CSci 4271W
Development of Secure Software Systems
Day 23: Identity and Electronic Voting

Stephen McCamant

University of Minnesota, Computer Science & Engineering

Outline

Error rate trade-offs, cont'd

Web authentication

Names and Identities

Elections and their security

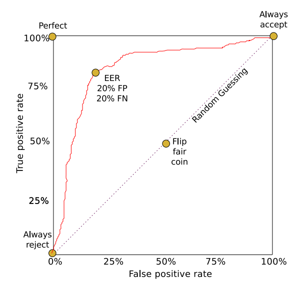
System security of electronic voting

End-to-end verification

Imperfect detection

- Many security mechanisms involve imperfect detection/classification of relevant events
- Biometric authentication
- Network intrusion detection
- Anti-virus (malware detection)
- Anything based on machine learning

Error rates: ROC curve



Extreme biometrics examples

- `exact_iris_code_match`: very low false positive (false authentication)
- `similar_voice_pitch`: very low false negative (false reject)

Where are these in ROC space?

B `return REJECT;`

E `return ACCEPT;`

F `if (rand() & 1) return ACCEPT; else return REJECT;`

C `if (iris()) return ACCEPT; else return REJECT;`

G `if (pitch()) return ACCEPT; else return REJECT;`

A `if (iris()) return REJECT; else return ACCEPT;`

D `if (iris() && pitch()) return ACCEPT; else return REJECT;`

H `if (iris() || pitch()) return ACCEPT; else return REJECT;`

Outline

Error rate trade-offs, cont'd

Web authentication

Names and Identities

Elections and their security

System security of electronic voting

End-to-end verification

Per-website authentication

- Many web sites implement their own login systems
  - \+ If users pick unique passwords, little systemic risk
  - – Inconvenient, many will reuse passwords
  - – Lots of functionality each site must implement correctly
  - – Without enough framework support, many possible pitfalls

Building a session

- HTTP was originally stateless, but many sites want stateful login sessions
- Built by tying requests together with a shared session ID
- Must protect confidentiality and integrity

Session ID: what

- Must not be predictable
  - Not a sequential counter
- Should ensure freshness
  - E.g., limited validity window
- If encoding data in ID, must be unforgeable
  - E.g., data with properly used MAC
  - Negative example: crypt(username || server secret)

Session ID: where

- Session IDs in URLs are prone to leaking
  - Including via user cut-and-paste
- Usual choice: non-persistent cookie
  - Against network attacker, must send only under HTTPS
- Because of CSRF, should also have a non-cookie unique ID

Session management

- Create new session ID on each login
- Invalidate session on logout
- Invalidate after timeout
  - Usability / security tradeoff
  - Needed to protect users who fail to log out from public browsers

Account management

- Limitations on account creation
  - CAPTCHA? Outside email address?
- See previous discussion on hashed password storage
- Automated password recovery
  - Usually a weak spot
  - But, practically required for large system

Client and server checks

- For usability, interface should show what's possible
- But must not rely on client to perform checks
- Attackers can read/modify anything on the client side
- Easy example: item price in hidden field

Direct object references

- Seems convenient: query parameter names resource directly
  - E.g., database key, filename (path traversal)
- Easy to forget to validate on each use
- Alternative: indirect reference like per-session table
  - Not fundamentally more secure, but harder to forget check

Function-level access control

- E.g. pages accessed by URLs or interface buttons
- Must check each time that user is authorized
  - Attack: find URL when authorized, reuse when logged off
- Helped by consistent structure in code

Outline

- Error rate trade-offs, cont'd
- Web authentication
- Names and Identities
- Elections and their security
- System security of electronic voting
- End-to-end verification

Accounts versus identities

- "Identity" is a broad term that can refer to a personal conception or an automated sytem
- "Name" is also ambiguous in this way
- "Account" and "authentication" refer unambiguously to institutional/computer abstractions
- Any account system is only an approximation of the real world

Real human names are messy

- Most assumptions your code might make will fail for someone
  - ASCII, length limit, uniqueness, unchanging, etc.
- So, don't design in assumptions about real names
- Use something more computer-friendly as the core identifier
  - Make "real" names or nicknames a presentation aspect

Zooko's triangle

- Claims (2001) it is hard/impossible for a naming scheme to be simultaneously:
  - Human-meaningful
  - Secure
  - Decentralized
- Too imprecise to be definitively proven/refuted
  - Blockchain-based name systems are highest-profile claimed counterexamples
- A useful heuristic for seeing design tensions

Identity documents: mostly unhelpful

- "Send us a scan of your driver's license"
  - Sometimes called for by specific regulations
  - Unnecessary storage is a disclosure risk
  - Fake IDs are very common

Identity numbers: mostly unhelpful

- Common US example: social security number
- Variously used as an identifier or an authenticator
  - Dual use is itself a cause for concern
- Known by many third parties (e.g., banks)
- No checksum, guessing risks
- Published soon after a person dies

"Identity theft"

- The first-order crime is impersonation fraud between two other parties
  - E.g., criminal trying to get money from a bank under false pretenses
- The impersonated "victim" is effectively victimized by follow-on false statements
  - E.g., by credit reporting agencies
  - These costs are arguably the result of poor regulatory choices
- Be careful w/ negative info from 3rd parties

Backup auth suggestion: use time

- Need for backup often comes for infrequently-used accounts
- May be acceptable to slow down recovery if it reduces attack risk
  - Account recovery is a hassle anyway
- Time can allow legitimate owner to notice malicious request

Outline

- Error rate trade-offs, cont'd
- Web authentication
- Names and Identities
- Elections and their security
- System security of electronic voting
- End-to-end verification

Elections as a challenge problem

- Elections require a tricky balance of openness and secrecy
- Important to society as a whole
  - But not a big market
- Computer security experts react to proposals that seem insecure

History of US election mechanisms

- For first century or so, no secrecy
  - Secret ballot adopted in late 1800s
- Punch card ballots allowed machine counting
  - Common by 1960s, as with computers
  - Still common in 2000, decline thereafter
- How to add more technology and still have high security?

Election integrity

- Tabulation should reflect actual votes
  - No valid votes removed
  - No fake votes inserted
- Best: attacker can't change votes
- Easier: attacker can't change votes without getting caught

Secrecy, vote buying and coercion

- Alice's vote can't be matched with her name (unlinkable anonymity)
- Alice can't prove to Bob who she voted for (receipt-free)
- Best we can do to discourage:
  - Bob pays Alice $50 for voting for Charlie
  - Bob fires Alice if she doesn't vote for Charlie

Election verifiability

- We can check later that the votes were tabulated correctly
- Alice, that her vote was correctly cast
- Anyone, that the counting was accurate
- In paper systems, "manual recount" is a privileged operation

Politics and elections

- In a stable democracy, most candidates will be "pro-election"
- But, details differ based on political realities
- "Voting should be easy and convenient"
  - Especially for people likely to vote for me
- "No one should vote who isn't eligible"
  - Especially if they'd vote for my opponent

Errors and Florida

- Detectable mistakes:
  - Overvote: multiple votes in one race
  - Undervote: no vote in a race, also often intentional
- Undetectable mistakes: vote for wrong candidate
- 2000 presidential election in Florida illustrated all these, "wake-up call"

Precinct-count optical scan

- Good current paper system, used here in MN
- Voter fills in bubbles with pen
- Ballot scanned in voter's presence
  - Can reject on overvote
- Paper ballot retained for auditing

Vote by mail

- By mail universal in OR, WA, CO, HI, UT
  - Many other states have lenient absentee systems
  - Some people are legitimately absent
  - Big for a one-time reason in 2020
- Security perspective: makes buying/coercion easy
  - Doesn't appear to currently be a big problem

Vote by web?

- An obvious next step
- But, further multiplies the threats
- No widespread use in US yet
- Unusual adversarial test in D.C. thoroughly compromised by U. Michigan team

DRE (touchscreen) voting

- "Direct-recording electronic": basically just a computer that presents and counts votes
- In US, touchscreen is predominant interface
  - Cheaper machines may just have buttons
- Simple, but centralizes trust in the machine

Adding an audit trail

- VVPAT: voter-verified paper audit trail
- DRE machine prints a paper receipt that the voter looks at
- Goal is to get the independence and verifiability of a paper marking system

Outline

Error rate trade-offs, cont'd
Web authentication
Names and Identities
Elections and their security
System security of electronic voting
End-to-end verification

Trusted client problem

- Everything the voter knows is mediated by the machine
  - (For Internet or DRE without VVPAT)
- Must trust machine to present and record accurately
- A lot can go wrong
  - Especially if the machine has a whole desktop OS inside
  - Or a bunch of poorly audited custom code

Should we use DRE at all?

- One answer: no, that's a bad design
- More pragmatic: maybe we can make this work
  - DREs have advantages in cost, disability access
  - If we implemented them well, they should be OK
  - Challenge: evaluating them in advance

US equipment market

- Voting machines are low volume, pretty expensive
- But jurisdictions are cost-conscious
- Makers are mostly small companies
  - One was temporarily owned by the larger Diebold
- Big market pressures: regulations, ease of administration

Security ecosystem

- Voting fraud appears to be very rare
  - Few elections worth stealing
  - Important ones are watched closely
  - Stiff penalties deter in-US attackers
- Downside: No feedback from real attacks
- Main mechanism is certification, with its limitations

Diebold case study

- Major manufacturer in early 2000s
  - During a post-2000 purchasing boom
  - Since sold and renamed
- Thoroughly targeted by independent researchers
  - Impolitic statement, blood in the water
- Later state-authorized audits found comprehensive problems
  - Your reading: from California

Physical security

- Locked case; cheap lock as in hotel mini-bar
- Device displays management menu on detected malfunction
  - Can be triggered in booth by unspecified use of paperclip
- Tamper-evident seals? Not a strong protection

Buffer overflows, etc.

- Format string vulnerability
  - `"Page %d of %d"`
- Was this audited?

```
TCHAR name;
_stprintf(&name,
          _T("\\Storage Card\\%s"),
          findData.cFileName);
```

Web-like vulnerabilities

In management workstation software:

- SQL injection
- Authentication logic encoded only in enabled/disabled UI elements
  - E.g., buttons grayed out if not administrator
  - Not quite as obviously wrong as in web context
  - But still exploitable with existing tools

OpenSSL mistakes

- Good news: they used OpenSSL
  - Bad news: old, buggy version
- Insufficient entropy in seeding PRNG
  - Good interface from desktop Windows missing in WinCE
- Every device ships with same certificate and password

Election definitions

- Integrity “protected” by unkeyed, non-crypto checksum
- Can change bounding boxes for buttons
  - Without changing checksum!
- Can modify candidate names used in final report
  - E.g. to fix misspelling; security implication mentioned in comment

Secrecy problems

- Limited, since the DRE doesn't see registration information
- But, records timestamp and order of voting
- Could be correlated with hidden camera or corrupted poll worker

Voting machine viruses

- Two-way data flow between voting and office machines
- Hijacking vuln's in software on both sides
- $\rightarrow$ can write virus to propagate between machines
- Leverage small amount of physical access

Subtle ways to steal votes

- Change a few votes your way, revert if the voter notices
  - Compare: flip coin to split lunch
- Control the chute for where VVPAT receipts go
- Exchange votes between provisional and regular voters

Outline

Error rate trade-offs, cont'd

Web authentication

Names and Identities

Elections and their security

System security of electronic voting

End-to-end verification

End-to-end integrity and verification

- Tabulation cannot be 100% public
- But how can we still have confidence in it?
- Cryptography to the rescue, maybe
  - Techniques from privacy systems, others
  - Adoption requires to be very usable

Commitment to values

- Two phases: commit, later open
  - Similar to one use of envelopes
- Binding property: can only commit to a single value
- Hiding property: value not revealed until opened

Randomized auditing

- How can I prove what's in the envelope without opening it?
- n envelopes, you pick one and open the rest
  - Chance $1/n$ of successful cheating
- Better protection with repetition

Election mix-nets

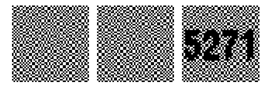
- Independent election authorities similar to remailers
- Multi-encrypt ballot, each authority shuffles and decrypts
- Extra twist: prove no ballots added or removed, without revealing permutation
  - Instance of "zero-knowledge proof"
- Privacy preserved as long as at least one authority is honest

Pattern voting attack

- Widely applicable against techniques that reveal whole (anonymized) ballots
- Even a single race, if choices have enough entropy
  - 3-choice IRV with 35 candidates: 15 bits
- Buyer says: vote first for Bob, then 2nd and 3rd for Kenny and Xavier
  - Chosen so ballot is unique

Fun tricks with paper: visual crypto

- Want to avoid trusted client, but voters can't do computations by hand
- Analogues to crypto primitives using physical objects
- One-time pad using transparencies:



Scantegrity II

- Designed as end-to-end add-on to optical scan system
- Fun with paper 2: invisible ink
- Single trusted shuffle
  - Checked by random audits of commitments