Verifiable Election Technologies

How Voters can Confirm that their Votes are Accurately Counted

Josh Benaloh

Senior Principal Cryptographer

Microsoft Research

Crisis of Confidence

- We have a crisis of confidence in U.S. elections today.

- Millions of Americans do not have confidence in the results of U.S. elections.

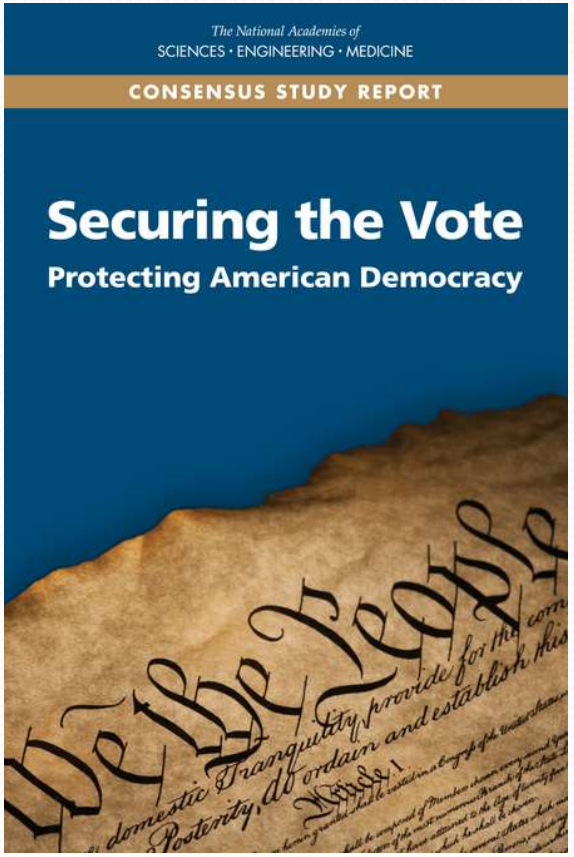
The Facts …

Regardless of how you view these concerns, there are some objective truths…

- We are not providing voters with substantive evidence that their votes have been correctly counted.
- Instead, we are asking voters to trust their local election officials, equipment vendors, etc.

National Academies of Science, Engineering, and Medicine

Issued September 2018



Findings and Recommendations

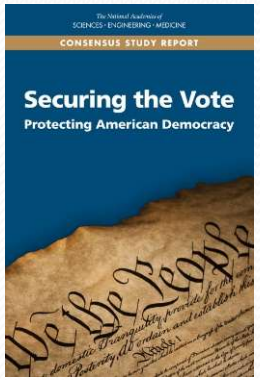
Over 8,000 election jurisdictions in the U.S.

The election equipment market is broken.

The certification process is broken.

Better funding is required.

The systems are *extremely* vulnerable.



Prof. J. Alex Halderman

– University of Michigan

“My undergraduate security class could have changed the results of the 2016 election.”


ELECTION HACKING & VOTING TECHNOLOGIES
University of California, Irvine
C-SPAN
c-span.org
@cspan
arch 13, 2018


Mike Lindell Presents:
ABSOLUTELY 9-0

Secret-Ballot Elections

Why are elections …

harder than banking?

harder than shopping?

different from everything else?

Ballot Privacy

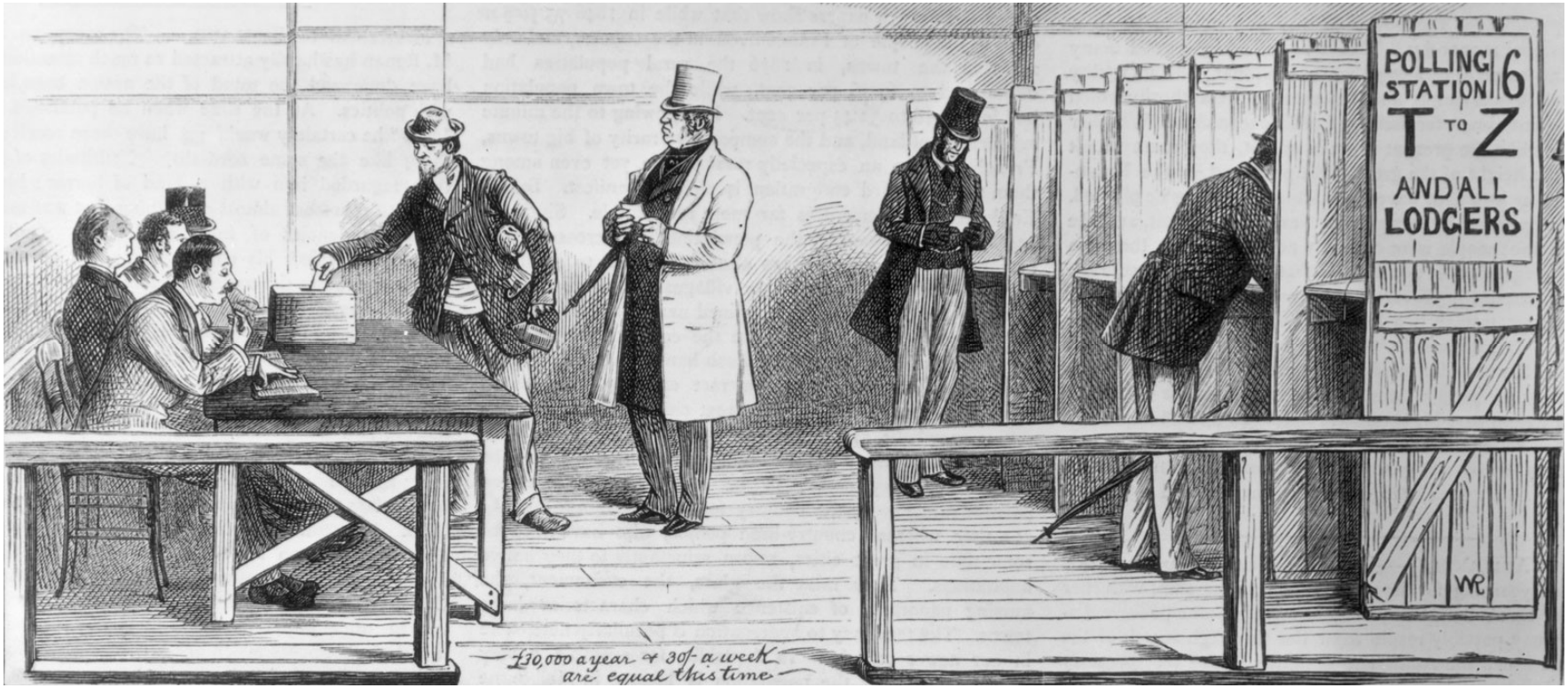
In a secret-ballot election, voters should not only *be able* to keep their votes private.

They should be *unable* to disclose their votes … even if they want to do so.

Elections Prior to Secret Ballots

The County Election – George Caleb Bingham 1852

The Australian Ballot



Election Transparency

- Secret ballots are critical, but we've paid a high price in transparency and integrity.
- With current elections, voters can do little more than deposit their ballots and hope …

The Ideal of Transparency

- We would like to be able to restore the same transparency the we had prior to the secret ballot.
- How close can we come?

What is Possible?

Technology exists that enables *any* inaccuracies and tampering of election tallies to be detected …

… not just by election officials, but also by any candidate, media outlet, voter, or other observer …

… and not just external tampering, but corruption by election officials, equipment vendors, and others.

This is known as *End-to-End (E2E) Verifiability*.

End-to-End Verifiability

End-to-End (E2E) Verifiability is the answer to the question

> *How can I trust the accuracy of an election outcome …*
>
> *when I don't trust the software, hardware, or personnel responsible for conducting the election?*

End-to-End Verifiable Elections

An election is *end-to-end verifiable* if

1. Voters can verify that their own selections have been correctly recorded.
2. Anyone can verify that the recorded votes have been correctly tallied.

A Public Election Ledger

Voter Name	Vote
Alice Smith	Jefferson
Bob Williams	Adams
Carol James	Adams
David Fuentes	Jefferson
Ellen Chu	Jefferson

Totals	
Jefferson	3
Adams	2

An End-to-End Verifiable Election

Voter Name	Vote
Alice Smith	Jefferson
Bob Williams	Adams
Carol James	Adams
David Fuentes	Jefferson
Ellen Chu	Jefferson

Totals	
Jefferson	3
Adams	2

A Secret-Ballot E2E-V Election

Voter Name	Vote	
Alice Smith	Jefferson	X37BM6YPM
Bob Williams	Adams	2J8CNF2KQ
Carol James	Adams	VRSF5JQWZ
David Fuentes	Jefferson	MW5B2VA7Y
Ellen Chu	Jefferson	8VPPS2L39

Totals	
Jefferson	3
Adams	2

A Secret-Ballot E2E-V Election

Voter Name	Vote	
Alice Smith	Jefferson	X37BM6YPM
Bob Williams	Adams	2J8CNF2KQ
Carol James	Adams	VRSF5JQWZ
David Fuentes	Jefferson	MW5B2VA7Y
Ellen Chu	Jefferson	8VPPS2L39

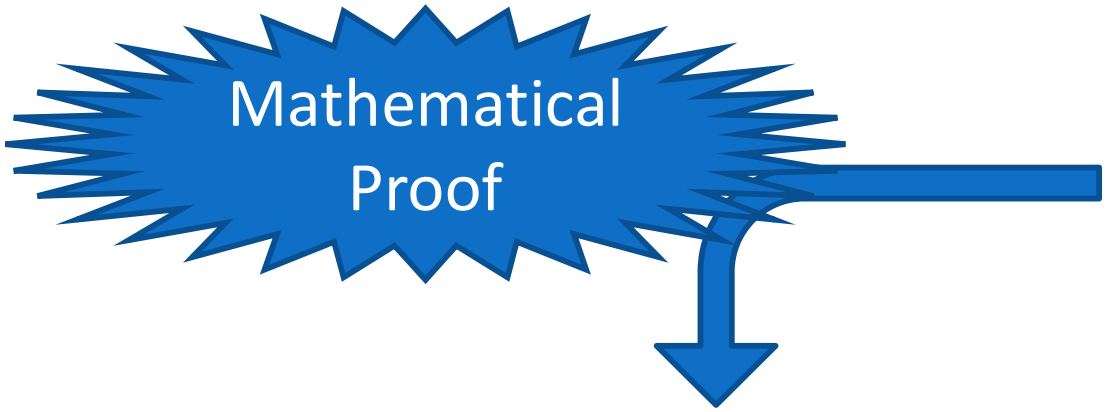
Totals	
Jefferson	3
Adams	2

A Secret-Ballot E2E-V Election

Voter Name	Vote	
Alice Smith	Jefferson	X37BM6YPM
Bob Williams	Adams	2J8CNF2KQ
Carol James	Adams	VRSF5JQWZ
David Fuentes	Jefferson	MW5B2VA7Y
Ellen Chu	Jefferson	8VPPS2L39

Totals	
Jefferson	3
Adams	2

A Secret-Ballot E2E-V Election



X37BM6YPM
2J8CNF2KQ
VRSF5JQWZ
MW5B2VA7Y
8VPPS2L39

Totals	
Jefferson	3
Adams	2

A Secret-Ballot E2E-V Election

Voter Name	Vote	
Alice Smith	Jefferson	X37BM6YPM
Bob Williams	Adams	2J8CNF2KQ
Carol James	Adams	VRSF5JQWZ
David Fuentes	Jefferson	MW5B2VA7Y
Ellen Chu	Jefferson	8VPPS2L39

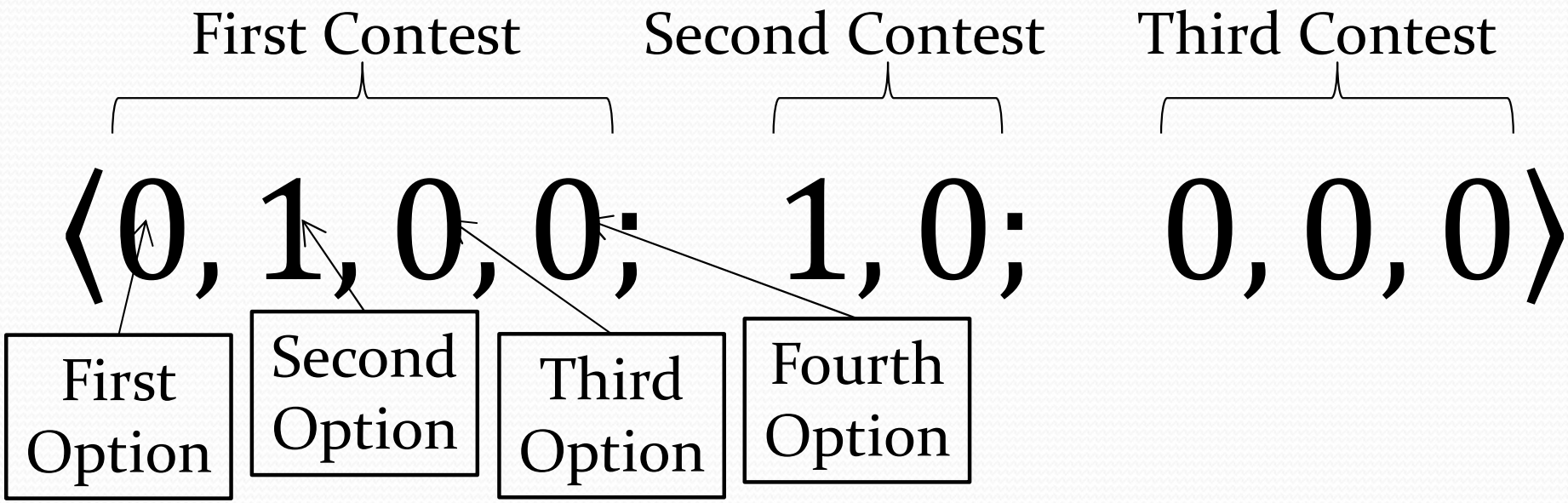
Totals	
Jefferson	3
Adams	2

End-to-End Verifiable Elections

Two questions must be answered …

1. How do voters reliably turn their preferences into encrypted votes?
2. How are voters convinced that the published set of encrypted votes corresponds the announced tally?

A Valid Vote


First Contest
Second Contest
Third Contest
⟨0, 1, 0, 0; 1, 0; 0, 0, 0⟩
First Option
Second Option
Third Option
Fourth Option

Election Tallying

Alice	$\langle 0, 1, 0, 0;\quad 1, 0;\quad 0, 0, 0\rangle$
Bob	$\langle 0, 0, 0, 1;\quad 1, 0;\quad 0, 1, 0\rangle$
Carol	$\langle 0, 0, 1, 0;\quad 0, 1;\quad 1, 0, 0\rangle$
David	$\langle 0, 1, 0, 0;\quad 1, 0;\quad 0, 0, 1\rangle$
Eve	$\langle 0, 0, 1, 0;\quad 0, 1;\quad 0, 0, 1\rangle$

Election Tallying

Alice	$\langle 0, 1, 0, 0;\quad 1, 0;\quad 0, 0, 0\rangle$
Bob	$\langle 0, 0, 0, 1;\quad 1, 0;\quad 0, 1, 0\rangle$
Carol	$\langle 0, 0, 1, 0;\quad 0, 1;\quad 1, 0, 0\rangle$
David	$\langle 0, 1, 0, 0;\quad 1, 0;\quad 0, 0, 1\rangle$
Eve	$\langle 0, 0, 1, 0;\quad 0, 1;\quad 0, 0, 1\rangle$
	$\downarrow\ \downarrow\ \downarrow\ +\ \downarrow\ \downarrow\ \downarrow$
Tally	$\langle 0, 2, 2, 1;\quad 3, 2;\quad 1, 1, 2\rangle$

Encrypted Election Tallying?

Alice	$\langle 0, 1, 0, 0;\quad 1, 0;\quad 0, 0, 0\rangle$
Bob	$\langle 0, 0, 0, 1;\quad 1, 0;\quad 0, 1, 0\rangle$
Carol	$\langle 0, 0, 1, 0;\quad 0, 1;\quad 1, 0, 0\rangle$
David	$\langle 0, 1, 0, 0;\quad 1, 0;\quad 0, 0, 1\rangle$
Eve	$\langle 0, 0, 1, 0;\quad 0, 1;\quad 0, 0, 1\rangle$

Traditional Static Encryption

The only thing you do with encrypted data

VRSF5JQWZ

is decrypt it.

Computing on Encrypted Data

Some modern encryption methods allows useful computation on encrypted data.

VRSF5JQWZ ⊗ MW5B2VA7Y

This is known as *Homomoprhic Encryption*.

Homomorphic Encryption

We can construct encryption functions such that if

A is *an* encryption of a and

B is *an* encryption of b then

$A \times B$ is *an* encryption of $a \times b$.

Homomorphic Encryption

We can also construct other encryption functions such that if

A is *an* encryption of a and

B is *an* encryption of b then

$A \times B$ is *an* encryption of $a + b$.

In Elections …

$$Z_1 = E(\text{Vote \#1})$$
$$Z_2 = E(\text{Vote \#2})$$
$$\vdots$$
$$Z_k = E(\text{Vote \#}k)$$

The *composition* of the *encryptions* of the votes is an *encryption* of the *sum* of the votes.

Requirements for Elections

- *Additively* Homomorphic Encryption
- Threshold Decryption
- Zero-knowledge Proofs of Ballot Properties
- Everything *must* be practical

Homomorphic Encryption

With RSA encryption,

$$Z_1 = E(M_1) = M_1^e$$
$$Z_2 = E(M_2) = M_2^e$$

$$Z_1 \times Z_2 = E(M_1) \times E(M_2) = M_1^e \times M_2^e$$
$$= (M_1 \times M_2)^e = E(M_1 \times M_2)$$

RSA is *multiplicatively homomorphic.*

Homomorphic Encryption

With some other encryption functions,

$$Z_1 = E(M_1) = g^{M_1}$$
$$Z_2 = E(M_2) = g^{M_2}$$

$$Z_1 \times Z_2 = E(M_1) \times E(M_2) = g^{M_1} \times g^{M_2}$$
$$= g^{M_1+M_2} = E(M_1 + M_2)$$

Such functions are *additively homomorphic.*

Multiplicative → Additive

RSA and ElGamal are multiplicatively homomorphic.

- To "additively" encrypt message m, compute $M = g^m \bmod n$ and encrypt M.
- Then $M_1 \times M_2 = g^{m_1} \times g^{m_2} = g^{m_1+m_2} \pmod{n}$.
- Recovering $m_1 + m_2$ requires computing a discrete log, but the plaintext space is small.

Homomorphic Encryption

A Brief History

- 1976 – Diffie-Hellman *New Directions in Cryptography*
- 1978 – Rivest, Shamir, Adleman (RSA)
- 1978 – Rivest, Adleman, Dertouzos – *On Databanks and Privacy Homomorphisms*
- 1985 – Benaloh – (Additive) Homomorphic Encryption
- 1999 – Pallier Encryption (Additive)

Homomorphic Encryption

Some Homomorphic Functions

- (×) RSA: $E(M) = M^e \bmod n$
- (×) ElGamal: $E(M, r) = (g^r, Mh^r) \bmod p$
- ($\oplus$) Goldwasser-Micali: $E(b, r) = r^2 g^b \bmod n$
- (+) Benaloh: $E(M, r) = r^e g^M \bmod n$
- (+) Pallier: $E(M, r) = r^n g^M \bmod n^2$

Homomorphic Encryption

Some Homomorphic Functions

- (×) RSA: $E(M) = M^e \bmod n$
- (×) ElGamal: $E(M, r) = (g^r, Mh^r) \bmod p$
- ($\oplus$) Goldwasser-Micali: $E(b, r) = r^2 g^b \bmod n$
- (+) Benaloh: $E(M, r) = r^e g^M \bmod n$
- (+) Pallier: $E(M, r) = r^n g^M \bmod n^2$

Multiplicative → Additive

RSA and ElGamal are multiplicatively homomorphic.

- To "additively" encrypt message m, compute $M = g^m \bmod n$ and encrypt M.
- Then $M_1 \times M_2 = g^{m_1} \times g^{m_2} = g^{m_1+m_2} \pmod{n}$.
- Recovering $m_1 + m_2$ requires computing a discrete log, but the plaintext space is small.

Exponential ElGamal Encryption

Fix constants g and p in advance.

Keyholder chooses random secret s and publishes public key $K = g^s \bmod p$.

To encrypt message m, select a random value r, and for the encryption pair

$$E(m, r) = (g^r \bmod p, g^m K^r \bmod p).$$

Exponential ElGamal Decryption

To decrypt a pair (A, B), compute (all mod p)

$$\frac{B}{A^s} = \frac{g^m K^r}{g^{rs}} = \frac{g^m g^{sr}}{g^{rs}} = g^m.$$

When the message is small, it can be derived from g^m by exhaustive search.

ElGamal Encryption

- Vast majority of web traffic is protected with ElGamal
- Basically just Diffie-Hellman key exchange – predates RSA
- Can be used to achieve an additive homomorphism
- Supports simple threshold encryption
- Supports simple ZK proofs of necessary properties
- Is extremely efficient

Homomorphic Tallying

Alice	⟨0, 1, 0, 0; 1, 0; 0, 0, 0⟩
Bob	⟨0, 0, 0, 1; 1, 0; 0, 1, 0⟩
Carol	⟨0, 0, 1, 0; 0, 1; 1, 0, 0⟩
David	⟨0, 1, 0, 0; 1, 0; 0, 0, 1⟩
Eve	⟨0, 0, 1, 0; 0, 1; 0, 0, 1⟩

Homomorphic Tallying

Alice	⟨0, 1, 0, 0; 1, 0; 0, 0, 0⟩
Bob	⟨0, 0, 0, 1; 1, 0; 0, 1, 0⟩
Carol	⟨0, 0, 1, 0; 0, 1; 1, 0, 0⟩
David	⟨0, 1, 0, 0; 1, 0; 0, 0, 1⟩
Eve	⟨0, 0, 1, 0; 0, 1; 0, 0, 1⟩
Encrypted	↓ ↓ ↓ × ↓ ↓ ↓
Tally	⟨0, 2, 2, 1; 3, 2; 1, 1, 2⟩

Homomorphic Tallying

Alice	$\langle 0, 1, 0, 0;\ 1, 0;\ 0, 0, 0 \rangle$
Bob	$\langle 0, 0, 0, 1;\ 1, 0;\ 0, 1, 0 \rangle$
Carol	$\langle 0, 0, 1, 0;\ 0, 1;\ 1, 0, 0 \rangle$
David	$\langle 0, 1, 0, 0;\ 1, 0;\ 0, 0, 1 \rangle$
Eve	$\langle 0, 0, 1, 0;\ 0, 1;\ 0, 0, 1 \rangle$
	$\downarrow\ \downarrow\ \downarrow\ +\ \downarrow\ \downarrow\ \downarrow$
Tally	$\langle 0, 2, 2, 1;\ 3, 2;\ 1, 1, 2 \rangle$

Homomorphic Tallying

Alice	$\langle 0, 1, 0, 0;\quad 1, 0;\quad 0, 0, 0\rangle$
Bob	$\langle 0, 0, 0, 1;\quad 1, 0;\quad 0, 1, 0\rangle$
Carol	$\langle 0, 0, 1, 0;\quad 0, 1;\quad 1, 0, 0\rangle$
David	$\langle 0, 1, 0, 0;\quad 1, 0;\quad 0, 0, 1\rangle$
Eve	$\langle 0, 0, 1, 0;\quad 0, 1;\quad 0, 0, 1\rangle$
	$\downarrow\ \downarrow\ \downarrow\ +\ \downarrow\ \downarrow\ \downarrow$
Tally	$\langle 0, 2, 2, 1;\quad 3, 2;\quad 1, 1, 2\rangle$

Who Can Decrypt?

- We don’t want there to be a single entity who can decrypt everything.
- The decryption capabilities should be split amongst members of a *canvassing board.*
- We therefore want to *split* the decryption key.

Split Key ElGamal

- Instead of a single $K = g^k$, each canvassing board member selects its own private key k_i and forms the corresponding public key $K_i = g^{k_i}$.
- The *joint* public key is simply $K = \prod_i K_i$.
- Each keyholder can perform its own decryption, and the partial decryptions are multiplied.

Threshold Homomorphic Encryption

In practice, it is better to use *threshold* homomorphic encryption which allows for some robustness by, for example, requiring only 3 of 5 canvassing board members to cooperate in order to perform a decryption.

Homomorphic Tallying

Alice	⟨0, 1, 0, 0; 1, 0; 0, 0, 0⟩
Bob	⟨0, 0, 0, 1; 1, 0; 0, 1, 0⟩
Carol	⟨0, 0, 1, 0; 0, 1; 1, 0, 0⟩
David	⟨0, 1, 0, 0; 1, 0; 0, 0, 1⟩
Eve	⟨0, 0, 1, 0; 0, 1; 0, 0, 1⟩
Encrypted	↓ ↓ ↓ × ↓ ↓ ↓
Tally	⟨0, 2, 2, 1; 3, 2; 1, 1, 2⟩

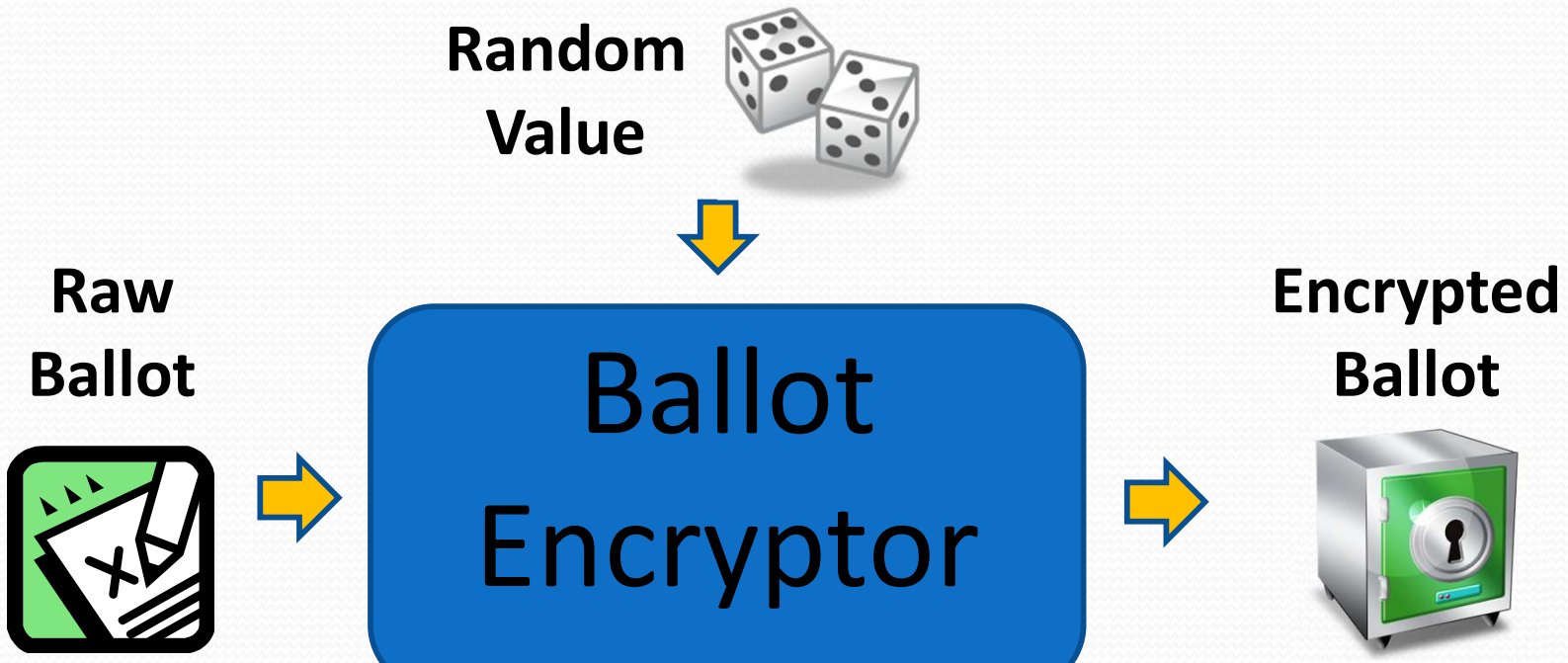
Homomorphic Tallying

Alice	$\langle 0, 1, 0, 0;\ 1, 0;\ 0, 0, 0 \rangle$
Bob	$\langle 0, 0, 0, 1;\ 1, 0;\ 0, 1, 0 \rangle$
Carol	$\langle 0, 0, 1, 0;\ 0, 1;\ 1, 0, 0 \rangle$
David	$\langle 0, 1, 0, 0;\ 1, 0;\ 0, 0, 1 \rangle$
Eve	$\langle 0, 0, 1, 0;\ 0, 1;\ 0, 0, 1 \rangle$
	$\downarrow\ \downarrow\ \downarrow\ +\ \downarrow\ \downarrow\ \downarrow$
Tally	$\langle 0, 2, 2, 1;\ 3, 2;\ 1, 1, 2 \rangle$

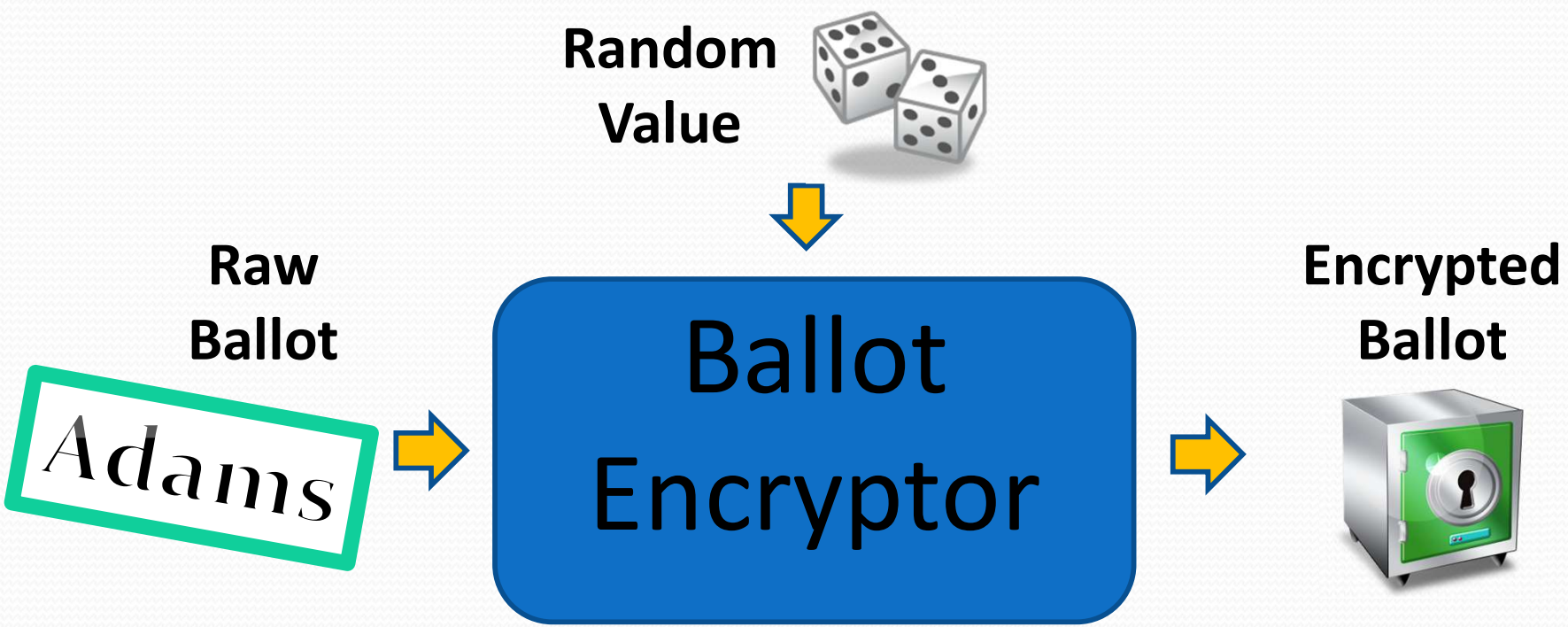
Randomized Encryption

- Ballot encryption *must* be “randomized”.
- Identical ballots should *not* have identical encryptions.

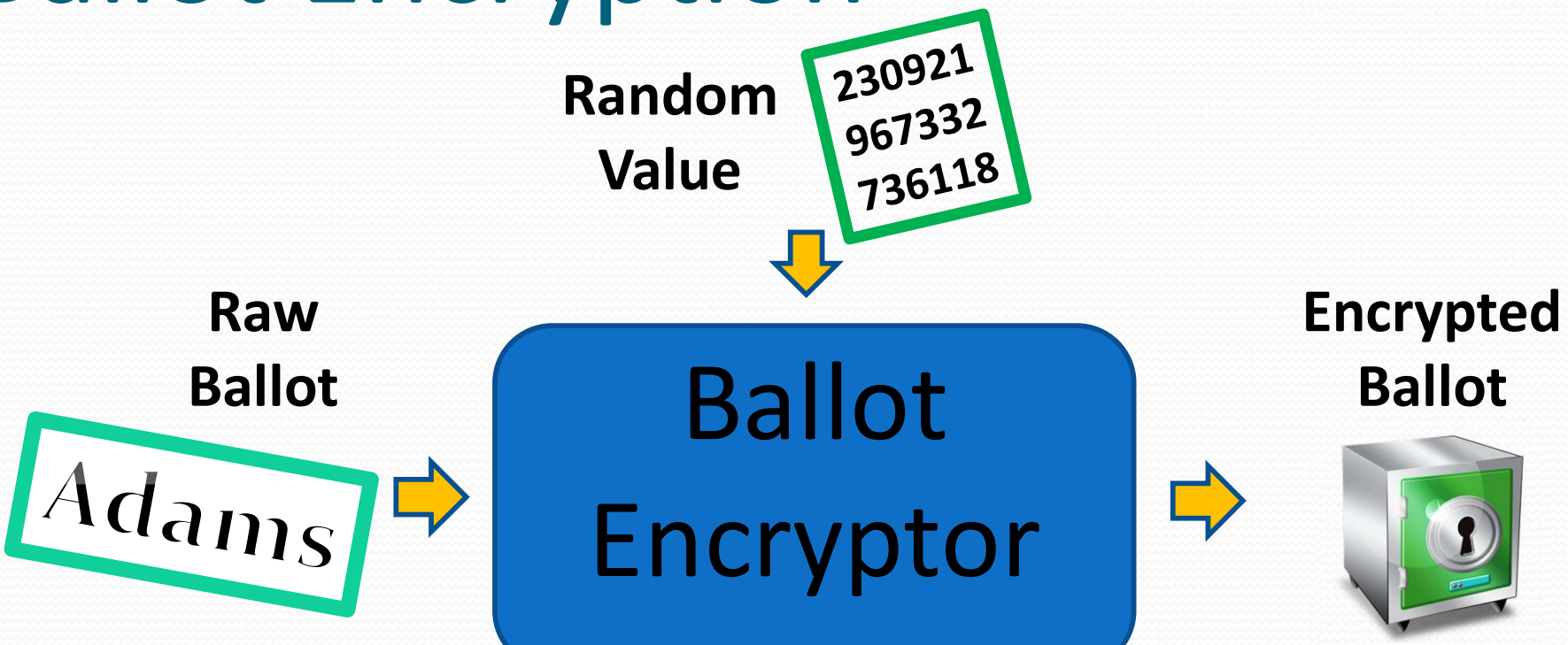
Ballot Encryption



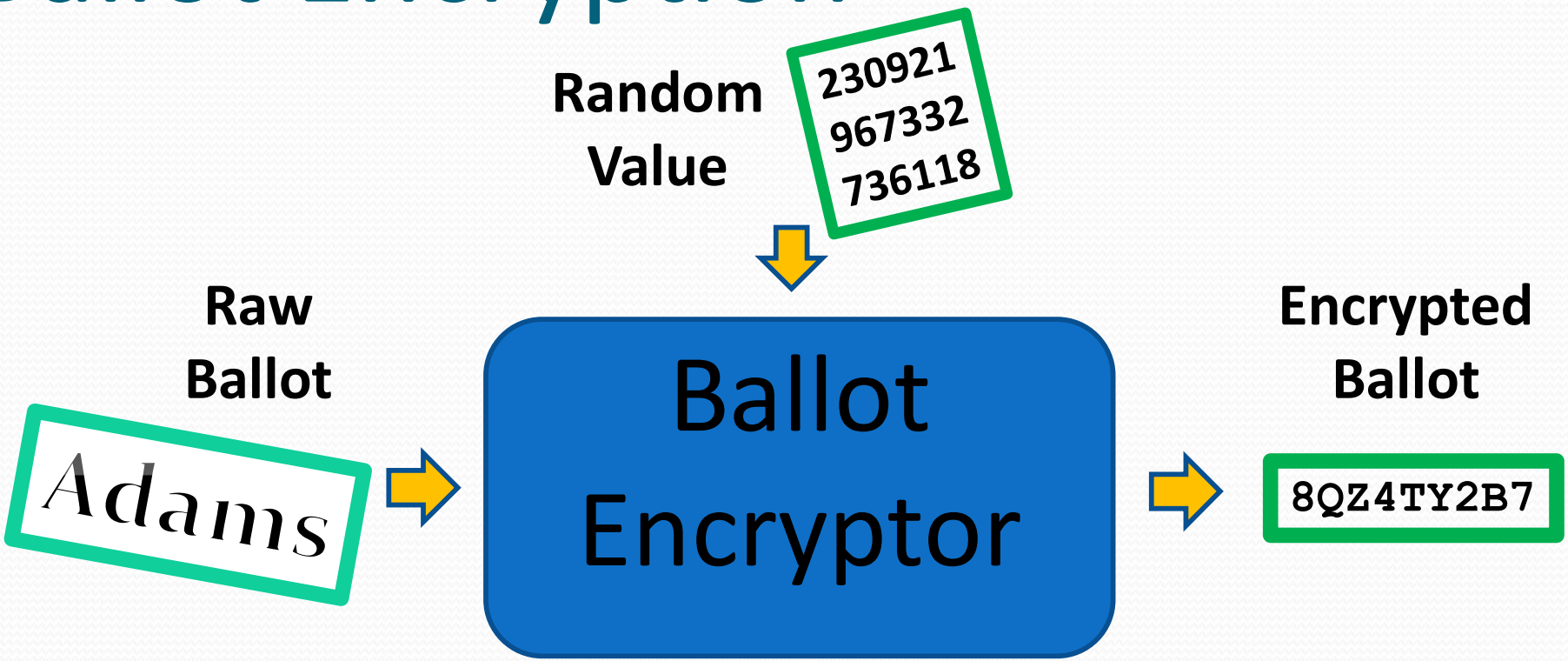
Ballot Encryption



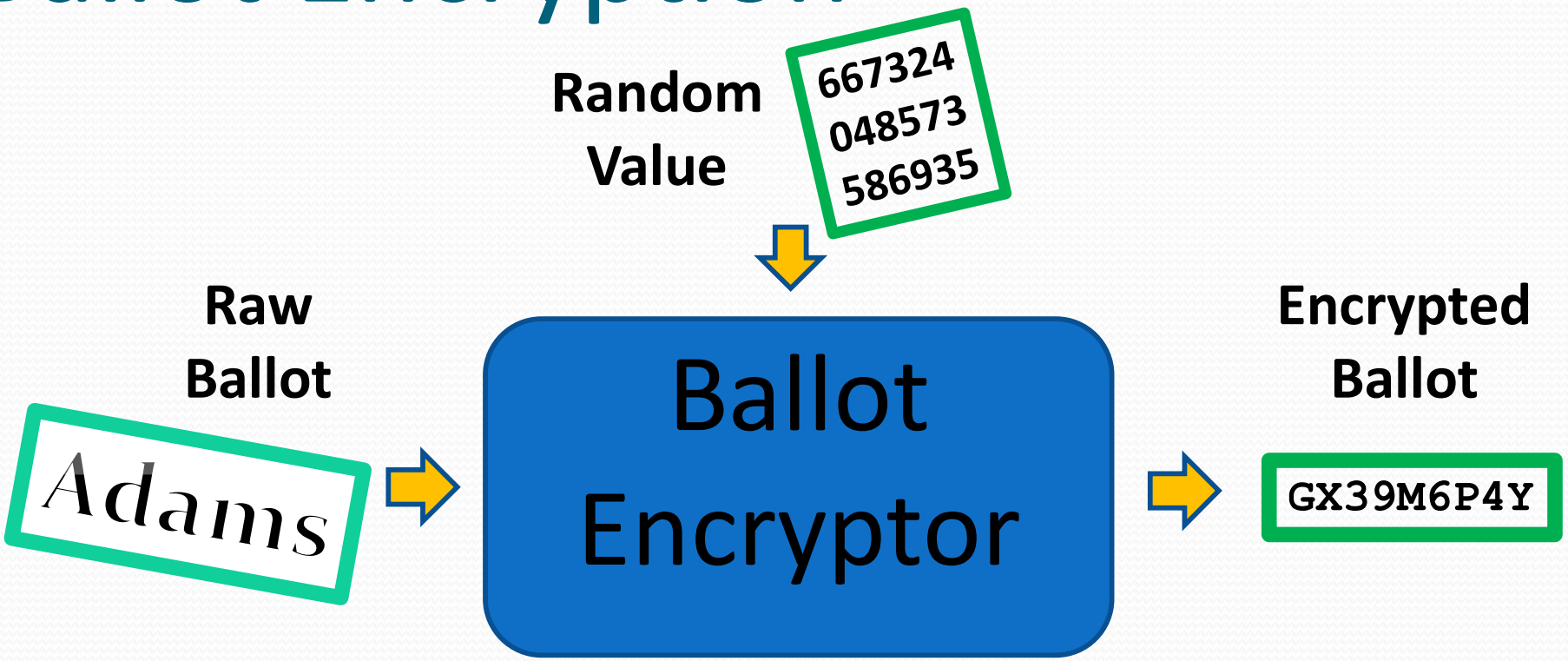
Ballot Encryption



Ballot Encryption



Ballot Encryption



Verifiable Decryption

The keyholders can't simply decrypt, they have to convince observers that they've decrypted correctly.

This can be done *without* revealing keys.

Interactive Proofs

A *Zero-Knowledge Interactive Proof (ZKIP)* is an exchange between a *prover* and a *verifier* wherein the prover convinces the verifier of a fact – without revealing additional information.

1. Prover Claim
2. Random Verifier Challenge
3. Prover Response

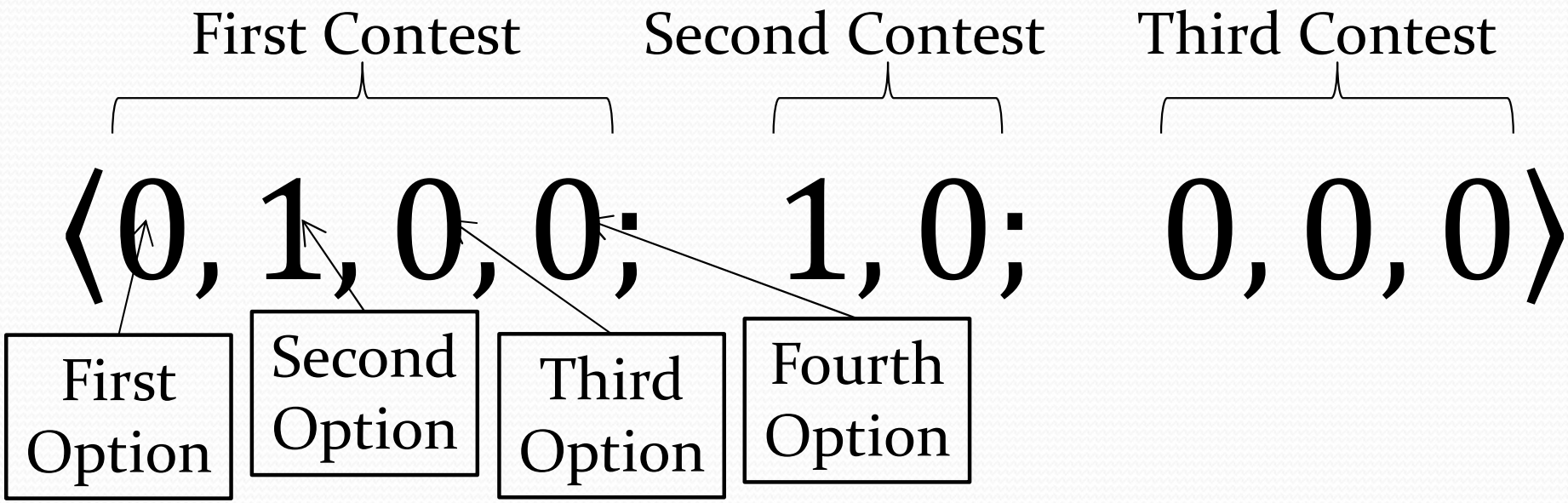
Non-Interactive ZK Proofs

Interactive proofs can often be made non-interactive by replacing the verifier with a one-way hash function.

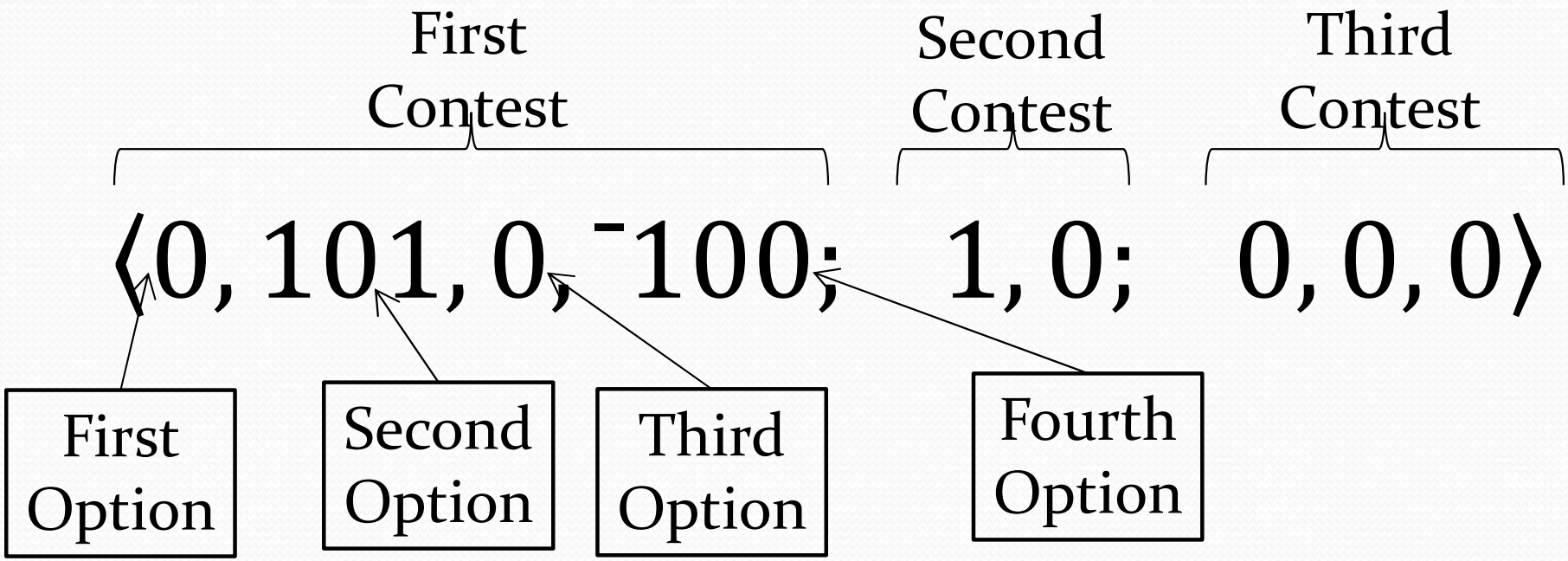
Typical Non-Interactive Zero-Knowledge (NIZK) Proofs

1. Prover Claim
2. Hash of Claim
3. Prover Response

A Valid Ballot



An Invalid Ballot



NIZK Proofs

- A Chaum-Pedersen interactive proof can be used to prove a precise ElGamal decryption.
- A Cramer-Damgård-Schoenmakers interactive proof can be used to prove a disjunction.
- The Fiat-Shamir heuristic can be applied to make this non-interactive.

End-to-End Verifiable Elections

Two questions must be answered …

1. How do voters reliably turn their preferences into encrypted votes?
2. How are voters convinced that the published set of encrypted votes corresponds the announced tally?

How do Humans Encrypt?

- If voters encrypt their votes with devices of their own choosing, they are subject to coercion and compromise.
- If voters encrypt their votes on “official” devices, how can they trust that their intentions have been properly captured?

The Human Encryptor

We need to find ways to engage humans in an *interactive proof* process to ensure that their intentions are accurately reflected in ballots encrypted on their behalf.

How Can Humans Verify Votes?

VRSF5JQWZ = Adams ?

Believing Without Seeing

I claim that all of the cards below are red.

Believing Without Seeing

I claim that all of the cards below are red.

Believing Without Seeing

I claim that all of the cards below are red.





Believing Without Seeing

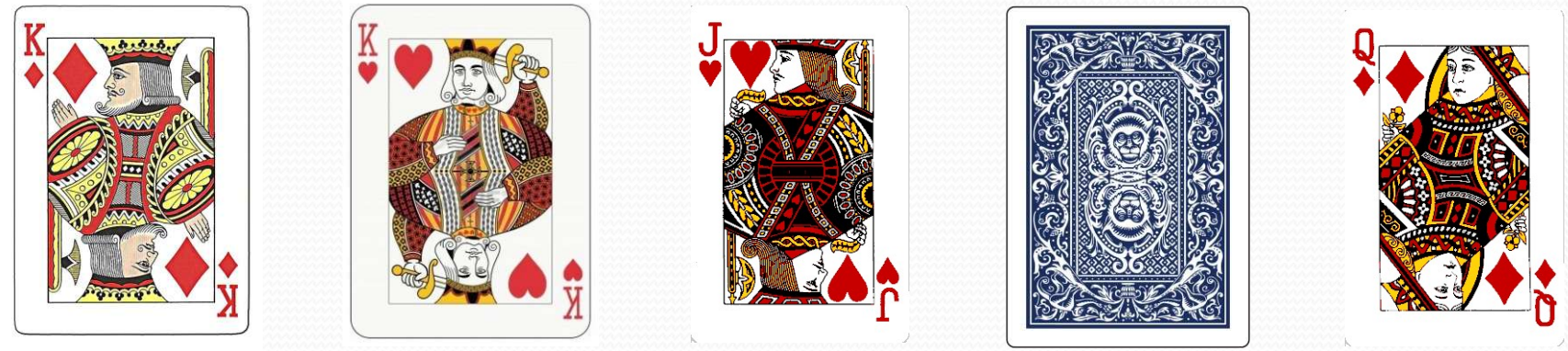
I claim that all of the cards below are red.





Believing Without Seeing

I claim that all of the cards below are red.



Believing Without Seeing

I claim that all of the cards below are red.

Believing Without Seeing

I claim that all of the cards below are red.

You've never seen this card.

Believing Without Seeing

I claim that all of the cards below are red.

You've never seen this card.

But you now have good reason to believe it's red.

Non-transferable Belief

Even though you now believe that this card is red, there's nothing that you can do to convince someone else.

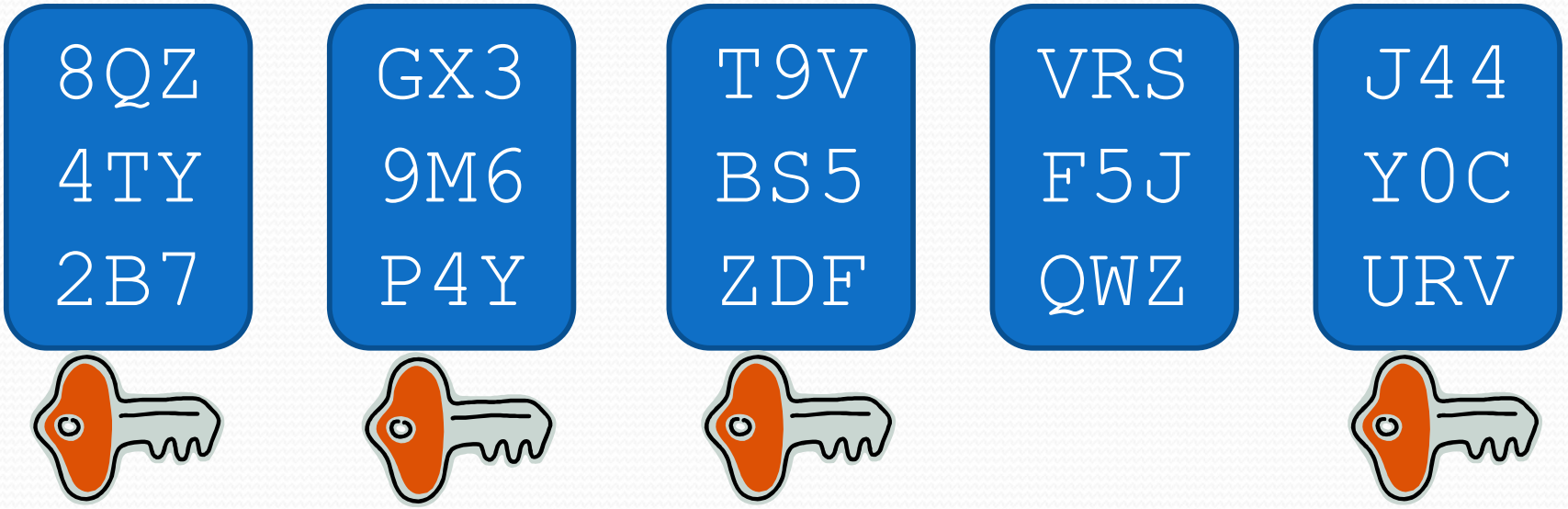
Believing Without Seeing

I claim that all of the encryptions below are votes for Adams.

8QZ 4TY 2B7	GX3 9M6 P4Y	T9V BS5 ZDF	VRS F5J QWZ	J44 Y0C URV

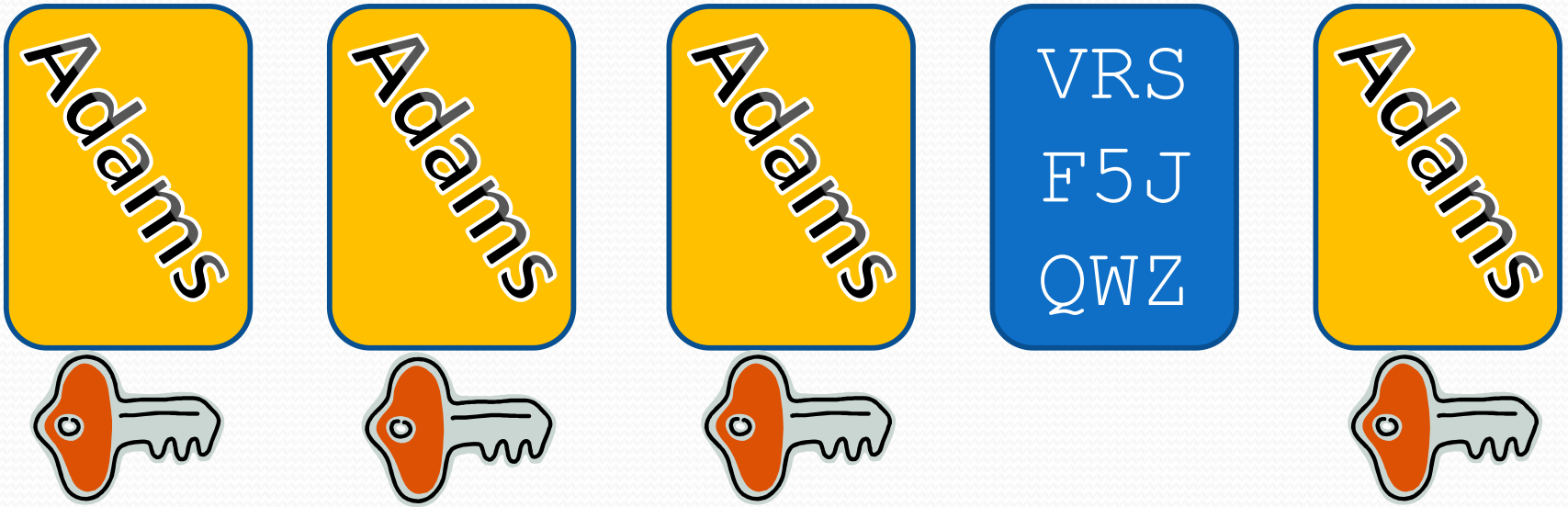
Believing Without Seeing

I claim that all of the encryptions below are votes for Adams.



Believing Without Seeing

I claim that all of the encryptions below are votes for Adams.



In practice …

- Even if very few voters each “spoil” a single ballot, very high integrity is assured.
- If 100 voters in a national election each spoil a single ballot, a malicious system would be unlikely to be able to alter even 1% of the votes without detection.

A Verifiable Election Record

Voter	Cast Ballots	Adams	Jefferson
Alice	X37BM6YPM	0	1
Bob	2J8CNF2KQ	1	0
Carol	VRSF5JQWZ	1	0
David	MW5B2VA7Y	0	1
Ellen	8VPPS2L39	0	1
	×	+	
	CM97JQX4D	2	3

Spoiled Ballots		
36PWY4MMB	0,1	Jefferson
8QZ4TY2B7	1,0	Adams
GX39M6P4Y	1,0	Adams

Totals	
Jefferson	3
Adams	2

Writing a Verifier

- Verify that the encrypted ballots are correctly multiplied to form encrypted tallies.
- Verify that the encrypted tally is correctly decrypted.
- Verify that the spoiled ballots are correctly decrypted.
- Verify that each encrypted ballot is “well-formed”.

The Voter's Perspective

Verifiable election systems can be built to look exactly like current systems …

… with one addition …

A Verifiable Receipt

Confirmation Code

Use this ticket to verify your ballot was counted.
Go to:
www.findmyballot.com

Scan with your phone

-or-

Enter this code:
4CCD3 6EDC2 CA933
7A632 25E08 9B3CE
2039B 886FE 6E667
62F7A 225B0 BD725
1876

The Voter's Perspective

Voters can …

- Use receipts to check their results are properly recorded on a public web site.
- Throw their receipts in the trash.

- Write and use their own election verifiers.
- Download applications from sources of their choice to verify the mathematical proof of the tally.
- Believe verifications done by their political parties, LWV, ACLU, etc.
- Accept the results without question.

Real-World Deployments

- Helios (www.heliosvoting.org) – Adida and others
  - Used to elect president of UC Louvain, Belgium.
  - Used in Princeton University student government.
  - Used by ACM, IACR, and other professional societies.
- Scantegrity II (www.scantegrity.org) – Chaum, Rivest, many others
  - Used for 2009 & 2011 municipal elections in Takoma Park, MD.
- STAR-Vote – Benaloh, Byrne, Eakin, Kortum, McBurnett, Pereira, Stark, Wallach
  - Designed for use in Travis County, Texas.

ElectionGuard

… a free, open-source software toolkit

Can be built into …

- Touch screen systems
- Optical scanners
- Vote by Mail
- (Even Internet voting)

GeekWire

Microsoft unveils ElectionGuard to help secure voting systems across the country

BY MONICA NICKELSBURG on July 17, 2019 at 3:22 pm

Microsoft demos ElectionGuard at the Aspen Security Forum. (Microsoft Photo)

ElectionGuard Partners

- Microsoft is working with vendors to encourage and help integrate *ElectionGuard* into new and existing systems.
- Microsoft is working with jurisdictions promote *ElectionGuard* and assist with its use.

ElectionGuard in Practice

First use in a public election Feb. 18, 2020 in Fulton, Wisconsin.

Another step in testing ElectionGuard

Feb 17, 2020 | Tom Burt - Corporate Vice President, Customer Security & Trust

Tomorrow I'll be in Fulton, Wisconsin, with a team of people from Microsoft taking one of many steps needed to prepare our ElectionGuard technology for broad adoption. Together with election officials from the state of Wisconsin and the election technology company VotingWorks, we will be piloting ElectionGuard in an actual election for the first time.

This could be Microsoft's most important product in 2020. If it works

ElectionGuard isn't designed to make voting machines safe from hackers. It's meant to make hacking them pointless.

Alfred Ng Feb. 18, 2020

This story is part of **Elections 2020**, CNET's coverage of the run-up to voting in November.

ElectionGuard in Practice

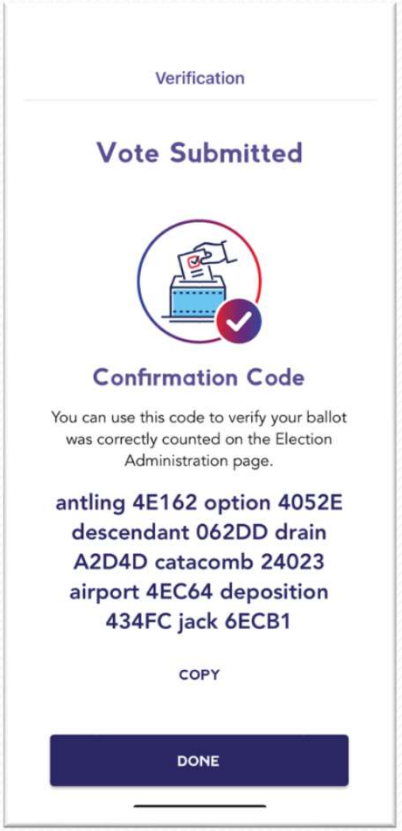
Nov. 2020 in Inyo County, California

ElectionGuard was used by VotingWorks to conduct a privacy-preserving risk-limiting audit.

ElectionGuard in Practice

Dec. 2020 with Markup

ElectionGuard was used U.S. House of Representatives Democratic Caucus to elect their leadership (Speaker, Whip, etc.).



ElectionGuard in Practice

June 3, 2021 Partnership with Hart InterCivic

Hart will integrate *ElectionGuard* into its *Verity* line of precinct-based optical scanners.

ElectionGuard in Practice

Nov. 2022 Hart Pilot – Preston, Idaho

The Preston Citizen

Franklin County precinct chosen to pilot new voting software

By TERESA CHIPMAN Citizen staff Nov 9, 2022 0

ElectionGuard in Practice

ElectionGuard lets voters confirm that their ballot was counted and provides an independent verification that the election results are correct.

https://www.collegeparkmd.gov/DocumentCenter/View/5221/

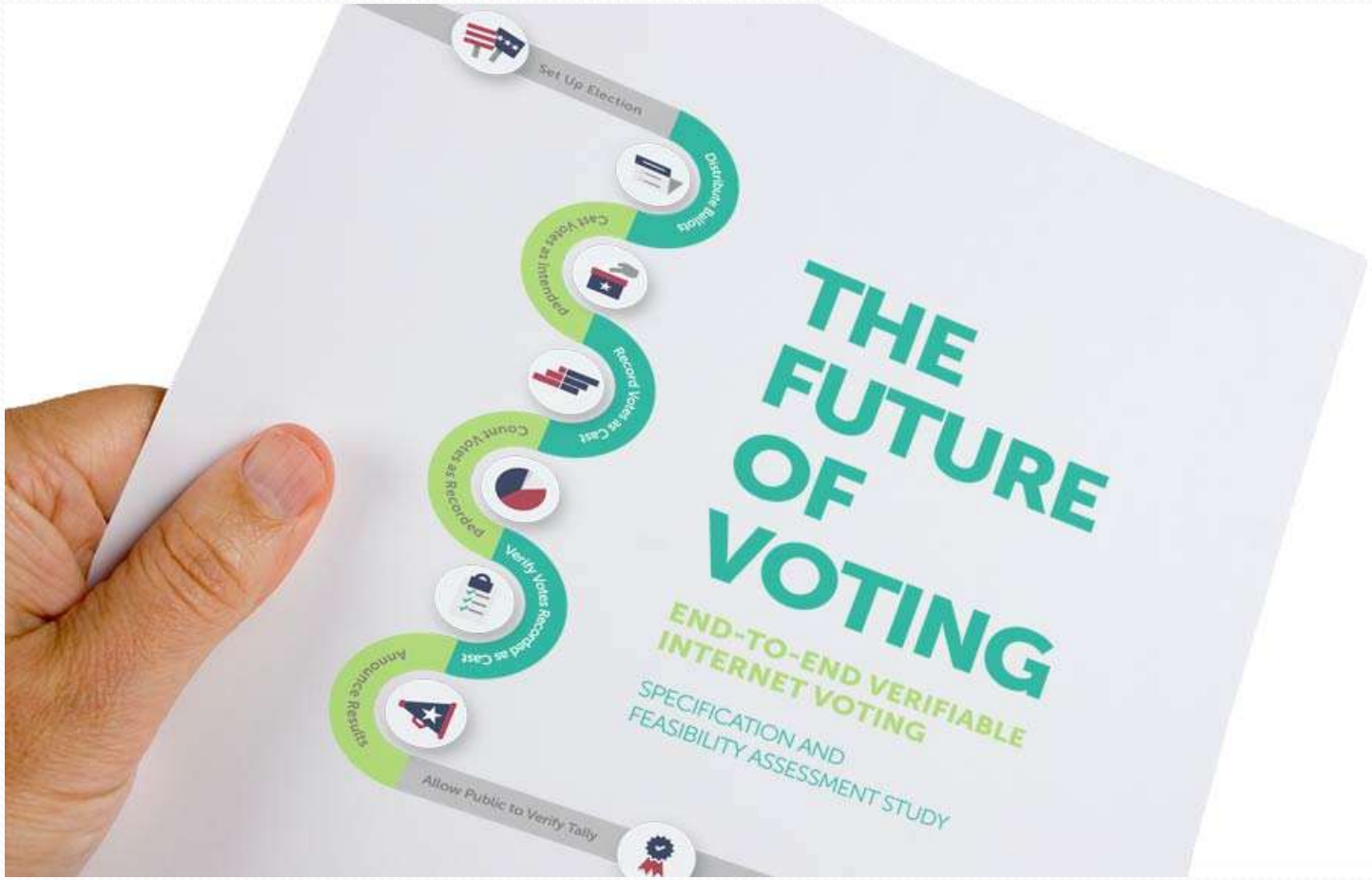
ElectionGuard in Practice

MITRE has worked with *ElectionGuard* since late 2021 to write a *premium* verifier.

What's Next?

Internet Voting?

- Some jurisdictions are beginning to explore Internet voting.
- There is a strong push towards IV from a variety of constituencies.


Set Up Election
Distribute Ballots
Cast Votes as Intended
Record Votes as Cast
Count Votes as Recorded
Verify Votes Recorded as Cast
Announce Results
Allow Public to Verify Tally
THE FUTURE OF VOTING
END-TO-END VERIFIABLE INTERNET VOTING
SPECIFICATION AND FEASIBILITY ASSESSMENT STUDY

References

- National Academies report
  https://www.nationalacademies.org/our-work/the-future-of-voting-accessible-reliable-verifiable-technology
- U.S. Vote Foundation report
  https://www.usvotefoundation.org/E2E-VIV
- Non-technical overview of E2E-verifiability
  https://arxiv.org/abs/1504.03778
- Microsoft Research 45-minute webinar
  https://note.microsoft.com/MSR-Webinar-ElectionGuard-Registration-on-demand.html
- Microsoft *ElectionGuard*
  https://github.com/microsoft/electionguard

Questions?