Trustworthy Elections

Joey Dodds

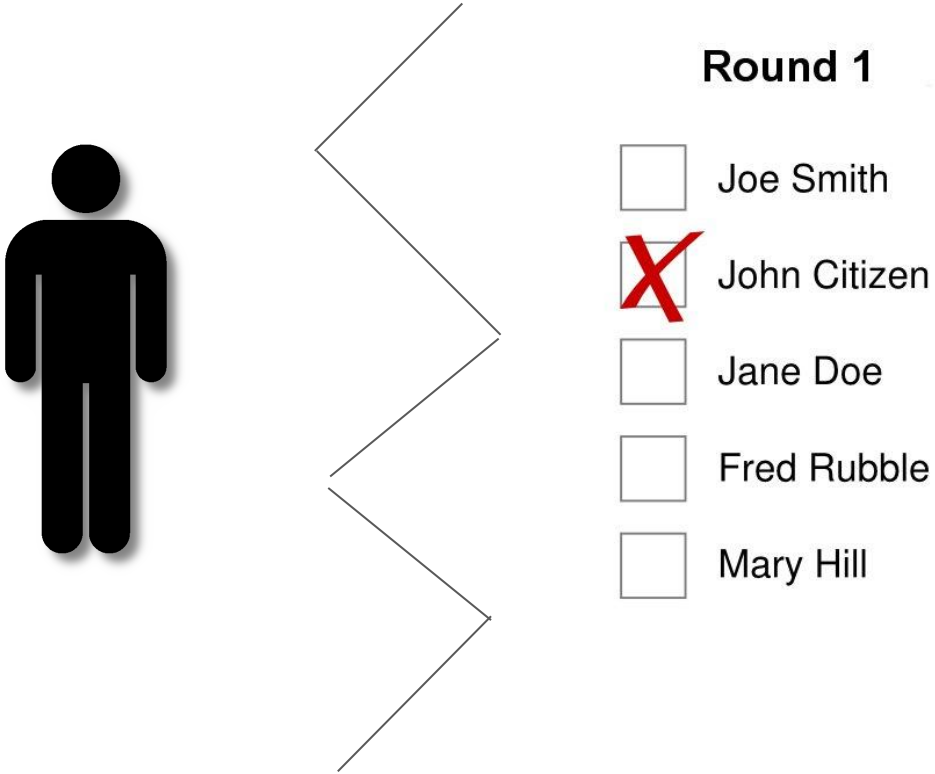
Don’t trust me.

Election Evidence

- Paper Trail
- Quality of paper trail
- Audits of elections
- Tabulation procedures
- Certification of devices and processes
- The outcome

Everyone who votes should be able to trust the outcome of an election

Elections are challenging

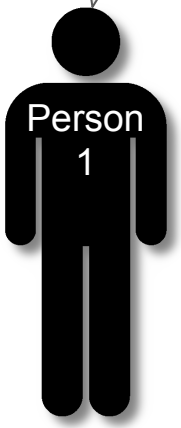


Elections are challenging

Person Name	Vote
Person 1	A
Person 2	B
Person 3	A

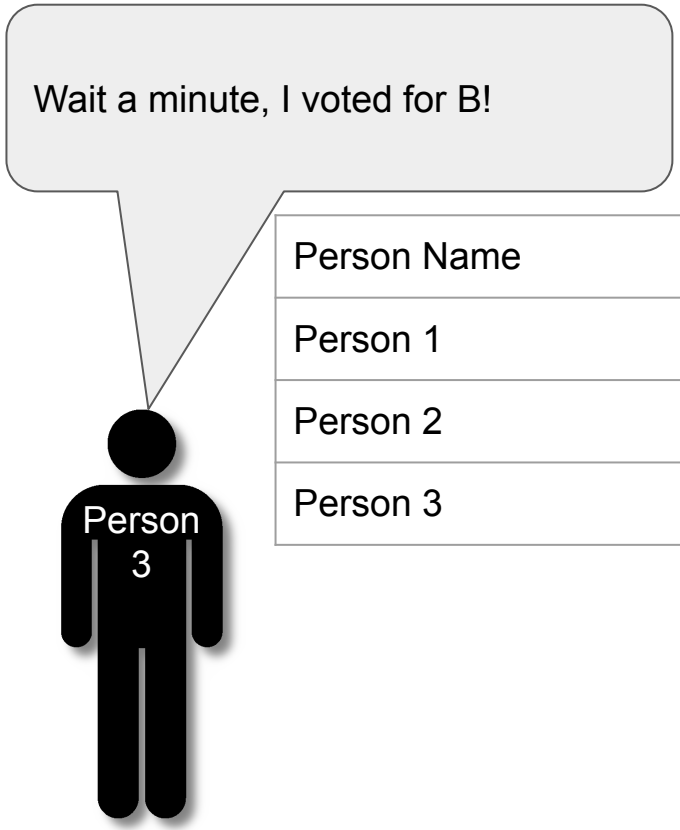
Elections are challenging

Looks good, and I see that A won as reported!



Person Name	Vote
Person 1	A
Person 2	B
Person 3	A

Elections are challenging

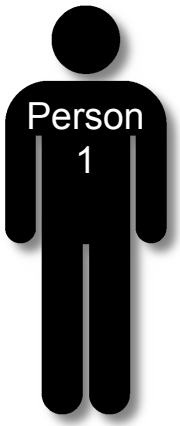


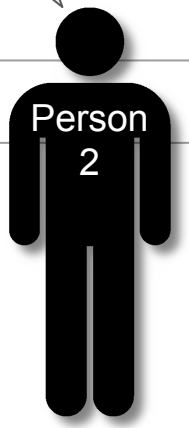
Person Name	Vote
Person 1	A
Person 2	B
Person 3	A

Elections are challenging

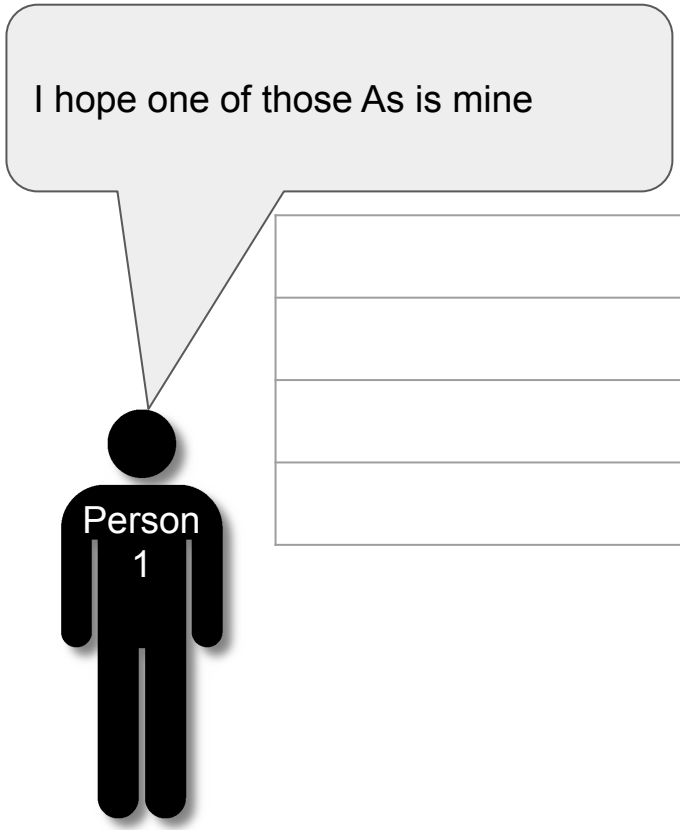
Why didn’t you vote for B? We can’t be friends any more

Person Name	Vote
Person 1	A
Person 2	B
Person 3	A





Elections are challenging



	Vote
	A
	B
	A

Risk Limiting Audits (RLAs)



Risk Limiting Audits (RLAs)



Risk Limiting Audits (RLAs)



Risk Limiting Audits (RLAs)



Risk Limiting Audits (RLAs)



Risk Limiting Audits (RLAs)

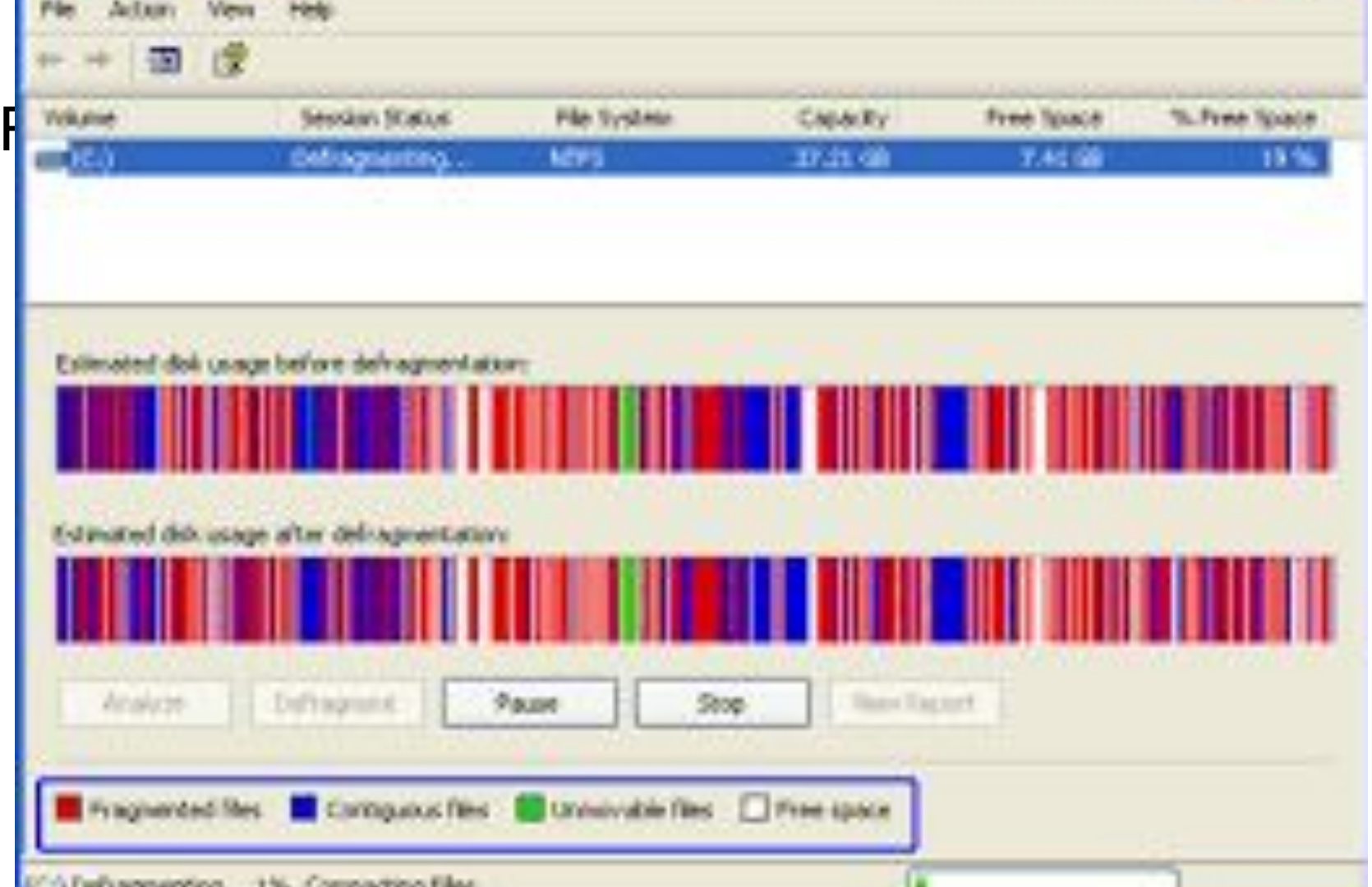


Risk Limiting Audits (RLAs)



Risk Limiting Audits (RLAs)




Pause
Stop

Risk Limiting Audits (RLAs)



Risk Limiting Audits (RLAs)



End to End Verifiable elections



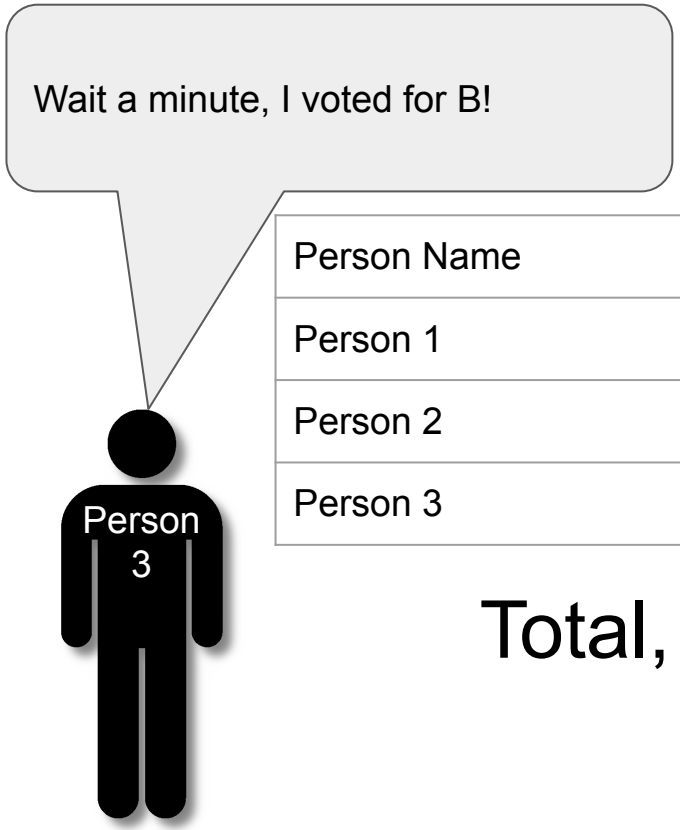
Person Name	Vote
Person 1	A
Person 2	B
Person 3	A

Person Name	Vote
Person 1	A
Person 2	B
Person 3	A

Total, 2 A 1 B

Person Name	Vote
Person 1	A
Person 2	B
Person 3	A * 1000

Total, 1001 A 1 B



Person Name	Vote
Person 1	A
Person 2	B
Person 3	A

Total, 2 A 1 B

Vote A	Vote B
E(1)	E(0)
E(0)	E(1)
E(1)	E(0)

Homomorphic Encryption

$$E(X) \oplus E(Y) = E(X+Y)$$

$$g^x * g^y = g^{x+y}$$

Vote A	Vote B
E(1)	E(0)
E(0)	E(1)
E(1)	E(0)

Total, D(E(1+0+1) A, D(E(0+1+0)) B







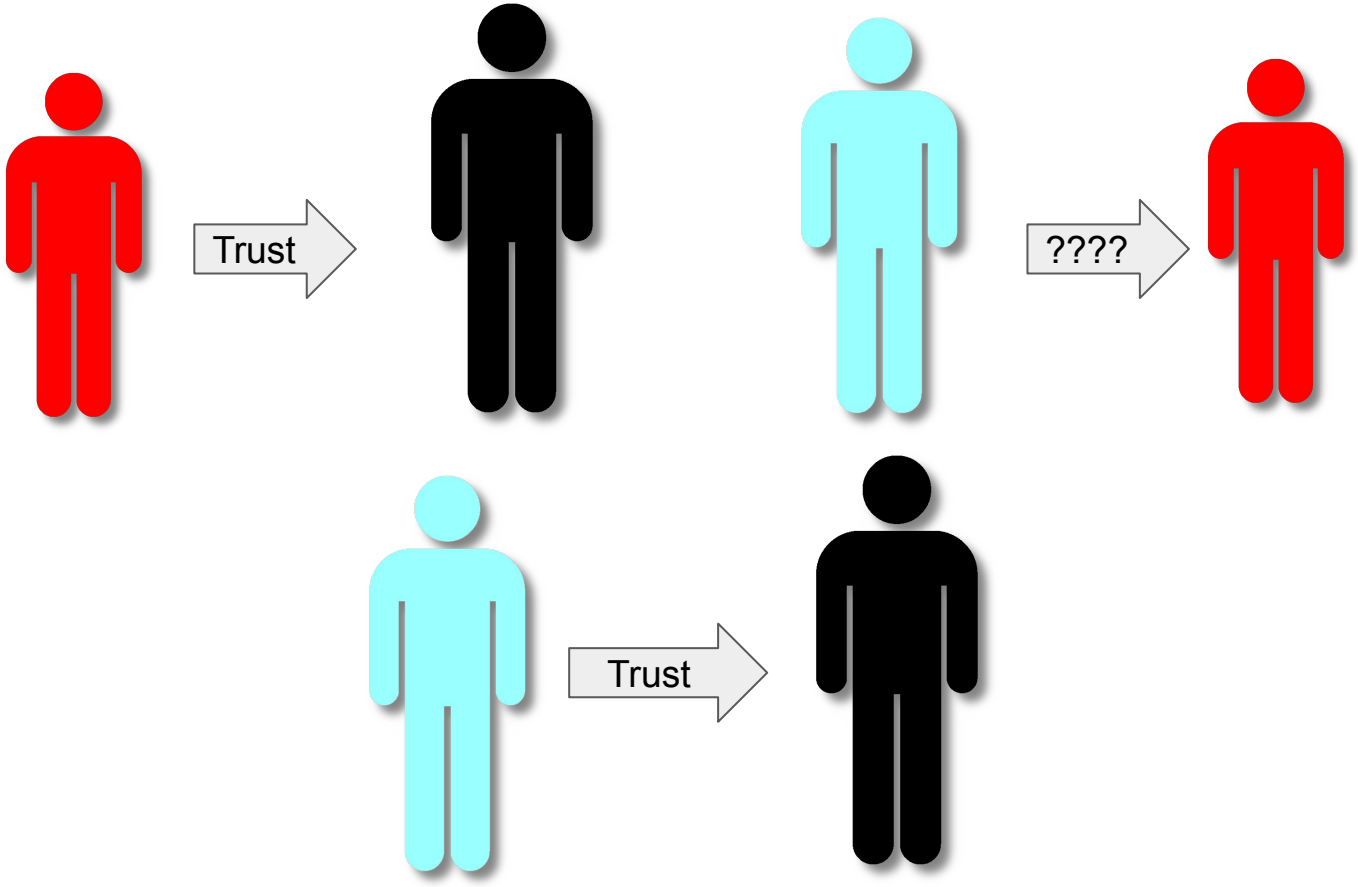


NIZK Proof that (α, β) is an encryption of one (given knowledge of encryption nonce R)

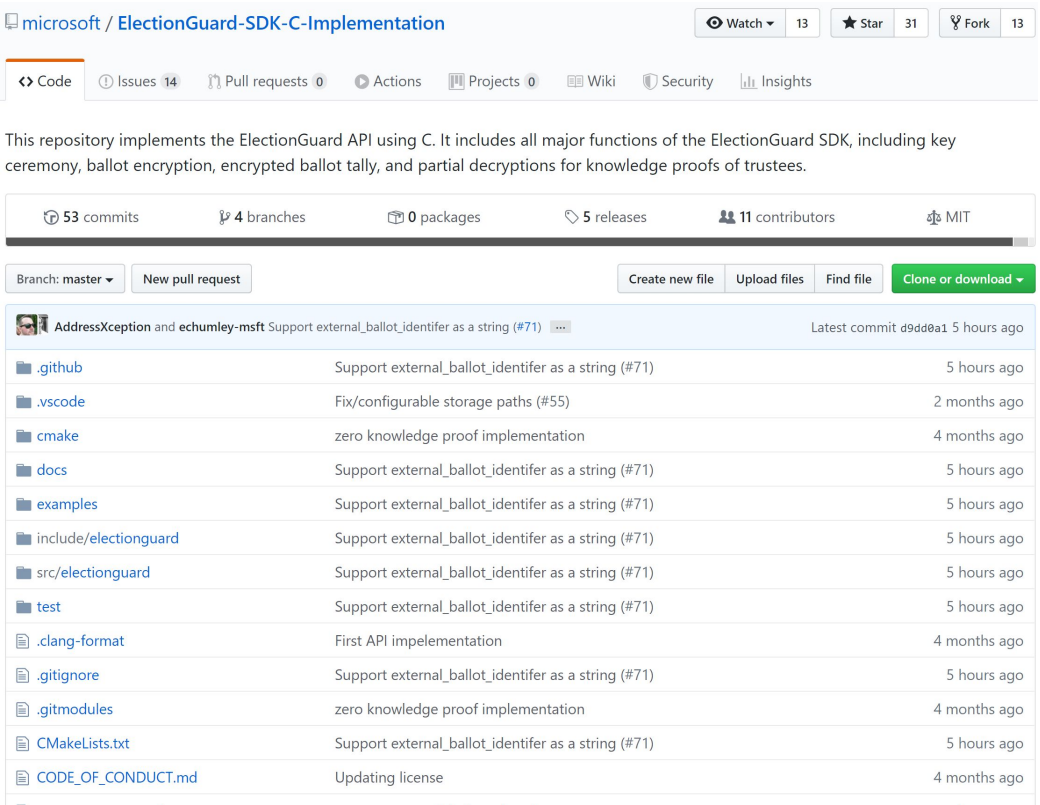
To prove that (α, β) is an encryption of one, $\frac{\beta}{g} \bmod p$ is substituted for β in the above. The verifier can be relieved of the need to perform a modular division by computing $\beta \bar{g} \bmod p$ rather than $\frac{\beta}{g} \bmod p$. As an alternative, the verifier can confirm that $g^c \cdot K^v \bmod p = b \cdot \beta^c \bmod p$ instead of $K^v \bmod p = b \cdot \left(\frac{\beta}{g}\right)^c \bmod p$.

As with many zero-knowledge protocols, if the prover knows a challenge value prior to making its commitment, it can create a false proof. For example, if a challenge c is known to be forthcoming, a prover can generate a random v in the range $0 \leq v < q$ and commit to $(a, b) = \left(\frac{g^v}{\alpha^c} \bmod p, \frac{K^v}{\beta^c} \bmod p\right) = (g^v \alpha^{q-c} \bmod p, K^v \beta^{q-c} \bmod p)$. This selection will satisfy the required checks for (α, β) to appear as an encryption of zero regardless of the values of (α, β). Similarly, setting $(a, b) = \left(\frac{g^v}{\alpha^c} \bmod p, \frac{K^v g^c}{\beta^c} \bmod p\right) = (g^v \alpha^{q-c} \bmod p, K^v g^c \beta^{q-c} \bmod p)$ will satisfy the required checks for (α, β) to appear as an encryption of one regardless of the values of (α, β). This quirk is what enables the Cramer-Damgård-Schoenmakers technique to prove a disjunction of two predicates.

Transitive trust



Election Guard



microsoft / ElectionGuard-SDK-C-Implementation

Watch 13 | Star 31 | Fork 13

Code | Issues 14 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights

This repository implements the ElectionGuard API using C. It includes all major functions of the ElectionGuard SDK, including key ceremony, ballot encryption, encrypted ballot tally, and partial decryptions for knowledge proofs of trustees.

53 commits | 4 branches | 0 packages | 5 releases | 11 contributors | MIT

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

AddressXception and echumley-msft Support external_ballot_identifer as a string (#71) ... Latest commit d9dd0a1 5 hours ago

.github	Support external_ballot_identifer as a string (#71)	5 hours ago
.vscode	Fix/configurable storage paths (#55)	2 months ago
cmake	zero knowledge proof implementation	4 months ago
docs	Support external_ballot_identifer as a string (#71)	5 hours ago
examples	Support external_ballot_identifer as a string (#71)	5 hours ago
include/electionguard	Support external_ballot_identifer as a string (#71)	5 hours ago
src/electionguard	Support external_ballot_identifer as a string (#71)	5 hours ago
test	Support external_ballot_identifer as a string (#71)	5 hours ago
.clang-format	First API impelementation	4 months ago
.gitignore	Support external_ballot_identifer as a string (#71)	5 hours ago
.gitmodules	zero knowledge proof implementation	4 months ago
CMakeLists.txt	Support external_ballot_identifer as a string (#71)	5 hours ago
CODE_OF_CONDUCT.md	Updating license	4 months ago

Why don’t we have these technologies today?

REPORT

Pilot Implementation Study of Risk-Limiting Audit Methods in the State of Rhode Island

SUMMARY: The risk-limiting audit is one of the most important tools states can use to bolster voter confidence and secure elections for 2020 and beyond.



RISK-LIMITING AUDIT CONCLUDES PAPER-BALLOT SYSTEM ACCURATE

(CARTERSVILLE, Ga.) – Georgia’s new, secure paper-ballot system gets high marks for accuracy in risk-limiting audits of last week’s election conducted Tuesday by the Secretary of State’s Office, Bartow County election workers and two nationally recognized organizations.

“An important part of the new voting system is the ability to audit with the use of paper ballots. This feature provides the confidence voters deserve,” said Secretary of State Brad Raffensperger.

Georgia is replacing its 17-year-old electronic voting machines with a system that produces paper ballots. The system’s first statewide use will be during the March 24 Presidential Preference Primary.

The first use of the new system in a Georgia election was Nov. 5 in local elections in six counties, Bartow, Carroll, Catoosa, Decatur, Lowndes and Paulding. More than 27,000 votes were cast in the pilot counties, roughly one out of 10 cast statewide.

Bartow was selected for the audit because of its size and proximity to Atlanta. The public process consisted of examining a random sample of ballots to ensure the ratio of votes for each candidate is represented in the total compiled by the scanners used in tabulating ballots from early voting and Election Day.

Key Election Dates and Information | Military and Overseas Voting

Register to Vote | Where do I vote? (MVP)

QUICK LINKS

Secure The Vote

2019 List Maintenance

Posted Rule for Public Comment

Vote Safe

FAQs

iVote - Students / Educators

FREE & FAIR

HOME PROJECTS ABOUT BLOG IN THE NEWS ARTICLES CAREERS CONTACT

Evidence-based post-election audits

Free & Fair has created a risk-limiting audit (RLA) system that was used statewide in Colorado beginning with the November 2017 general election. First developed in 2008, RLAs promote evidence-based confidence in election outcomes by comparing a random sampling of paper ballots to their corresponding digital versions. This was the first time anywhere in the United States that risk-limiting audits are conducted on a regular, statewide basis. Learn more about RLAs and how the new Colorado system works here.

VotingWorks Voting Machines Risk-Limiting Audits News Donate Careers

Risk-Limiting Audits with Arlo

Democracy depends upon trust in the electoral process. In the United States, nearly every jurisdiction relies on machines to tally votes, so trusting the process means trusting the machines that scan and tally our paper ballots. Given the high stakes of our elections, and our adversaries' demonstrated interest in them, what can we do to double-check the equipment producing the tallies?

The best safeguard we have against hacked or otherwise faulty voting systems is a Risk-Limiting Audit (RLA). Experts believe that RLAs are the most efficient way to gain certainty that the winner reported by the voting system is correct.

The RLA is efficient because of its insight that we do not need to manually recount every vote to gain certainty in the outcome of an election. Instead, the number of votes we need to manually count depends on the margin of victory in the contest we're auditing. A typical state-wide RLA might require checking fewer than 1000 ballots to validate an entire state's election results.

In 2017, Colorado became the first state to implement rigorous RLAs, auditing one race in each of 50 of its 64 counties. Colorado's experience helped to create the requirements for software that is used to run RLAs today. VotingWorks is proud to lead the development of this software which has been renamed Arlo. Arlo is built on the pioneering work done by Colorado and its technical partner, Democracy Works.

One Hundred Seventh Congress of the United States of America

AT THE SECOND SESSION

Begun and held at the City of Washington on Wednesday, the twenty-third day of January, two thousand and two

An Act

To establish a program to provide funds to States to replace punch card voting systems, to establish the Election Assistance Commission to assist in the administration of Federal elections and to otherwise provide assistance with the administration of certain Federal election laws and programs, to establish minimum election administration standards for States and units of local government with responsibility for the administration of Federal elections, and for other purposes.

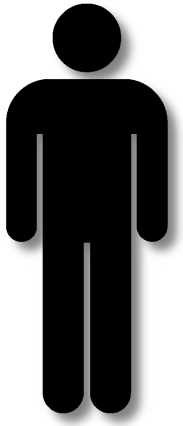
Be it enacted by the Senate and House of Representatives of the United States of America in Congress assembled,

SECTION 1. SHORT TITLE; TABLE OF CONTENTS.

(a) SHORT TITLE.—This Act may be cited as the "Help America Vote Act of 2002".

(b) TABLE OF CONTENTS.—The table of contents of this Act is as follows:

I need you to develop a mission-critical system. Here, have 2 million dollars



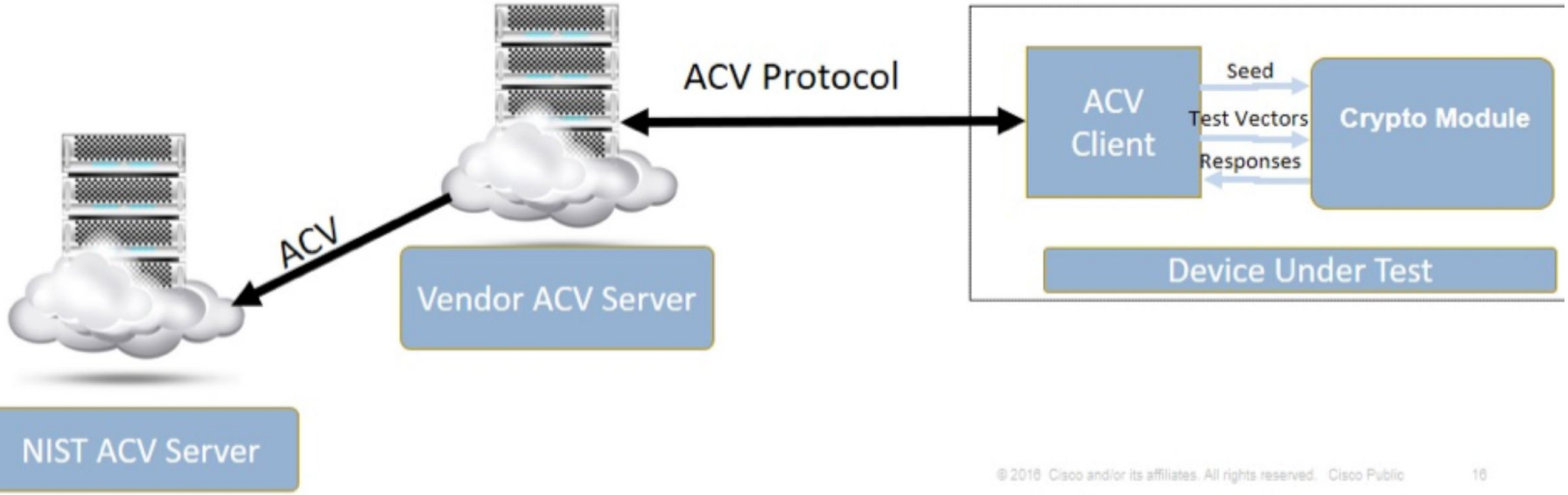


Why do we certify?

Voting System (Name/Version)	Manufacturer	Testing Standard	Date Certified
Assure 1.2	Premier Election Solutions, Inc. (formerly Diebold Election Systems, Inc.)	VSS 2002	2009-08-06
Assure 1.3 Modification	Dominion Voting Systems Corp	VSS 2002	2012-06-29
ClearVote 1.4	Clear Ballot Group, Inc.	VVSG 1.0 (2005)	2018-02-08
ClearVote 1.5	Clear Ballot Group, Inc.	VVSG 1.0 (2005)	2019-03-19
ClearVote 2.0	Clear Ballot Group, Inc.	VVSG 1.0 (2005)	2019-10-21
Democracy Suite 4.0	Dominion Voting Systems Corp	VVSG 1.0 (2005)	2012-05-10
Democracy Suite 4.14 Modification	Dominion Voting Systems Corp	VVSG 1.0 (2005)	2013-07-18

Proxy/Validation Authority Architecture

Automated Cryptographic Validation System



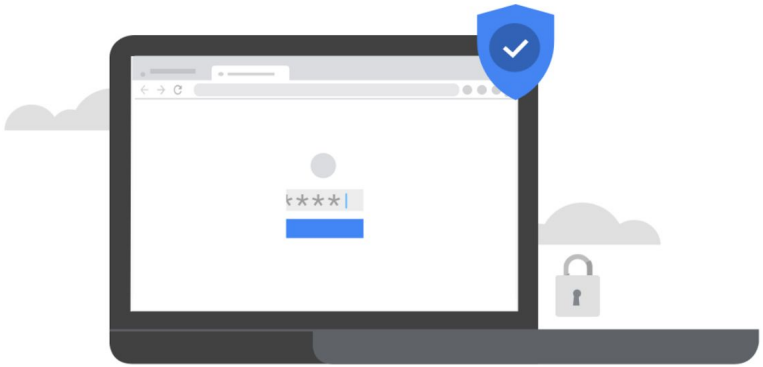
© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Public

What can we do

Chrome Browser enterprise security

Chrome Browser empowers organizations to provide a secure and consistent browsing experience to ensure employees stay productive and corporate data stays protected.

Get Chrome Browser



Reduce web risks and stay productive

Limit web attacks with Chrome Browser's built-in security.

Progress in San Francisco's Open Source Elections Project

Submitted by administrator on Mon, 2018-10-29 16:50



As many within the OSI community may know, the OSI has been following closely—and supporting—San Francisco's efforts to develop an open source elections system. Below is an update from Commissioner Chris Jerdonek of the San Francisco Elections Commision.

Schemas

The schemas in this directory represent all of the data formats needed as the output of the end-to-end verifiable elections run using ElectionGuard. These documents, together with the specification of the protocol, can be used as a complete specification for a verifier.

We will soon add an additional document, containing only the parts of the protocol spec necessary for implementing a verifier.

Get Started

The top-level file in this directory is schemas/election_record.schema.json. From there, references to other files can be followed. For example the election parameters can be found in schemas/election_parameters.schema.json:

```
"parameters": {
      "$ref": "election_parameters.schema.json"
    },
```

We provide scripts for viewing, formatting, and validating the schemas, as well as generating random instances and generating code for parsing instances. To install the dependencies of the scripts locally, run

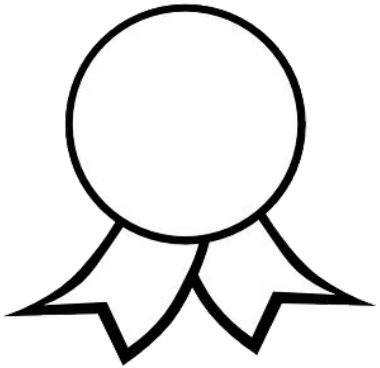
```
npm install
```

Viewing

To create an interactive viewer for the schemas, run

```
npm run viewer:serve
```

Certificate



Joey Dodds

jdodds@galois.com

Galois, Free and Fair