ECE 455: CYBERSECURITY

Lecture #1

Daniel Gitzel

Announcements

- **Lab #0 due next week**
- **Security Review #0 due in two weeks**
- **Read papers for discussion next week**

Survey Results

- Penetration Testing
- Detection and Mitigation
- Program Analysis and Verification
- Databases (SQL injection, prevention)
- Mobile OS
- AI (training data, adversarial inputs)
- Distributed Systems (SCADA, IoT, mesh networks)
- Embedded Systems and Hardware
- Medical Devices
- Satellites (base station, comms, hardware)
- Blockchain Technology and Cryptocurrencies

REVIEW

Security Goals

- **Confidentiality** is concealment of information.
- **Integrity** is the prevention of unauthorized changes.
- **Authenticity** is knowing who you're talking to.
- **Availability** is ability to use information or resources.

Threat models

- **Assets**
  - What are we protecting? How valuable is this stuff?
- **Adversaries**
  - Who is attacking, and why?
- **Vulnerabilities**
  - How might the system be weak?
- **Threats**
  - What actions would an adversary take?
- **Risk**
  - How important are the assets? How likely is the exploit? Economic incentives?
- **All influence possible defenses**

Approaches to Security

- **Prevention**
  - Stop an attack
- **Detection**
  - Detect an ongoing or past attack
- **Response**
  - Respond to attacks
  - The threat of response may deter attackers: “Beware of Dog”

Whole System is Critical

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

- **Security on as strong as the weakest link**
  - Why attack the strongest part? “Backdoor” or weak spots

Asymmetry Advantage

From Policy to Implementation

- **Realizing a security policy has challenges:**
  - Requirement bugs
    - Conflicting or wrong goals
  - Design bugs
    - Poor use of cryptography
    - Poor source of randomness
  - Implementation bugs
    - Traditional software bugs
  - Usability
    - Can a normal human actually use this?

An ecosystem of participants

- **Many parties involved**
  - System developers
  - Companies/contractors deploying the system
  - End users
  - Adversaries
- **Security is a *people* problem**
  - Social engineering is a powerful attack method!

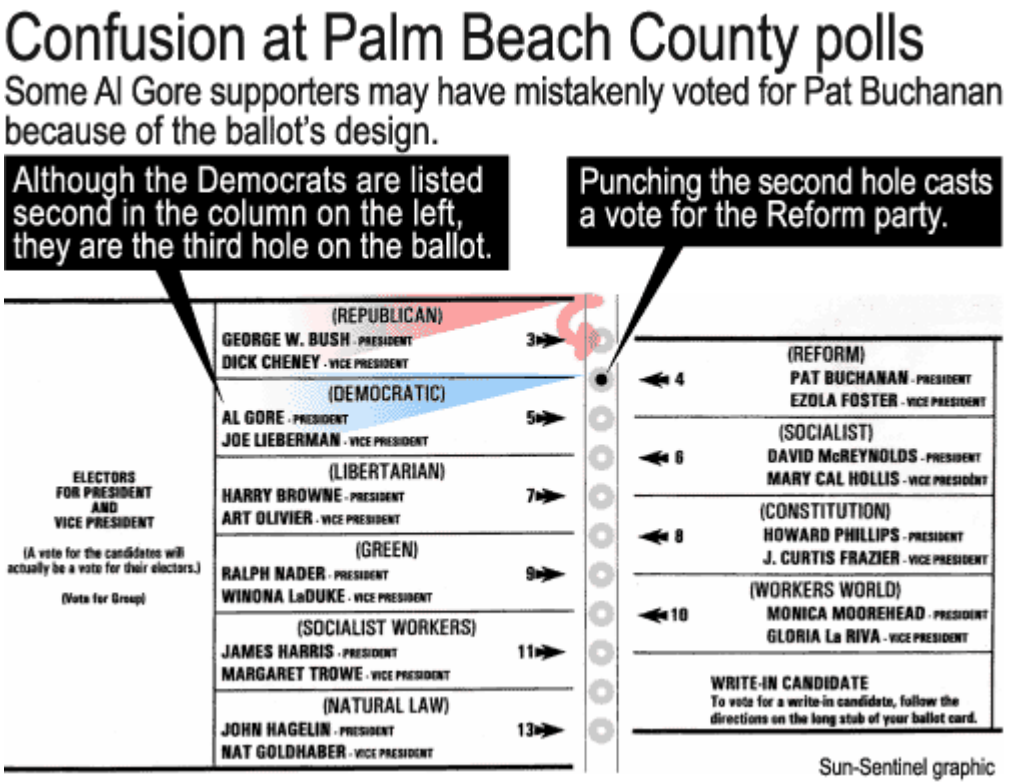
THREAT MODEL EXAMPLE

ELECTRONIC VOTING

Slides by Tadayoshi Kohno (2004 paper author)

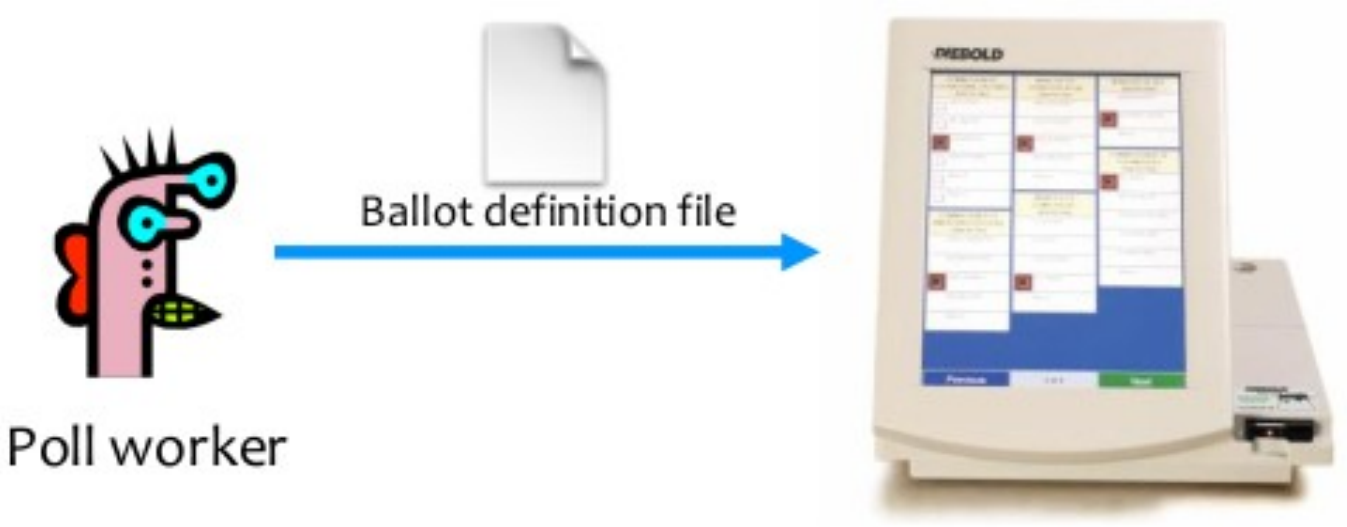
Electronic Voting

- **Popular alternative to paper ballot voting**

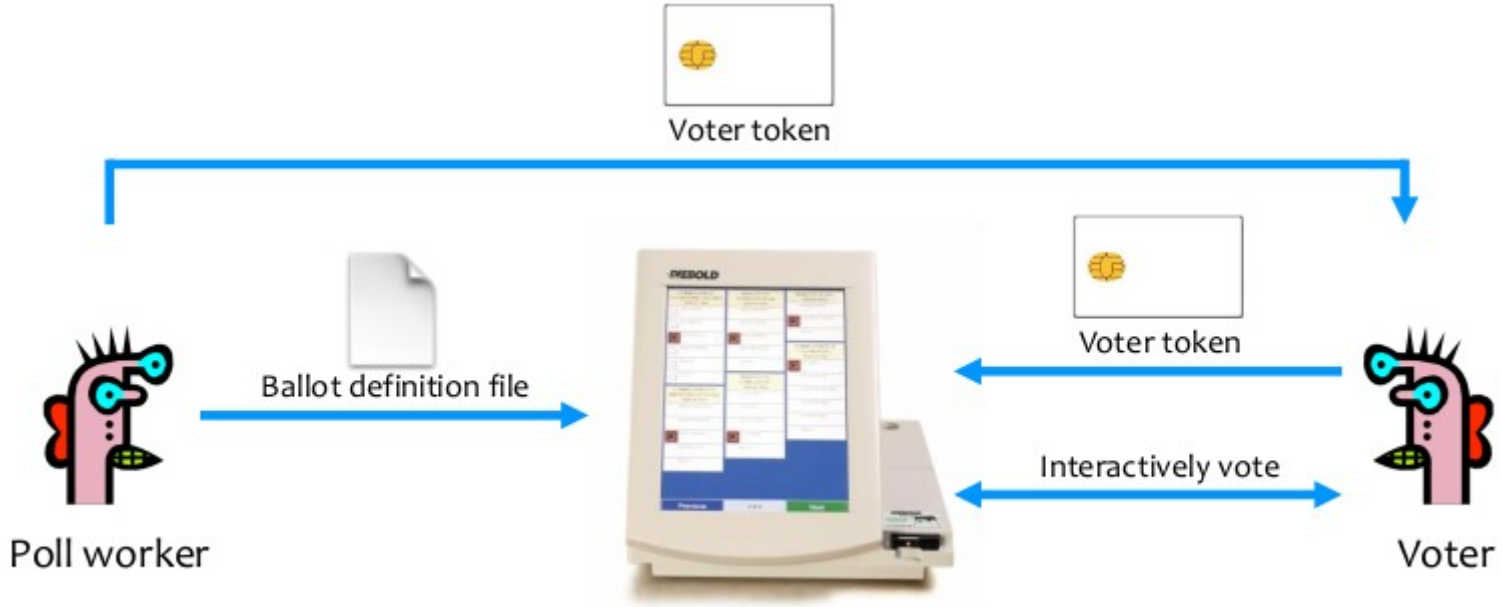


Diebold System

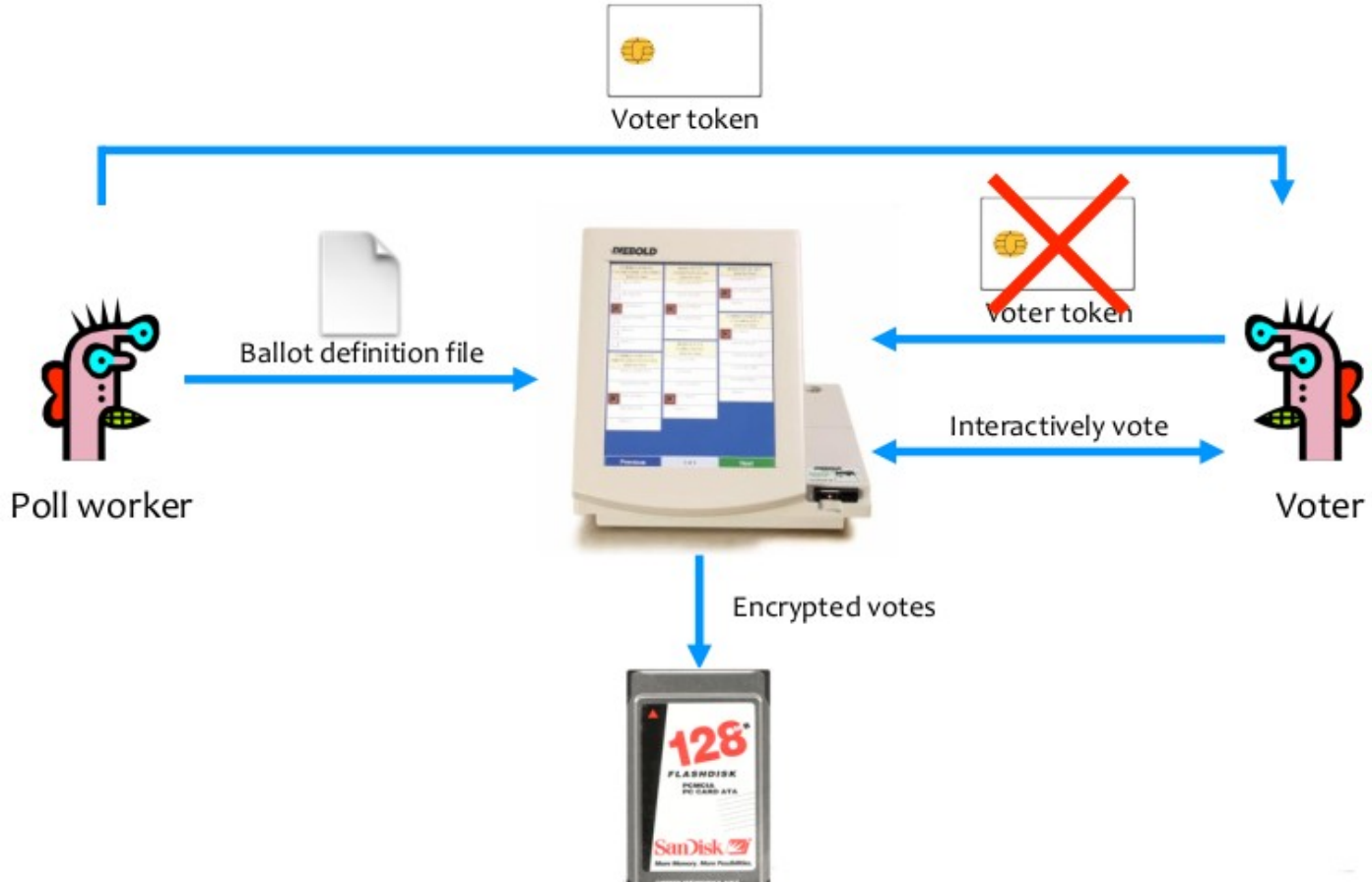
- **Pre-Election**
  - Poll workers load “ballot definition files” on voting machine.



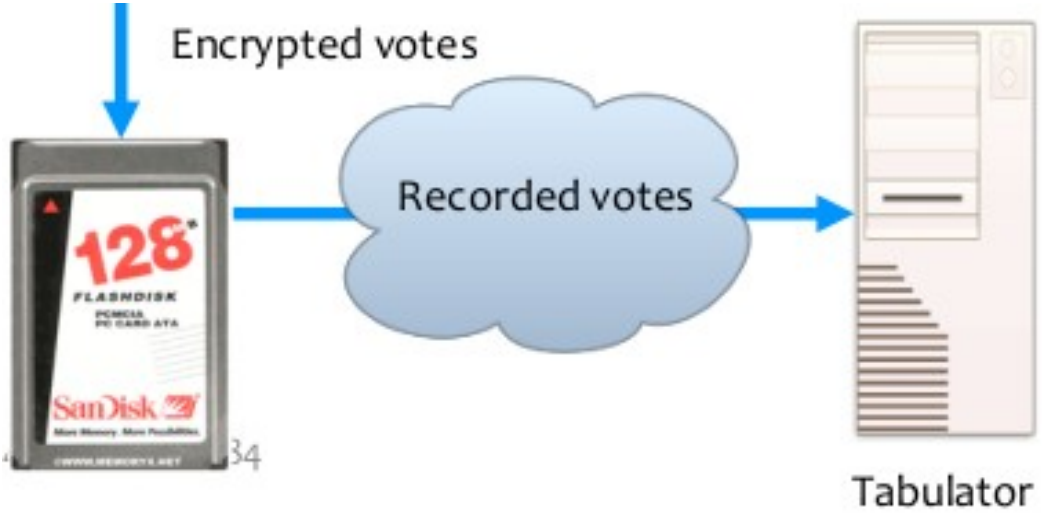
- **Active voting:**
  - Voters obtain single-use tokens from poll workers.
  - Voters use tokens to activate machines and vote.



- **Active Voting**
  - Votes encrypted and stored
  - Voter token canceled



- **Post-Election**
  - Votes uploaded to tabulation server



System Goals

- **Functionality goals:**
  - Easy to use, reduce mistakes/confusion
- **Security goals:**
  - Adversary should not be able to tamper with the election outcome:
    - By changing votes (**integrity**)
    - By voting on behalf of someone (**authenticity**)
    - By denying voters the right to vote (**availability**)
  - Adversary should not be able to figure out how voters voted (**confidentiality**)

What Issues Do You See?

Potential Adversaries

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

What (Whose) Software is Running?

- **Adversary**
  - Poll worker, software developer, or company representative
- **Vulnerability**
  - Control over the software or the underlying hardware
- **Threat**
  - Broad power to change ballot or votes

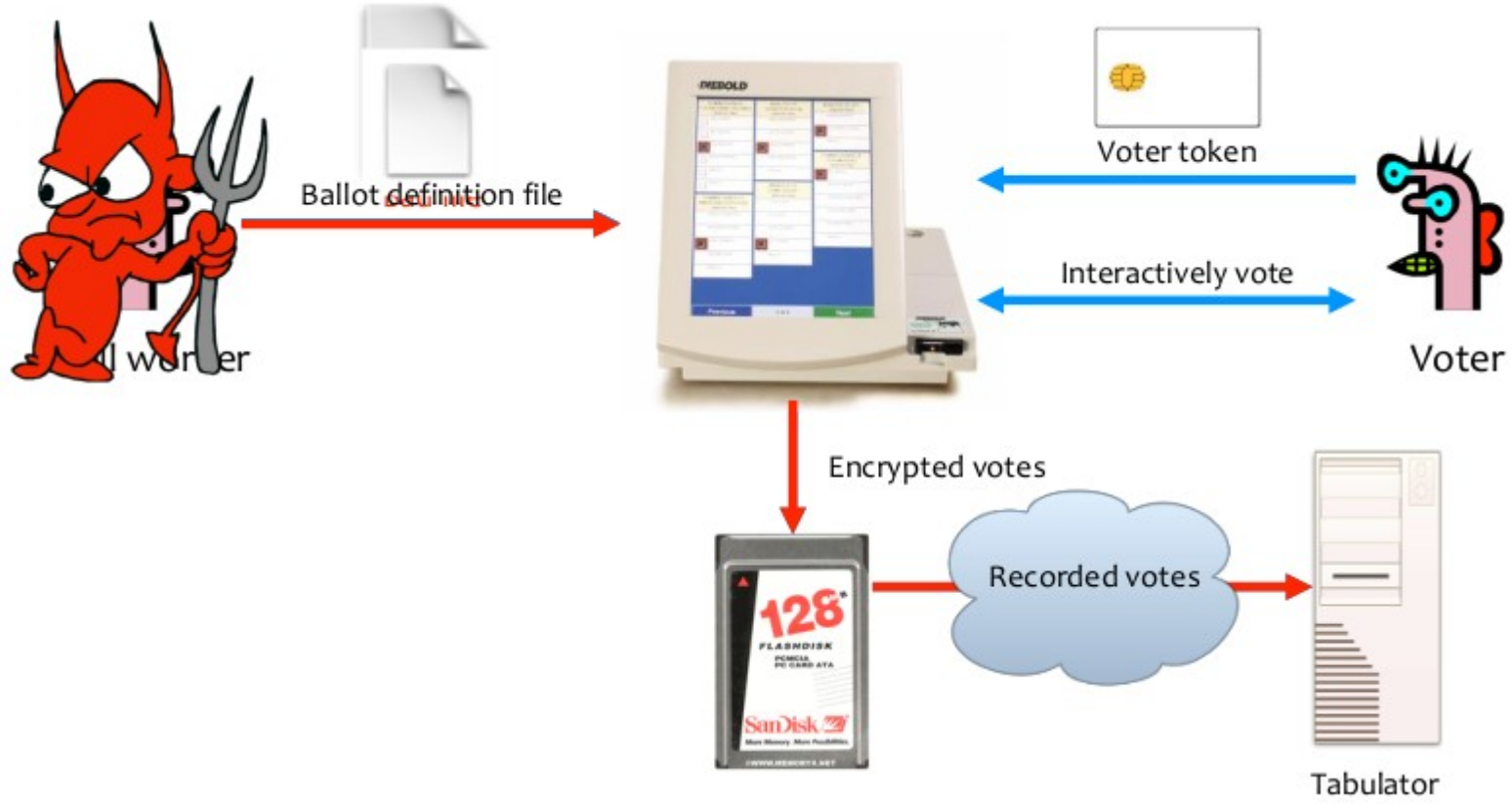
Who Has Physical Access?

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.

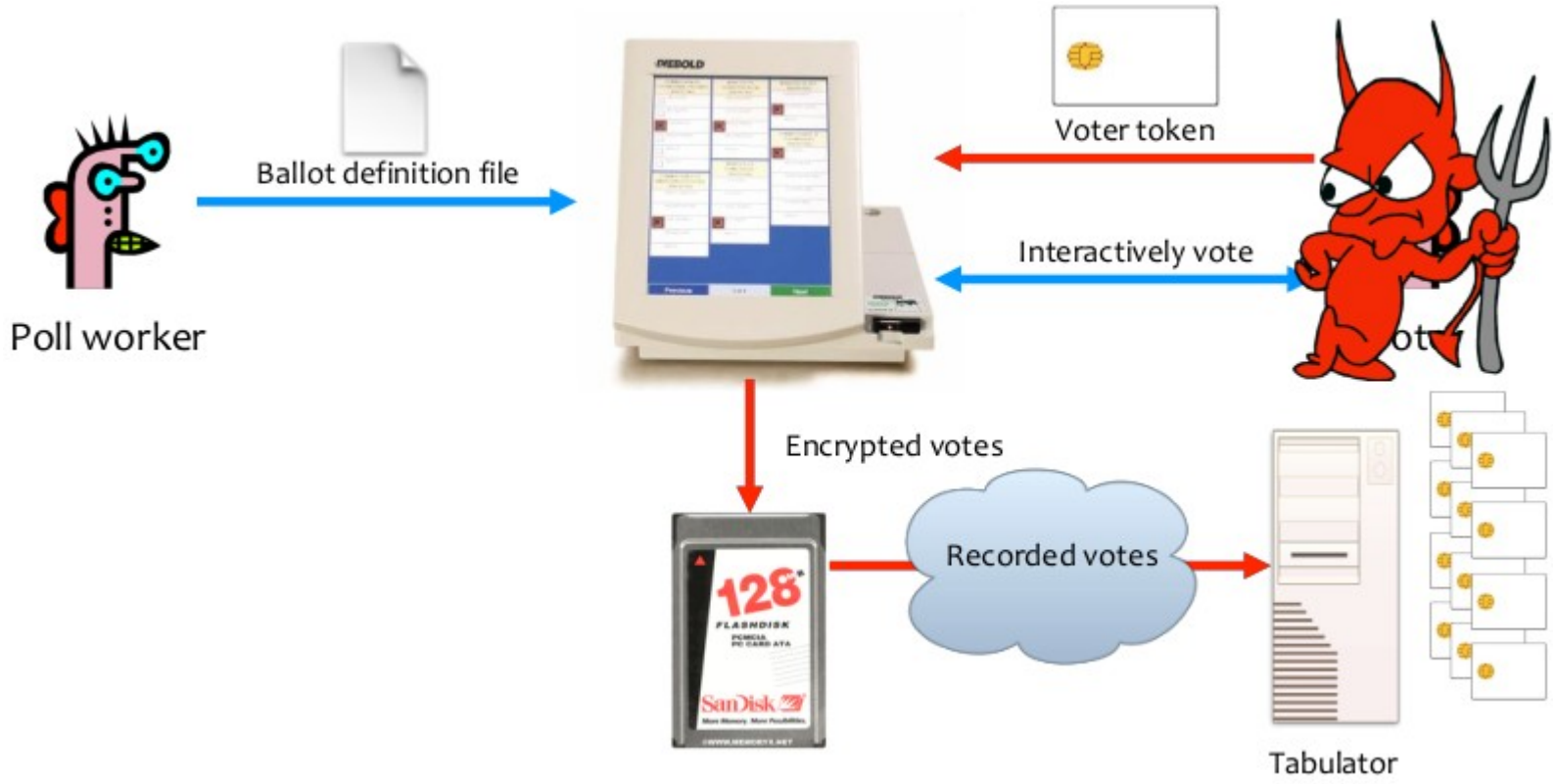
Unauthenticated Software

- **Problem:**
  - Ballot definition files are not authenticated.
- **Example attack:**
  - A malicious poll worker could modify ballot definition files
  - Votes cast for “Candidate A” are recorded for “Candidate B”


Ballot definition file
Voter token
Interactively vote
Voter
Encrypted votes
Recorded votes
Tabulator

Unauthenticated Tokens

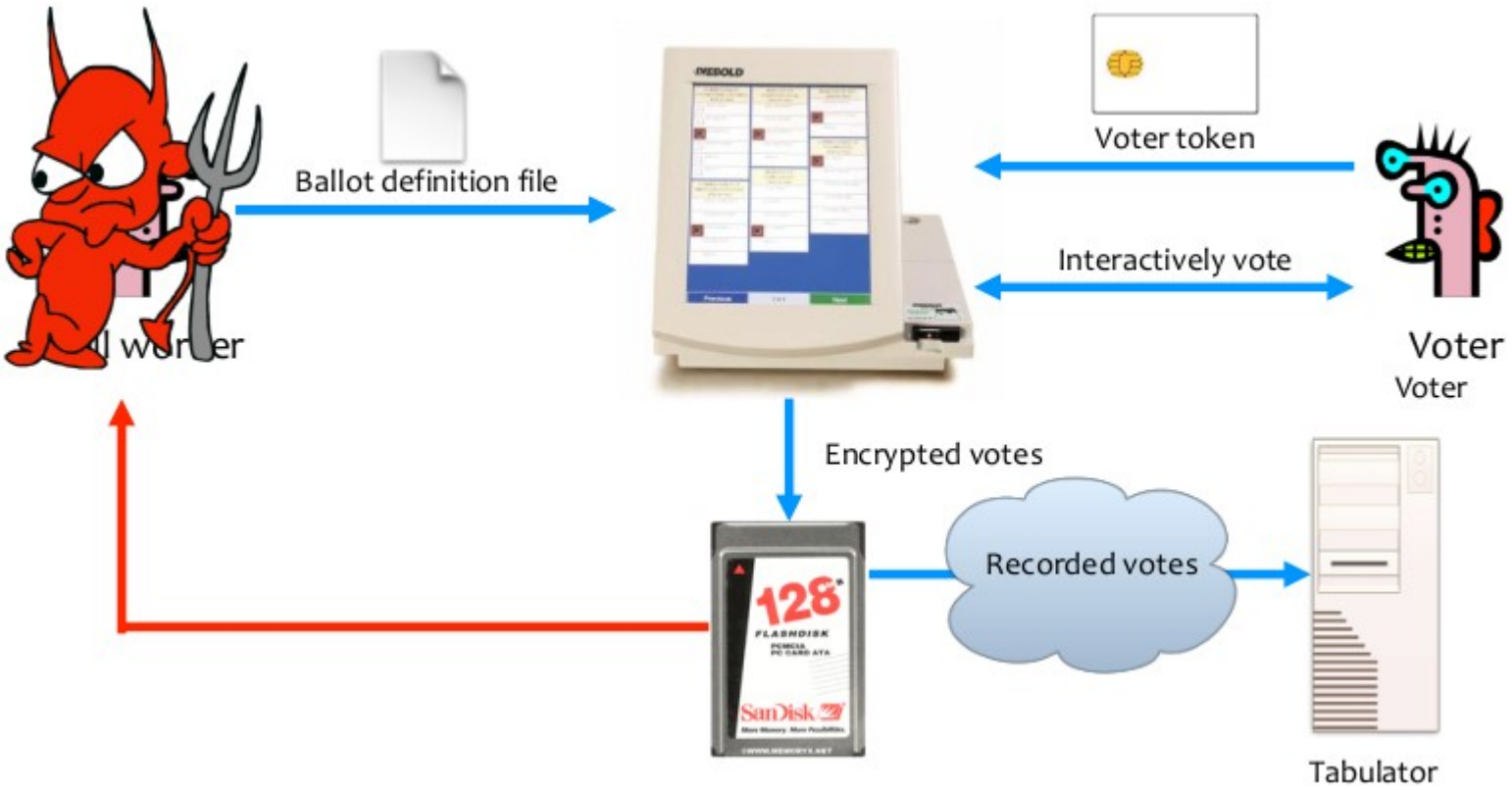
- **Problem:**
  - There is no authentication from voter token to terminal.
- **Example attack:**
  - A regular voter could make his or her own voter tokens and vote multiple times.


Poll worker
Ballot definition file
Voter token
Interactively vote
Encrypted votes
Recorded votes
Tabulator

Hardcoded Keys

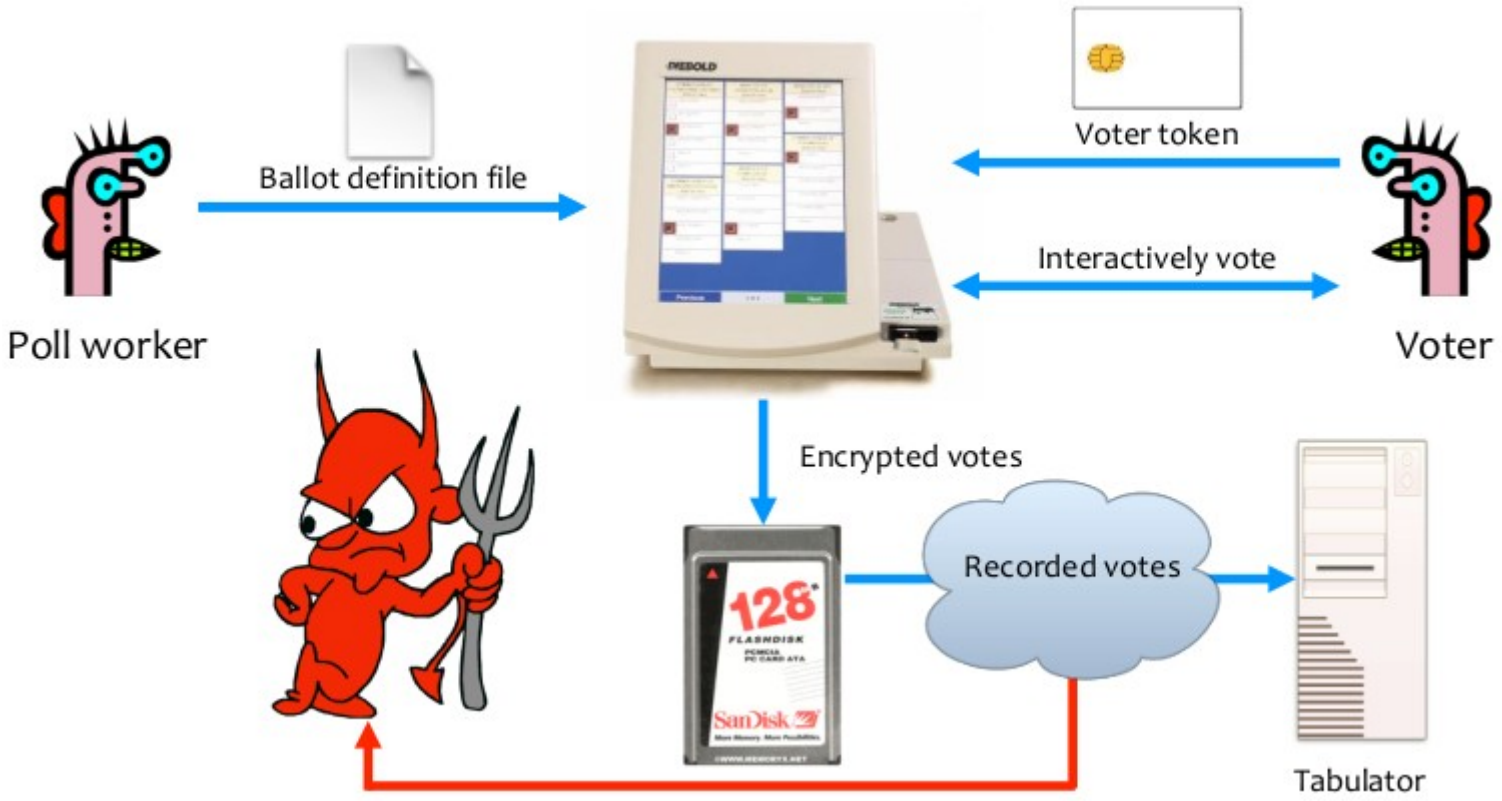
- **Problem:**
  - Encryption key (“F2654hD4”) hard-coded into the software since (at least) 1998.
  - Votes stored in the order cast.
- **Example attack:**
  - A poll worker could determine how voters vote.

#define DESKEY ((des_key*)"F2654hD4")


Ballot definition file
Voter token
Interactively vote
Voter
Voter
Encrypted votes
Recorded votes
Tabulator

Transmissions in the Clear

- **Problem:**
  - When votes transmitted to tabulator over the Internet, they are decrypted first; the clear-text results are sent to the tabulator.
- **Example attack:**
  - A sophisticated outsider could determine how voters vote.


Ballot definition file
Voter token
Interactively vote
Poll worker
Voter
Encrypted votes
Recorded votes
Tabulator

Tracking Attacks

Potential Attackers	Manufacturer	Poll Worker	Voter	Company Employee
Voter Privacy				
Voter Integrity				
...				

- **What can each party do?**
  - Each cell has an action that these parties can try
  - Parties may collaborate

Tracking Parties

Potential Attackers	Modify software	Make fake voter tokens	Steal flash drive	Intercept network packets
Voter Privacy				
Voter Integrity				
...				

- **What different attacks exist?**
  - Each cell contains the attack details
- **Who could mount these attacks?**
- **How easy would it be to implement each attack?**

Table from Paper

	Voter (with forged smartcard)	Poll Worker (with access to storage media)	Poll Worker (with access to network traffic)	Internet Provider (with access to network traffic)	OS Developer	Voting Device Developer	Section
Vote multiple times using forged smartcard	•	•	•				3.2
Access administrative functions or close polling station	•	•			•	•	3.3
Modify system configuration		•			•	•	4.1
Modify ballot definition (e.g., party affiliation)		•	•	•	•	•	4.2
Cause votes to be miscounted by tampering with configuration		•	•	•	•	•	4.2
Impersonate legitimate voting machine to tallying authority		•	•	•	•	•	4.3
Create, delete, and modify votes		•	•	•	•	•	4.3, 4.5
Link voters with their votes		•	•	•	•	•	4.5
Tamper with audit logs		•			•	•	4.6
Delay the start of an election		•	•	•	•	•	4.7
Insert backdoors into code					•	•	5.3

Table 1: This table summarizes some of the more important attacks on the system.

- **Nov 4, 2002:**
  - State of Georgia votes on Diebold DREs.
- **March 18, 2003:**
  - Diebold source code leaks.
- **July 23, 2003:**
  - Tadayoshi Kohno, Adam Stubblefield, Avi Rubin,Dan Wallach, “Analysis of an Electronic Voting System”.

Broward Vote-Counting Blunder Changes Amendment Result

POSTED: 1:34 pm EST November 4, 2004

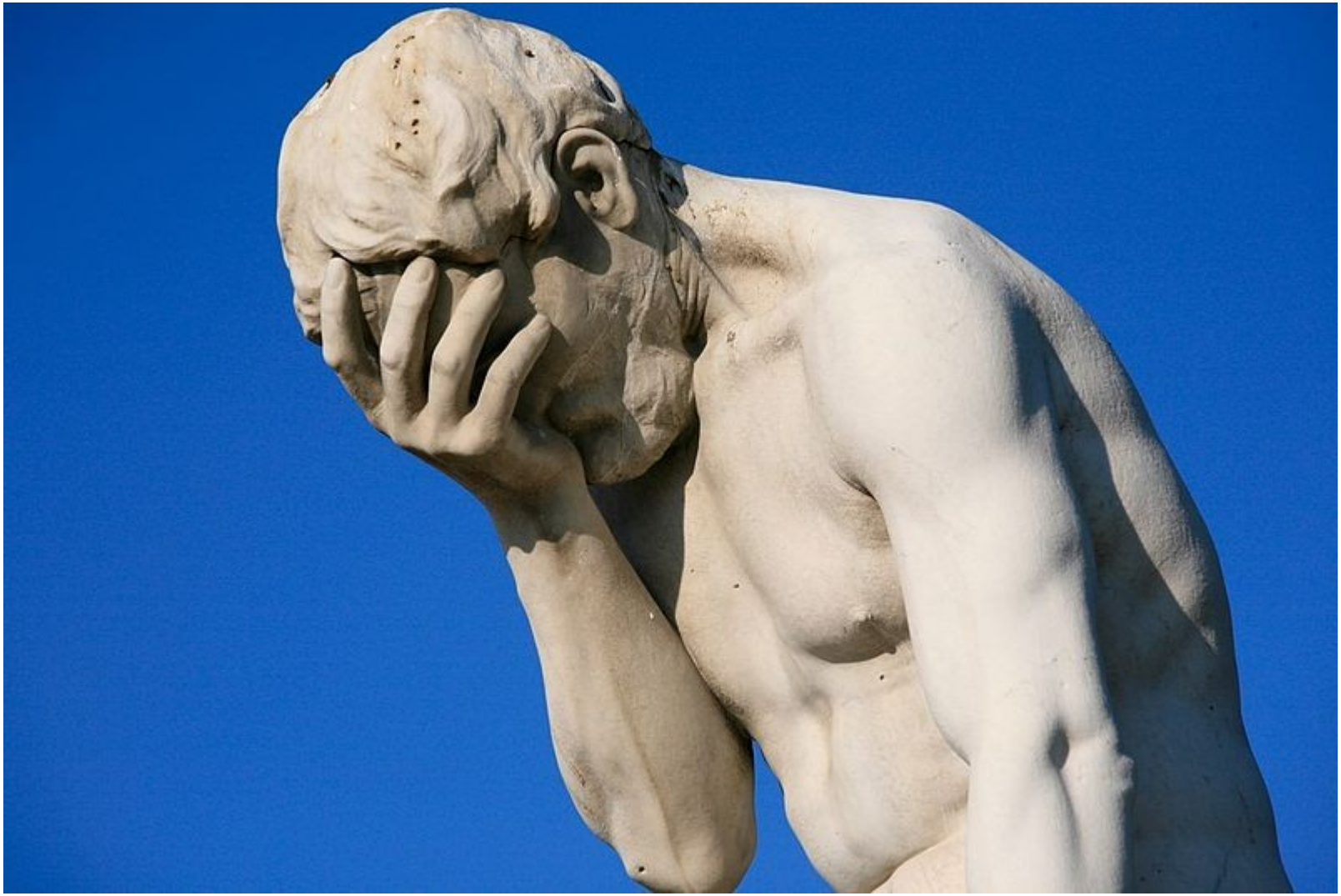
BROWARD COUNTY, Fla. -- The Broward County Elections Department has egg on its face today after a computer glitch misreported a key amendment race, according to WPLG-TV in Miami.

Amendment 4, which would allow Miami-Dade and Broward counties to hold a future election to decide if slot machines should be allowed at racetracks, was thought to be tied. But now that a computer glitch for machines counting absentee ballots has been exposed, it turns out the amendment passed.

"The software is not geared to count more than 32,000 votes in a precinct. So what happens when it gets to 32,000 is the software starts counting backward," said Broward County Mayor Ilene Lieberman.

Broward County Mayor Ilene Lieberman says voting counting error is an "embarrassing mistake."

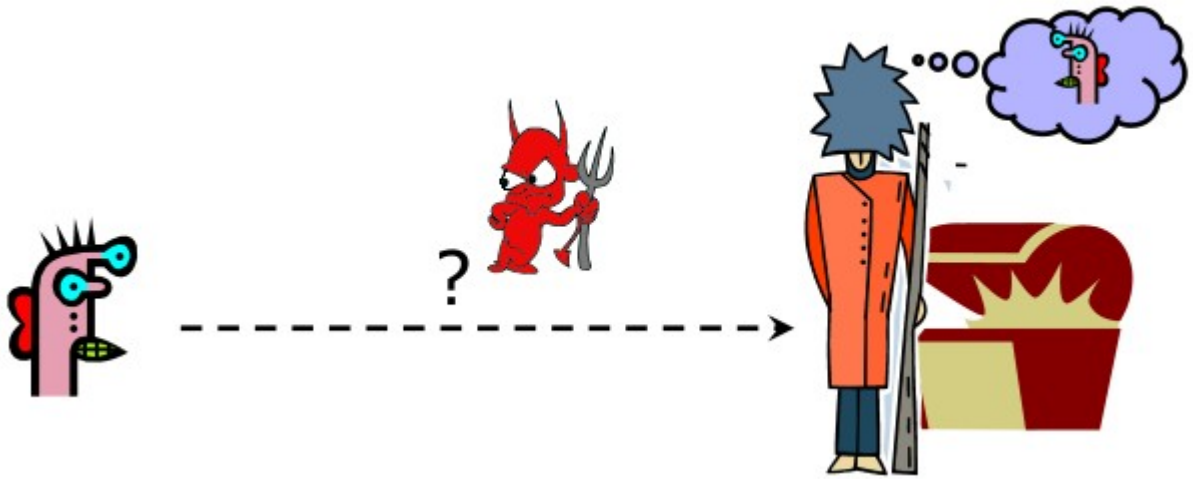
That means that Amendment 4 passed in Broward County by more than 240,000 votes rather than the 166,000-vote margin reported Wednesday night. That increase changes the overall statewide results in what had been a neck-and-neck race, one for which recounts had been going on today. But with news of Broward's error, it's clear amendment 4 passed.



IDENTIFICATION & AUTHENTICATION

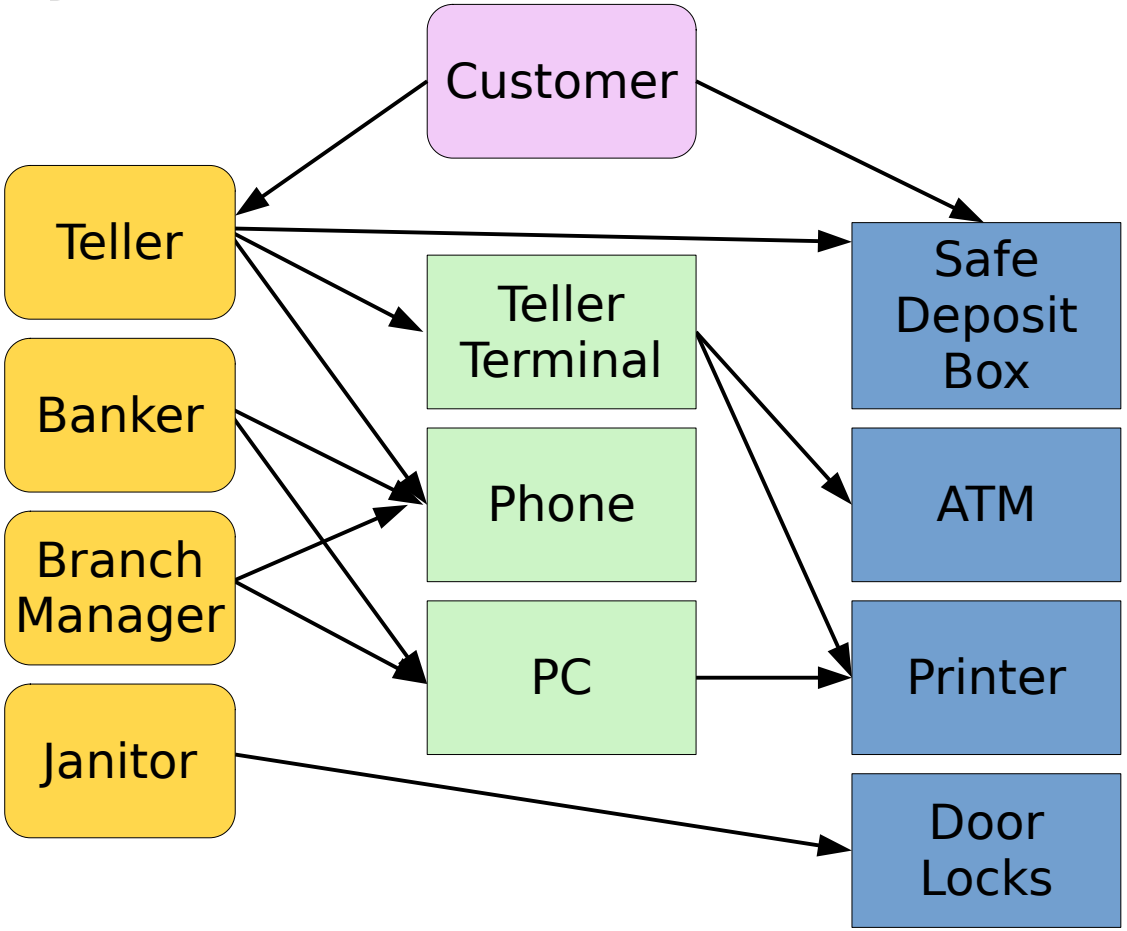
Basic Problem

- **How do you prove to someone that you are who you claim to be?**
  - Any system with access control must solve this problem.



Large Systems Have Multiple Points

- **Users and systems must authenticate each other**



Security Goals

- **Accountability** is the ability to identify and authenticate users and audit actions.
- **Non-repudiation** is unforgeable evidence that a specific action has occurred.

Who Goes There?

- **Parking garage fob**
  - Radio transmitter, key, memory, counter
- **Fob: T → G: T, $\{T, N\}_{KT}$**
  - T, serial number from ROM
  - KT encryption key from ROM
  - N nonce (number, used ***once***) from counter
- **Garage**
  - Derive K_T from K_M and T
  - Synchronize counter?
- **Can you think of any issues?**

Cars and Immobilizers

- **Protocol**
  - E → F: N
  - F → E: T, $\{T,N\}_K$
  - N nonce or challenge
  - T, serial number from ROM
  - K encryption key from ROM
- **Engine**
  - Issue challenge for a given key
  - Compute correct response
  - Synchronize keys?
- **Can you think of any issues?**

Common Problems

- **Lack of entropy**
- **Lack of key diversity**
- **Re-use of nonce (16-bit counter)**
- **Small key size (40-bit keys)**

How to Prove Who You Are

- **What you know**
  - Passwords
  - Answers to questions that only you know
- **Where you are**
  - IP address, geolocation
- **What you are**
  - Biometrics
- **What you have**
  - Secure tokens, mobile devices

Simple Idea!

- **Register New User**
  - Create password & username
  - System stores list of usernames & passwords
- **System checks credentials at logon**
  - User authenticated

- **Can you think of any issues?**

Password Storage

- **Protecting the password file**
  - Don’t store plain-text passwords (obviously)
  - Don’t use encrypted passwords (dictionary attacks)
  - Use *hashed* passwords
- **Hash a salt along with the password**
  - Store the salt and the hashed salt+password on the server
  - Users with same password will have different password+salt!
- **Hash function (simple definition)**
  - Given x, $f(x)$ is **easy** to compute
  - Given $f(x)$, x is **hard** to compute

Password Storage

- **Need to protect password file**
  - Use OS access control
    - /etc/shadow vs. /etc/passwd
  - Shadow
    - Hash, Salt, Encrypted Password, Time Left, etc.
  - daniel:6d5IKst7M$aWFALmMbbAF72Y8o/nFfXr.0ojd7rIM5Up9Gvj40uvt7SOiMy/dqpcf6n6IjYeJ37zv85Gejl1hDZAScpnJBi.:17304:0:99999:7:::

```
MyLinuxBox root ~ > ll /etc/passwd
-rw-r--r--. 1 root root 1725 Jul 31 23:02 /etc/passwd

MyLinuxBox root ~ > ll /etc/shadow
-rw-------. 1 root root 1187 Jul 16 09:10 /etc/shadow
```

- **Prevent off-line guessing**
  - Hash + salt (otherwise could hash dictionary and match!)
  - Slow hash function (slow down brute force and dictionary attacks)

Passwords and Computer Security

- **In 2012, 76% of network intrusions exploited weak or stolen credentials (username/password)**
  - Source: Verizon Data Breach Investigations Report
- **First step after any successful intrusion: install sniffer or key-logger to steal more passwords**
- **Second step: run cracking tools on password files**
  - Cracking needed because modern systems usually do not store passwords in the clear (how are they stored?)
- **In Mitnick's "Art of Intrusion" 8 out of 9 exploits involve password stealing and/or cracking**

Default Passwords

- **Examples from Mitnick's "Art of Intrusion"**
  - U.S. District Courthouse server: "public" / "public"
  - NY Times employee database: pwd = last 4 SSN digits
- **Mirai IoT botnet**
  - Weak and default passwords on routers and other devices
  - Exploited to form enormous network of compromised devices

Weak Passwords

- **RockYou hack**
  - “Social gaming” company
  - Database with 32 million user passwords from partner social networks
- **Passwords stored in the clear**
  - December 2009: entire database hacked using an SQL injection attack and posted on the Internet
- **One of many such examples!**

Sidebar: SQL Injection

- **Web app does not filter or validate the user input**
- **An attacker sends malformed SQL query to the underlying database:**
  - `statement = "SELECT * FROM users WHERE name = '" + userName + "';"`
  - `userName = ' OR '1'='1' –`
  - `SELECT * FROM users WHERE name = '' OR '1'='1' -- ';`
  - Query selects all and comments out other code!
- **Top web app security problem**

Weak Passwords

- **Hacked from rockyou**

Password Popularity – Top 20

Rank	Password	Number of Users with Password (absolute)
1	123456	290731
2	12345	79078
3	123456789	76790
4	Password	61958
5	iloveyou	51622
6	princess	35231
7	rockyou	22588
8	1234567	21726
9	12345678	20553
10	abc123	17542

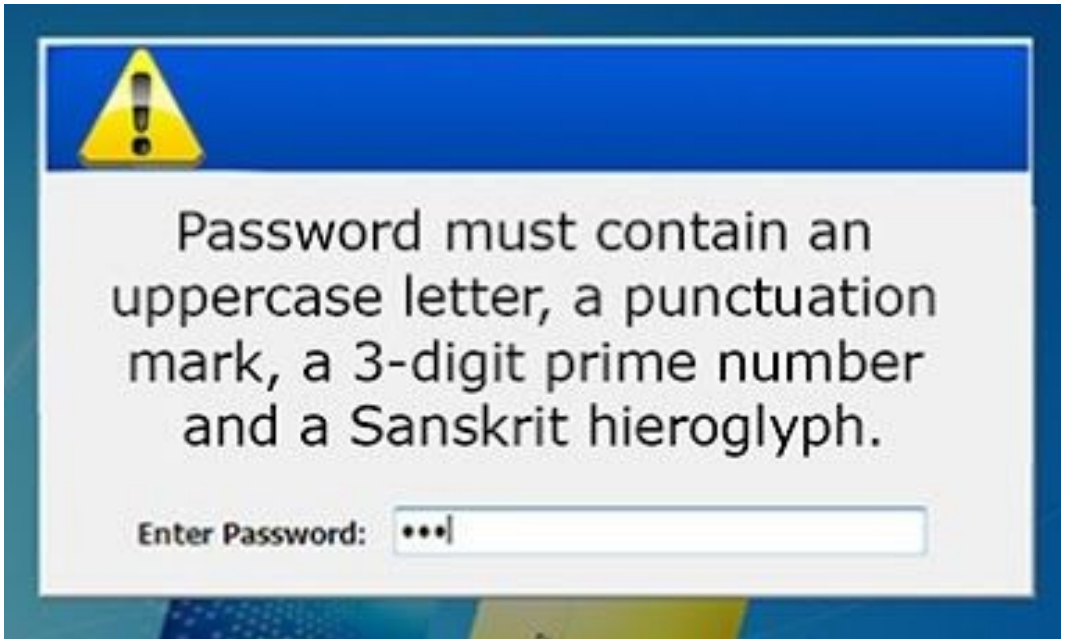
Rank	Password	Number of Users with Password (absolute)
11	Nicole	17168
12	Daniel	16409
13	babygirl	16094
14	monkey	15294
15	Jessica	15162
16	Lovely	14950
17	michael	14898
18	Ashley	14329
19	654321	13984
20	Qwerty	13856

Weak Passwords

Type	Name (Order by: Uploaded, Size, ULed by, SE, LE) View: **Single** / Double	SE	LE
Other (Other)	Pwned Passwords v1.0 from Troy Hunt Uploaded 08-04 2017, Size 5.31 GiB, ULed by lewisje	23	0
Other (E-books)	How to Hack Wi-Fi Passwords Uploaded 07-18 2018, Size 201.89 KiB, ULed by bookflare	21	1
Other (Other)	Pwned Passwords Update 1 from Troy Hunt Uploaded 08-04 2017, Size 250.15 MiB, ULed by lewisje	11	0
Other (Other)	Pwned Passwords Update 2 by Troy Hunt Uploaded 08-08 2017, Size 7.63 MiB, ULed by lewisje	10	8
Other (Other)	wpa.1.2.billion.passwords.for.wifi.wpa.pentesting Uploaded 05-24 2016, Size 13.45 GiB, ULed by marcola15	10	1
Other (Other)	CrackStation.Human.Passwords.Only Uploaded 02-23 2013, Size 246.02 MiB, ULed by crackstation	5	1
Other (Other)	Learn Cracking Wi-fi Passwords Keys (WEP WPA WPA2) Uploaded 03-31 2017, Size 113.43 MiB, ULed by sumi	4	0
Other (Other)	Android Password Cracking Expert - Crack passwords on the go Uploaded 01-09 07:03, Size 289.93 MiB, ULed by tuts756	4	1
Applications (Windows)	Xampp 1.7.8 (32Bit) Installer Full Version + Passwords - {RedDra Uploaded 07-04 2011, Size 63.23 MiB, ULed by RedDr4gon	4	0
Other (Other)	5.7 million passwords list 2015 to 2019 (August 2019) [Ny2rogen] Uploaded 08-25 00:30, Size 26.03 MiB, ULed by Ny2rogen	3	0
Other (Other)	Linkedin SHA1 passwords Uploaded 06-06 2012, Size 240.2 MiB, ULed by *Anonymous*	2	0

Password Usability

- **Classic recommendation:**
  - > 8 characters
  - At least 3: digits, lower/upper case, symbol
  - No dictionary words
  - Change every 3 months
  - Can’t repeat passwords
-



Password Usability

- **But …**
  - Frustrated users and less security
- **Burdens of devising, learning, forgetting passwords**
  - Users construct passwords insecurely, write them down
  - Small changes to old passwords (classic OldPassword + Number)
  - Heavy password re-use across systems
- **Password managers can help**

More Password Issues

- **Credential Stuffing**
  - using stolen credentials on other sites
- **No rate limiting**
  - Website allows brute force (automated guesses)
- **No multi-factor authentication**
  - Just password is enough
- **Weak password recovery mechanisms**
  - Remember the Palin email hack?
- **Application timeouts too long**
  - Did you know that sudo lasts 15 minutes?

Even More Password Issues

- **Keystroke loggers**
  - Hardware
  - Software (spyware)
- **Shoulder surfing**
- **Same password at multiple sites**
  - One breach becomes many!
- **Broken implementations**
  - TENEX timing attack

Examples from One Company

TENEX Timing Attack

- **Old 1970's OS**
- **Char-by-char comparison**
  - AAAAAA vs. SECRET: stop at 1.
  - SAAAAA vs. SECRET: stop at 2.
- **Attacker sees time taken to compare**
  - Implementation language makes the wrong thing simple and the right thing complex.

```
for (i = 0;i < 16;++i)
  if (x[i] != y[i]) return 0;
return 1;
```

```
uint32 diff = 0;
for (i = 0;i < 16;++i)
  diff |= x[i] ^ y[i];
return 1 & ((diff-1) >> 8);
```

TENEX Timing Attack

- **Objection!**
  - Timings are noisy!
- **Can noise stop all attacks?**
  - Noise must stop *all* information flow
- **Attacker can use statistics to reduce noise**
- **Attacker uses methods to amplify signal**
  - Cross page boundary (page fault is much longer)

Examples of Successful Attacks

- **2005 Tromer–Osvik–Shamir:**
  - 65ms to steal Linux AES key used for hard-disk encryption.
- **2013 Al Fardan–Paterson:**
  - “LuckyThirteen: breaking the TLS and DTLS record protocols” steals plain-text using decryption timings.
- **2014 van de Pol–Smart–Yarom:**
  - Steals Bitcoin key from timings of 25 OpenSSL signatures.
- **2016 Yarom–Genkin–Heninger:**
  - “CacheBleed” steals RSA secret key via timings of OpenSSL.

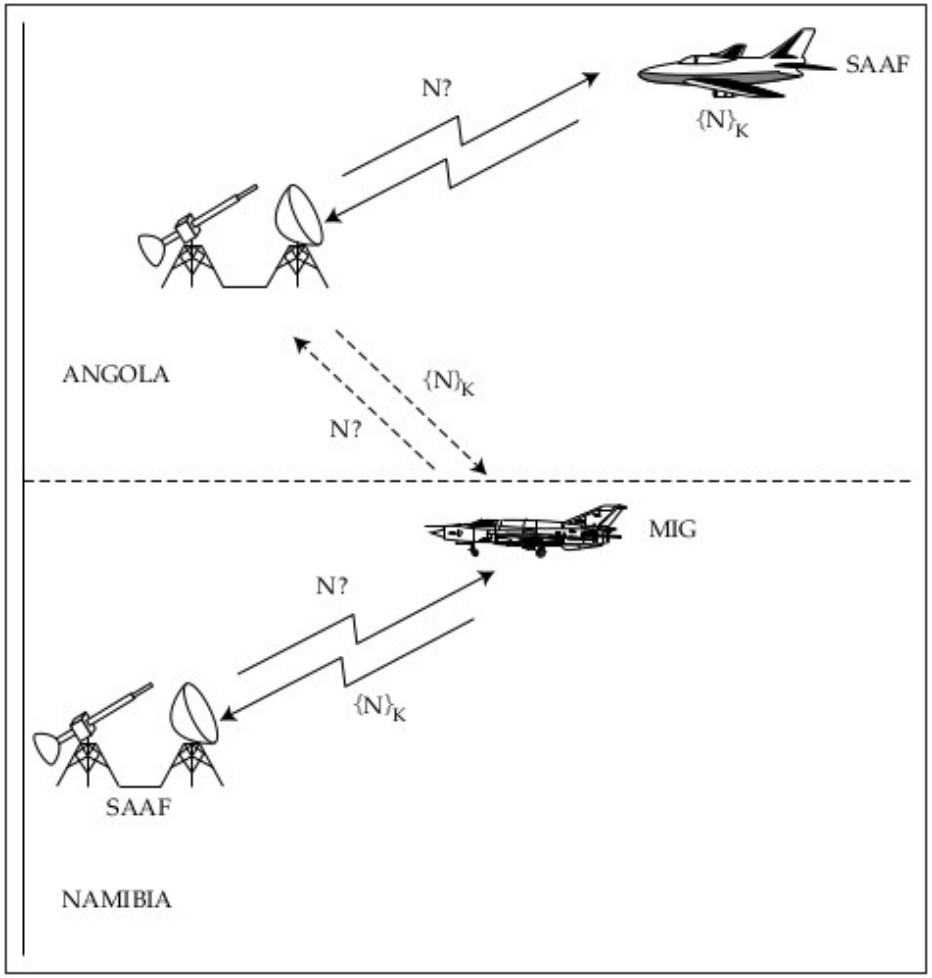
Even More Issues

- **Usability**
  - Hard-to-remember passwords?
  - Carry a physical object all the time?
- **Denial of service**
  - Attacker tries to authenticate as you, account locked after three failures
- **Social engineering**

Protocol Weakness “MIG-in-the-middle”

- **IFF (identify friend or foe)**
  - Authenticate friendly aircraft to air defense
  - Electronic warfare
- **South African Border War**
  - Multi-party warfare: Nambia, South Africa, Angola, Cuba
  - SAAF deployed bombers with IFF
  - Cuban and Angolan allies tailed SAAF
  - Suddenly Nambian allies break through SAAF defenses

“MIG-in-the-middle”



1) MIG records IFF challenge
2) Transmits challenge to Angolan ground forces
3) Ground challenges bomber which replies
4) Ground transmits password to MIG
5) MIG flies into enemy territory

MAT HONAN’S EPIC HACK

Consequences of Linking Accounts

Summary of Mat Honan’s Wired article

Mat Honan

- **Mat Honan is a senior staff writer with WIRED.**
- **Reasonably tech savvy**
  - Early adopter Twitter, Gmail, Amazon
  - Long random passwords
  - Setup important backup emails
- **Journalist, so personal info is semi-public**

The Hack

- **In >1 hour, complete takeover**
  - Google Account taken over and deleted
  - Apple Account taken over iPhone, iPad, MacBook wiped and locked
  - Twitter taken over, tweeting racist slurs

How Did This Happen?

- **Timeline**
  - **4:33 PM** spoofed call to Apple care support
  - **4:50** reset Apple @me.com email
  - **4:52** used @me.com to reset Gmail password
  - **5:00** used “Find My iPhone” to wipe phone
  - **5:01** wiped iPad
  - **5:02** used Gmail to reset Twitter password
  - **5:05** wiped Macbook
  - **5:10** deleted Google account

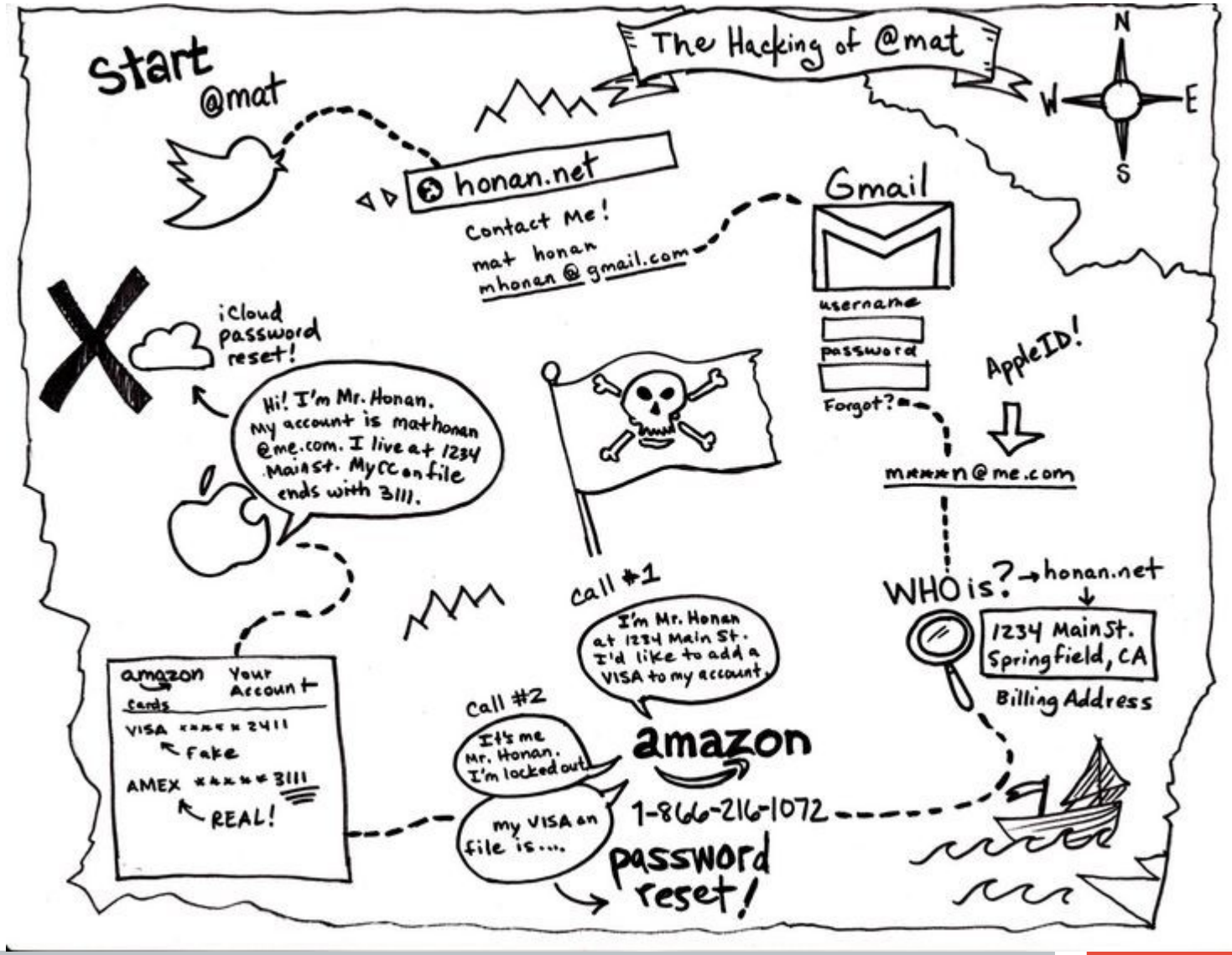
How Did This Happen?

- **Hacker did a lot of background:**
  - Twitter links to blog (with Gmail contact info)
  - Tries out Gmail's password recovery
  - Gmail shows m******n@me.com (obviously Apple email)
  - Got address from whois lookup (on Mat's blog)
  - Got credit card number from Amazon

How Did This Happen?

- **Accessing Apple Account:**
  - Name
  - Email address (just the address, not access)
  - Billing address
  - Last 4 digits of credit card
- **Accessing Amazon Credit Card Number:**
  - Name
  - Email address
  - Billing address

Treasure Map!



Remedies

- **Entangled accounts might backfire!**
  - Each vendor has a different policy!
  - No standardization of assets
  - Attacker can piece together enough info by exploiting this
- **Use multi-factor authentication!**
  - Just SMS + Password on Gmail would have stopped this attack
- **Back-up devices**
  - Remote wipes and locks make modern devices susceptible to attacks
  - Run you own back up on a separate device

What was the point?

- **Twitter DM from hacker “Phobia”:**
  - We wanted the “cool” 3 character handle @mat
  - We wanted to “f things up”
- **Lesson**
  - One man’s trash…
    - Your assets might be worth more to others than to you
  - It’s easier than you think…
    - Your pizza delivery guy can do this (name, address, email, CC#)
    - We hand out this info to everyone all the time

Improving Passwords

- **Add biometrics**
  - For example, keystroke/mouse dynamics or voice print
- **Graphical passwords**
  - Goal: easier to remember? no need to write down?
- **Password managers**
  - Examples: LastPass, built into browsers
  - Can have security vulnerabilities...
- **Two-factor authentication**
  - Leverage phone (or other device) for authentication

Improving Passwords (More)

- **Mutual Authentication**
  - User authenticates *and* site authenticates (prevent phishing)
- **Trusted Path**
  - Guarantee user only communicates with OS (CTRL+ALT+DEL)
- **Display number of failed attempts**
  - First try fails, second succeeds
  - But OS shows one login attempt!
- **Timeouts and Limits**
  - Prevent online guessing

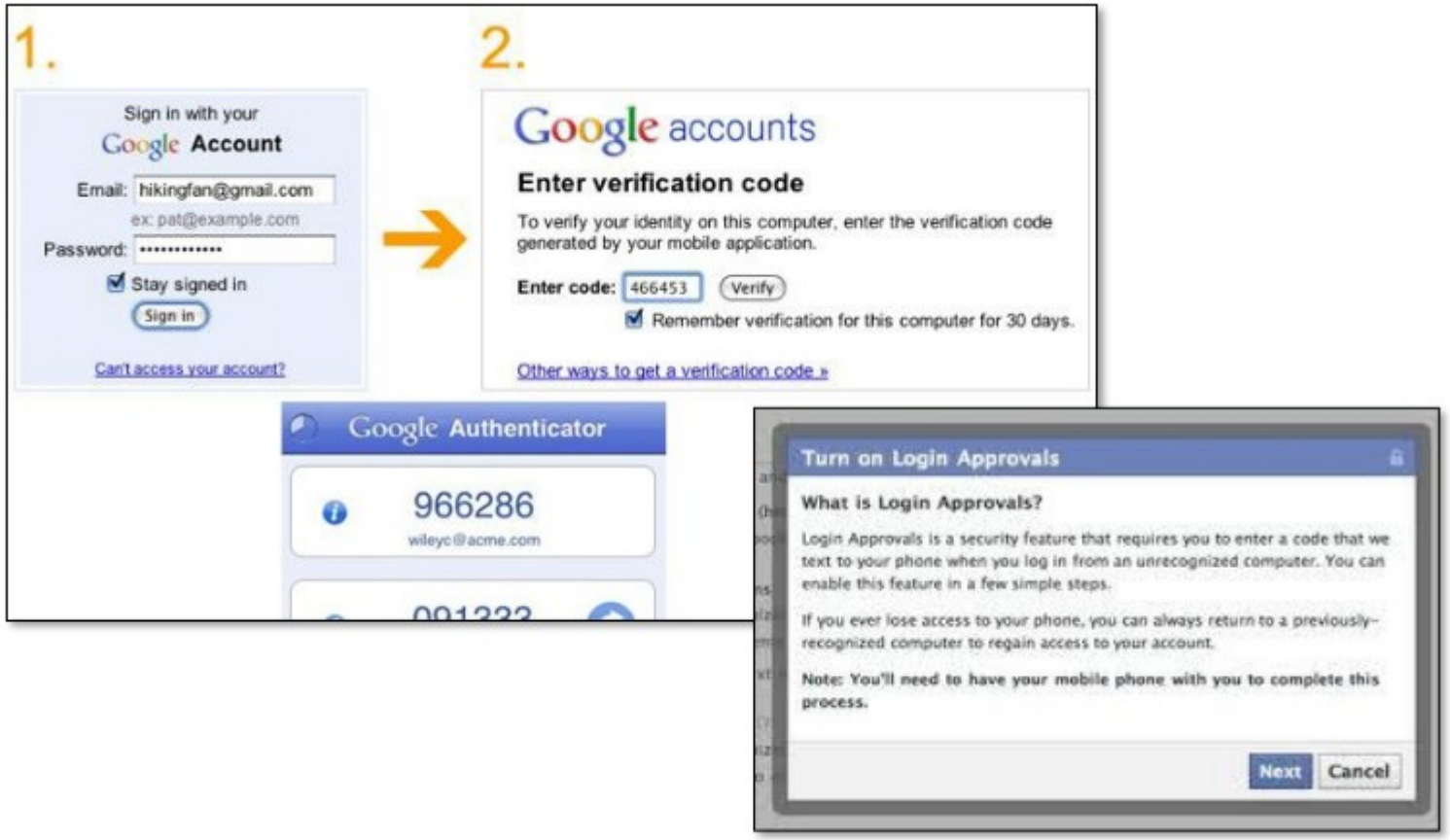
Multifactor Authentication

- **Is it multi-factor?**
  - Password + Secret Question
  - Password + SMS Text
  - Fingerprint + Password
  - Smart-card + Fingerprint

Classic Bank Multi-factor

- **Protocol uses Server, Password Generator, User**
  - S → U: N
  - U → P: N, PIN
  - P → U: $\{N, PIN\}_K$
  - U → S: $\{N, PIN\}_K$
- **User knows: PIN**
- **User has: Password Generator**

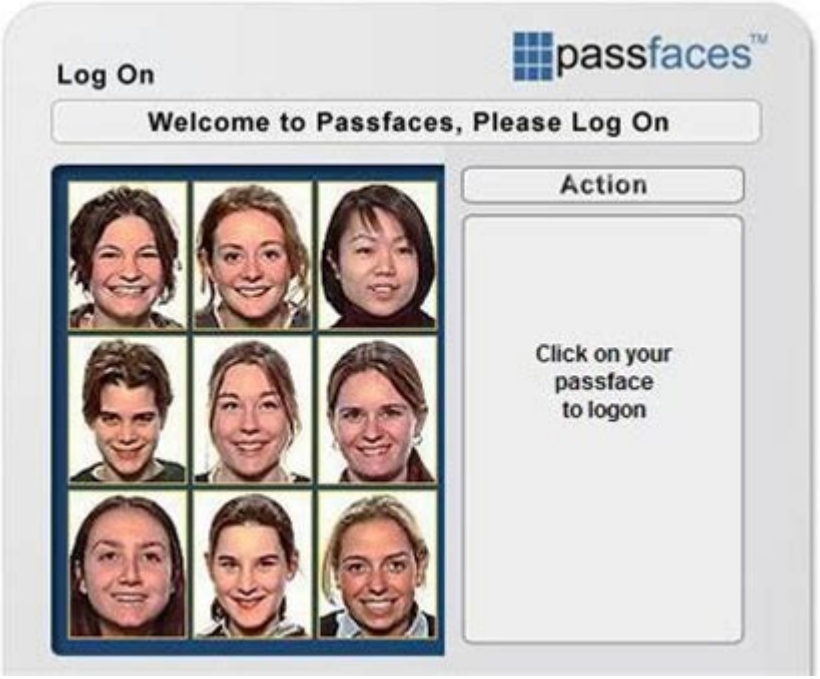
Multifactor Authentication



Hardware Two-Factor

Graphical Passwords

- **Many variants… one example:**
  - Passfaces assumption: easier to recall faces



- **Problem:**
  - Users choose predictable faces

Graphical Passwords

- **Draw on the image (Windows 8)**
- **Users choose predictable points/lines**

Unlock Patterns

- **Problems:**
  - Predictable patterns
  - (sound familiar by now??)
- **Smear patterns**
- **Side channels:**
  - accelerometer and gyroscope

What About Biometrics?

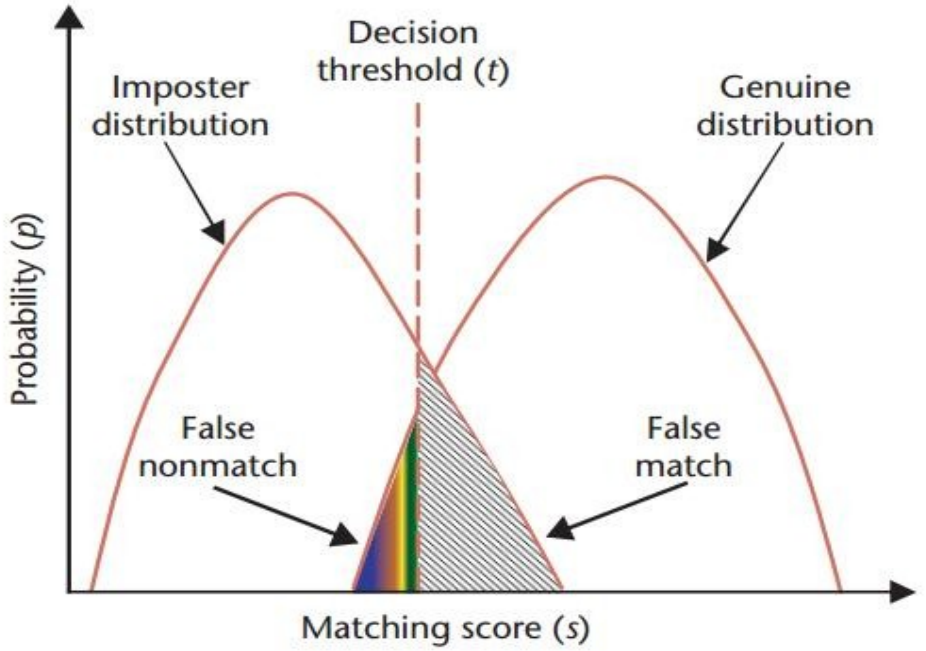
- **Authentication: What you are**
- **Unique identifying characteristics**
  - Biological and physiological: Fingerprints, iris scan
  - Behaviors characteristics: Handwriting, typing, gait
- **Advantages:**
  - Nothing to remember
  - Passive
  - Can’t share (generally)
  - With perfect accuracy, could be fairly unique

Issues with Biometrics

- **Private, but not secret**
  - Maybe left on your glass, door handle, etc.
  - Shared between multiple systems?
- **Revocation is impossible?**
  - Sorry, your iris has been compromised, please create a new one...
- **Physically identifying**
  - Can prevent private use or trace users across systems
- **Birthday paradox**
  - With false accept rate of 1 in a million, probability of false match is above 50% with only 1609 samples

Issues with Biometrics

- **Trade-offs**
  - FMR = # false matches / # false attempts
  - FNMR = # rejected true matches / # true attempts



Madrid Train Bombing (2004)

- **Ultimate FMR!**
- **False positive identification rate with n persons:**
  - FPIR = 1 - (1 – FMR)^n (assuming all prints can be acquired)
- **Fingerprint in the Madrid bombing compared against 530 million entries**
- **A match found by four experts with 100% confidence to a US citizen (B. Mayfield).**
- **They were wrong! Mayfield had not left the US.**
- **Criteria for matching features had to be re-appraised.**

New Attacks Everyday!

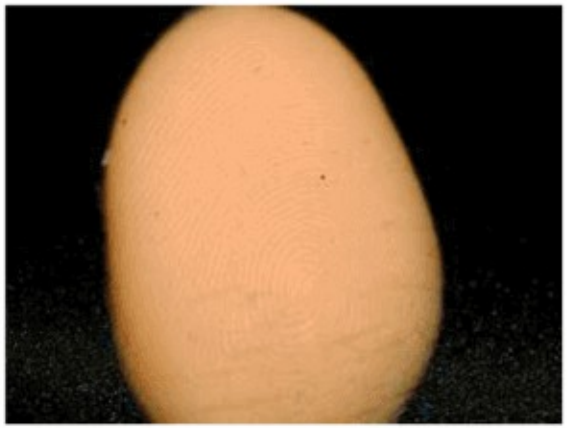
(a)

(b)

(c)

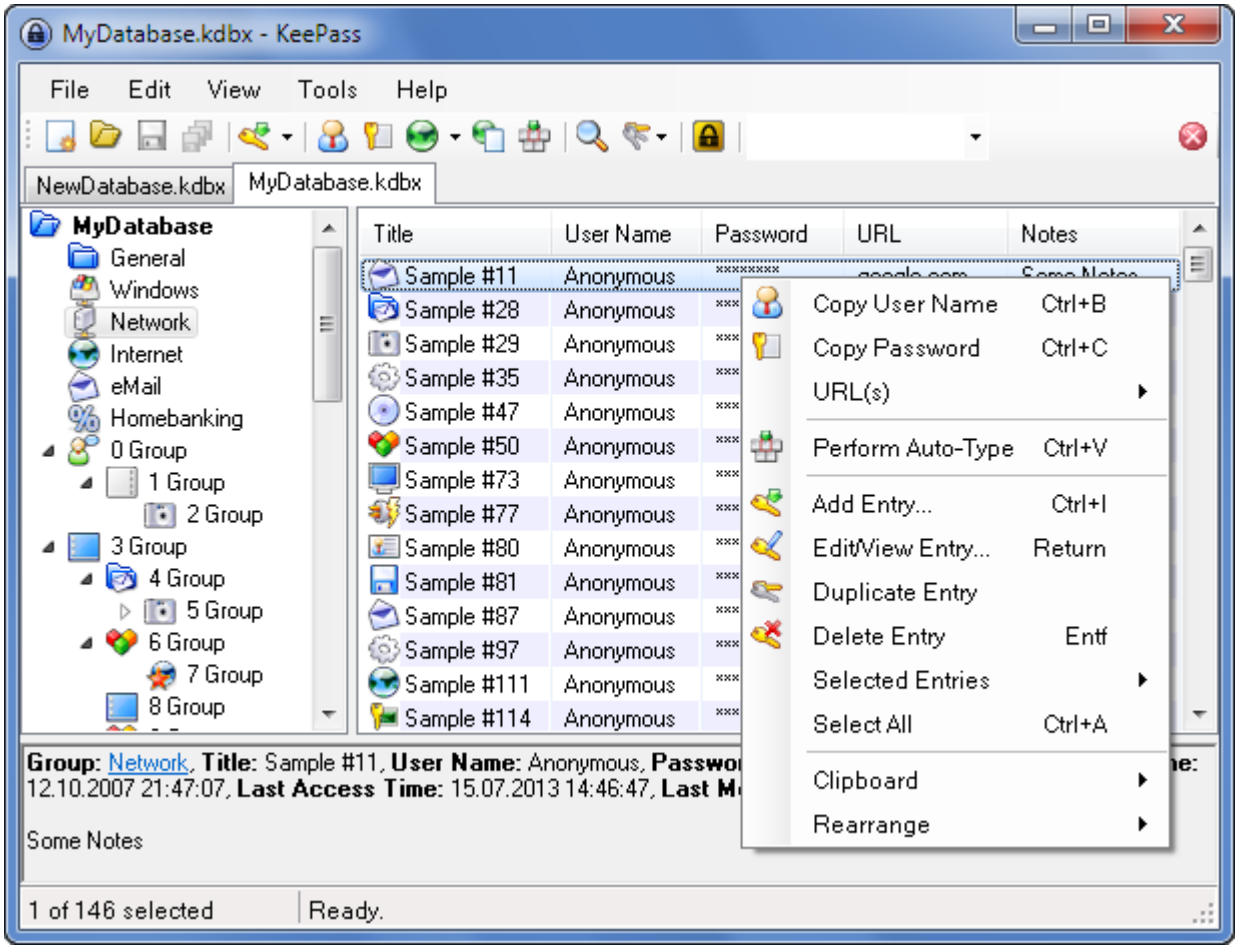
(d)

(e)

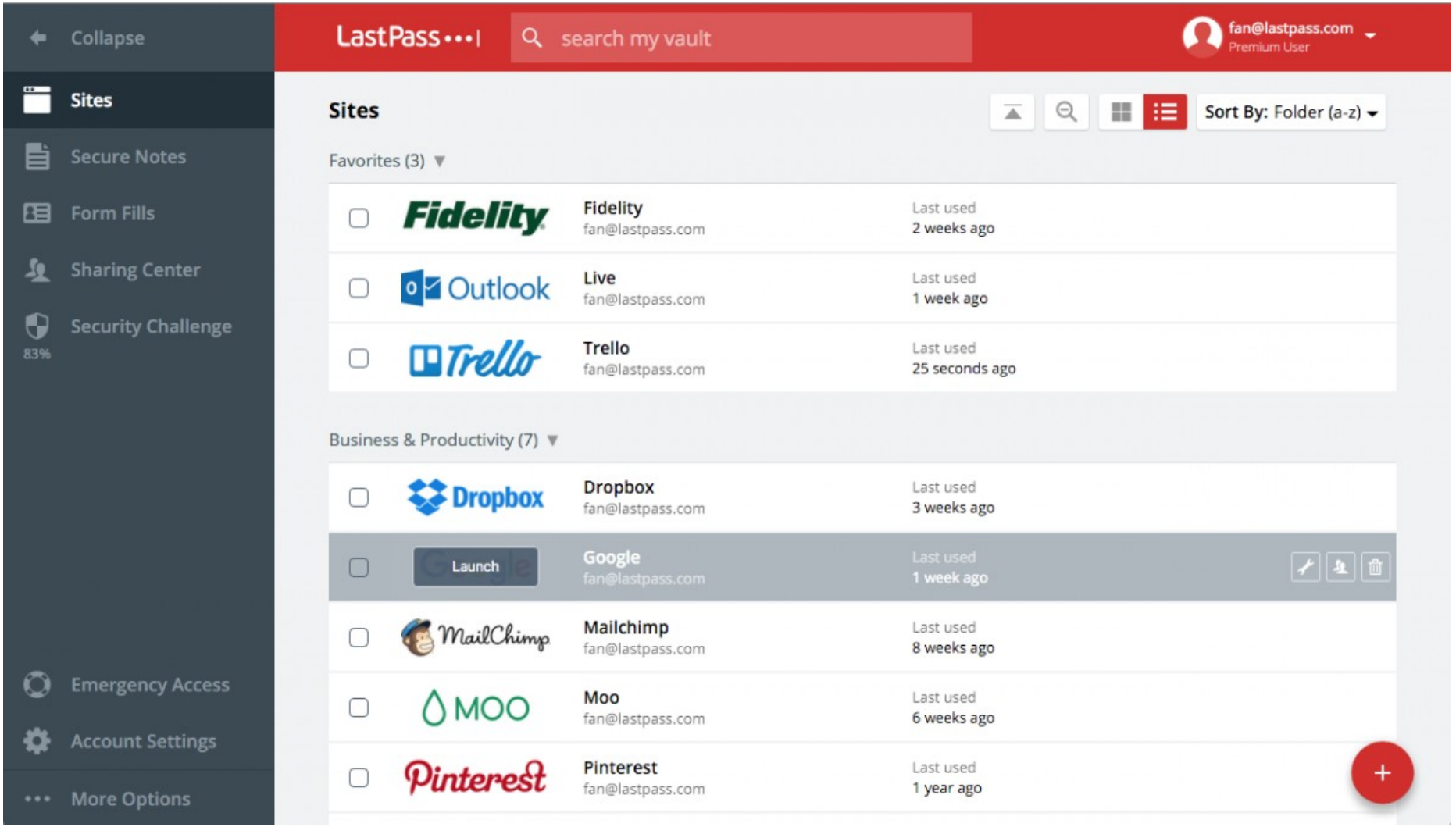


(f)

Password Managers



Password Managers



Password Managers

- **Advantages:**
  - Generate secure passwords
    - Arbitrary rules and lengths
  - Remember old passwords
  - Auto-fill via Browser extension
- **Drawbacks:**
  - Tasty target! All passwords in one place
  - Implementation bugs, break all passwords
  - Trust cloud database with your secrets?