CSE 484 / CSE M 584 (Spring 2012)

Computer Security and Privacy

Tadayoshi Kohno

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

Administrivia

- Reminder: Ethics form before Wednesday
- Guest lecture on Friday
- Lab 1 out next week

- Assigned Reading: Daswani et al, Chapter 1.
- Assigned Video: Long, No Tech Hacking: http://www.youtube.com/watch?v=5CWrzVJYLWw

Alexei Czeskis

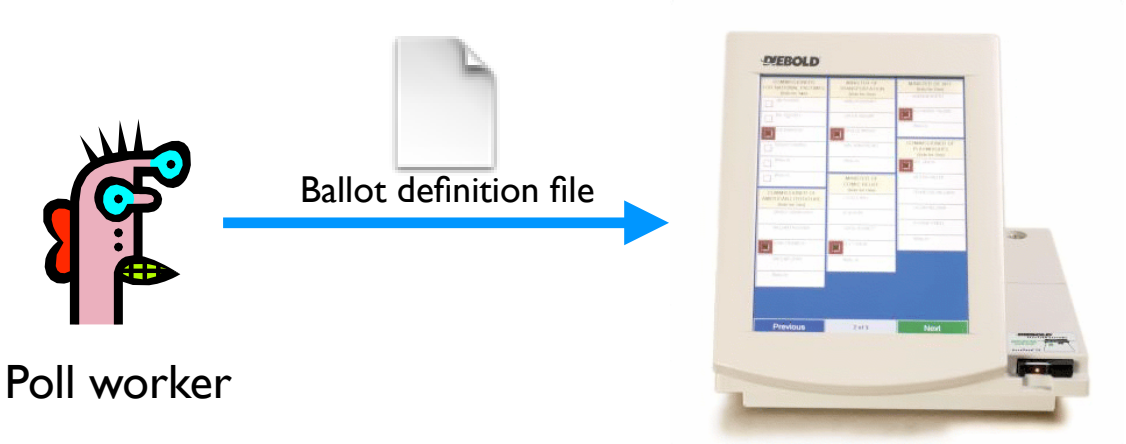
- aczeskis@cs.washington.edu

Example: Electronic Voting

- Popular replacement to traditional paper ballots

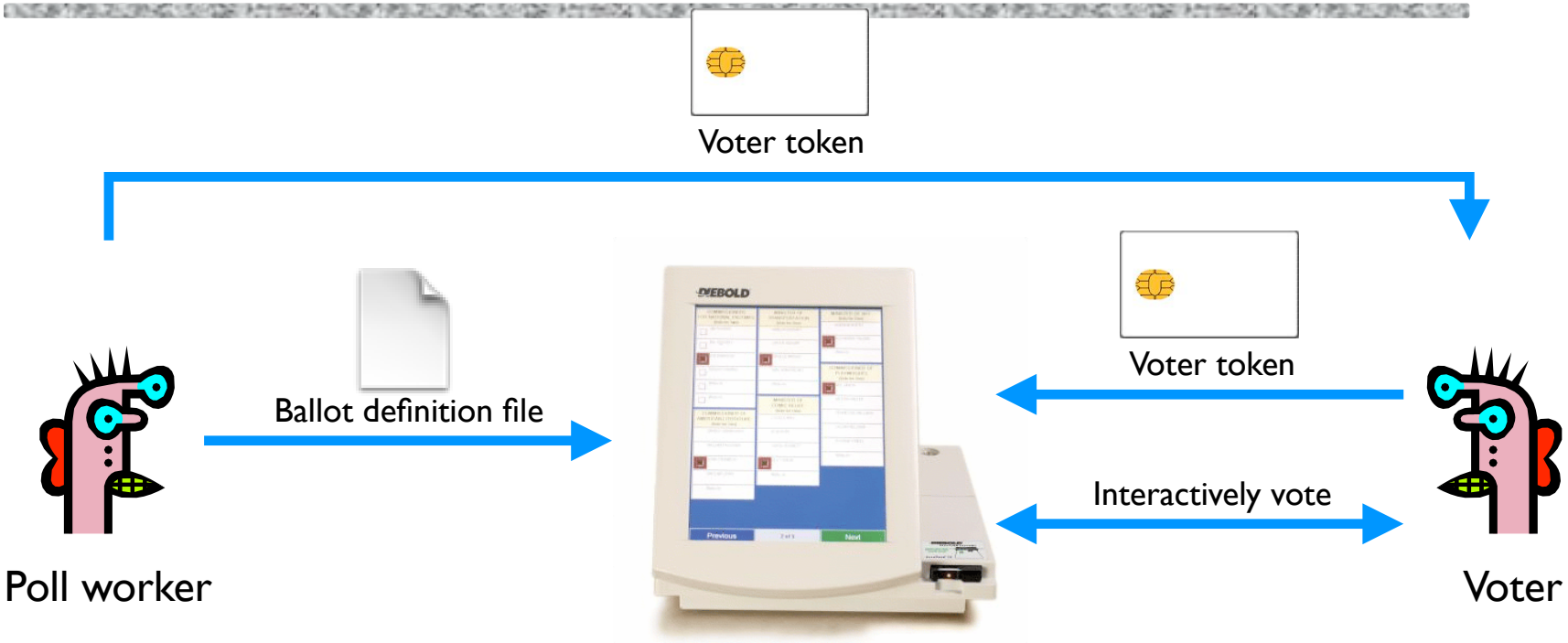
AP

Pre-Election



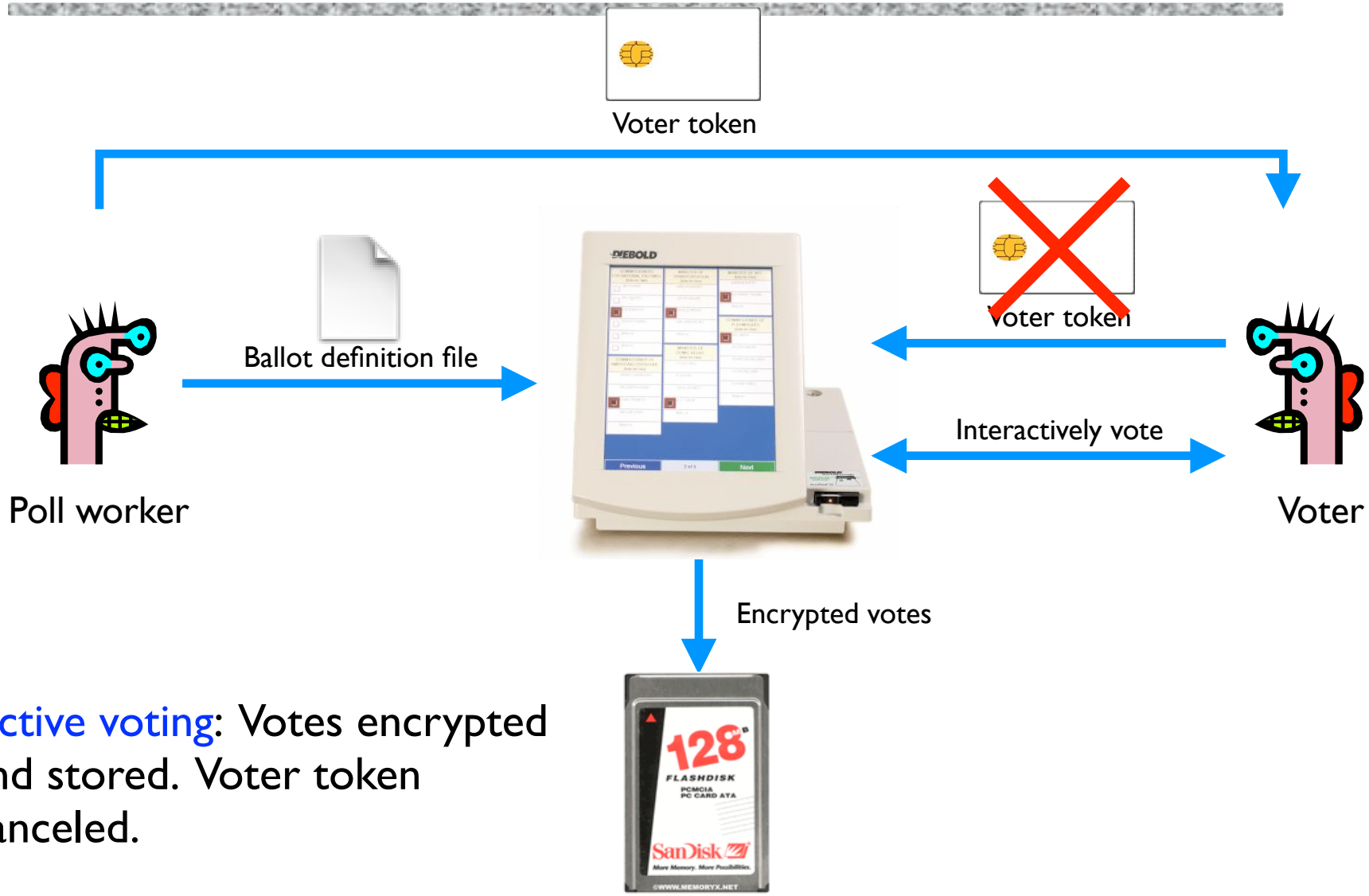
Pre-election: Poll workers load "ballot definition files" on voting machine.

Active Voting



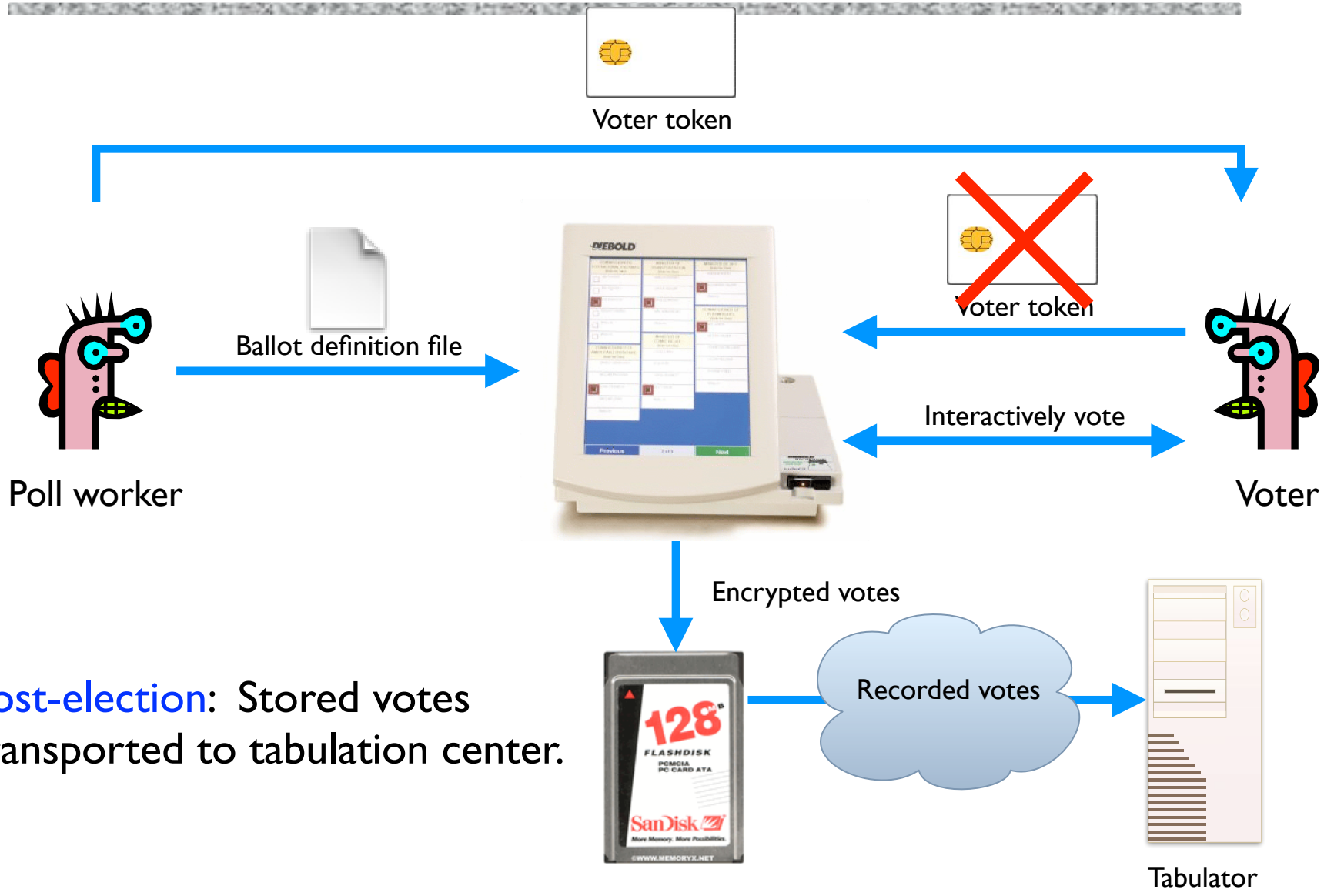
Active voting: Voters obtain single-use tokens from poll workers. Voters use tokens to active machines and vote.

Active Voting



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

Post-Election



Post-election: Stored votes transported to tabulation center.

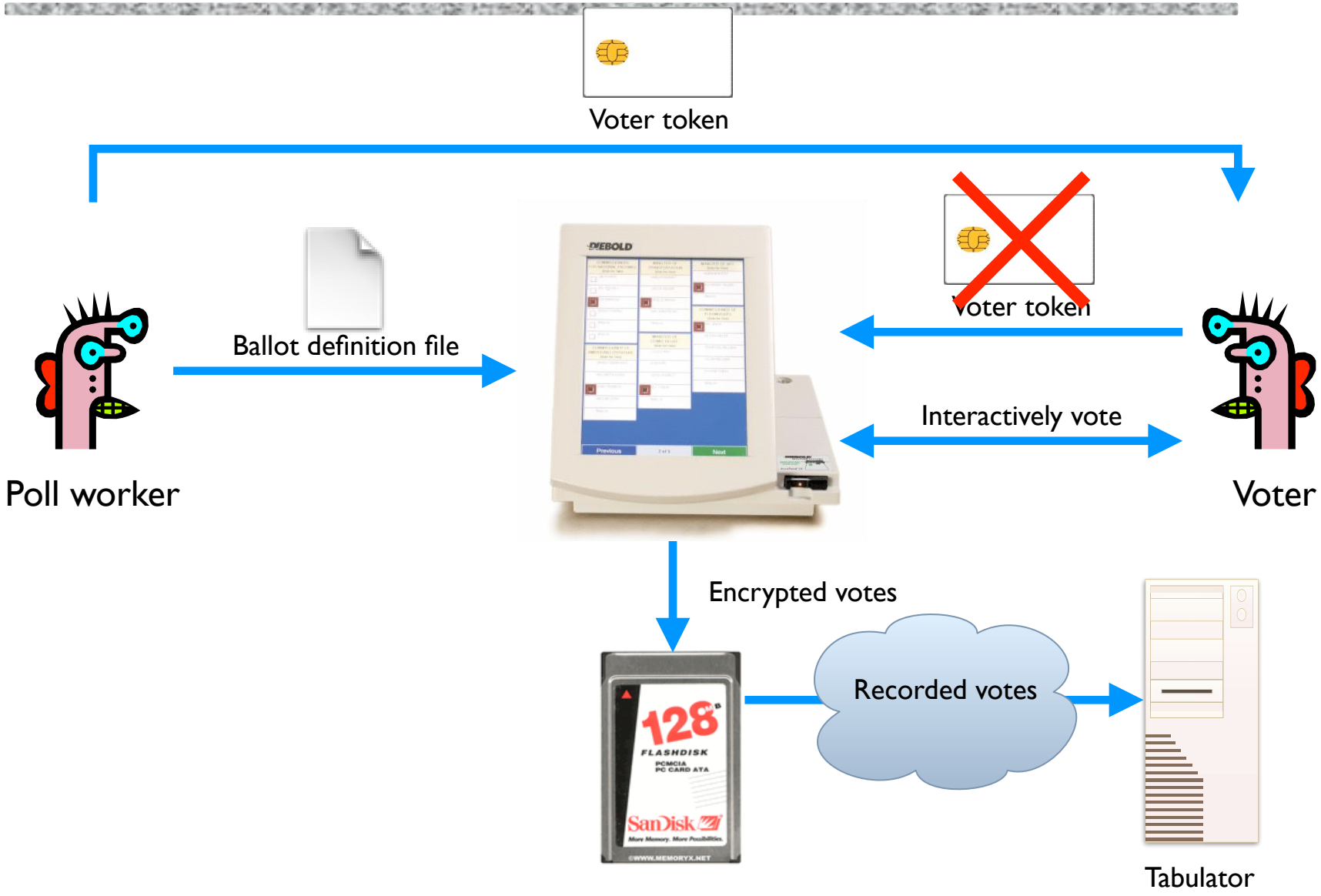
Security and E-Voting (Simplified)

- Functionality goals:
  - Easy to use
  - People should be able to cast votes easily, in their own language or with headphones for accessibility

Security 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
  - Is it OK if an adversary can do the above, assuming you can catch him or her or them?
  - Adversary should not be able to figure out how voters vote

Can You Spot Any Potential Issues?



Potential Adversaries

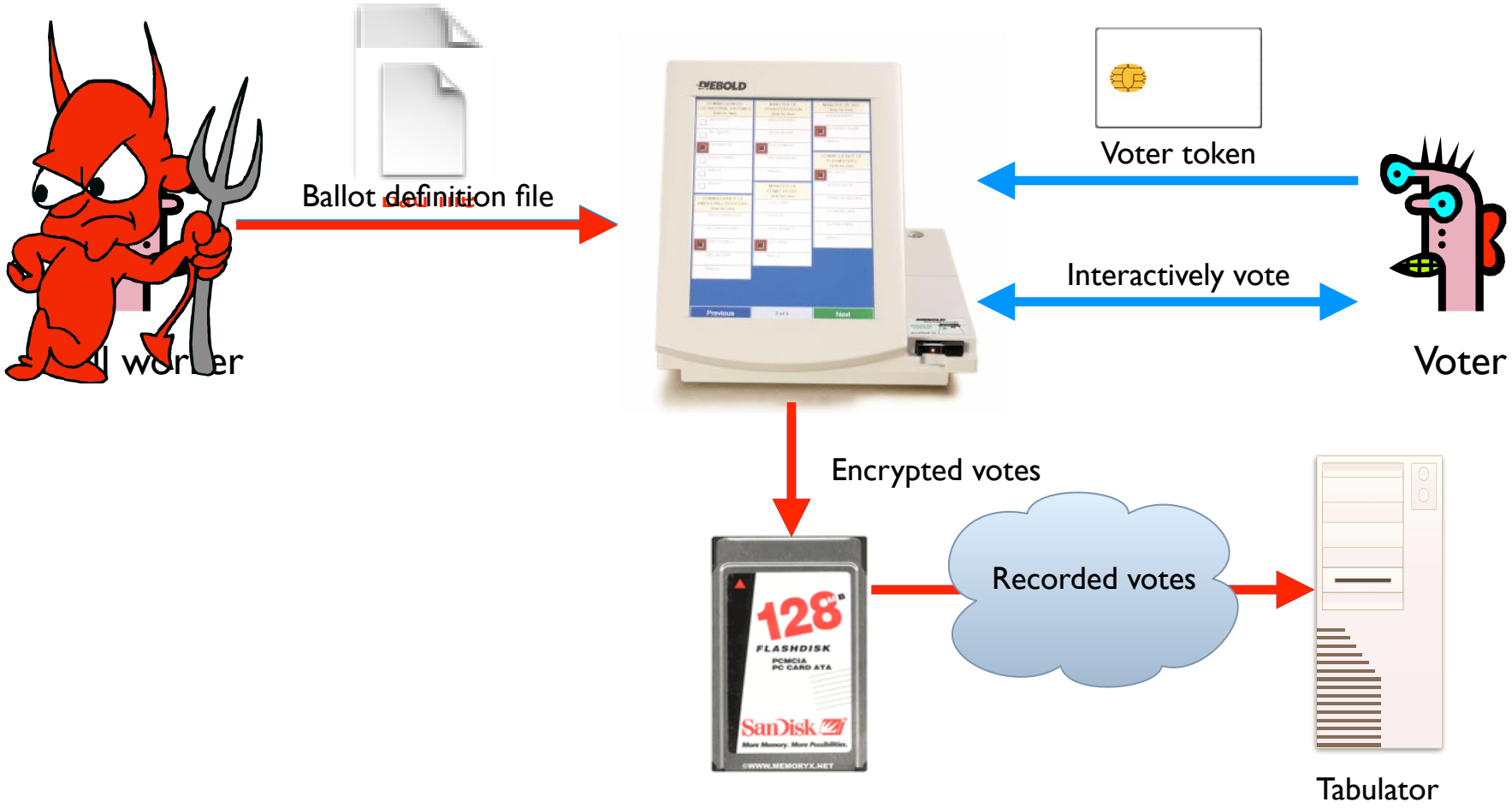
- Voters
- Election officials
- Employees of voting machine manufacturer
  - Software/hardware engineers
  - Maintenance people
- Other engineers
  - Makers of hardware
  - Makers of underlying software or add-on components
  - Makers of compiler
- ...
- Or any combination of the above

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.

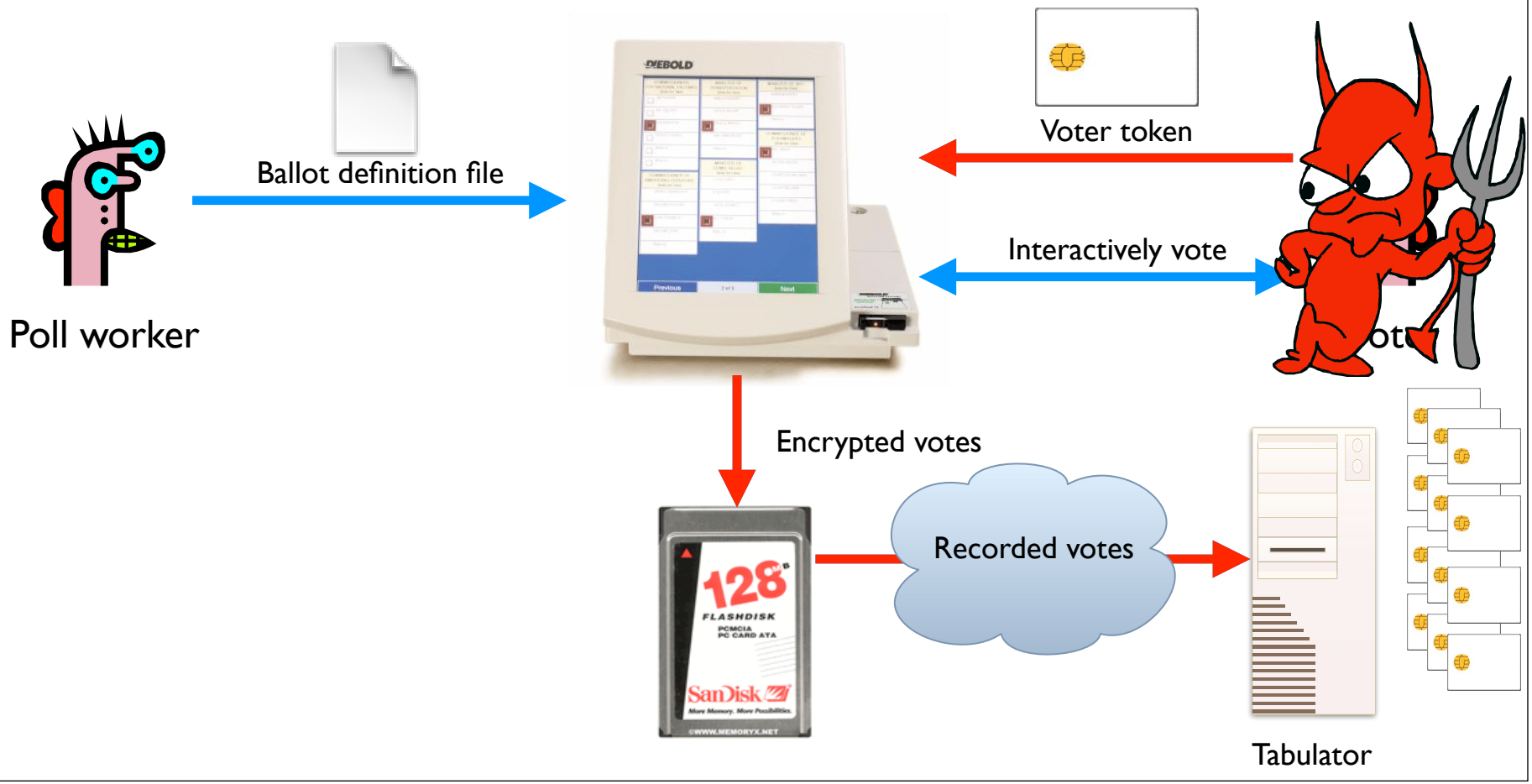
Problem: Ballot definition files are not authenticated.

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



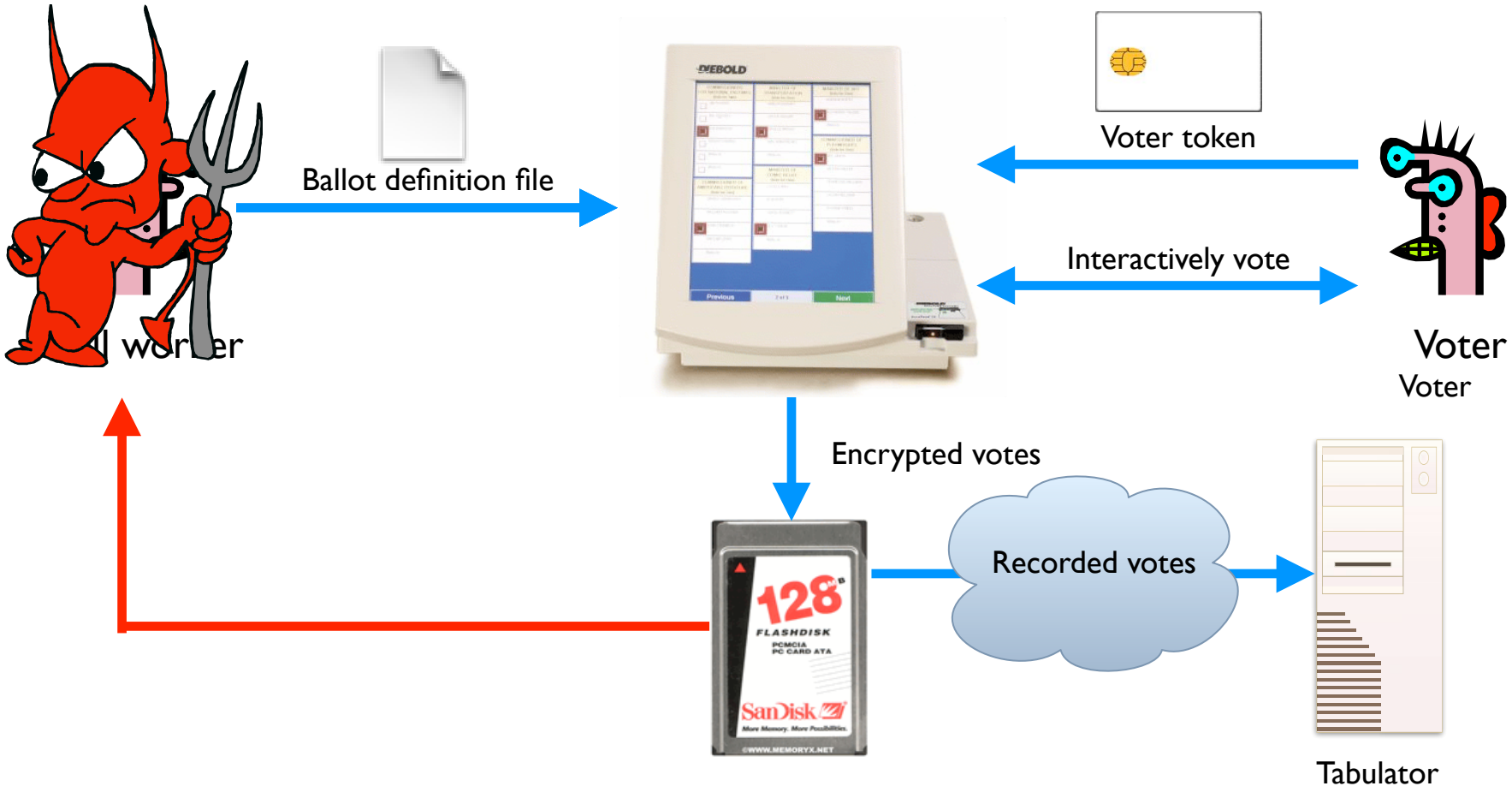
Problem: Smartcards can perform cryptographic operations. But there is no authentication from voter token to terminal.

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



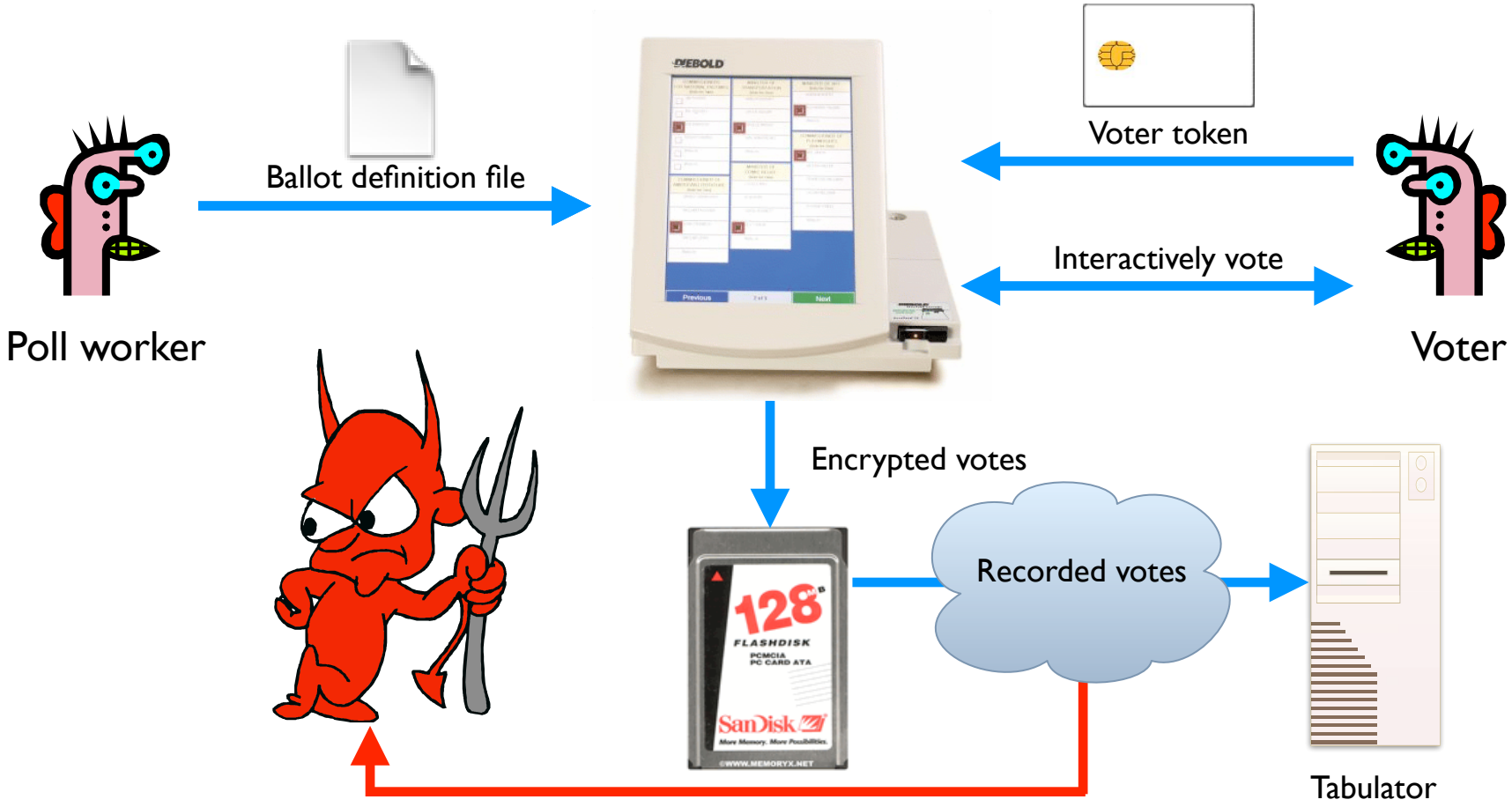
Problem: Encryption key ("F2654hD4") hard-coded into the software since (at least) 1998. Votes stored in the order cast.

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

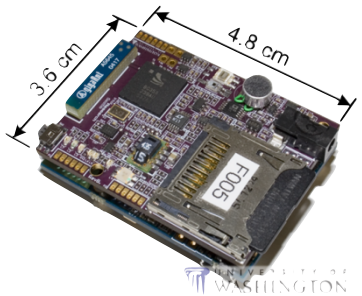


Problem: When votes transmitted to tabulator over the Internet or a dialup connection, they are decrypted first; the cleartext results are sent the the tabulator.

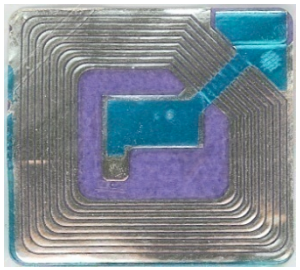
Example attack: A sophisticated outsider could determine how voters vote.



Security not just for PCs



mobile sensing platforms



RFID

EEG Gaming

large displays

ambient displays

smart phones

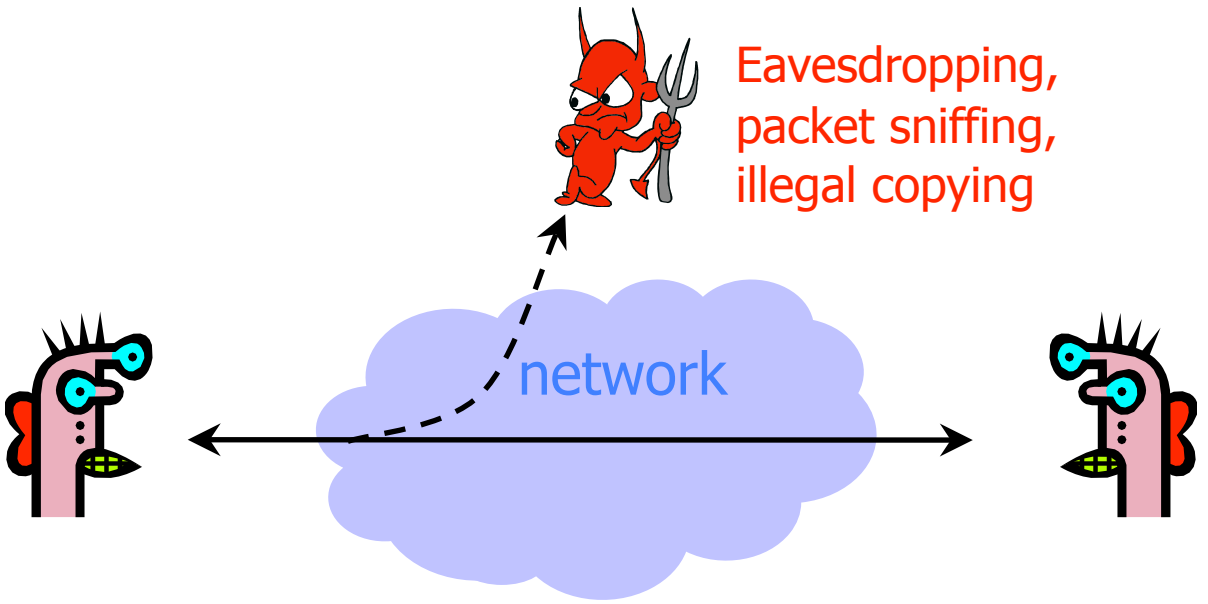
wearables

health displays

Security Goals

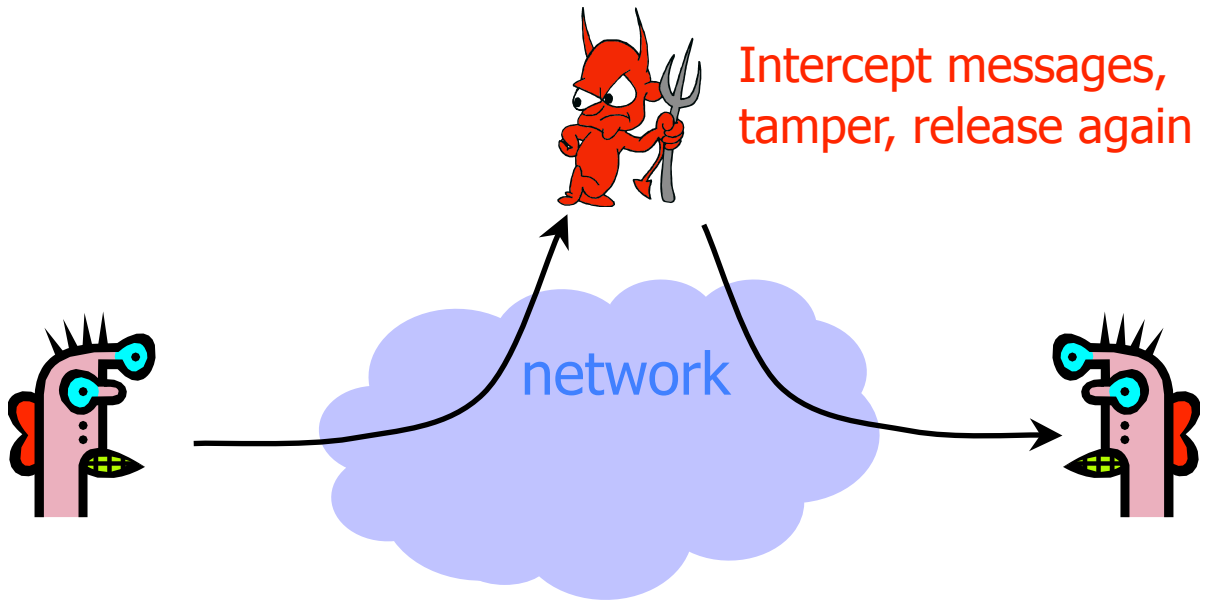
Confidentiality (Privacy)

- Confidentiality is concealment of information



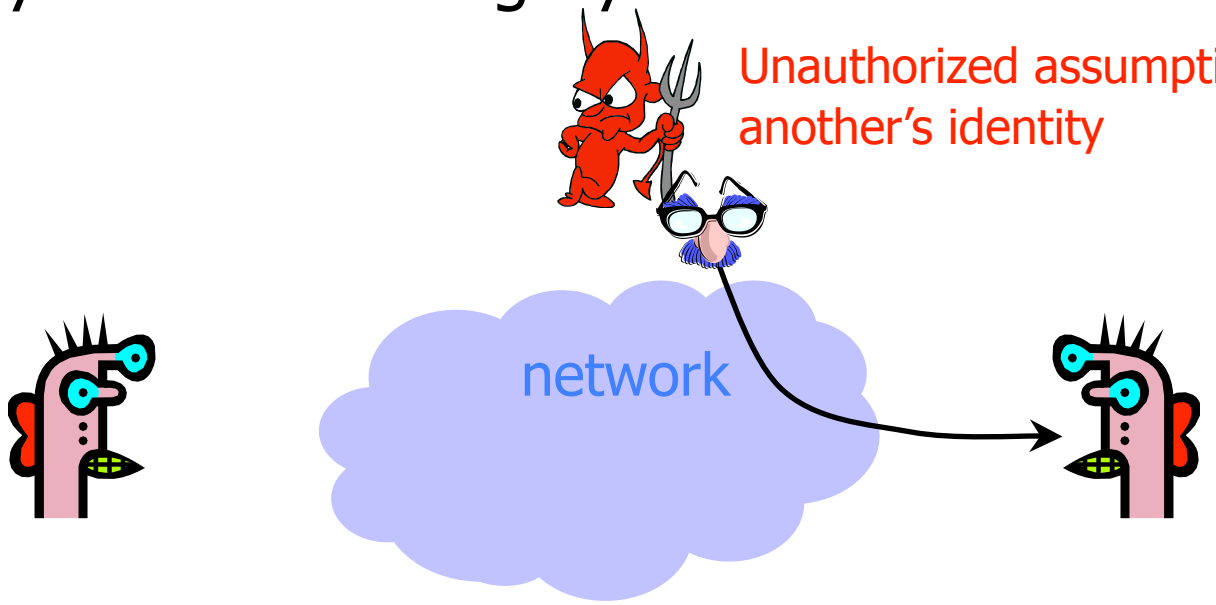
Integrity

- Integrity is prevention of unauthorized changes



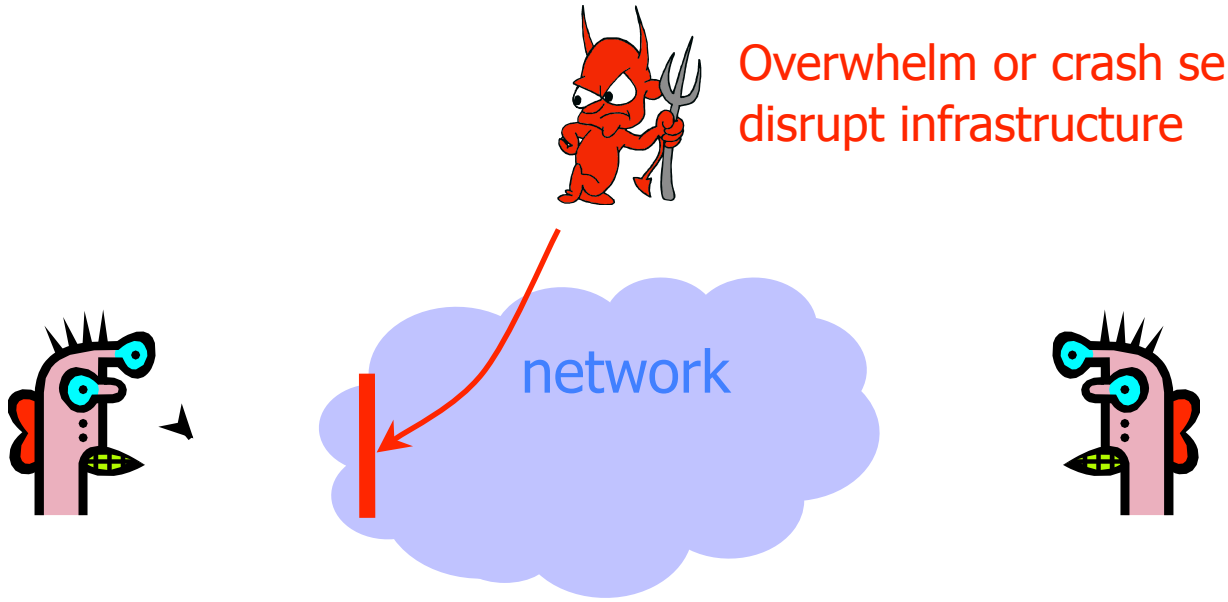
Authenticity

- Authenticity is identification and assurance of origin of information
- Highly related to integrity



Availability

- Availability is ability to use information or resources desired



Security of a system

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)

Analyzing the Security of a System

- First thing: Summarize the system as clearly and concisely as possible
  - Critical step. If you can't summarize the system clearly and concisely, how can you analyze it's security?
  - Summary can be hierarchical
- Next steps:
  - Identify the assets: What do you wish to protect?
  - Identify the adversaries
  - Identify the threats
  - Identify vulnerabilities: Weaknesses in the system
  - Calculate the risks

Assets

- Need to know what you are protecting!
  - Data and information: Data for running and planning your business, design documents, data about your customers, data about your identity
  - Reputation, brand name
  - Responsiveness
  - Personal safety
  - Hardware: Laptops, servers, routers, PDAs, phones, ...
  - Software: Applications, operating systems, database systems, source code, object code, ...
- Assets should have an associated value (e.g., cost to replace hardware, cost to reputation, how important to business operation)

Adversaries

- National governments
- Organized crime
- Terrorists
- Thieves
- Business competitors
- Your supplier
- Your consumer
- The New York Times
- Your family members (parents, children)
- Your friends
- Your ex-friends
- ...

Threats

- Threats are actions by adversaries who try to exploit vulnerabilities to damage assets
  - Spoofing identities: Attacker pretends to be someone else
  - Tampering with data: Change outcome of election
  - Crash machines: Attacker makes voting machines unavailable on election day
  - Elevation of privilege: Regular voter becomes admin
- Specific threats depend on environmental conditions, enforcement mechanisms, etc
  - You must have a clear, simple, accurate understanding of how the system works!

Threats

- Several ways to classify threats
  - By damage done to the assets
    - Confidentiality, Integrity, Availability
  - By the source of attacks
    - (Type of) insider
    - (Type of) outsider
    - Local attacker
    - Remote attacker
    - Attacker resources
  - By the actions
    - Interception
    - Interruption
    - Modification
    - Fabrication

Vulnerabilities

- Weaknesses of a system that could be exploited to cause damage
  - Accounts with system privileges where the default password has not been changed (Diebold: 1111)
  - Programs with unnecessary privileges
  - Programs with implementation flaws
  - Problems with cryptography
  - Weak firewall configurations that allow access to vulnerable services
  - ...
- Sources for vulnerability updates: CERT, SANS, Bugtraq, the news, ...

Risks Analyses: Lots of Options

Risk Exposure Risk Impact

- Quantitative risk analysis

Probability

  - Example: Risk = Asset × Threat × Vulnerability
  - Monetary value to assets
  - Threats and vulnerabilities are probabilities
  - (Yes: Difficult to assign these costs and probabilities)
- Qualitative risk analysis
  - Assets: Critical, very important, important, not important
  - Vulnerabilities: Very likely, likely, unlikely, very unlikely
  - Threats: Very likely, likely, unlikely, very unlikely

Helpful Tables

Asset	Confidentiality	Integrity	Availability
Hardware			
Software			
Data			
Personal Safety			
…			

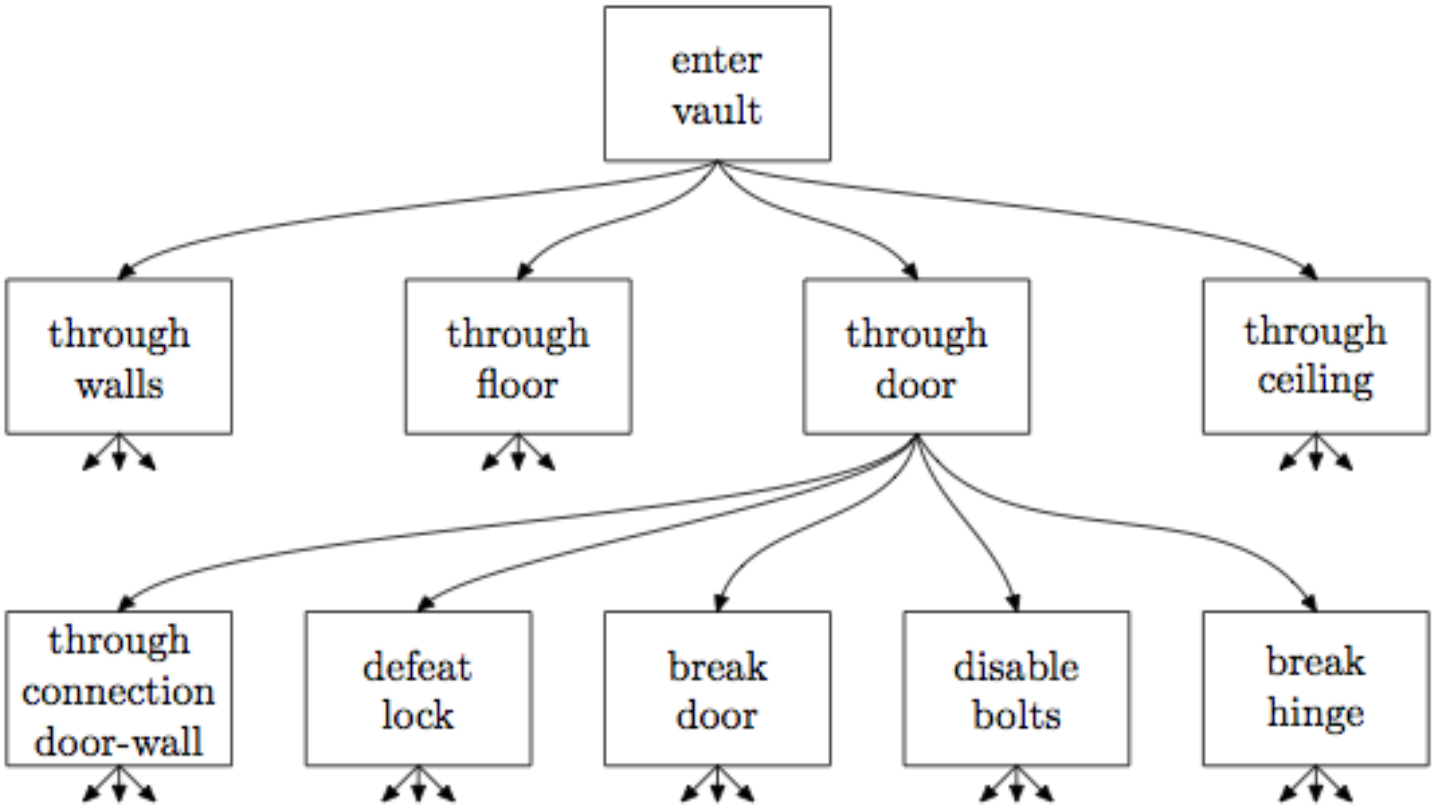
Helpful Tables

	Voter	Election official	...
Privacy of vote			
Integrity of vote			
Availability of voting system			
Confidence in election			
...			

Helpful Tables

	Create New Voter Cards	Decrypt voting record	...
Privacy of vote			
Integrity of vote			
Availability of voting system			
Confidence in election			
...			

Attack Trees



Security is Subtle

- Security attacks can be subtle
- Can't provably and accurately identify / quantify all risks, vulnerabilities, threats.
- So need to think careful!
  - And keep the whole system in mind
- Phishing one example
  - If attacker can trick user into entering private information, then no protection mechanism will help
  - (So research tries to focus on helping users not be tricked)

On Modularity and Complexity

- Modular design may increase vulnerability
  - Abstraction is difficult to achieve in security: what if the adversary operates below your level of abstraction?
- Modular design may increase security: small TCB (trusted computing base)
- Complexity may increase vulnerability

Not So Great News

- Security may not be a primary consideration
  - Performance and usability take precedence
- Feature-rich systems are hard to understand
  - Higher-level protocols make mistaken assumptions
- Implementations can be buggy
  - Buffer overflows, XSS vulnerabilities, ...
- Networks can be left open and accessible
  - Increased exposure, easier to cover tracks
- No matter what technical mechanisms a system has, people may circumvent them
  - Phishing, impersonation, write down passwords, ...
- Attackers may be very powerful
  - ISPs, governments, ...
    - –

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

Course and Assignments

Tentative Syllabus

- Thinking about security; the "big picture"
  - The hardest part: Getting the "security mindset"
- Software security (including buffer overflow attacks)
- Web security (including XSS attacks)
- Cryptography
- Network security
- Botnets and malware
- The users (including usability)
- Anonymity

Field broad. All parts interconnected, so we will "bounce" around in a methodical way

Forum

- Help you develop the "security mindset"
- Best way to learn a foreign language: move to that country and immerse yourself in the language.
- Same thing applies to "security thinking"
- Forum: opportunity to think about security on a regular basis -- outside of class
  - Current events
  - New product announcements
  - Security in your everyday life

Current Events

- Important for computer security practitioners (and all computer scientists) to be able to
  - Reflect on the broader context of technology
  - Guide future development of technology
  - Guide future policy
- For the course blog
  - Summarize current event
  - Discuss why event arose
  - Reflect on what could have been done prior to the event arising (to prevent, deter, or change consequences)
  - Describe broader issues surrounding current event (ethical, societal)
  - How should people respond to the event (policy makers, the public, companies, etc.)

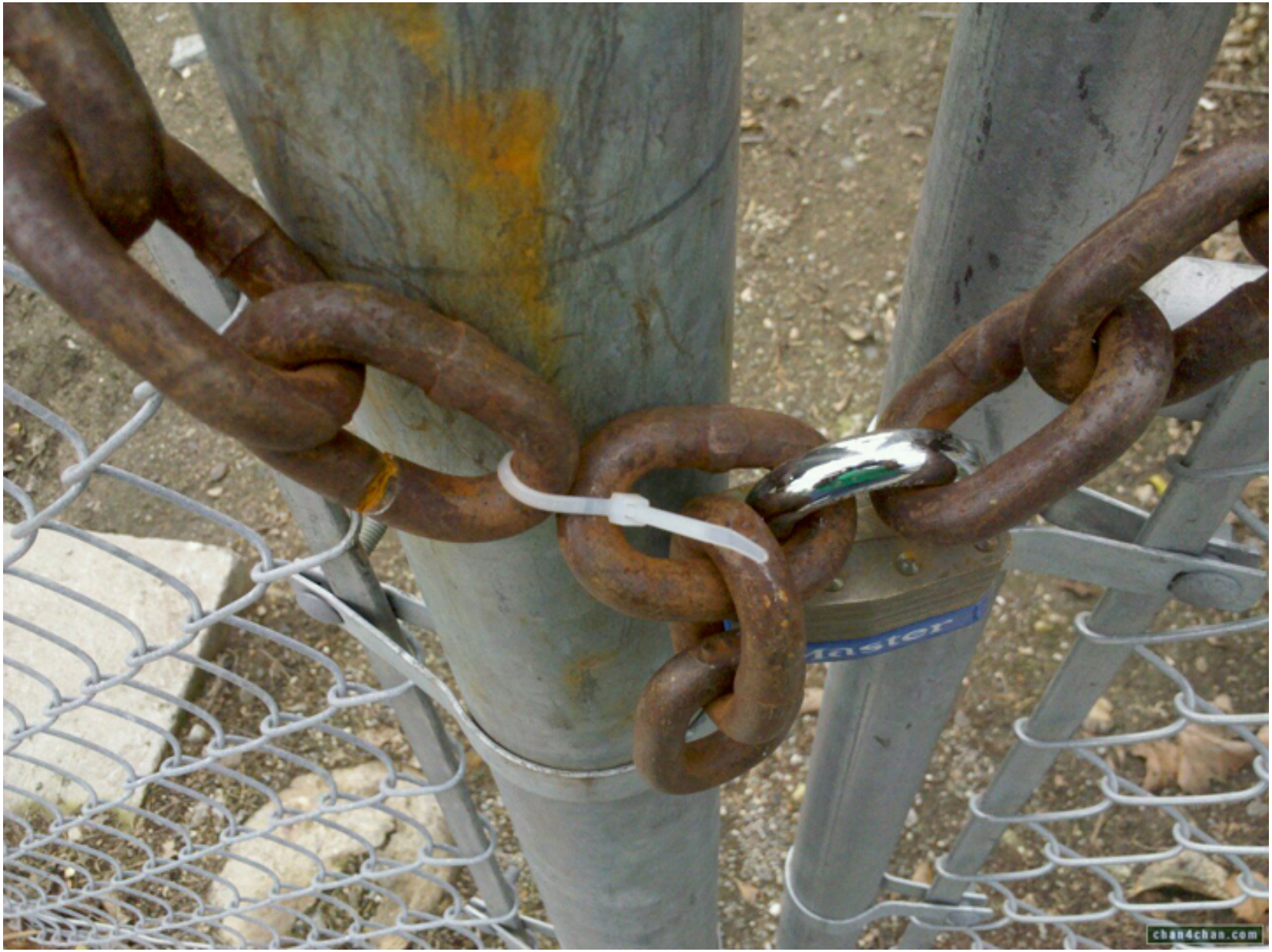
Security Reviews

- Summary of system/product
- Assets
- Adversaries
- Threats
- Potential weaknesses (OK to speculate, but make it clear that you are speculating)
- Potential defenses
- Risks
- Conclusions
- Important: Also has a Catalyst dropbox

Security in your life

- Take and share security-related photos and stories and obsevations (anecdotes, videos, audio, etc.) on the forum
- Explain what you were capturing and how it relates to security

- *__Stay within legal limits__*---for instance, Washington State is a "2-Party State", which means you can't record communications without both sides' consent/notification.
  (All-party for multi-way communications)


Master
chan4chan.com