Defeating Malicious Terminals in an Electronic Voting System

Daniel Hanley

Andre dos Santos

Jeff King

Georgia Tech Information Security Center

Overview

- Motivation
- Related Work
- Protocol
- Examples
- Analysis

Motivation

- The Voting Problem
- Traditional Approach
- Electronic Voting

Motivation: The Voting Problem

- Scenario: Alice, a human, wishes to transmit message $c \in C$ to central tallier, Trent.
- Security requirements
  - Anonymity
  - Accuracy
  - etc.

Motivation: Traditional Approach

- Paper-based systems
  - Alice creates physical vote record and relays the vote to Trent.
- Disadvantages
  - Inaccurate
  - Expensive
- Advantages
  - Simple, usable
  - Secure (?)

Motivation: Electronic Voting

- Current state of electronic voting systems
  - Systems entrust untrustworthy voting terminals, volunteers
  - Security policy dictates isolation and physical controls
- Advantages
  - Relatively inexpensive
  - Accurate
- Disadvantages
  - Fails to use public infrastructure
  - Vulnerable to automated attacks
  - Vulnerable to undetectable attacks

Motivation: Electronic Voting

- Current state of electronic voting systems
  - Systems entrust untrustworthy voting terminals, volunteers
  - Security policy dictates isolation and physical controls
- Advantages
  - Relatively inexpensive
  - Accurate
- Disadvantages
  - Fails to use public infrastructure
  - Vulnerable to automated attacks
  - Vulnerable to undetectable attacks

Motivation: Electronic Voting

- **Solution**: Blind signature protocol with trustworthy hardware
  - Direct communication with Trent – infeasible!
  - Trustworthy voting terminals – costly!
  - Personal tamper resistant device – yes!
- **Problem**: How can we establish a trusted path between Alice and her voting device?
  - Direct I/O? Form factor prohibits this.
  - Via voting terminal? No!
  - **CAPTCHA-Voting Protocol?**
- Other schemes (Chaum, Prêt-à-Voter, KHAP)
  - Voter performs verification and auditing steps.

Related Work

- Completely Automated Publicly Available Turing Tests to tell Computers and Humans Apart (CAPTCHAs)
- One-time random substitution

Protocol: Actors

Alice *a human voter*

Trent *a central tallier, trusted to perform complex, anonymous operations on Alice's behalf*

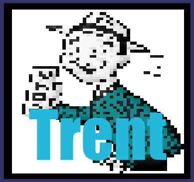
Mallory *an untrusted voting terminal*

Protocol

- Public list of candidates $C = [\, c_1, c_2, \ldots, c_n \,]$
- Public, random set $R = [\, r_1, r_2, \ldots, r_m \,]$ such that $m \geq n$
- Random mapping of candidates to random elements $K : C \rightarrow R$ such that
  - $P(\, K(c) = r_i \,) = P(\, K(c) = r_j \,)$ for all i, j
  - $K^{-1} : R \rightarrow C$
- CAPTCHA transformation function *T(m)* such that Mallory cannot derive *m* from *T(m)*, while Alice may infer *m* from *T(m)*
  - Trent may encode *K* using *T*. This is denoted by *T(K).*

Protocol

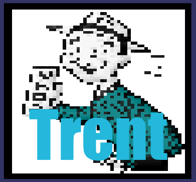
1. Trent generates and sends a CAPTCHA-encrypted ballot.



1.1. $K : C \rightarrow R$

Protocol

1. Trent generates and sends a CAPTCHA-encrypted ballot.

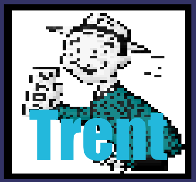


1.1. $K : C \rightarrow R$

1.2. $T(K)$

Protocol

1. Trent generates and sends a CAPTCHA-encrypted ballot.



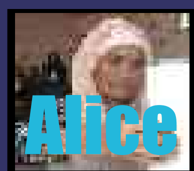
1.1. $K : C \rightarrow R$

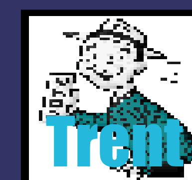
1.2. $T(K)$

1.3. $T(K)$

Protocol

2. Alice responds with the encrypted candidate.





1.1. $K : C \rightarrow R$

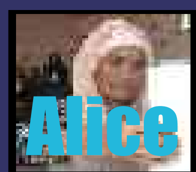
1.2. $T(K)$

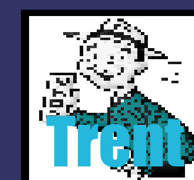
1.3. $T(K)$

2.1. $T^{-1}(\ T(K)\ ) = K$

Protocol

2. Alice responds with the encrypted candidate.





1.1. $K : C \rightarrow R$

1.2. $T(K)$

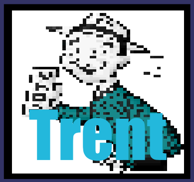
1.3. $T(K)$

2.1. $T^{-1}(\ T(K)\ ) = K$

2.2. $K(c) = r$

Protocol

2. Alice responds with the encrypted candidate.



1.1. $K : C \rightarrow R$

1.2. $T(K)$

1.3. $T(K)$

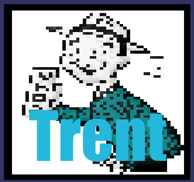
2.1. $T^{-1}(\ T(K)\ ) = K$

2.2. $K(c) = r$

2.3. r

Protocol

3. Trent decrypts Alice's preferred candidate.



1.1. $K : C \rightarrow R$

1.2. $T(K)$

1.3. $T(K)$ $\leftarrow$

2.1. $T^{-1}(T(K)) = K$

2.2. $K(c) = r$

2.3. r $\rightarrow$

3.1. $K^{-1}(r) = c$

Examples

- Text CAPTCHA
- 3D Animation CAPTCHA
- Audio CAPTCHA

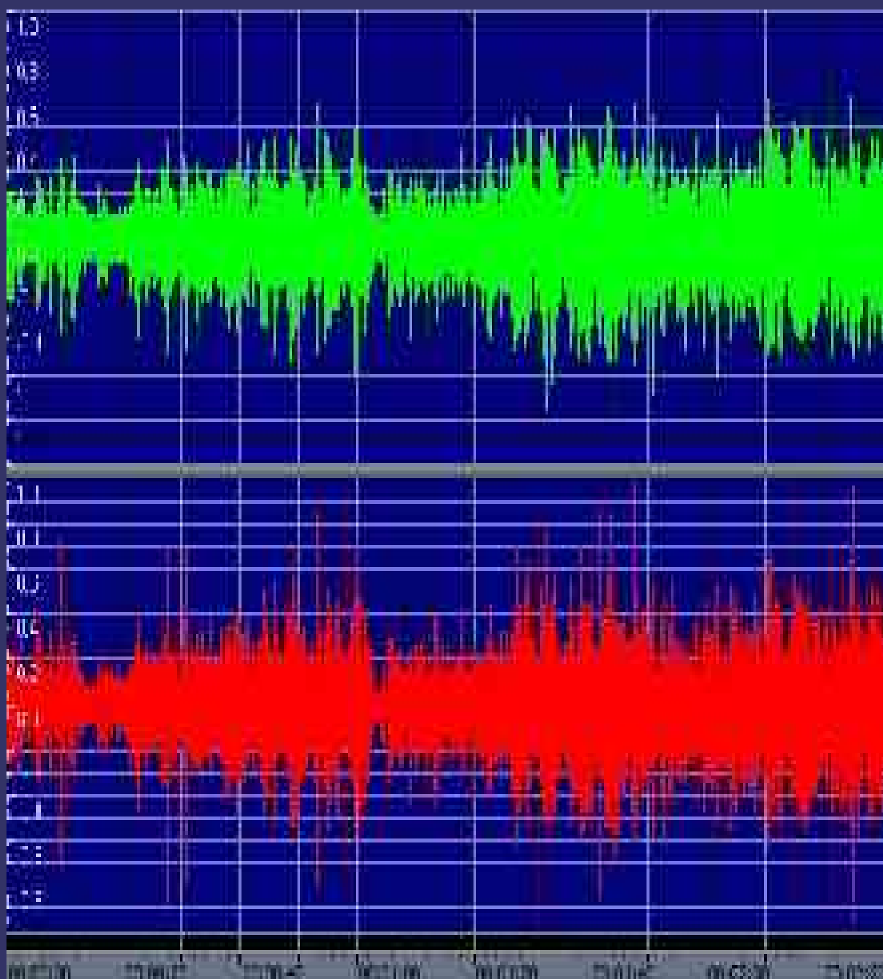
Example: Text CAPTCHA

- *R* consists of distinct regions in image.
- *T* renders mapping as image and contributes noise.

Example: 3D Animation CAPTCHA

- *R* consists of equally sized, contiguous sets of frames.
- *T* renders candidate names in animation.

Example: Audio CAPTCHA



- *K* is a similar, temporal mapping of candidates.
- Audio noise thwarts Mallory.

Analysis

- Fabricated votes
- Human adversaries
- Selective denial of service

Analysis: Fabricated Votes

- Fabricated vote through guessed K
  - Mallory attempts to vote for c' through selection of arbitrary r''.
  - If $|R| = |C|$, then $P(\ K^{-1}(r'') = c'\ ) = 1\ /\ n$.
  - If $|R| > |C|$, then $P(\ K^{-1}(r'') = c'\ ) = 1\ /\ m$.
    - Probability that $K^{-1}(r'')$ is undefined: $(m - n)\ /\ m$
    - Invalid vote → detected attack!
- Fabricated vote through cracked T
  - Mallory increases probability that $P(\ K^{-1}(r'') = c'\ )$.
  - **Solution**: Find a better CAPTCHA?

Analysis: Human Adversary

- Transmission of *T(K)* to a human collaborator
- Time-dependent protocol
- Increased likelihood of detection
- Architectural solutions

Analysis: Selective DoS

- Selective DoS: Mallory discards Alice's vote if it is likely that $c \neq c'$.
- Mallory must learn Alice's preference.
  - Alice and Mallory's location
  - Alice's previous votes
    - **Solution**: Single ballot
  - Fabricated ballot
- Detection of selective denial of service
- Educated guessing

Conclusion

- Human interaction required – no efficient automated attacks
- Easy detection of large-scale attacks
- Comparison to traditional voting systems
- Future work
  - Usability data
  - Broader applications, using this protocol (possibly combined with KHAP) to form a trusted path

Questions?

Questions?