CSE 484 / CSE M 584 (Winter 2013)

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 ...

High-level information

- Instructor:
  - Tadayoshi Kohno (Yoshi)
- TAs:
  - Franzi Roesner, Daseul Lee, Ian Smith
- Course website
  - Assignments, reading materials, ...
- Course email list
  - Announcements
- Course forums
  - Discussion

High-level information

- Recitation sections:
  - Thursday, 1:30-2:20 and 2:30-3:20 (EEB 125)
- Yoshi’s, Franzi’s, Daseul’s, and Ian’s office hours:
  - TBD

Prerequisites (CSE 484)

- Required: Data Structures (CSE 326) or Data Abstractions (CSE 332)
- Required: Hardware/Software Interface (CSE 351) or Machine Org and Assembly Language (CSE 378)
- Assume: Working knowledge of C and assembly
  - One of the labs will involve writing buffer overflow attacks in C
  - You must have detailed understanding of x86 architecture, stack layout, calling conventions, etc.
- Assume: Working knowledge of software engineering tools for Unix environments (gdb, etc)
- Assume: Working knowledge of Java and JavaScript

Prerequisites (CSE 484)

- Strongly recommended: Computer Networks; Operating Systems
  - Will help provide deeper understanding of security mechanisms and where they fit in the big picture
- Recommended: Complexity Theory; Discrete Math; Algorithms
  - Will help with the more theoretical aspects of this course.

Prerequisites (CSE 484)

- Most of all: Eagerness to learn!
  - This is a 400 level course.
  - We expect you to push yourself to learn as much as possible.
  - We expect you to be a strong, independent learner capable of learning new concepts from the lectures, the readings, and on your own.

Course Logistics (CSE 484)

- Lectures: Mon, Wed: 2:30-3:50pm ;
  Sections: Thurs: 1:30-2:20pm and 2:30-3:20pm
- Security is a contact sport!
- Labs (45% of the grade)
  - Labs involve a lot of programming
  - Can generally be done in teams of 3 students (see specific lab descriptions for details)
- Homework (30% of grade)
- Final (25% of the grade)
- Participation (up to 10% Extra Credit)

Exceptional work may be rewarded with extra credit

No make-up or substitute exams!
If you are not sure you will be able to take the exam on the assigned date and time, **do not take this course!**

Course Logistics (CSE M 584)

- Same as before, but...
- Labs (40% of the grade) [-5%]
- Homework (25% of grade) [-5%]
- Final (25% of the grade)
- Research readings (10%) [+10%]
  - Read research papers (~1 paper per week)
- Participation (up to 10% Extra Credit)

Late Submission Policy

- Late assignments will (generally) be dropped 20% per calendar day.
  - Late days will be rounded up
  - So an assignment turned in 26 hours late will be downgraded 40%
  - See website for exceptions -- some assignments must be turned in on time
- Many assignments due on Friday

Participation Grade (Extra Credit)

- Main source of extra credit:
  - Regular contributions to class forums
    - (You can pick a pseudonym, though course staff will still know who owns each pseudonym)
  - (Don’t be silent for 9 weeks and then make 10 posts on the last day of the quarter)
- Also an option (but harder in a large class):
  - Participation in class
    - I would like to learn everyone’s name!
    - But ~90 students may overflow my buffer, without some form of assistance
    - I’m toying with the idea of name cards or a seating chart -- and will make a decision about that now that I’ve seen the classroom layout.

Course Materials

- <u>Textbook</u>:
  - Daswani, Kern, Kesavan, "Foundations of Security"
  - Additional materials linked to from course website
- Attend lectures.
  - Lectures will <u>not</u> follow the textbook
  - Lectures will focus on "big-picture" principles and ideas
  - Lectures will cover some material that is <u>not</u> in the textbook – and you will be tested on it!
  - Much of the crypto work will come from "Cryptography Engineering" (Ferguson et al), but you shouldn't need to buy the book (come to lectures)
  - (Also make sure to read the forum)

Other Helpful Books (online)

- Ross Anderson, “Security Engineering” (1st edition)
  - Focuses on design principles for secure systems
  - Wide range of entertaining examples: banking, nuclear command and control, burglar alarms
  - You should all at least look at the Table of Contents for this book.
  - (2nd edition available for purchase)
- Menezes, van Oorschot, and Vanstone, “Handbook of Applied Cryptography”
- Many many other useful books exist (not all online)

Others books, movies, ...

- Pleasure books include:
  - Little Brother by Cory Doctorow
    - Available online here http://craphound.com/littlebrother/download/
  - Cryptonomicon by Neal Stephenson
  - REAMDE by Neal Stephenson
  - Many more -- please feel free to post your favorites on the forum!
- Movies include:
  - Hackers
  - Sneakers
  - Die Hard 4
  - WarGames
  - Many more -- please feel free to post your favorites on the forum!
- Historical texts include:
  - The Codebreakers by David Kahn
  - The Code Book by Simon Singh

Ethics

- In this class you will learn about how to attack the security and privacy of (computer) systems.
- Knowing how to attack systems is a critical step toward knowing how to protect systems.
- But one must use this knowledge in an ethical manner.
- In order to get a non-zero grade in this course, you must electronically sign the “Security and Privacy Code of Ethics” form by Thursday, Jan 17.

Mailing List

- Make sure you’re on the mailing list
  - We’ll send a test mail after class; everyone enrolled should receive it
- URL for mailing list on course website:
  - http://www.cs.washington.edu/education/courses/cse484/13wi/administrivia/overview.html
- Used for announcements

Forum

- We’ve set up a forum for this course to discuss assignments
  - https://catalyst.uw.edu/gopost/board/kohno/31351/
- Please use it to discuss the homework assignments and labs and other general class materials

- We set up another forum for exercising the “security mindset”
  - https://catalyst.uw.edu/gopost/board/kohno/31352/
  - (Including discussions of movies, books, and security in the real world)

Labs

- General plan (tentative):
  - 3 labs (timeline TBD, most likely due on Fridays)
    - First lab out approximately next Wednesday
  - Submit to Catalyst system (URL on course page)
  - Groups of three generally allowed (check each project page for details)
- http://www.cs.washington.edu/education/courses/cse484/13wi/labs/

Labs (tentative plan)

- First lab: Software security
  - Buffer overflow attacks, double-free exploits, format string exploits, …
- Second lab: Web security
  - XSS attacks, …
- Third lab: TBD

Homework

- Approximately 4 homework assignments distributed across the quarter
  - http://www.cs.washington.edu/education/courses/cse484/13wi/homework/
- Do now: sign ethics form

What does “security” mean to you?

Two key themes of this course

- How to **think** about security
  - The Security Mindset - “new” way to think about systems
  - Threat models, security goals, assets, risks, adversaries
  - Connection between security, technology, politics, ethics, ...
  - The first few lectures, and the “security mindset” forum
    - http://cubist.cs.washington.edu/Security/ (previous years)
    - http://slashdot.org/
- **Technical aspects** of security
  - Attack techniques
  - Defenses

How to think about security

- Several approaches for developing “The Security Mindset” and for exploring the broader contextual issues surrounding computer security
  - Forum: Current event reflections
  - Forum: Security reviews
  - Control-Alt-Hack game
  - In class discussions
  - Additional participation in forums (e.g., critiquing movies)

Forum: Current events and security reviews (A look toward homework #1)

- One current event post
- One security review post
- 12 points each
- May work in groups of up to 3 people.
  - Working in groups is actually encouraged.
  - Recall: security is a contact sport -- lots of value in discussing security with other people
- Please participate in follow-up discussions on forum

Forum: Current events and security reviews

- Past blog URL: http://cubist.cs.washington.edu/Security/
- Past Security Reviews: http://cubist.cs.washington.edu/Security/category/security-reviews/

Technical Themes

- Vulnerabilities of computer systems
  - Software problems (buffer overflows); crypto problems; network problems (DoS, worms); people problems (usability, phishing)
- Defensive technologies
  - Protection of information in transit: cryptography, security protocols
  - Protection of networked applications: firewalls and intrusion detection
  - Least privilege, “Defense in depth”

What This Course is Not About

- Not a comprehensive course on computer security
  - Computer security is a broad discipline!
  - Impossible to cover everything in one quarter
  - So be careful in industry or wherever you go!
- Not about all of the latest and greatest attacks
  - Read news
- Not a course on ethical, legal, or economic issues
  - We will touch on ethical issues, but the topic is huge
- Not a course on how to “hack” or “crack” systems
  - Yes, we will learn about attacks ... but the ultimate goal is to develop an understanding of attacks so that you can build more secure systems

How Systems Fail

- Systems may fail for many reasons, including
- Reliability deals with accidental failures
- Usability deals with problems arising from operating mistakes made by users
- Security deals with intentional failures created by intelligent parties
  - Security is about computing in the presence of an adversary
  - But security, reliability, and usability are all related

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

Security is a Big Problem

- Security very often on the front page of Slashdot and other media outlets

Challenges: What is "Security?"

- What does security mean?
  - Often the hardest part of building a secure system is figuring out what security means
  - What are the assets to protect?
  - What are the threats to those assets?
  - Who are the adversaries, and what are their resources?
  - What is the security policy?
- Perfect security does not exist!
  - Security is not a binary property
  - Security is about risk management

Current events, security reviews, the game, and other discussions are designed to exercise our thinking about these issues

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
    - True?
  - 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

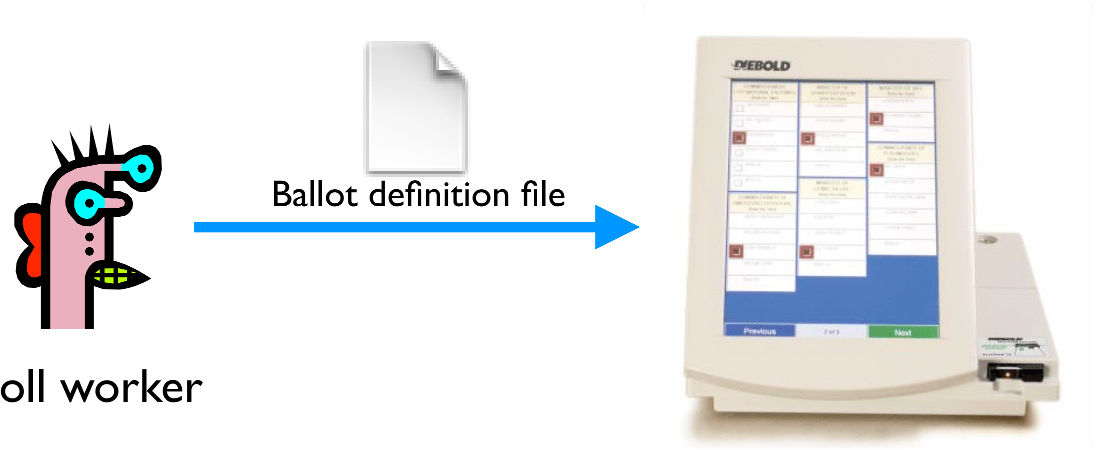
- The threat of a response may be enough to deter some attackers

Example: Electronic Voting

- Popular replacement to traditional paper ballots

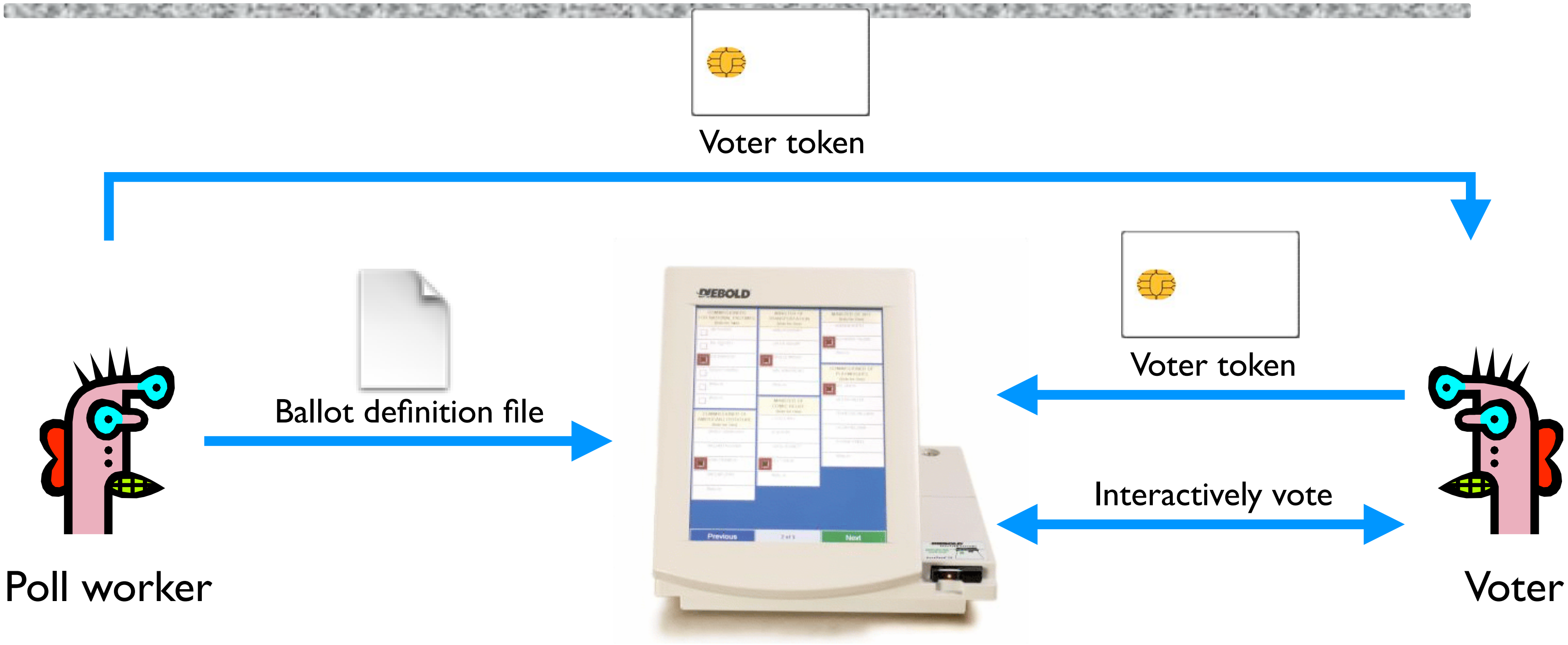
AP

Pre-Election



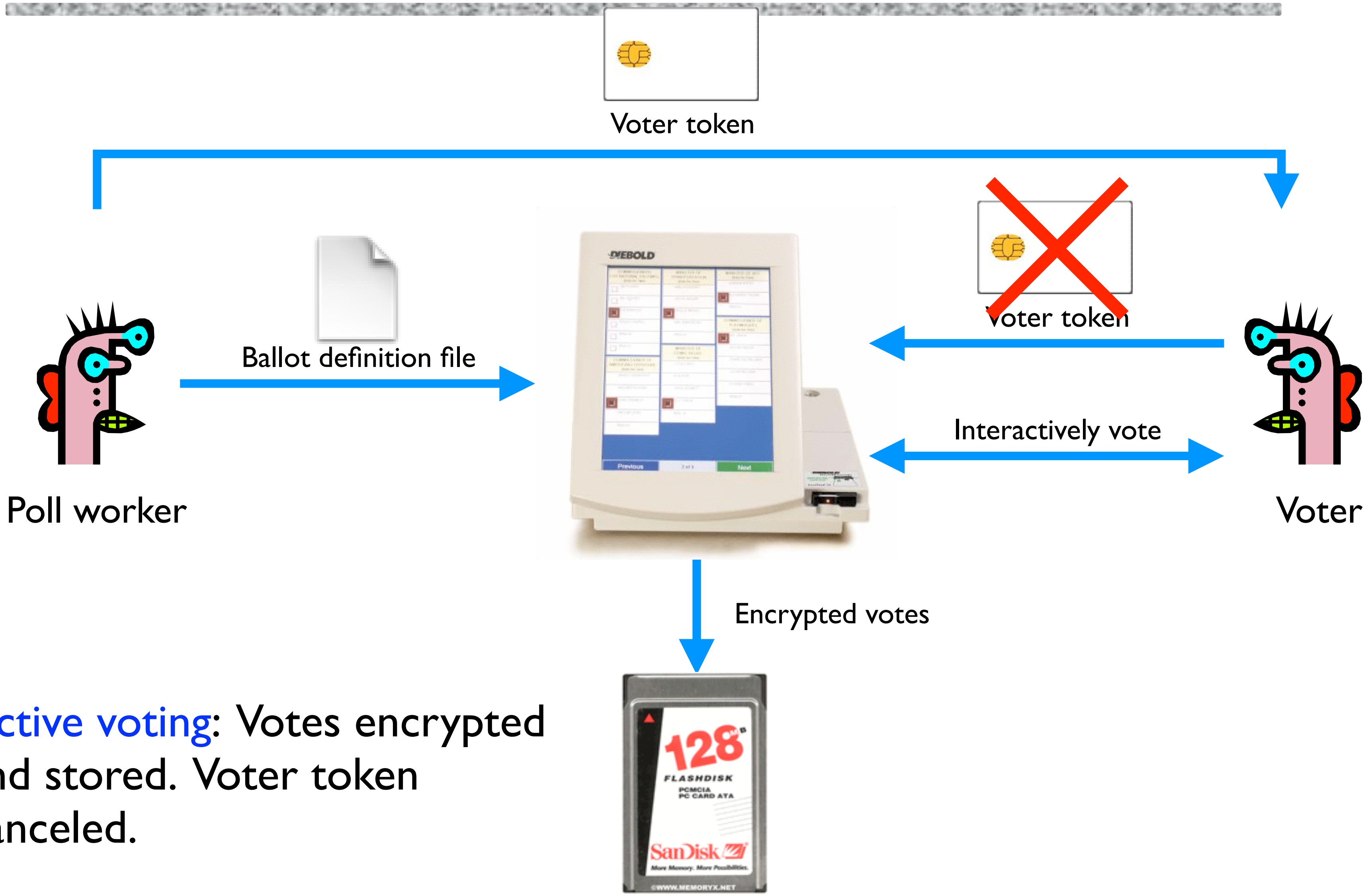
Pre-election: Poll workers load "ballot definition files" on voting machine.

Active Voting



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

Active Voting



Voter token

Ballot definition file

Voter token

Interactively vote

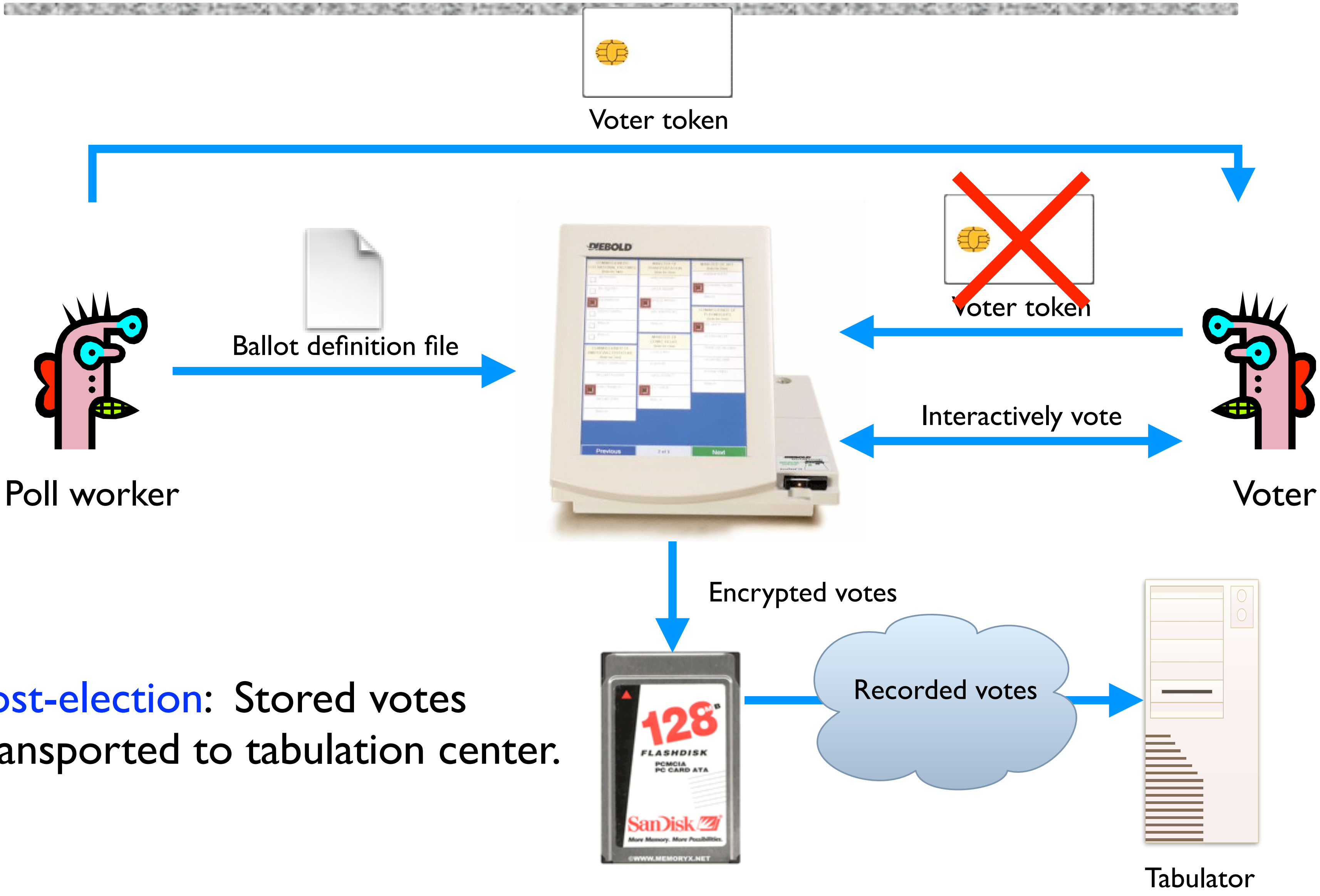
Poll worker

Voter

Encrypted votes

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

Post-Election



Post-election: Stored votes transported to tabulation center.

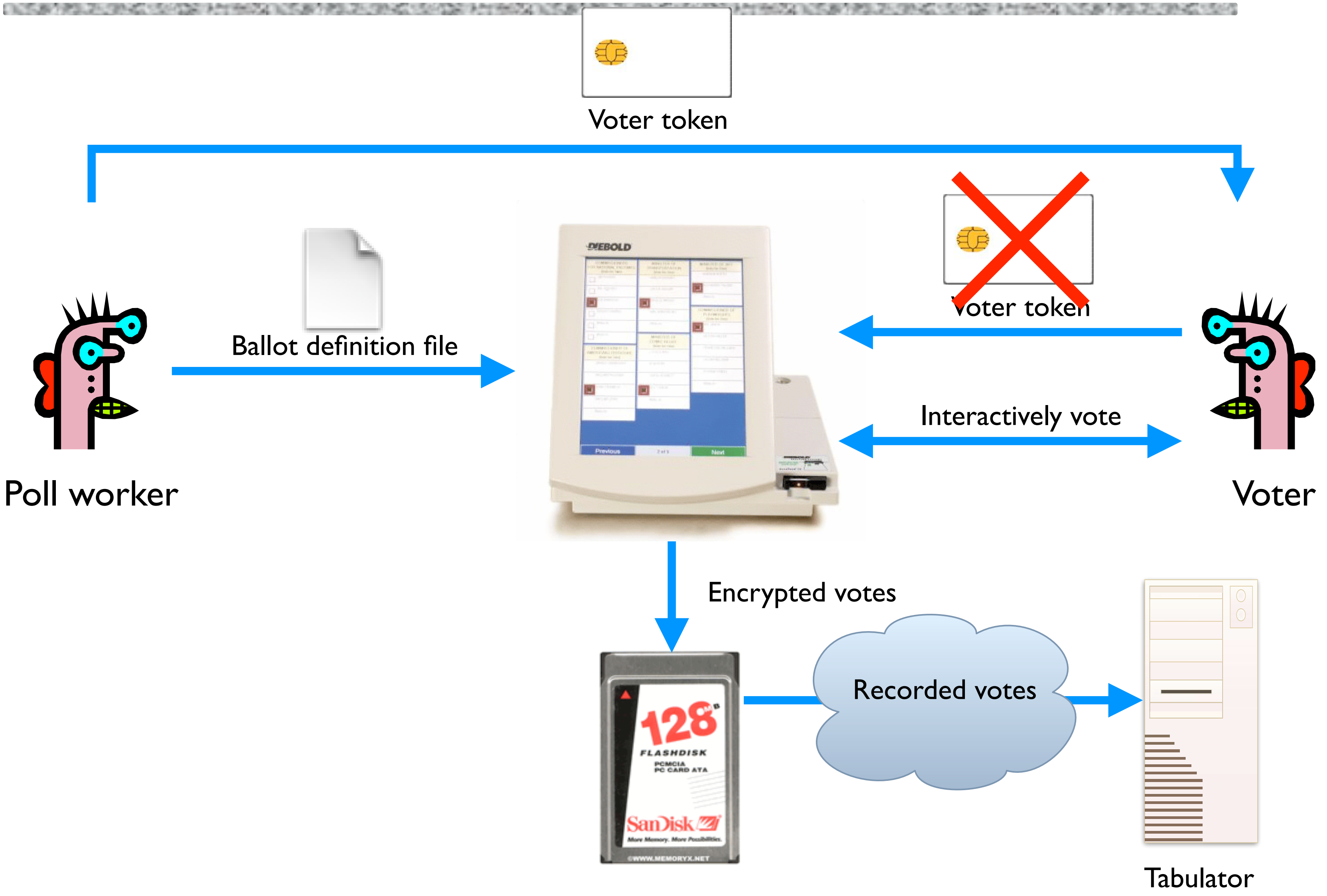
Security and E-Voting (Simplified)

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

Security 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?



Potential Adversaries

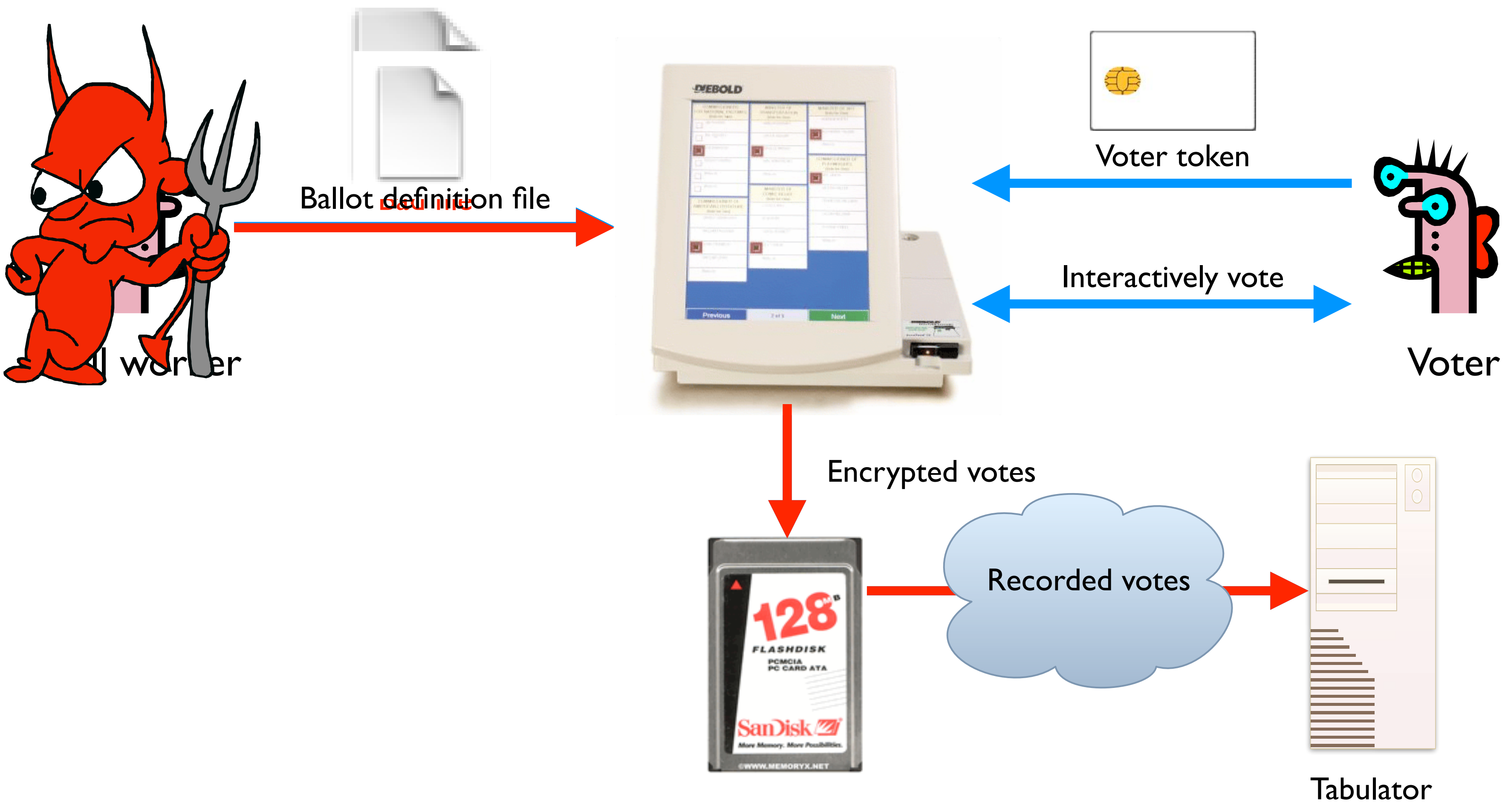
- Voters
- Election officials
- Employees of voting machine manufacturer
  - Software/hardware engineers
  - Maintenance people
- Other engineers
  - Makers of hardware
  - Makers of underlying software or add-on components
  - Makers of compiler
- …
- Or any combination of the above

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.

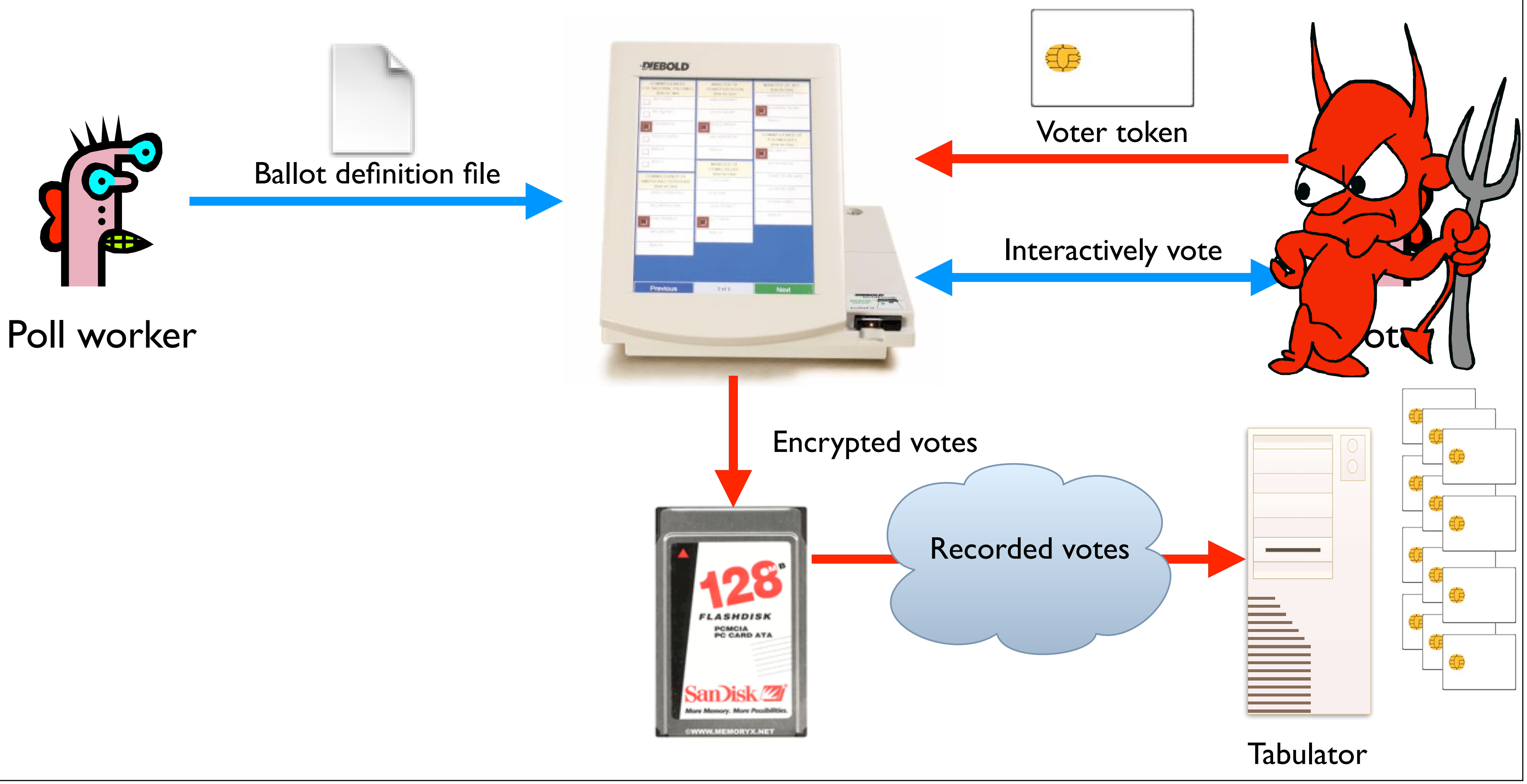
Problem: Ballot definition files are not authenticated.

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



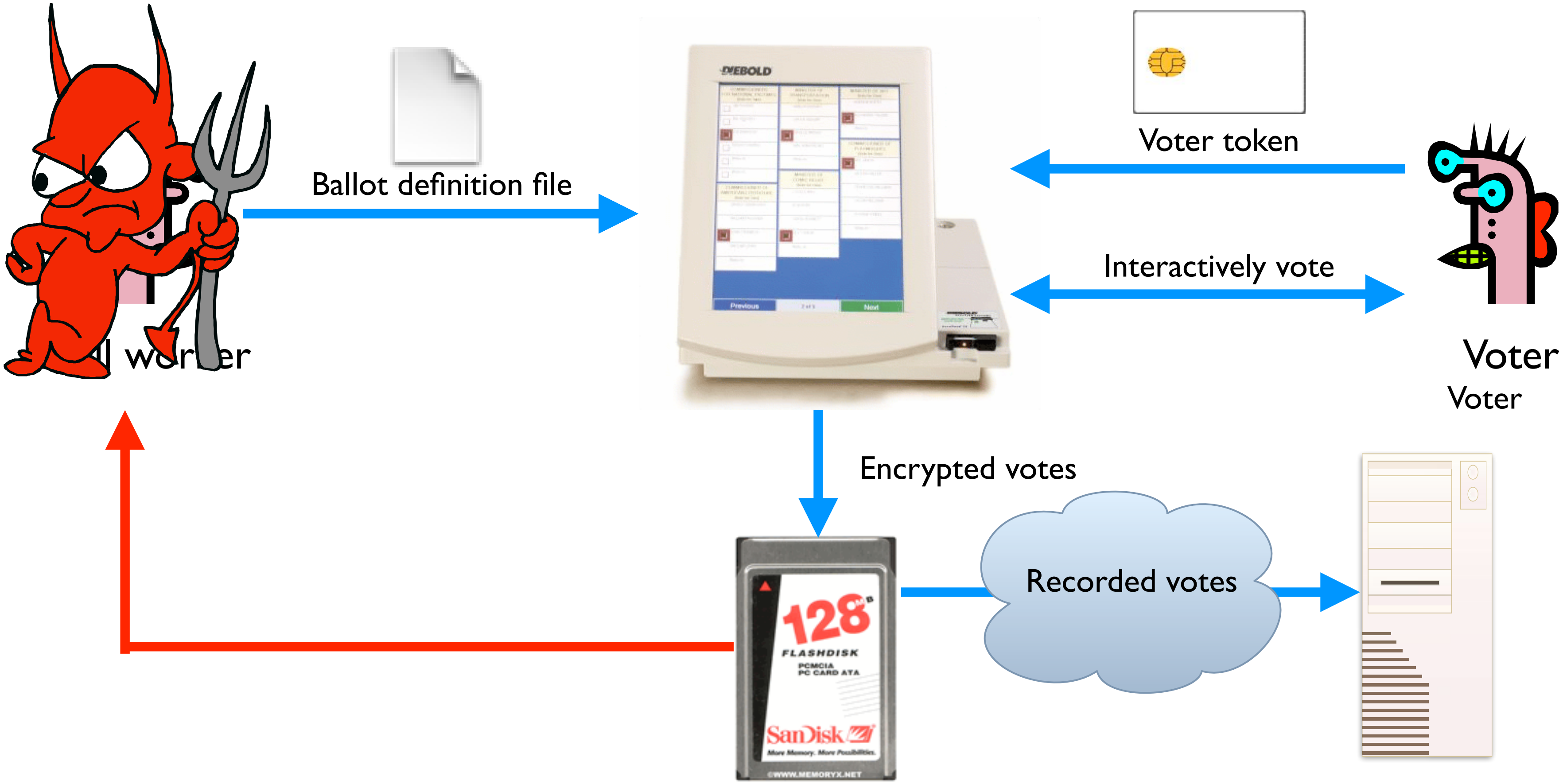
Problem: Smartcards can perform cryptographic operations. But there is no authentication from voter token to terminal.

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.



ll wor er

Ballot definition file

Voter token

Interactively vote

Voter

Voter

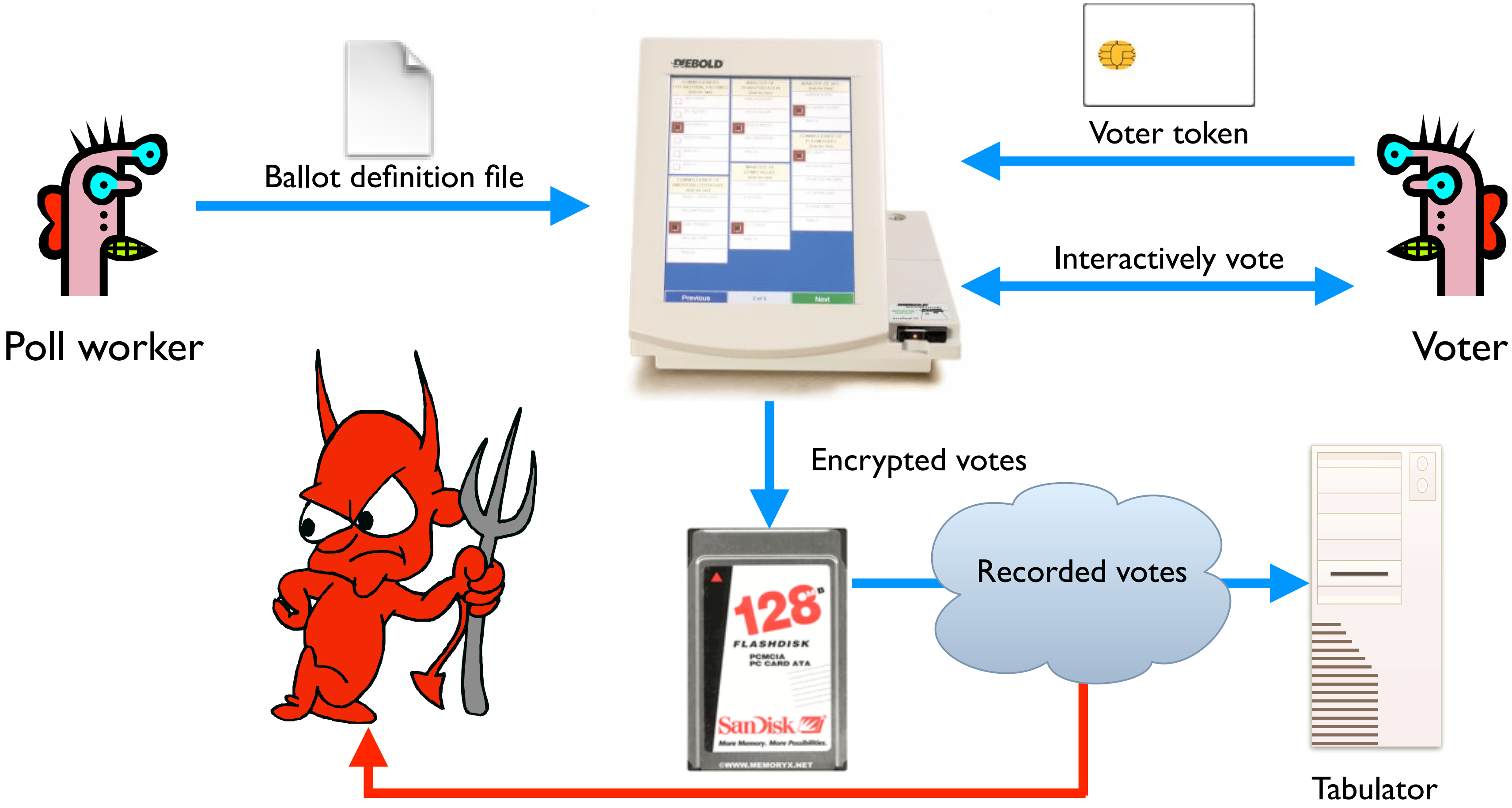
Encrypted votes

Recorded votes

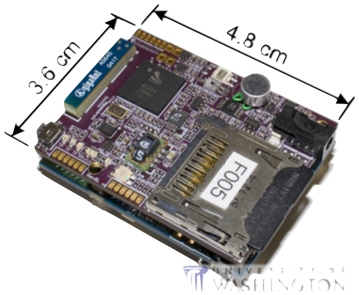
Tabulator

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.

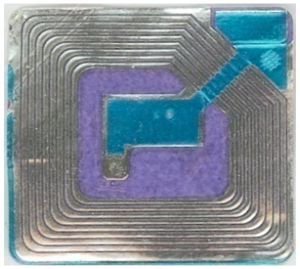
Example attack: A sophisticated outsider could determine how voters vote.



Security not just for PCs



mobile sensing platforms



RFID

EEG Gaming

large displays

ambient displays

smart phones

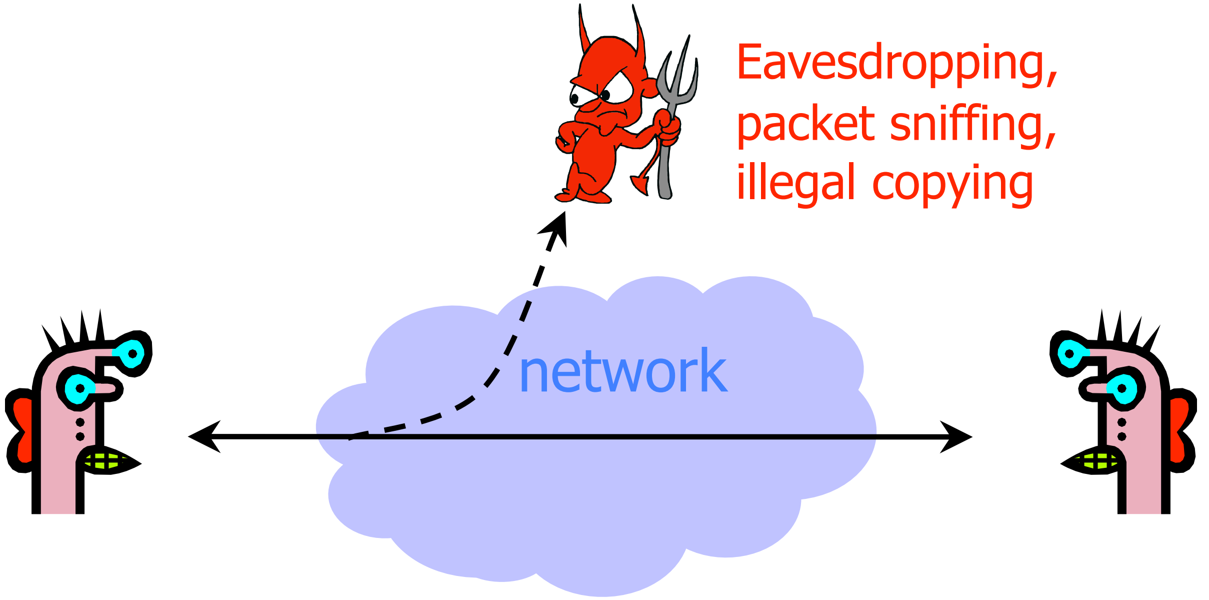
wearables

health displays

Security Goals

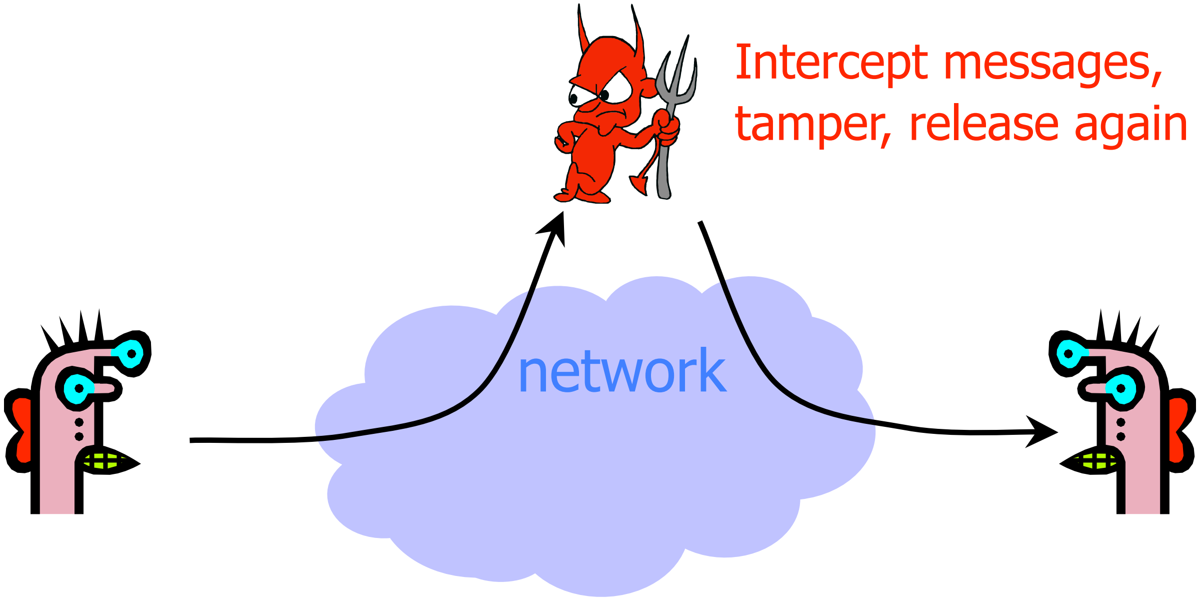
Confidentiality (Privacy)

- Confidentiality is concealment of information



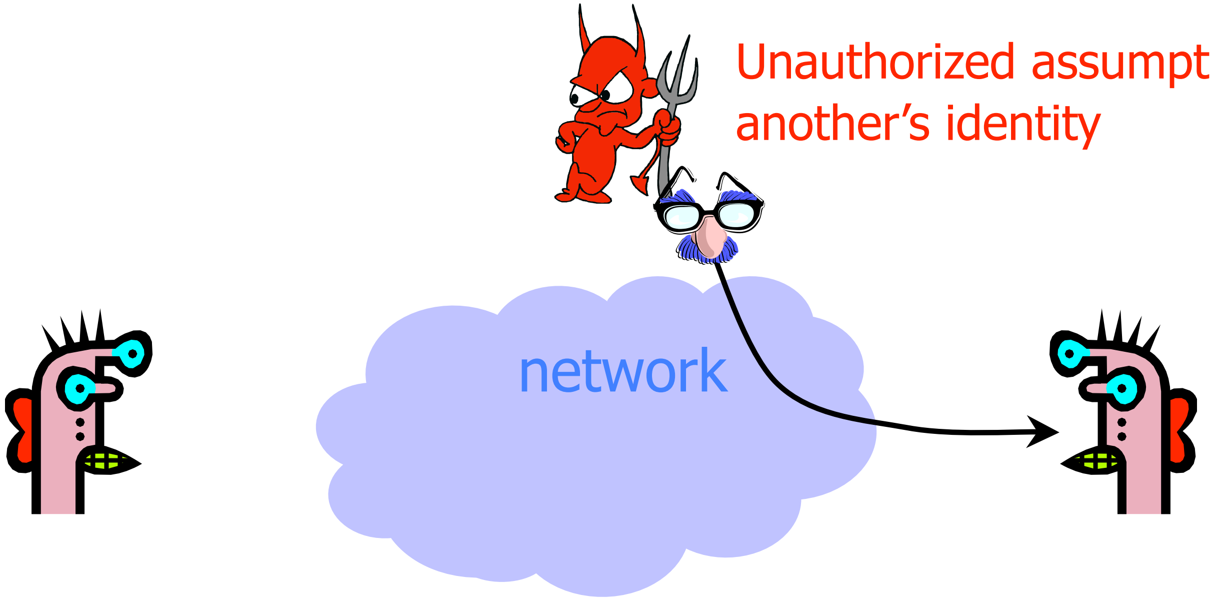
Authenticity / Integrity (1)

- Authenticity / integrity is prevention of unauthorized changes



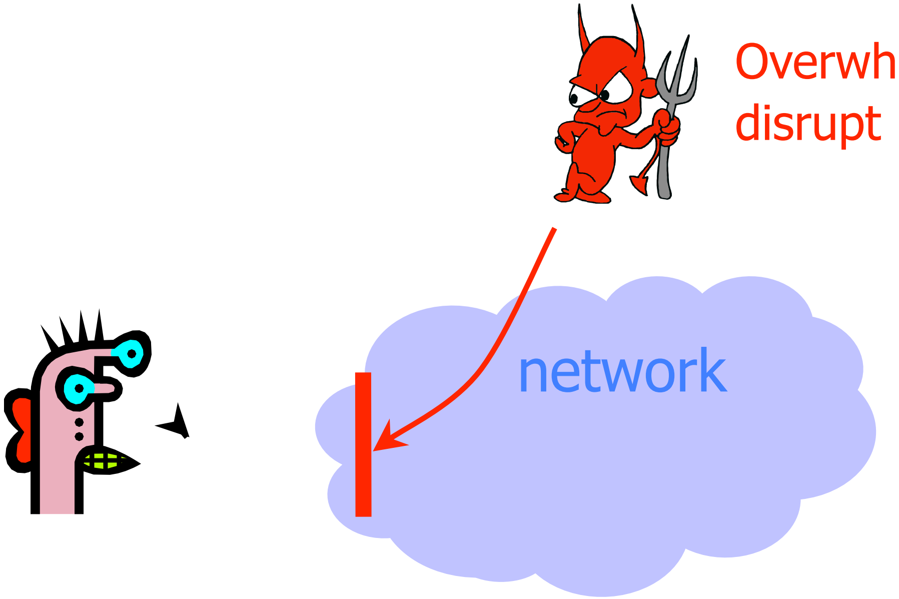
Authenticity / Integrity (2)

- Identification and assurance of origin of information



Availability

- Availability is ability to use information or resources desired



Overwhelm or crash servers, disrupt infrastructure

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 and 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, software
  - Reputation, brand name
  - Responsiveness
  - Personal safety
  - Physical resources: Laptops, servers, routers, PDAs, phones, ...
- 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 (None of them Great)

- Quantitative risk analysis
  - Example: $\overset{\text{Risk Exposure}}{\boxed{\text{Risk}}} = \overset{\text{Risk Impact}}{\boxed{\text{Asset}}} \times \overset{\text{Probability}}{\boxed{\text{Threat} \times \text{Vulnerability}}}$
  - 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
Reputation			
Responsiveness			
Data			
Personal Safety			
…			

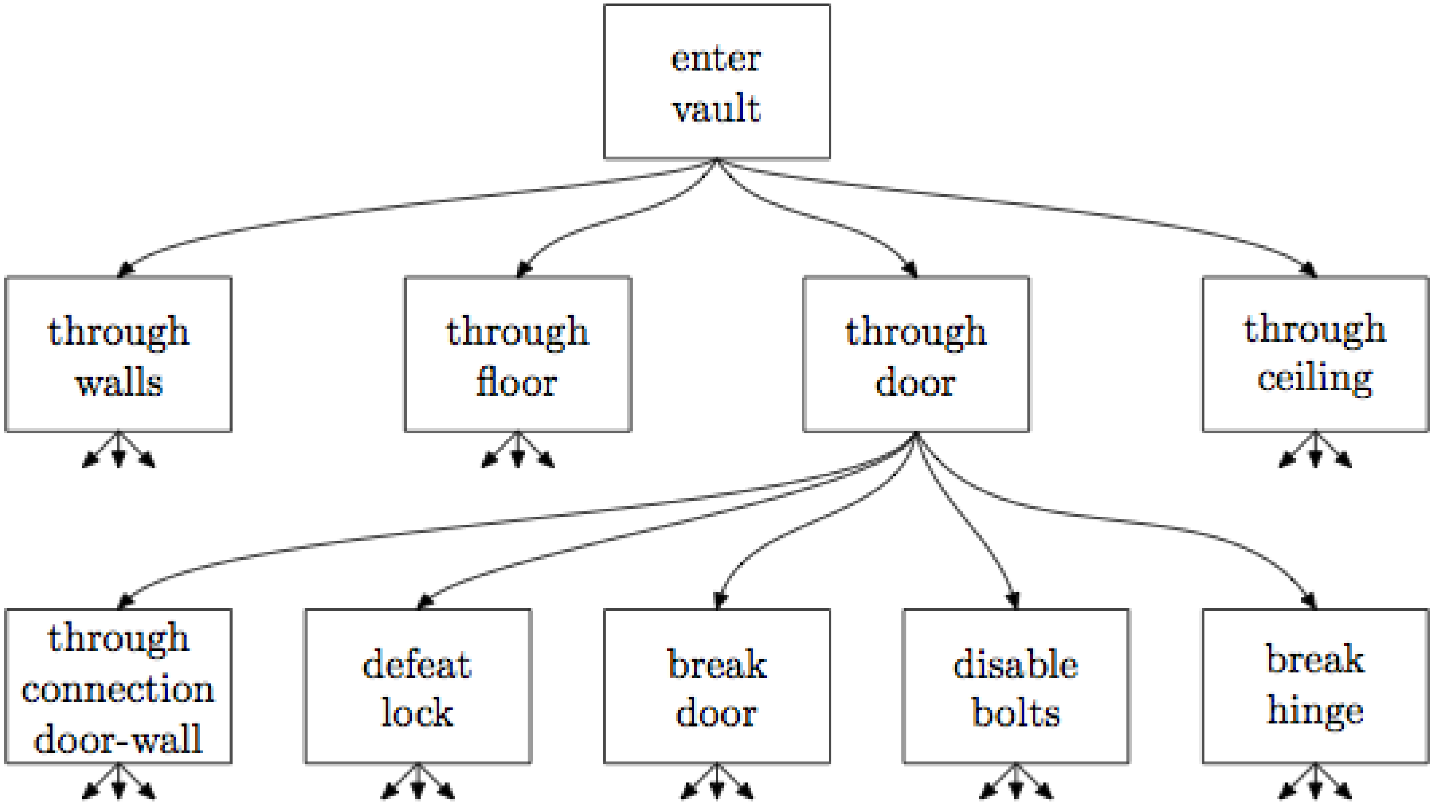
Helpful Tables

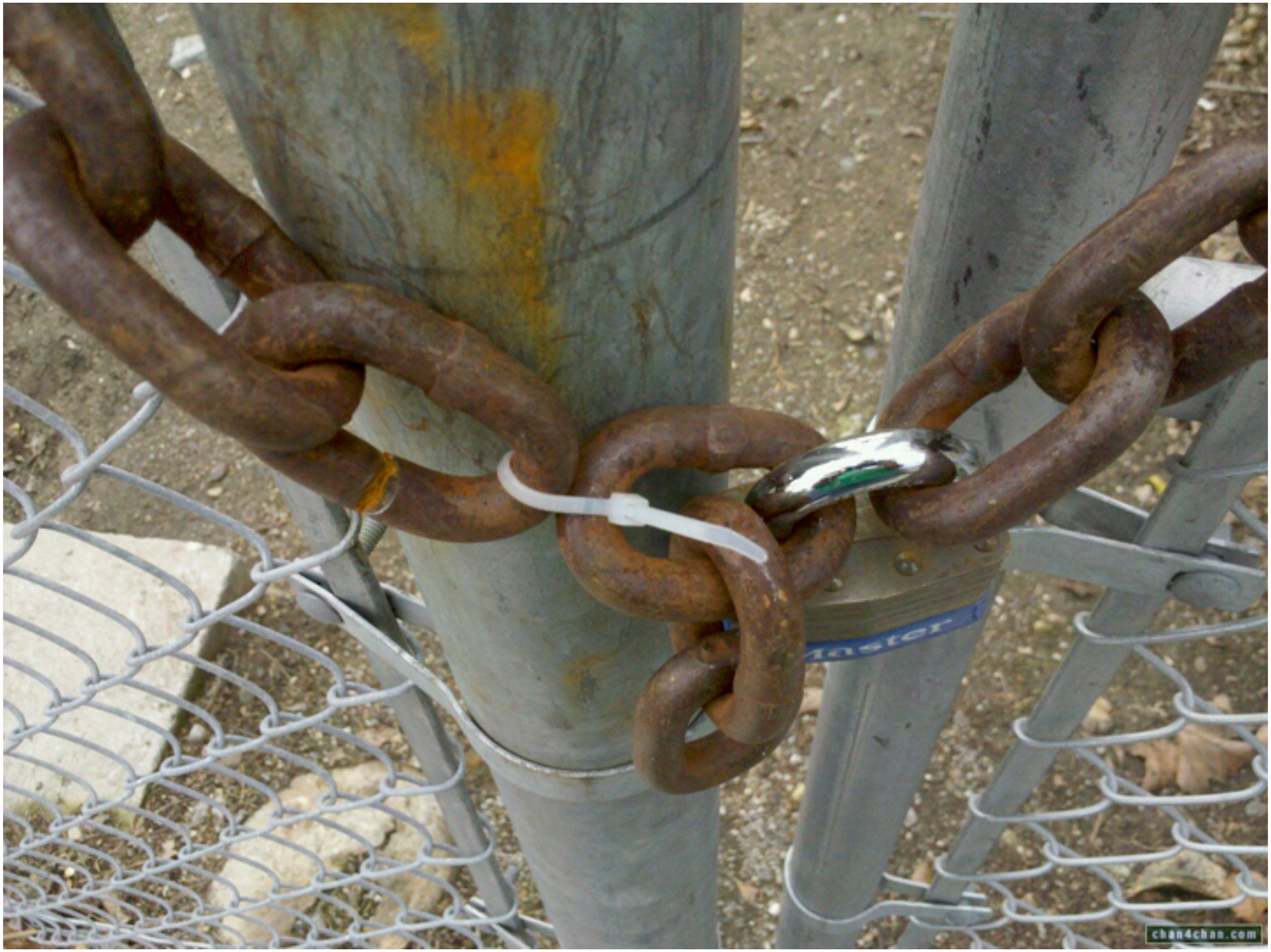
	Voter	Election official	…
Privacy of vote			
Integrity of vote			
Availability of voting system			
Confidence in election			
…			

Helpful Tables

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

Attack Trees




Master
chan4chan.com