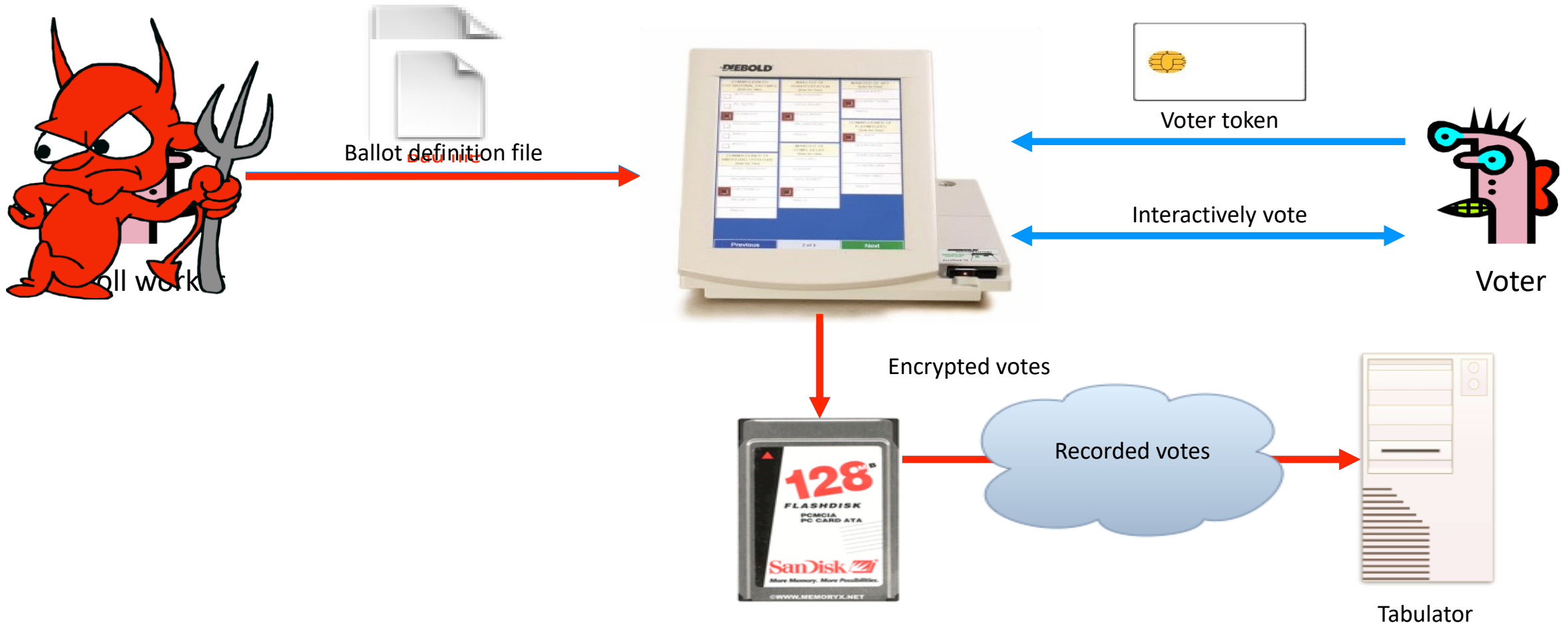


### **KEYS TO THE KINGDOM**

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.

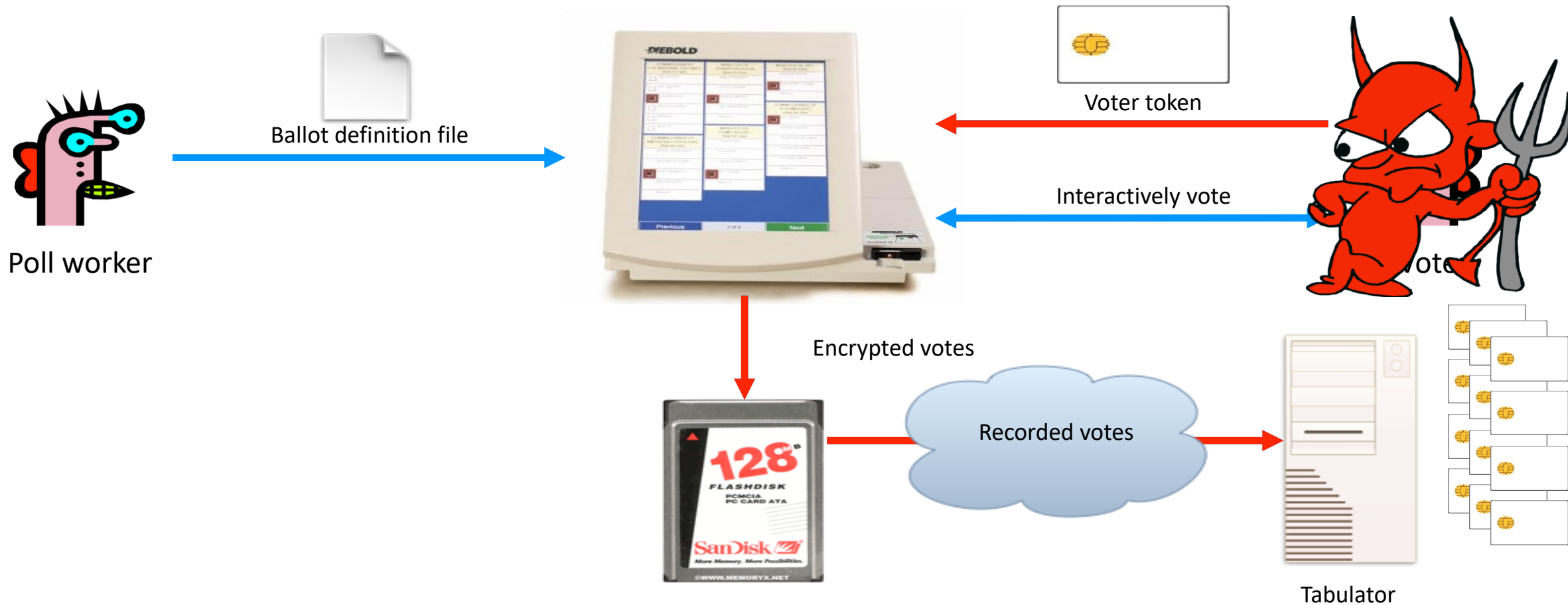
Problem: Ballot definition files are not authenticated.

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.

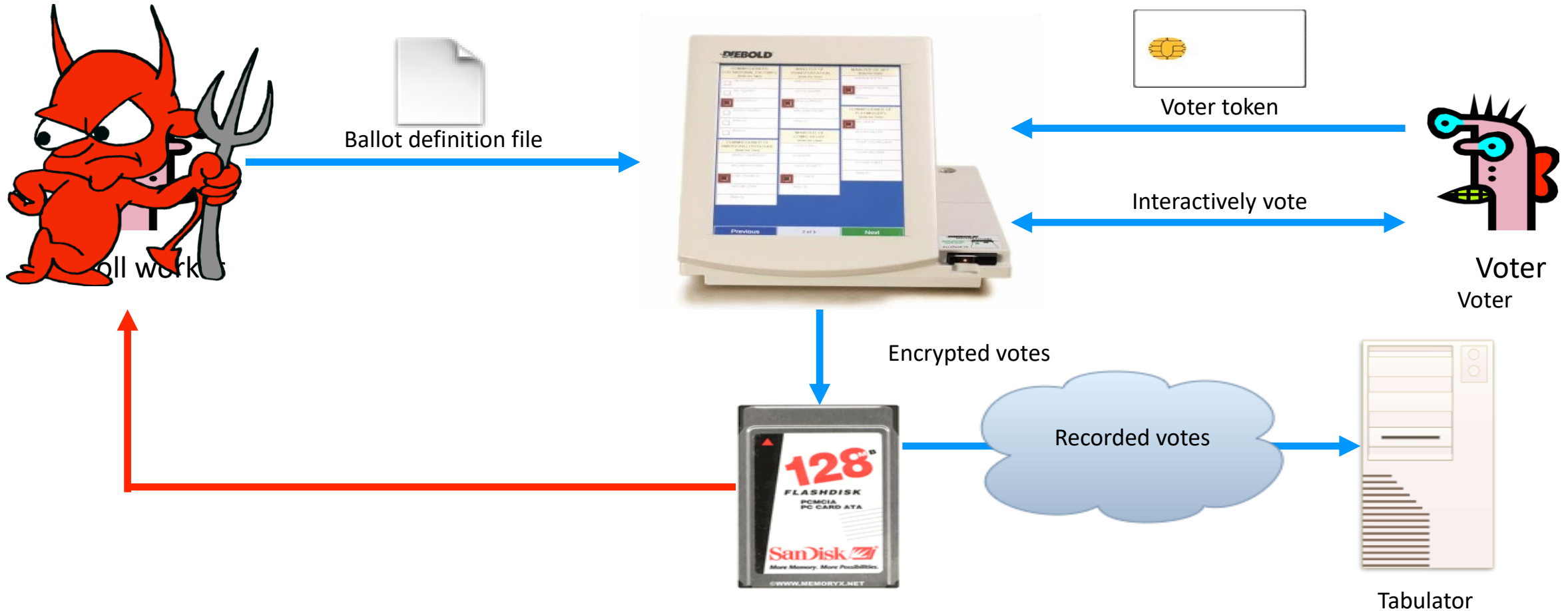
**Example attack:** A regular voter could make their own voter token and **vote multiple times**.





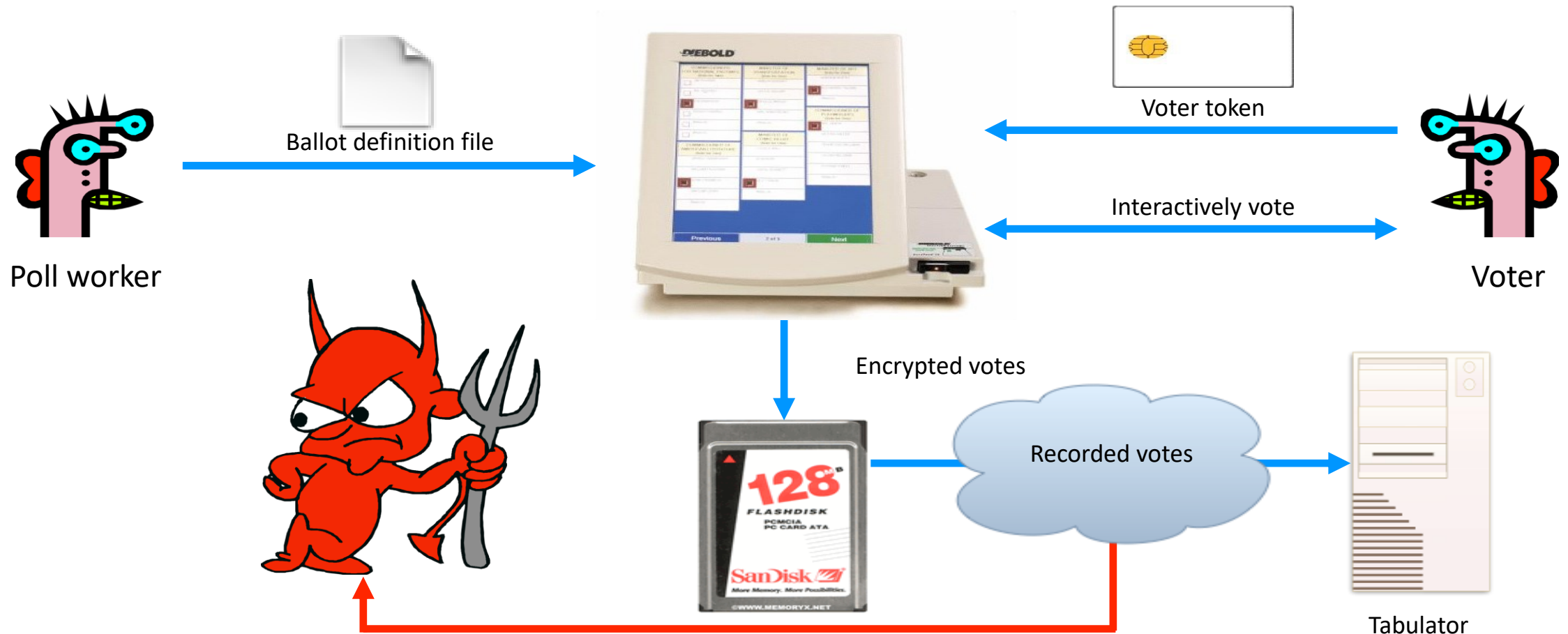
**Problem:** Encryption key (“F2654hD4”) hard-coded into the software since (at least) 1998. Votes stored in the order cast.

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



**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.



# TOWARDS DEFENSES

# Approaches to Security

- Prevention
  - Stop an attack
- Detection
  - Detect an ongoing or past attack
- Response and Resilience
  - Respond to / recover from 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.

# Whole System is Critical

- Securing a system involves a **whole-s**
  - 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.

# Whole System is Critical

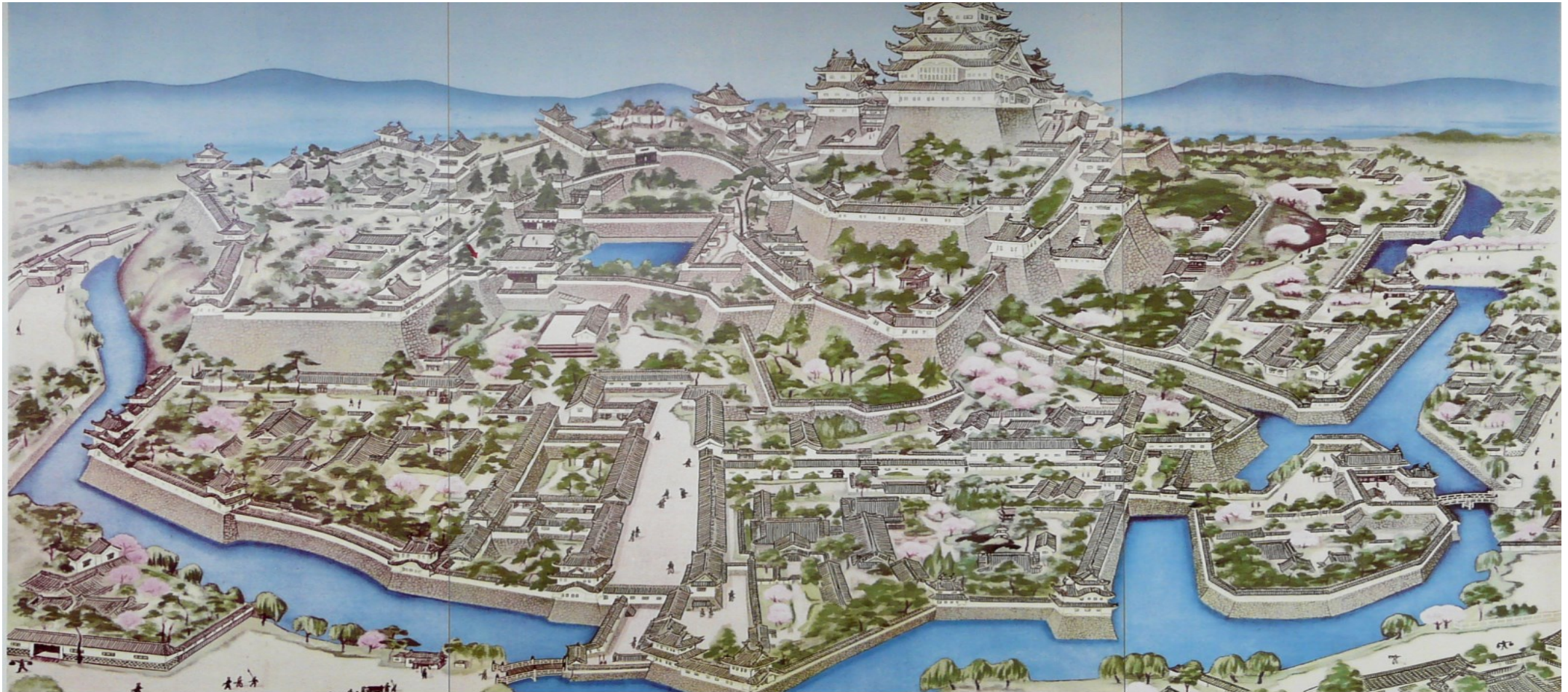
**Security Fail**

[failblog.org](http://failblog.org)





# Attacker's Asymmetric Advantage





# Attacker's Asymmetric Advantage



- Attacker only needs to win in one place
- Defender's response: **Defense in depth**

# From Policy to Implementation

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

# 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
    - Side question: Do system developers / companies really understand the needs and values of all their users? Or all stakeholders who might be impacted by the system?
  - But the relationship between these goals is quite complex (e.g., will customers choose features or security?)

# 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

