

Crypto and e-Voting: Homomorphisms, Zero-Knowledge Proofs

Dan S. Wallach (Rice University)



Administrivia

Project 1: Going online tomorrow

Some coding, some written bits — get started early

Proper use of Piazza

Please don't post partial solutions (code, hash values, etc.) without marking those posts as “private” to the instructors.

Our last crypto lecture!

On your own: go read up on how RSA works

Modern cryptographers seem to have concluded that DH-style is the way to go

Instead, this lecture is a whirlwind through several modern crypto topics

Zero-knowledge proofs

Mixnets

Homomorphic cryptography

Application to electronic voting

All of these things are deep topics

Your mission: understand some of the clever things that crypto can do

Reminder: Diffie- Hellman & ElGamal Crypto

Modular arithmetic 101

We're working in \mathbb{Z}_p^* , the integers in $[1, p)$

$$2 + 3 = 5 \pmod{7}$$

$$2 + 4 = 6 \pmod{7}$$

$$2 + 5 = 0 \pmod{7} \leftarrow \text{Forbidden!}$$

$$2 * 3 = 6 \pmod{7}$$

$$2 * 4 = 1 \pmod{7}$$

$$6 * 6 = 1 \pmod{7}$$

Note: \mathbb{Z}_p^* is closed under multiplication but not addition.

Modular arithmetic 101

In \mathbb{Z}_p^* , we want to find *generators* such that

$$g^1, g^2, \dots, g^{p-1}$$

cover all the elements in the group.

Example, for $p=7$:

$g=2$ is not a generator, but $g=3$ is.

Discrete logarithms

Back to the regular integers, say I give you $q = 5^{8437591243259543}$
and ask you to take $\log_5 q$

Logarithms, over integers, are tractable. But what about in \mathbb{Z}_p^* ?

No known efficient solution to DLog problem.

Diffie-Hellman (1976)

Alice : *random* $a \in \mathbb{Z}_p^*$

Bob : *random* $b \in \mathbb{Z}_p^*$

Public : *generator* $g \in \mathbb{Z}_p^*$

$A \rightarrow B$: g^a

$B \rightarrow A$: g^b

Alice : computes $(g^b)^a = g^{ab}$

Bob : computes $(g^a)^b = g^{ab}$

Eve : knows g^a, g^b , cannot compute g^{ab}

ElGamal encryption (1984)

Non-deterministic cryptosystem (different r every time)

$$E(g^a, r, M) = \langle g^r, (g^a)^r M \rangle$$

$$D(g^r, g^{ar} M, a) = \frac{g^{ar} M}{(g^r)^a}$$
$$= M$$

g	group generator
M	plaintext (message)
r	random (chosen at encryption time)
a	(private) decryption key
g^a	(public) encryption key

Digital signature algorithm (1991)

Similar idea to ElGamal encryption, just a bit more involved

$$k = \text{random} \in [1, q)$$

$$r = (g^k \bmod p) \bmod q$$

$$s = k^{-1} (H(M) + xr) \bmod q$$

$$\text{Sign}(x, k, M) = \langle r, s \rangle$$

$\text{Verify}(y, r, s) :$

$$w = s^{-1} \bmod q$$

$$u_1 = H(M) \cdot w \bmod q$$

$$u_2 = r \cdot w \bmod q$$

$$v = ((g^{u_1} y^{u_2}) \bmod p) \bmod q$$

$$\text{valid if } v = r$$

g	group generator
p	large prime (1000+ bits)
q	large-ish prime (~160 bits) divisible by $p-1$
M	plaintext (message)
k	random (chosen at encryption time)
x	(private) decryption key (chosen randomly $< q$)
$y=g^x$	(public) encryption key (mod p)

Digital signature algorithm (1991)

Similar idea to ElGamal encryption, just a bit more involved

$$k = \text{random} \in [1, q)$$

$$r = (g^k \bmod p) \bmod q$$

$$s = k^{-1} (H(M) + xr) \bmod q$$

$$\text{Sign}(x, k, M) = \langle r, s \rangle$$

$\text{Verify}(y, r, s) :$

$$w = s^{-1} \bmod q$$

$$u_1 = H(M) \cdot w \bmod q$$

g	group generator
p	large prime (1000+ bits)
q	large-ish prime (~160 bits) divisible by $p-1$
M	plaintext (message)
k	random (chosen at encryption time)
x	(private) decryption key (chosen randomly $< q$)
$y=g^x$	(public) encryption key (mod p)

Fun: if k isn't random and unique, attacker can recover x . (PS3 attack, Dec. 2010). Solution: derive k from x and $H(M)$ deterministically. [RFC 6979]

Homomorphic property

Anybody can combine two ciphertexts to get a new one.

$$\begin{aligned} E(M_1) \oplus E(M_2) &= \langle g^{r_1}, (g^a)^{r_1} M_1 \rangle \oplus \langle g^{r_2}, (g^a)^{r_2} M_2 \rangle \\ &= \langle g^{r_1} g^{r_2}, (g^a)^{r_1} M_1 (g^a)^{r_2} M_2 \rangle \\ &= \langle g^{r_1+r_2}, g^{a(r_1+r_2)} M_1 M_2 \rangle \\ &= E(M_1 M_2) \end{aligned}$$

g	group generator
M	plaintext (message)
r	random (chosen at encryption time)
a	(private) decryption key
g^a	(public) encryption key

Homomorphic vote tallying

Change messages to counters, additive in exponent of g .

“Exponential ElGamal”

$$\begin{aligned} E(v_1) \oplus E(v_2) &= \langle g^{r_1}, (g^a)^{r_1} g^{v_1} \rangle \oplus \langle g^{r_2}, (g^a)^{r_2} g^{v_2} \rangle \\ &= \langle g^{r_1+r_2}, g^{a(r_1+r_2)} g^{v_1+v_2} \rangle \\ &= E(v_1 + v_2) \end{aligned}$$

g	group generator
v	plaintext (counters)
r	random (chosen at encryption time)
a	(private) decryption key
g^a	(public) encryption key

Violation of encryption semantics?

**If I know M_1 and M_2 and $E(\textcolor{blue}{M}_1) \oplus E(\textcolor{red}{M}_2) = E(\textcolor{blue}{M}_1\textcolor{red}{M}_2)$
then I can find other messages where
I know their encryption!**

Solution: Padding

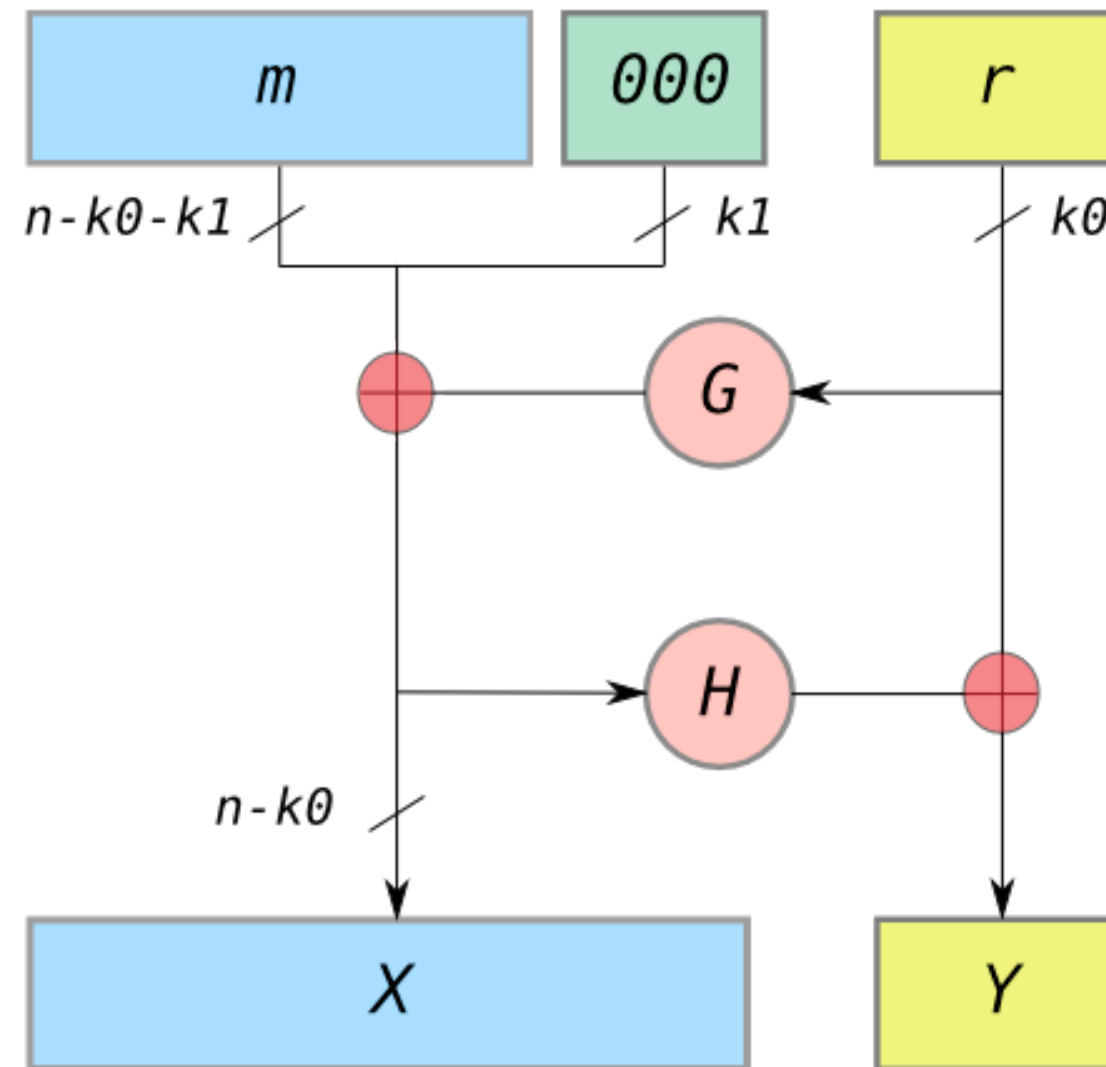
Optimal Asymmetric Encryption Padding (OAEP) - *Belare and Rogaway (1995)*

m - message (plaintext)

r - random number

G, H - cryptographic hash functions

X, Y - the message that gets encrypted



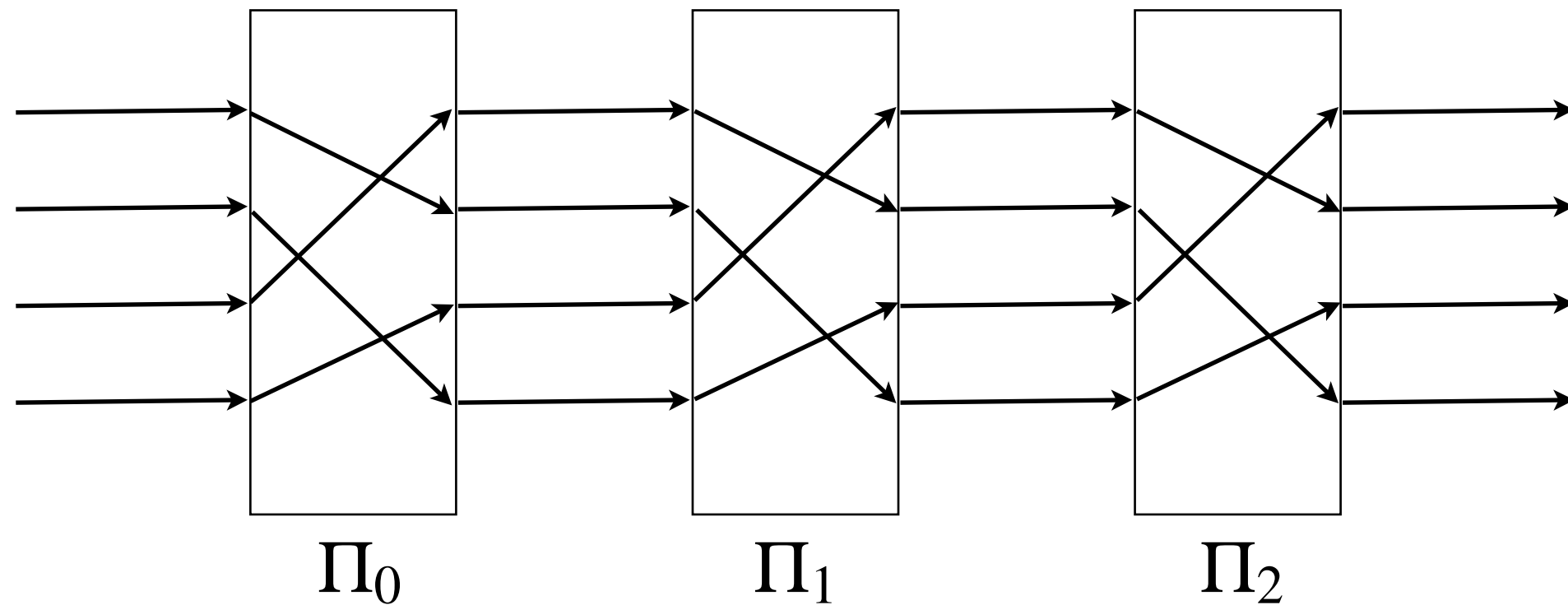
Cool trick: reencryption

$$E(M) \oplus E(0) = E(M)^*$$

Anybody can “reencrypt” a message.
(New random number introduced from $E(0)$.)

Reencryption mixnets

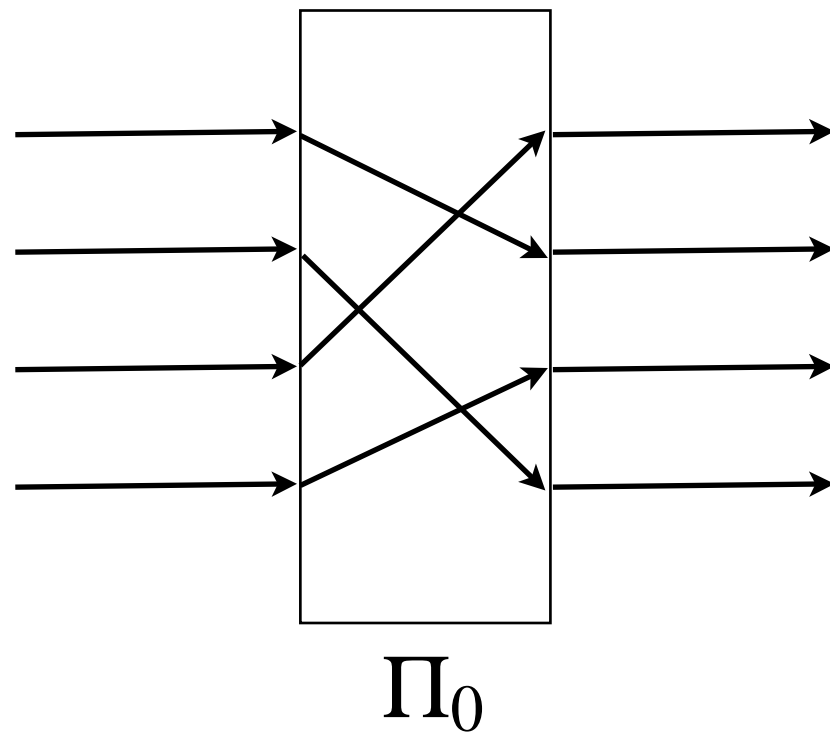
Permutations Π_i , where output is reencrypted.



**Each mix permutes/reencrypts.
Must prove output corresponds to input.**

Non-solution: reveal the mix

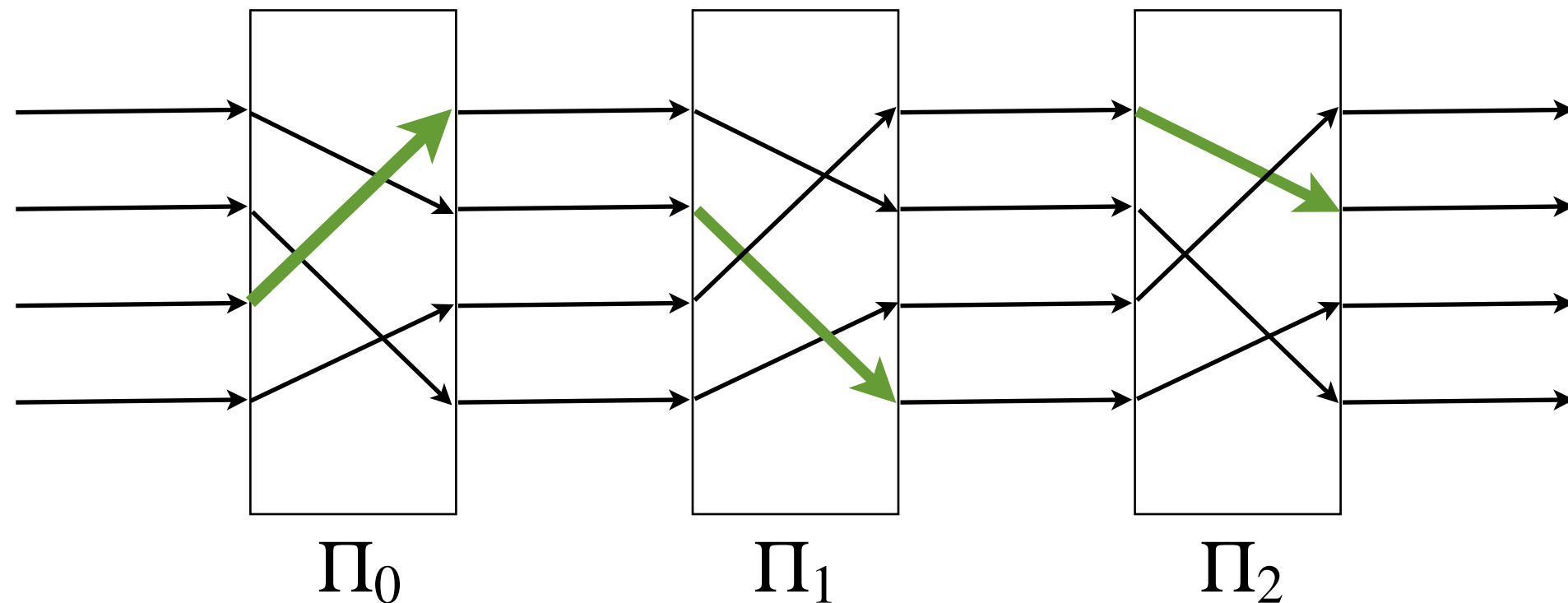
Publish the random numbers and the permutation.



Eliminates benefit of randomization.

Randomized partial checking

Effective across larger mixes.
(Jakobsson, Jules, Rivest '02)



Say we're mixing 1 million ballots, each mix reveals 1%. After five mixes, 99.99% chance that all ballots reencrypted at least once.

Zero-knowledge proofs (ZKP)

want to prove you know something

while revealing nothing

generalized format

prover: commit to something (e.g., reencryption mix output)

verifier: *challenge* the prover

prover: respond to the challenge

Example: Hamiltonian paths

Prover: “I know a HP over graph G .” Compute graph isomorphism H . Publish G , H .

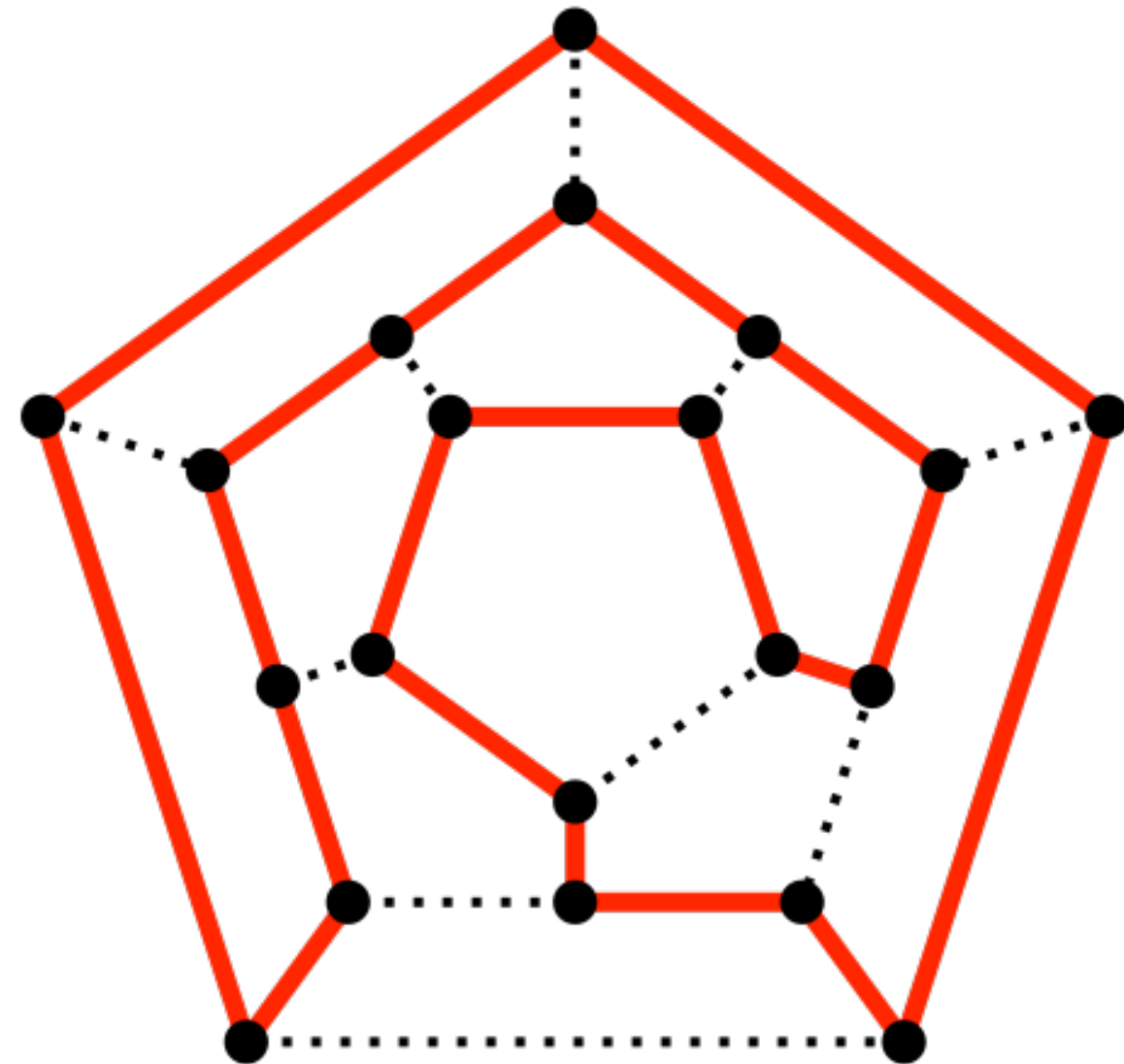
Verifier: Coin toss.

Heads: tell HP over H .

Tails: tell isomorphism G to H .

(Repeat N times.)

If prover doesn't know HP,
verifier catches with high
probability.

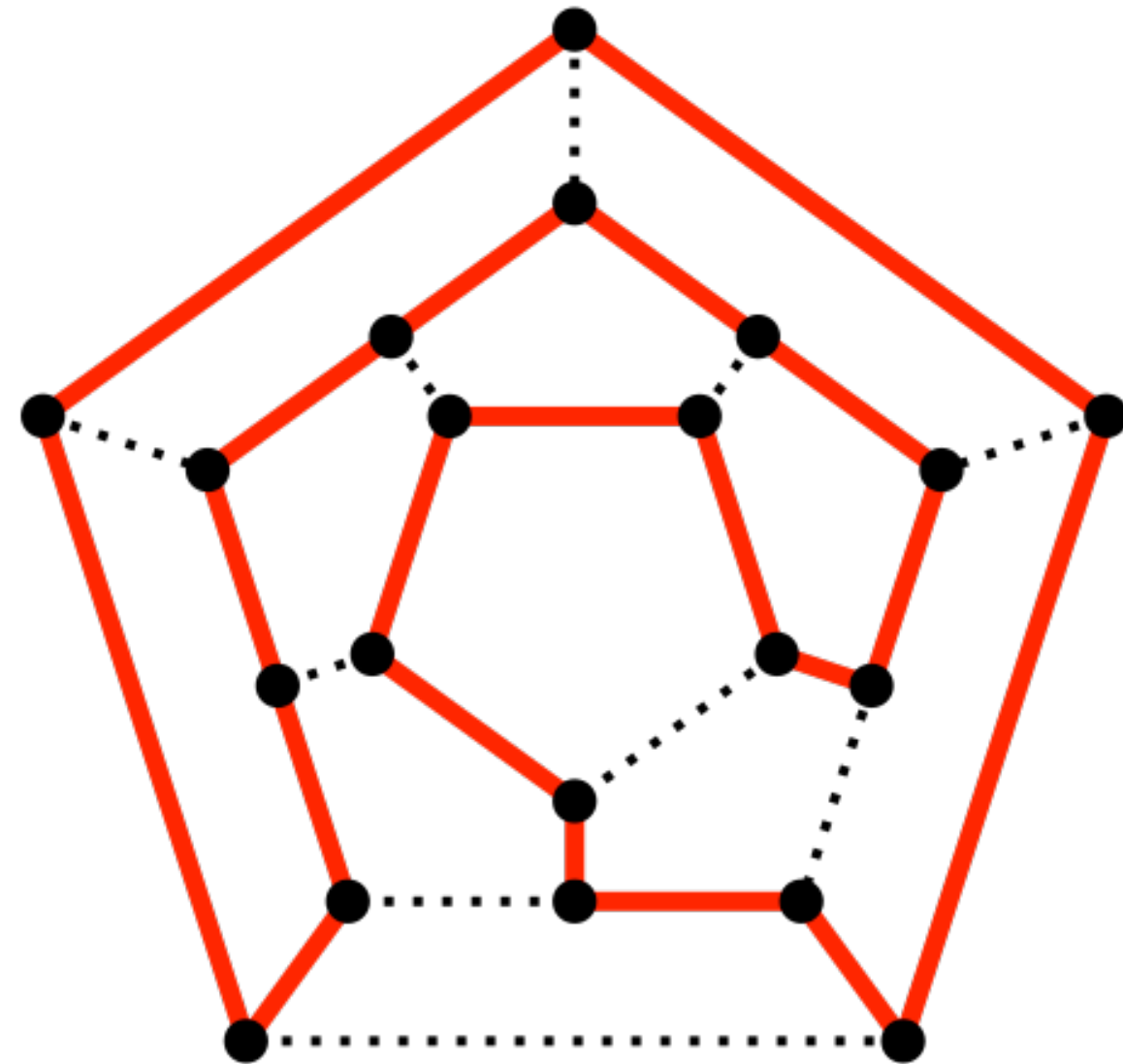


Non-interactive ZK proofs

Prover: Precompute N isomorphisms (H_1 to H_N) and hash them. Hash function yields coin tosses for virtual challenger. Then output the results.

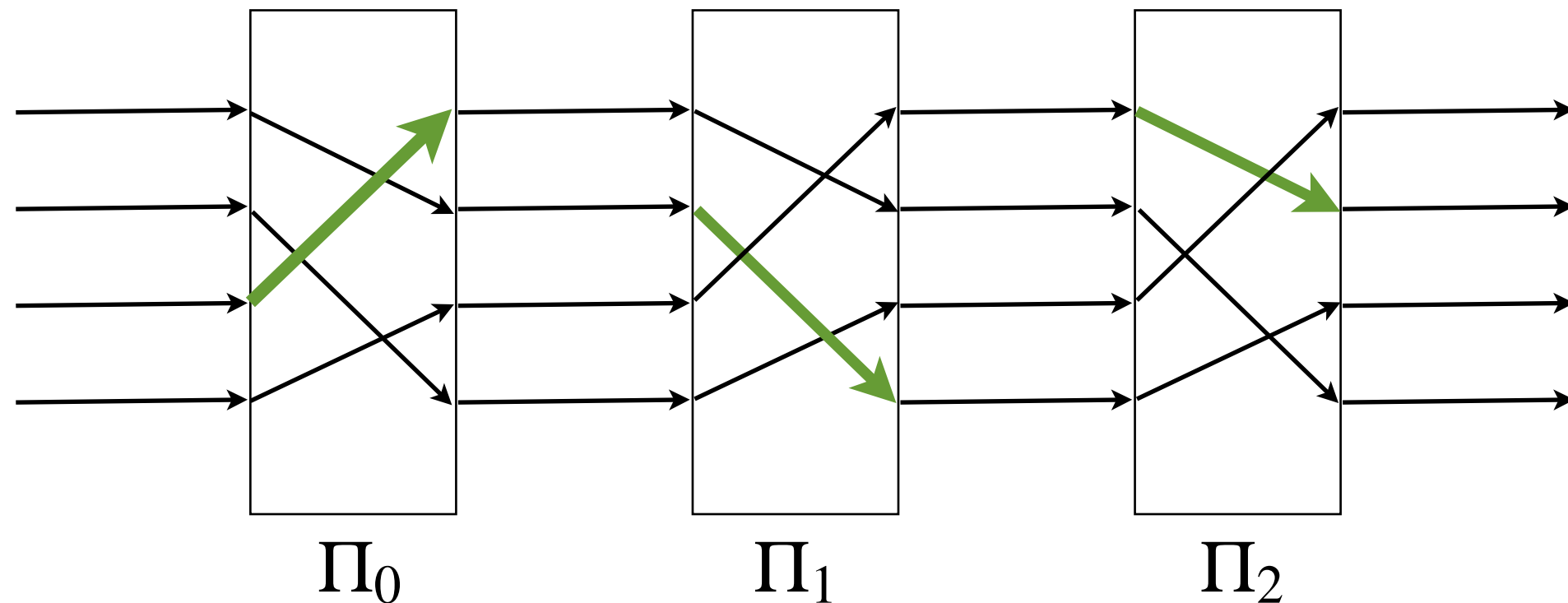
(Assumes good hash functions.)

This is an example of the *Fiat-Shamir heuristic* (1986).



NIZK variant for mixes

Hash the output of the permutation/reencryption. Use those bits to select which edges get revealed.



Say we're mixing 1 million ballots, each mix reveals 1%. After five mixes, 99.99% chance that all ballots reencrypted at least once.

Evil machine: E(bignum)?

Must prove ciphertext corresponds to well-formed plaintext. (Example, prove counters are zero or one.)

We need another ZK tool: Chaum-Pedersen proofs.

Prover knows: $(g, g^x), (h, h^x)$

Wants to prove that these two tuples share x

But x is a secret!

Chaum-Pedersen proofs (1992)

Goal: demonstrate $(g, g^x), (h, h^x)$

P: choose random $w \in \mathbb{Z}_p^*$, compute $(A = g^w, B = h^w)$

Send (A, B) to V

V: pick a random number c (challenge), send to P

P: compute $R = w + xc$

send R to V

V: Compute

$$\begin{aligned} A(g^x)^c &= g^w g^{xc} \\ &= g^{w+xc} \\ &= g^R \end{aligned}$$

$$\begin{aligned} B(h^x)^c &= h^w h^{xc} \\ &= h^{w+xc} \\ &= h^R \end{aligned}$$

Fake C-P proofs?

Goal: demonstrate $(g, g^x), (h, h^x)$

P: choose random $w \in \mathbb{Z}_p^*$, compute $(A = g^w, B = h^w)$

Send (A, B) to V

V: pick a random number c (challenge), send to P

P: compute $R = w + xc$

send R to V

V: Compute

$$\begin{aligned} A(g^x)^c &= g^w g^{xc} \\ &= g^{w+xc} \\ &= g^R \end{aligned}$$

$$\begin{aligned} B(h^x)^c &= h^w h^{xc} \\ &= h^{w+xc} \\ &= h^R \end{aligned}$$

Fake C-P proofs?

Goal: demonstrate $(g, g^x), (h, h^x)$

P: choose random $w \in \mathbb{Z}_p^*$, compute $(A = \cancel{g^w}, B = \cancel{h^w})$

Send (A, B) to V

V: pick a random number c (challenge), send to P

P: compute $R = w + xc$

send R to V

V: Compute

$$\begin{aligned} A(g^x)^c &= g^w g^{xc} \\ &= g^{w+xc} \\ &= g^R \end{aligned}$$

$$\begin{aligned} B(h^x)^c &= h^w h^{xc} \\ &= h^{w+xc} \\ &= h^R \end{aligned}$$

Fake C-P proofs?

Goal: demonstrate $(g, g^x), (h, h^x)$

P: choose random $w \in \mathbb{Z}_p^*$, compute $(A = g^w, B = h^w)$

Send (A, B) to **V** **P** chooses fake c, R s.t. $A = g^R (g^{xc})^{-1}$.

V: pick a random number c (challenge), send to **P**

P: compute $R = w + xc$
send R to **V**

V: Compute

$$\begin{aligned} A(g^x)^c &= g^w g^{xc} \\ &= g^{w+xc} \\ &= g^R \end{aligned}$$

$$\begin{aligned} B(h^x)^c &= h^w h^{xc} \\ &= h^{w+xc} \\ &= h^R \end{aligned}$$

Fake C-P proofs?

Goal: demonstrate $(g, g^x), (h, h^x)$

P: choose random $w \in \mathbb{Z}_p^*$, compute $(A = g^w, B = h^w)$

Send (A, B) to **V** **P** chooses fake c, R s.t. $A = g^R (g^{xc})^{-1}$.

V: pick a random number c (challenge), send to **P**

P: compute $R = w + xc$

send R to **V**

Observer can compute $A(g^x)^c \dots$

V: Compute

$$\begin{aligned} A(g^x)^c &= g^w g^{xc} \\ &= g^{w+xc} \\ &= g^R \end{aligned}$$

$$\begin{aligned} B(h^x)^c &= h^w h^{xc} \\ &= h^{w+xc} \\ &= h^R \end{aligned}$$

Fake C-P proofs?

Goal: demonstrate $(g, g^x), (h, h^x)$

P: choose random $w \in \mathbb{Z}_p^*$, compute $(A = g^w, B = h^w)$

Send (A, B) to V **P** chooses fake c, R s.t. $A = g^R (g^{xc})^{-1}$.

V: pick a random number c (challenge), send to P

P: compute $R = w + xc$

send R to V

Observer can compute $A(g^x)^c \dots$

V: Compute

$$\begin{aligned} A(g^x)^c &= g^w g^{xc} \\ &= g^{w+xc} \\ &= g^R \end{aligned}$$

$$\begin{aligned} B(h^x)^c &= h^w h^{xc} \\ &= h^{w+xc} \\ &= h^R \end{aligned}$$

ZK protocols only work when “live” (or use Fiat-Shamir heuristic for non-interactive)

C-P for vote testing

Can I prove a vote is zero or one? First, how about “proving” it’s zero using C-P.

Want to verify $\langle g^r, g^{ar} g^v \rangle$ for a specific value of v ?

“Prove we know r ” (or we could symmetrically prove we know a)

Do C-P protocol where $(g, g^x), (h, h^x)$ becomes $(g, g^r), \left(g^a, \frac{g^{ar} g^v}{g^v}\right)$

We could do this for any value of v

Cramer-Damgård-Schoenmakers (1996)

Can run two Chaum-Pedersen (or any two ZK proofs like this) simultaneously, one “real” and one “simulated”.

First, fake a proof (e.g., for $v = 1$) in advance.

Then, announce the first message for both protocols. Challenger sends c , prover announced a split c_0, c_1 where $c_0 + c_1 = c$, then executes both ZK protocols.

Verifier cannot tell which one was real vs. simulated, but knows that **one** of them was real.

Crypto summary

At the end of the day, **any** election observer can now:

- verify every single ballot for being “well-formed”
(valid ElGamal tuple, encrypted zero-or-one, etc.)
- add together all the ballots (homomorphically)
- verify a proof of the tally (Chaum-Pedersen again)
(only the election authority can generate this)

But we have no idea if the original ciphertext corresponded to the **intent of the voter** (versus evil machine flipping votes).

One newer, useful trick:

ballot challenge

Catch a machine if it cheats!

Benaloh challenges [2006]

Catch a machine if it cheats!

Benaloh challenges [2006]

voter makes selections

Catch a machine if it cheats!

Benaloh challenges [2006]

voter makes selections

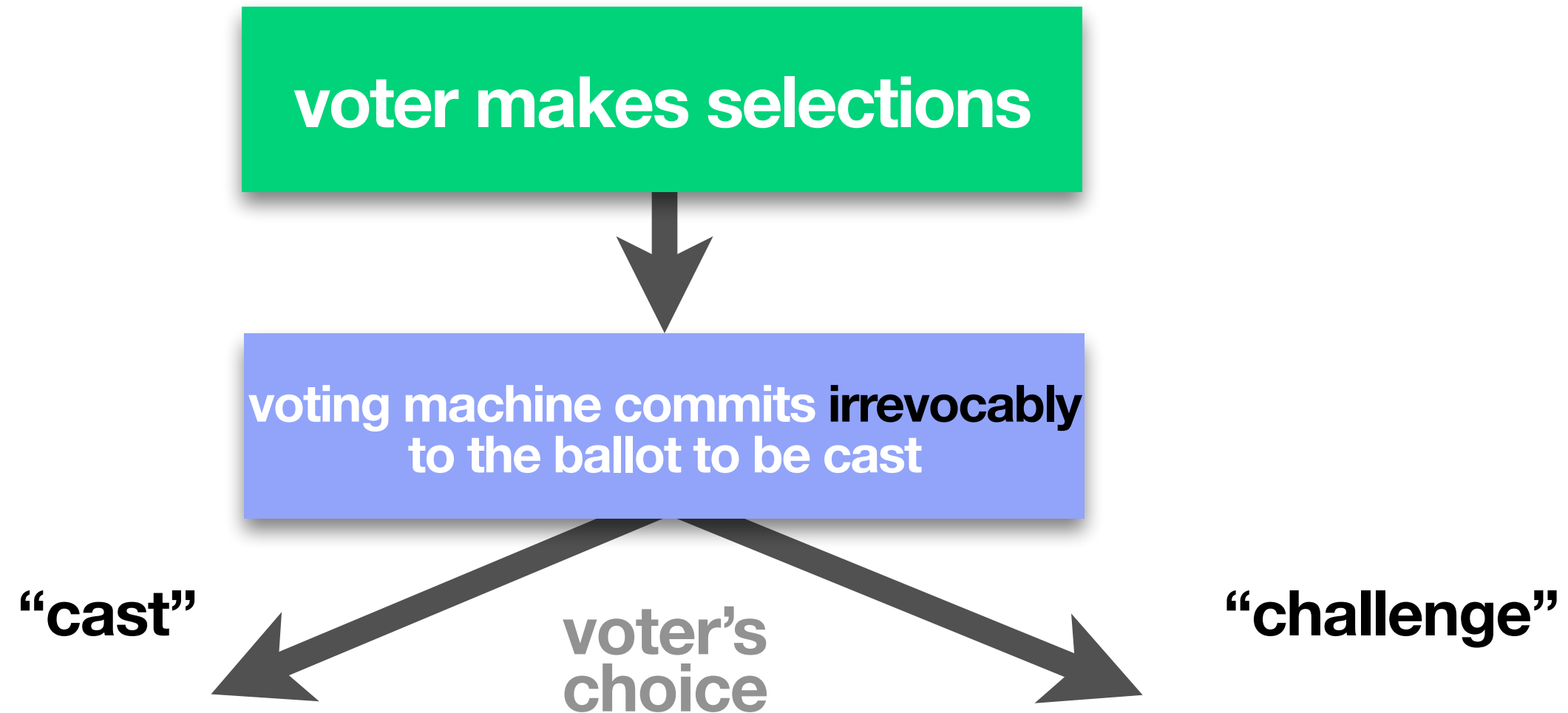


```
graph TD; A[voter makes selections] --> B[voting machine commits irrevocably to the ballot to be cast];
```

voting machine commits **irrevocably**
to the ballot to be cast

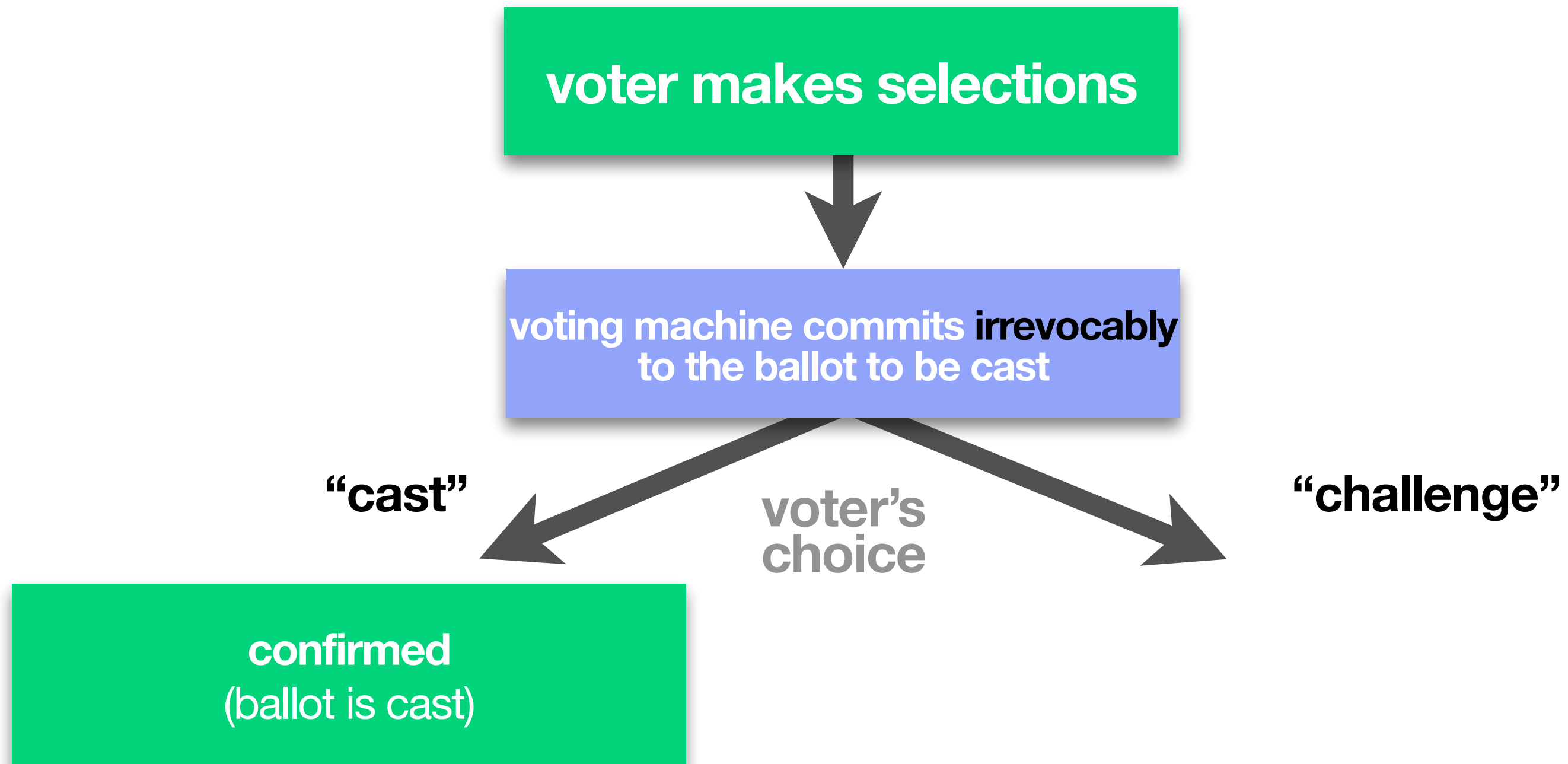
Catch a machine if it cheats!

Benaloh challenges [2006]



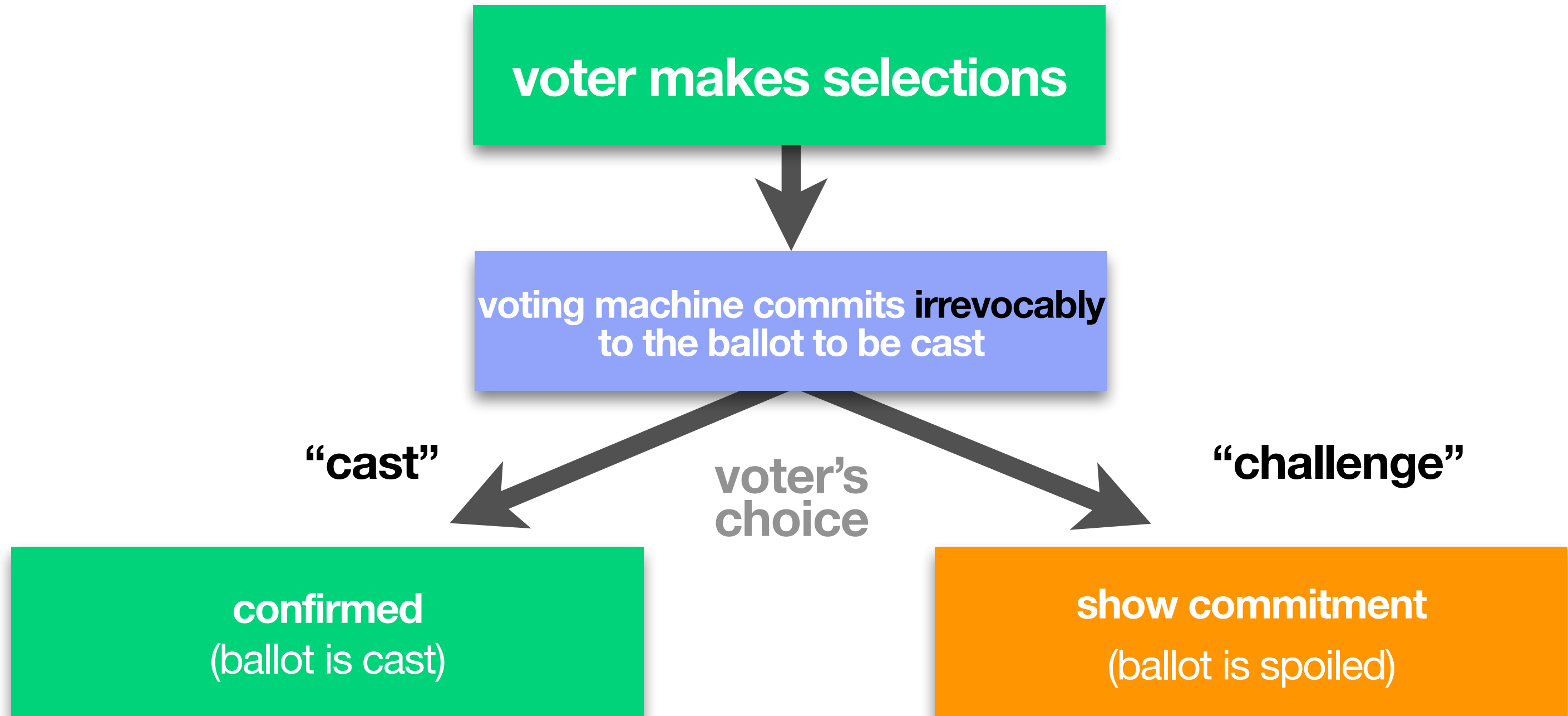
Catch a machine if it cheats!

Benaloh challenges [2006]



Catch a machine if it cheats!

Benaloh challenges [2006]



ElGamal extra goodies

Two ways to decrypt:

$$E(g^a, r, M) = \langle g^r, (g^a)^r M \rangle$$

$$D(g^r, g^{ar} M, a) = \frac{g^{ar} M}{(g^r)^a}$$

$$D(g^r, g^{ar} M, r) = \frac{g^{ar} M}{(g^a)^r}$$

g group generator

M plaintext (message)

r random (chosen at encryption time)

a (private) decryption key

g^a (public) encryption key

ElGamal extra goodies

Two ways to decrypt:

$$E(g^a, r, M) = \langle g^r, (g^a)^r M \rangle$$

$$D(g^r, g^{ar} M, a) = \frac{g^{ar} M}{(g^r)^a}$$

$$D(g^r, g^{ar} M, r) = \frac{g^{ar} M}{(g^a)^r}$$

Only election officials (or trustees) have a (or shares of a) to decrypt totals.

g group generator
 M plaintext (message)
 r random (chosen at encryption time)
 a (private) decryption key
 g^a (public) encryption key

ElGamal extra goodies

Two ways to decrypt:

$$E(g^a, r, M) = \langle g^r, (g^a)^r M \rangle$$

$$D(g^r, g^{ar} M, a) = \frac{g^{ar} M}{(g^r)^a}$$

$$D(g^r, g^{ar} M, r) = \frac{g^{ar} M}{(g^a)^r}$$

g	group generator
M	plaintext (message)
r	random (chosen at encryption time)
a	(private) decryption key
g^a	(public) encryption key

When challenged, machine reveals random nonces (r), observer can decrypt.

Challenging the machine

When challenged, the machine must reveal r

We can then decrypt this ballot (only) and see if it's what we expected to see

In Benaloh, the encrypted ballot is on paper

An **irrevocable** output medium

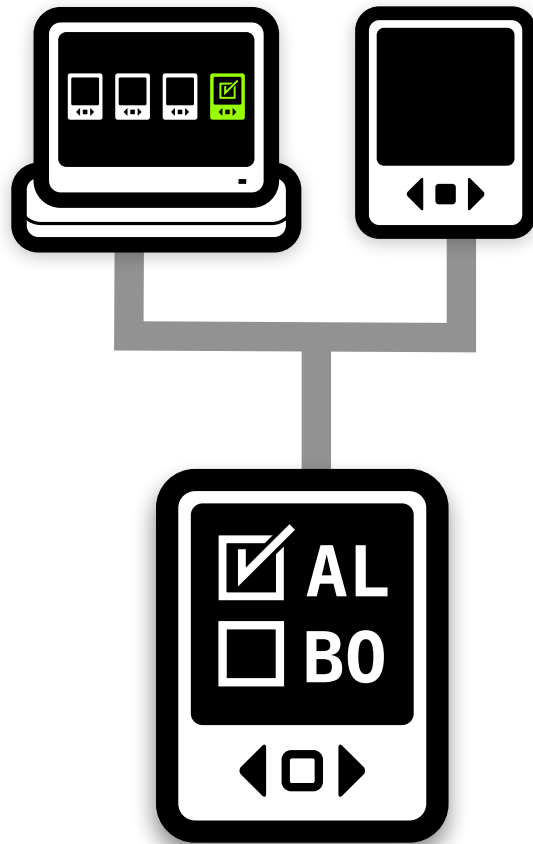
decrypting requires additional equipment

VoteBox (2008) has an irrevocable publishing system

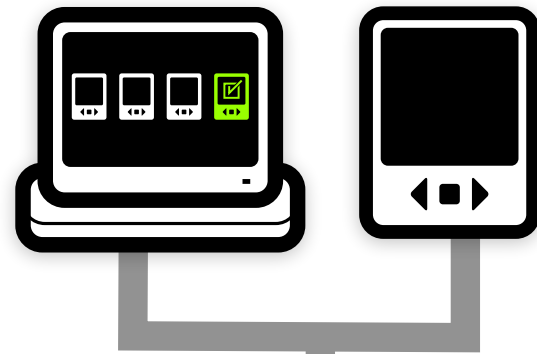
In-precinct LAN, where all machines replicate the history, plus hash entanglement.

Helios (also 2008) just showed it to you on the web

polling place

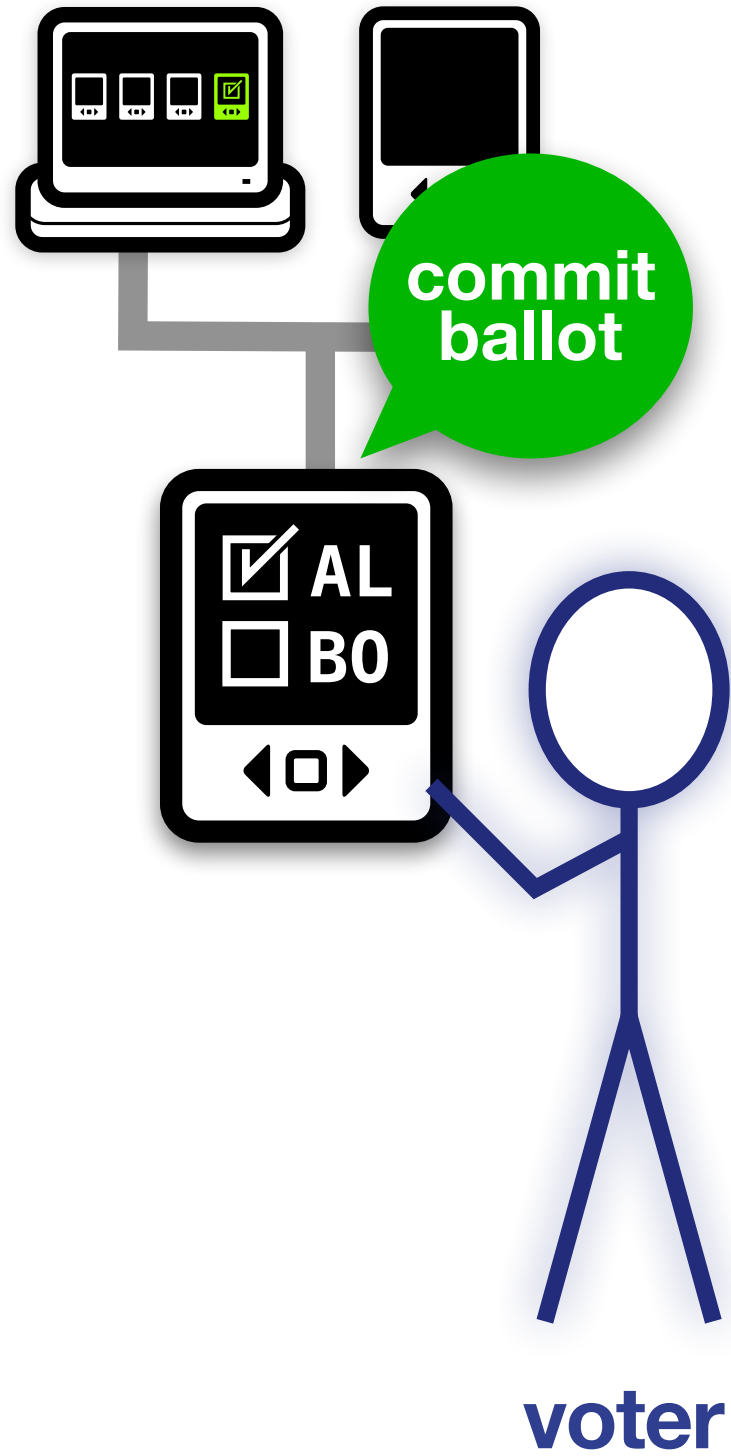


polling place

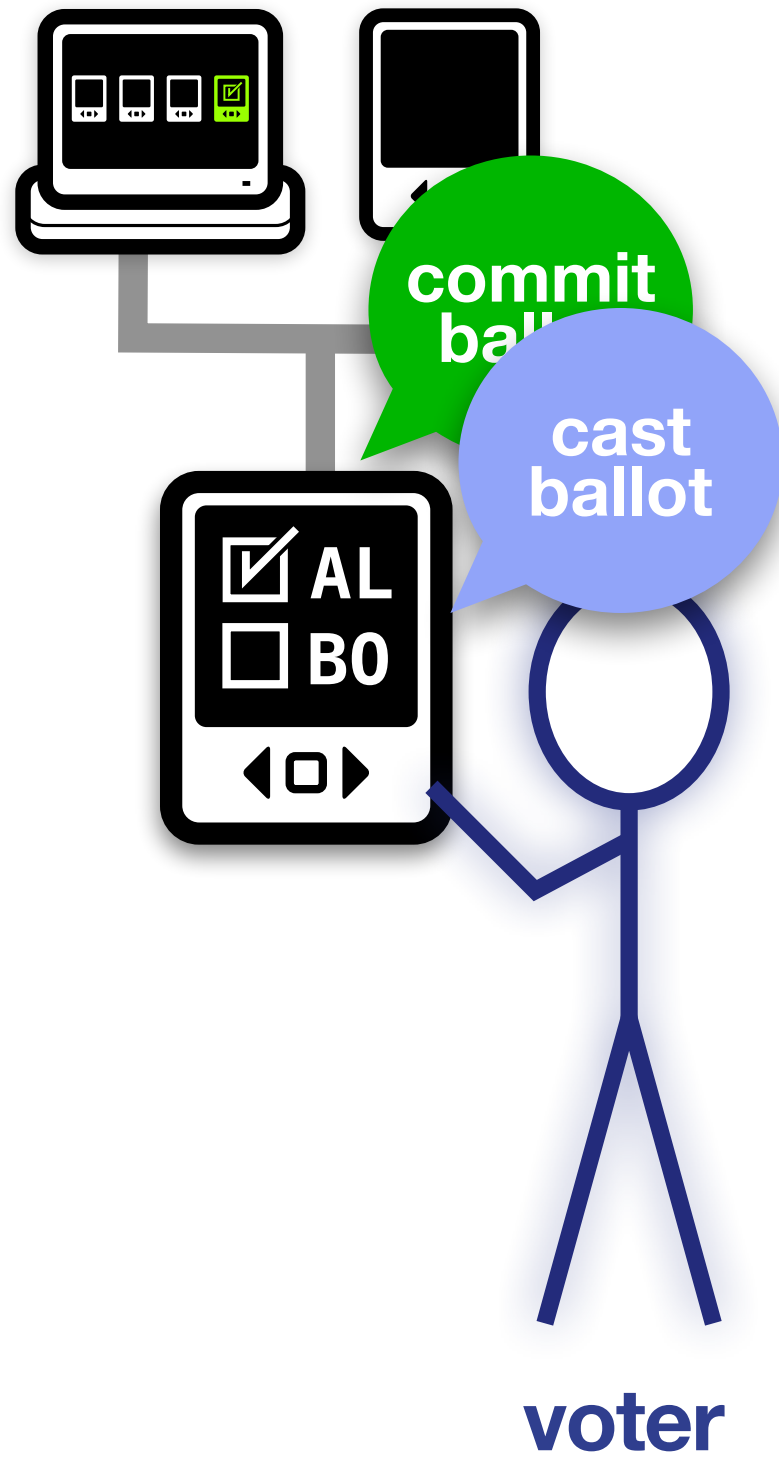


voter

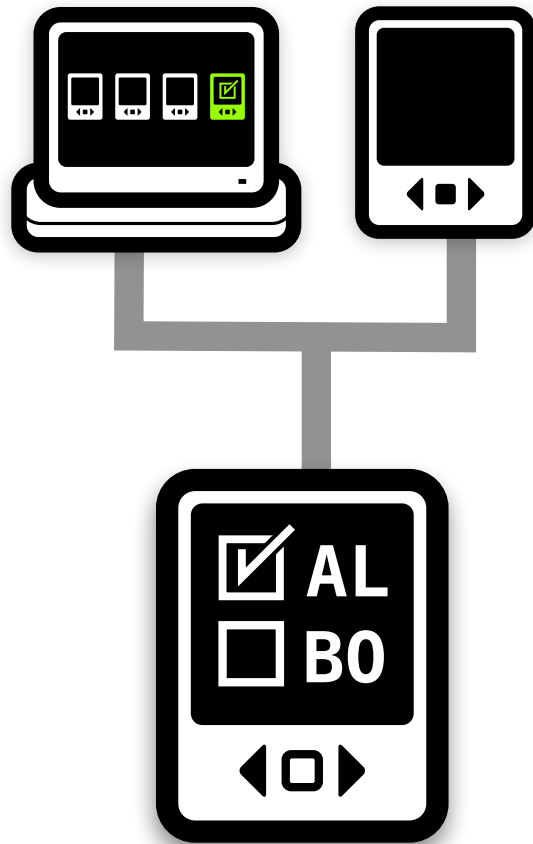
polling place



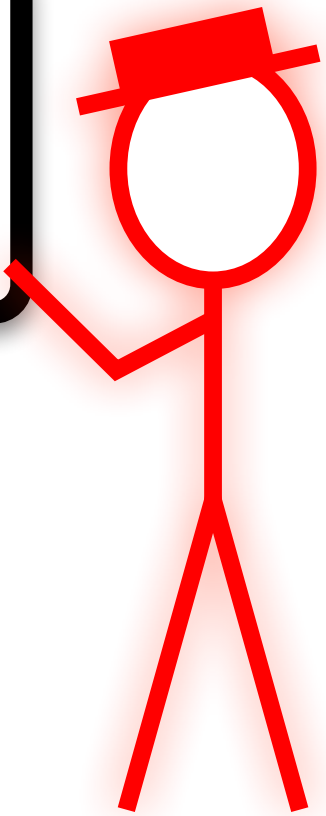
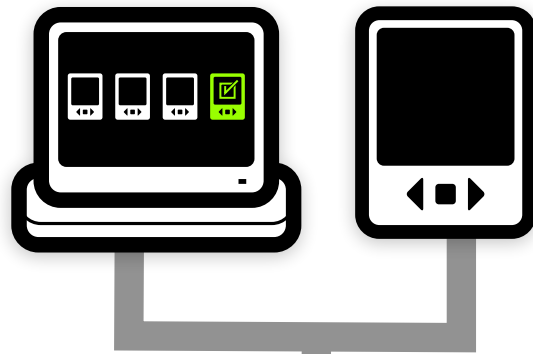
polling place



polling place

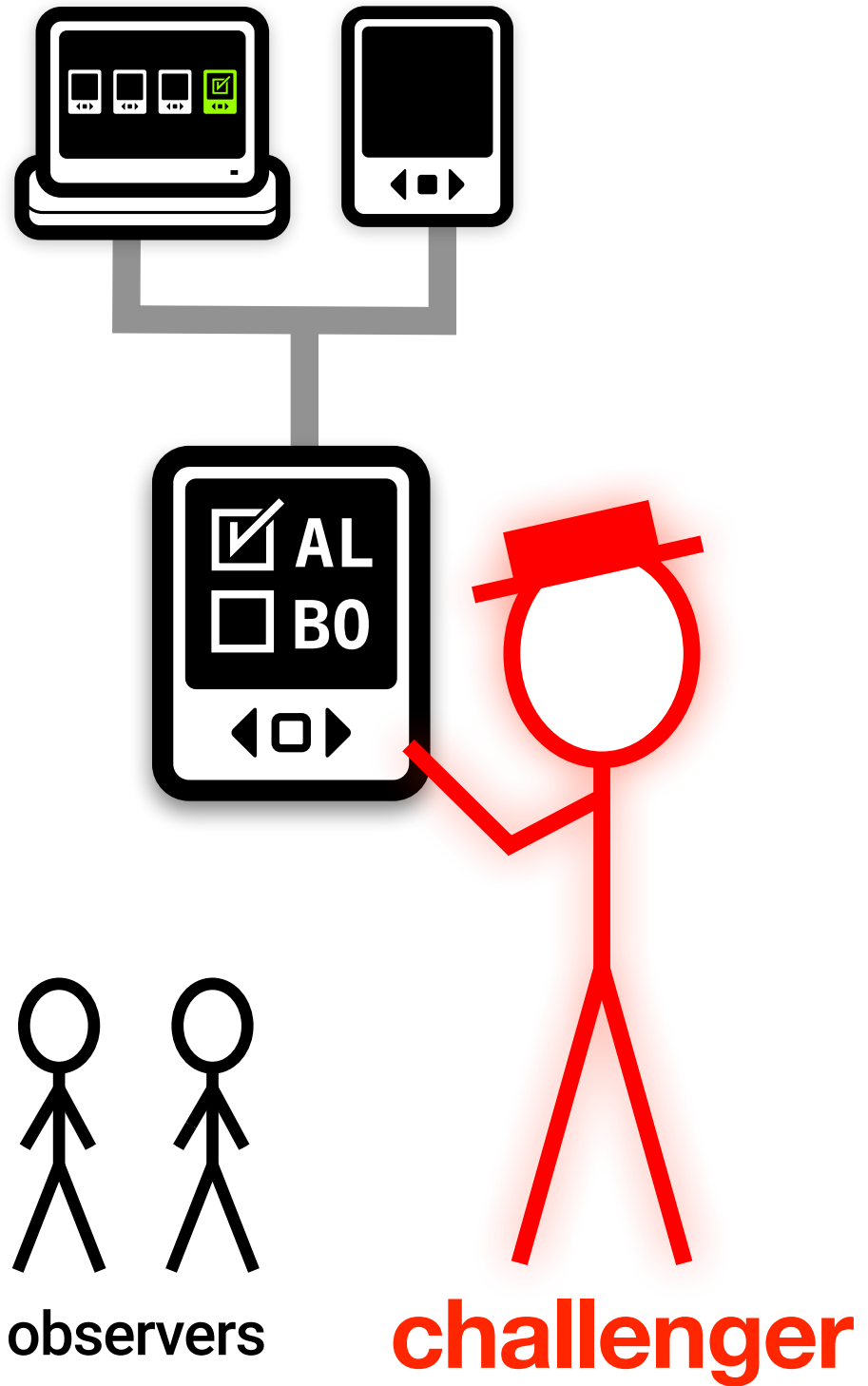


polling place

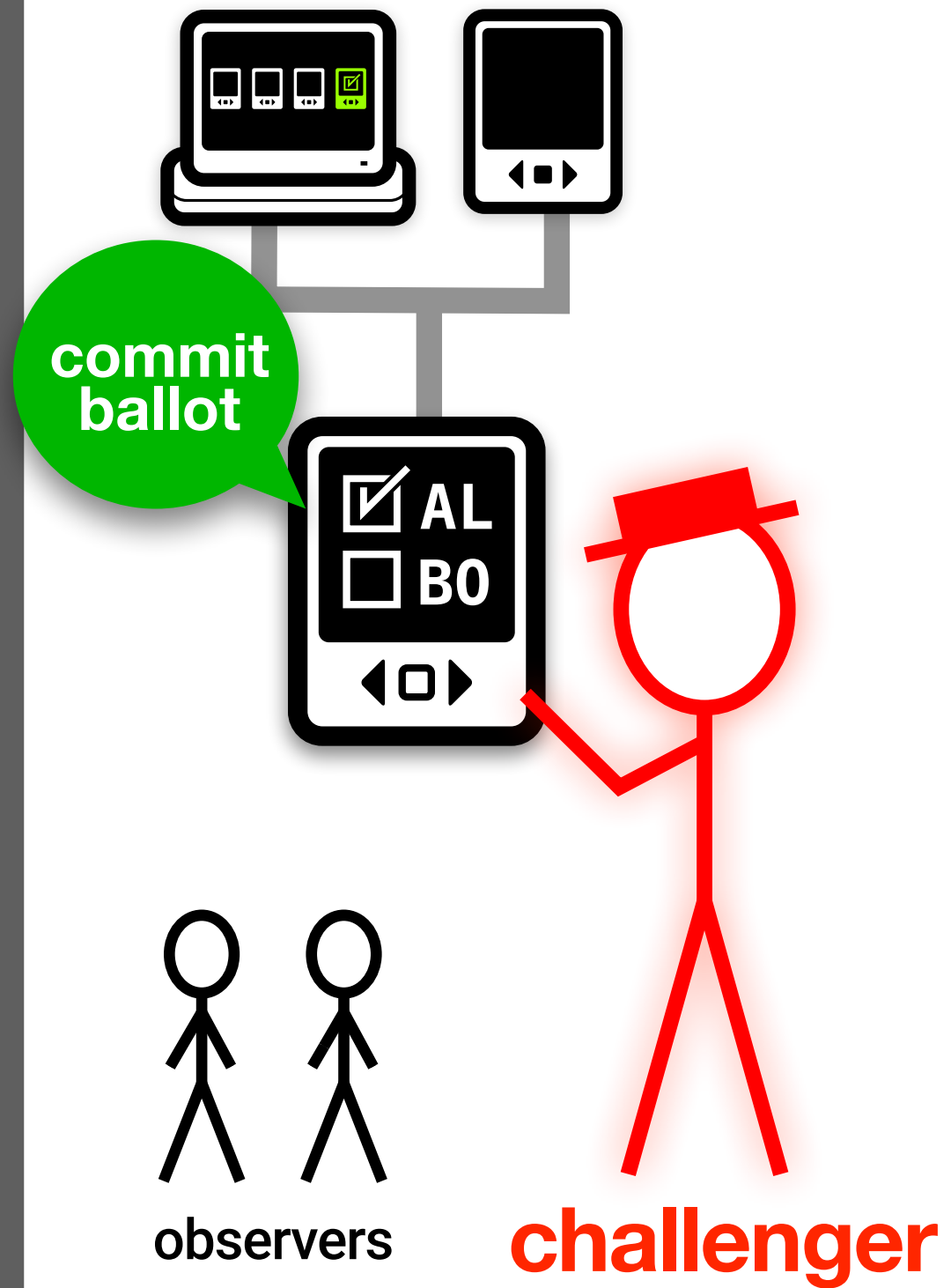


challenger

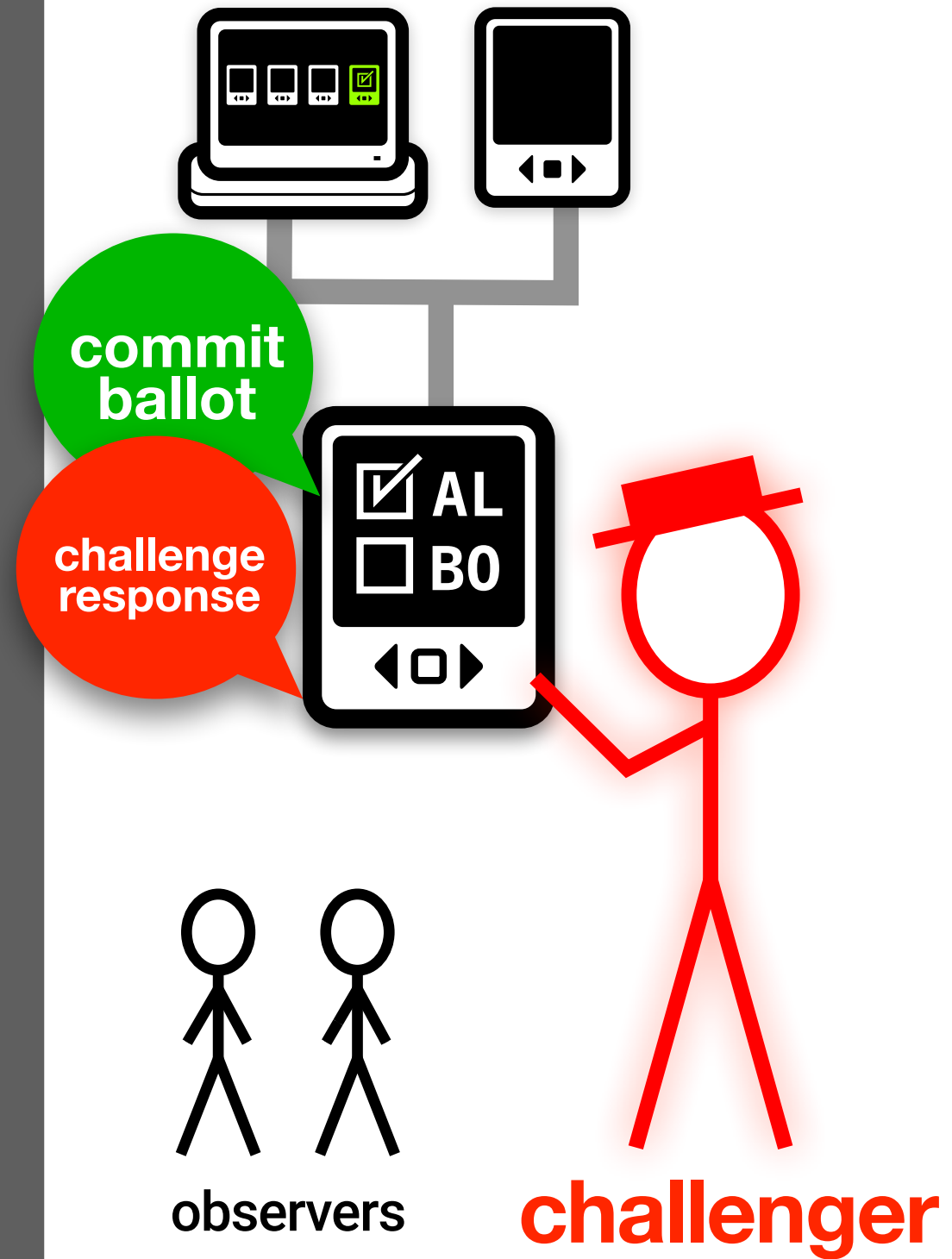
polling place



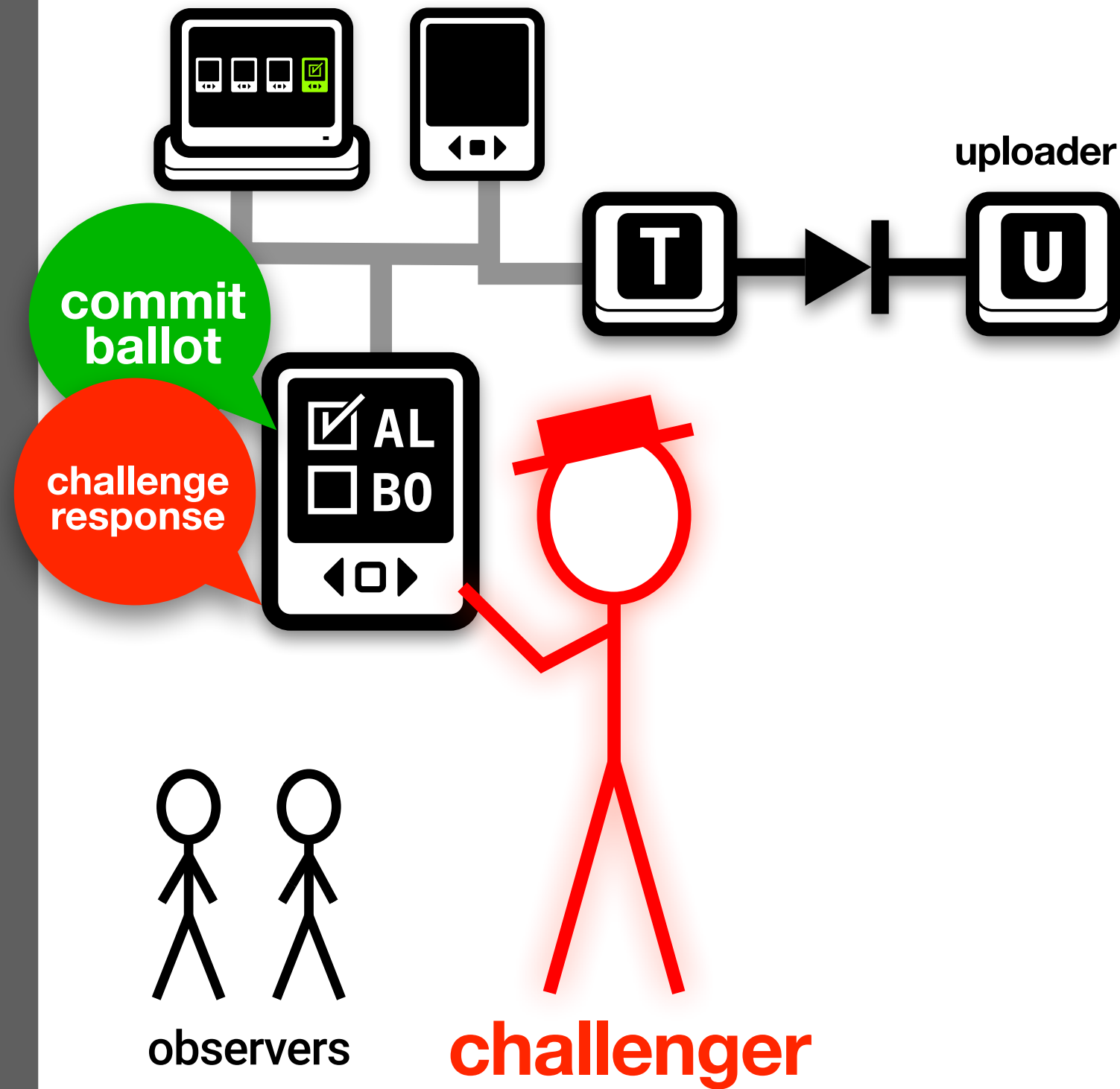
polling place



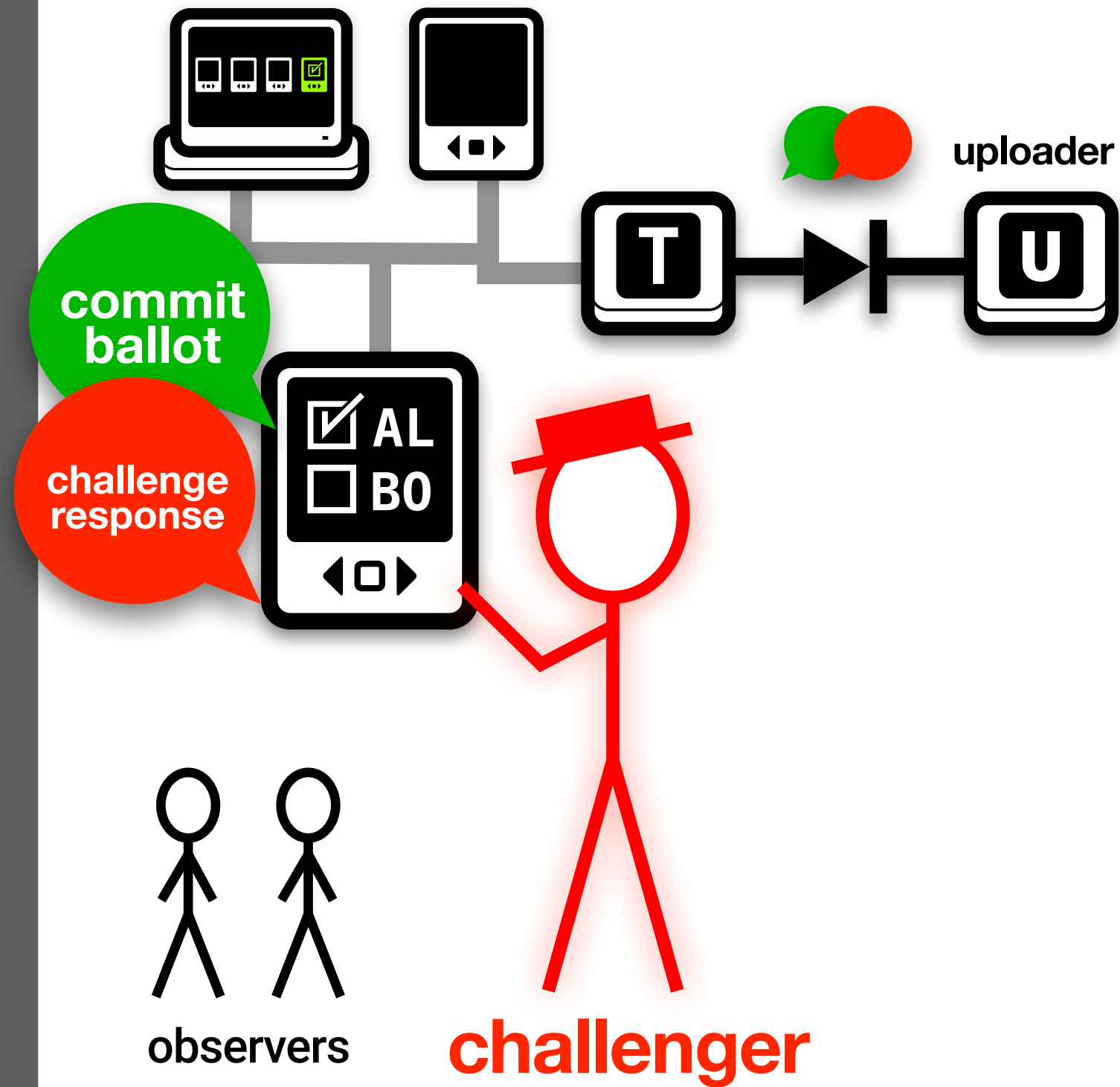
polling place



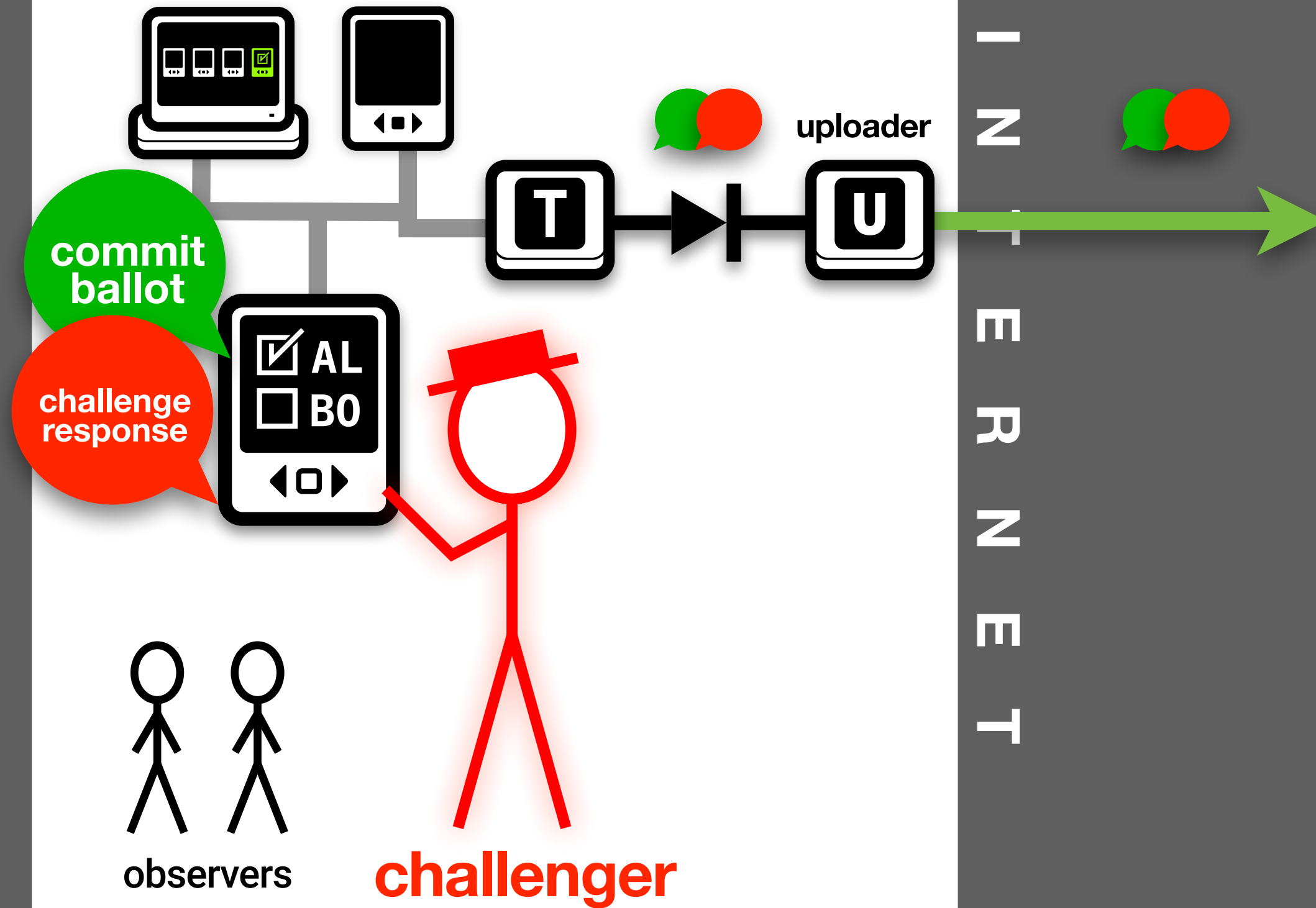
polling place

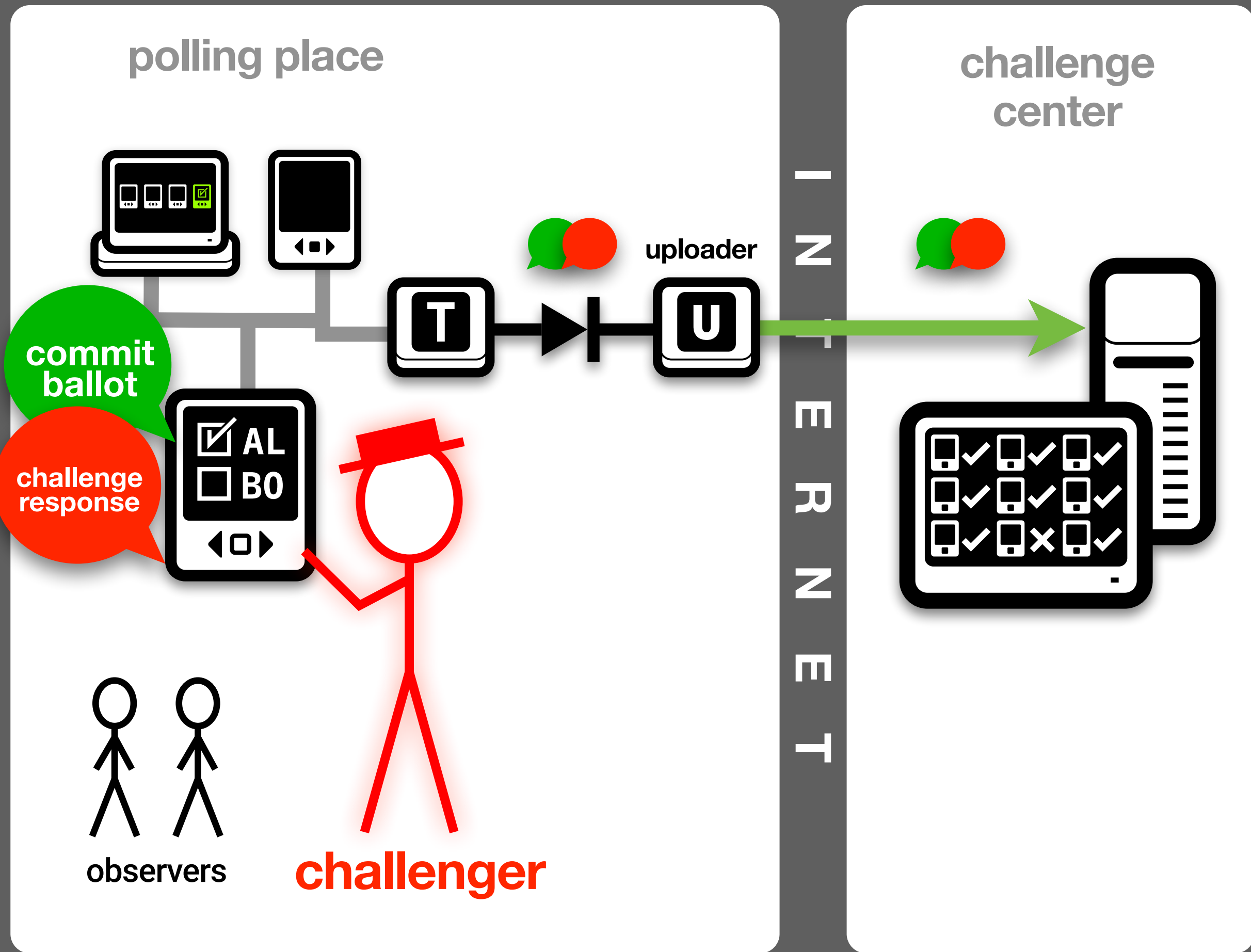


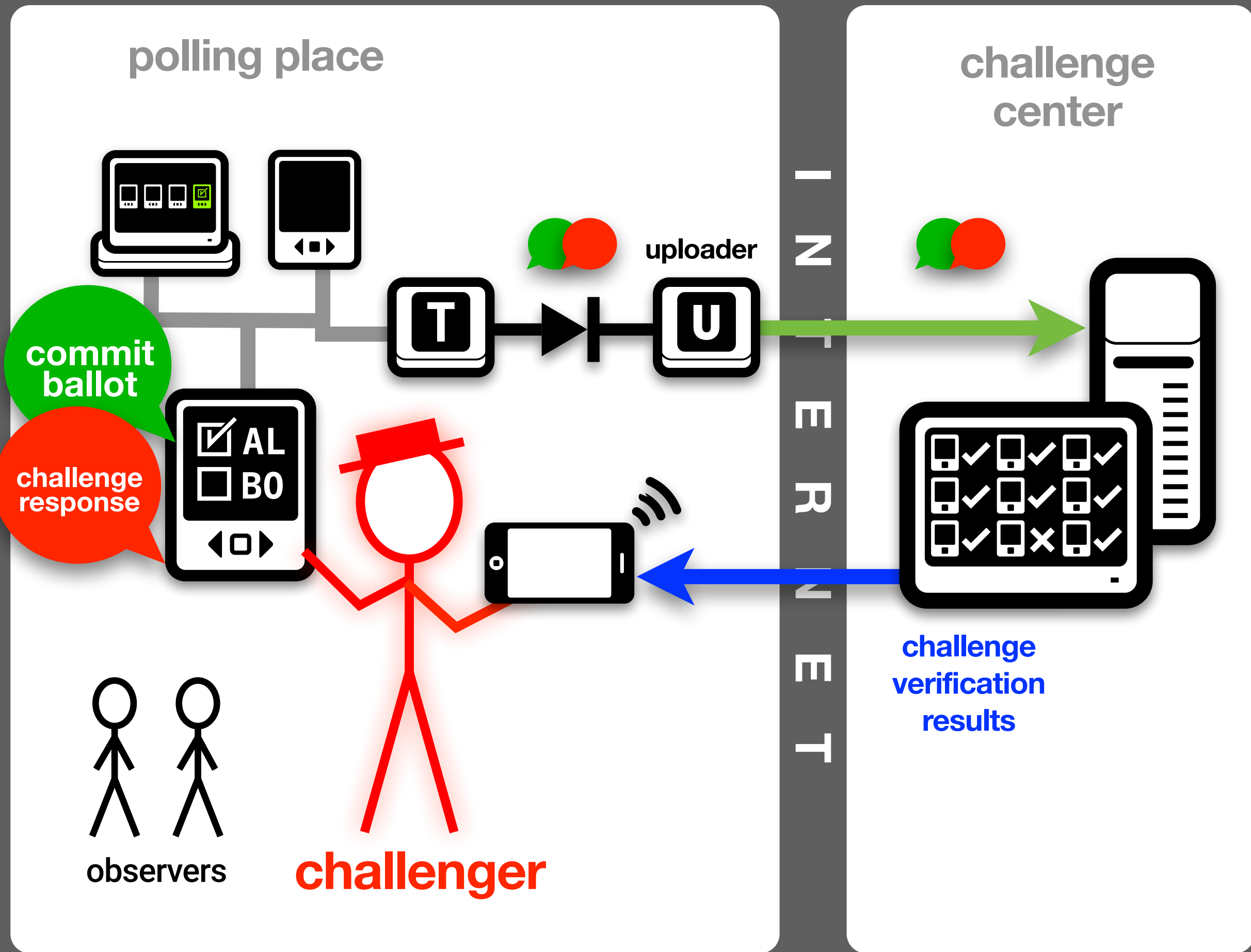
polling place

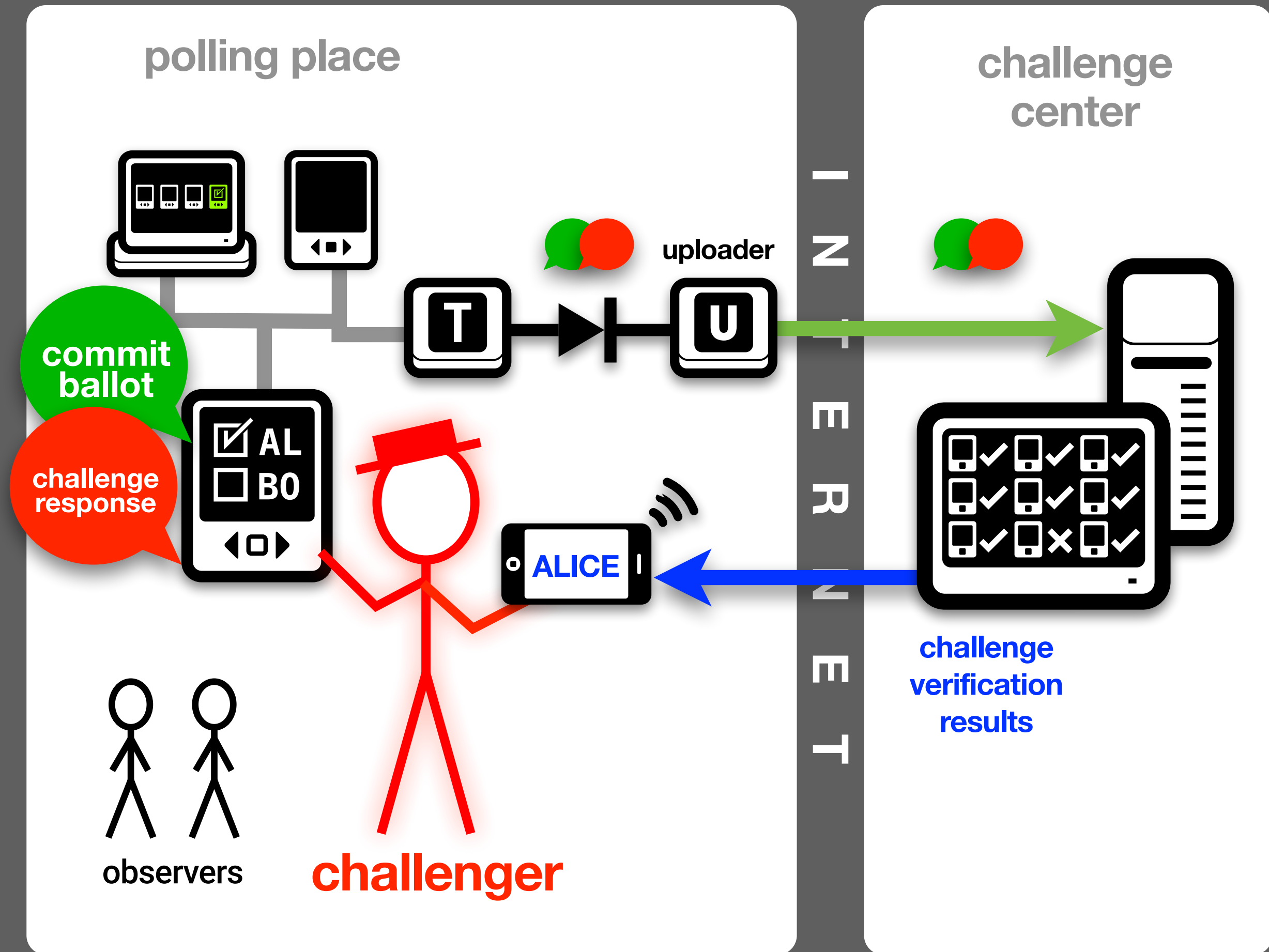


polling place









STAR challenges (2013)

Commitment: ciphertext broadcast to terminals

Happens when the ballot is printed, just like VoteBox

Challenge: voter deposits or keeps ballot

Challenger takes home printed ballot

Ballot randomness posted, anyone can decrypt (ballots not counted!)

Procedurally: same as a spoiled ballot

Big usability win

No need to ask the voter a challenge question

Simple “live parallel testing”

STAR Workflow: Registration

Registration



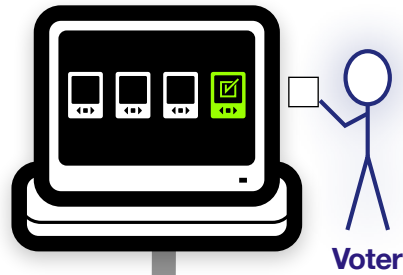
STAR Workflow: Registration

Registration



STAR Workflow: Registration

Registration



Workflow: Authorization

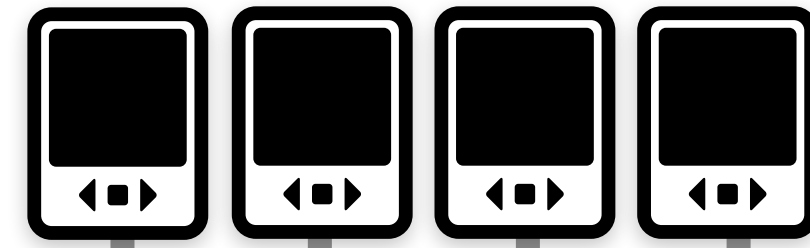
Registration



Controller



Voting terminals

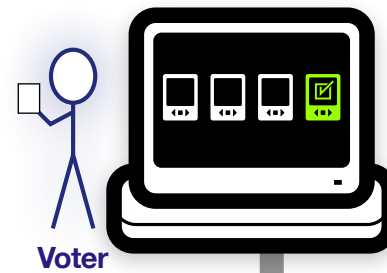


Workflow: Authorization

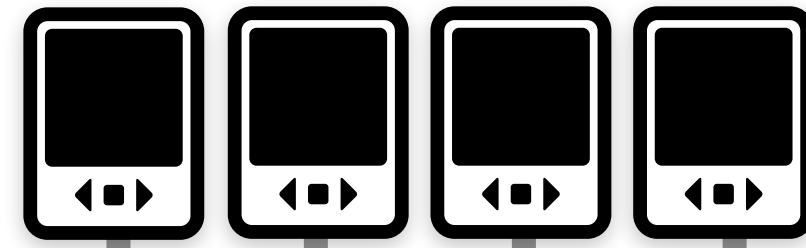
Registration



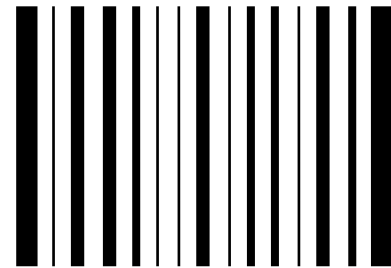
Controller



Voting terminals



Precinct 101A

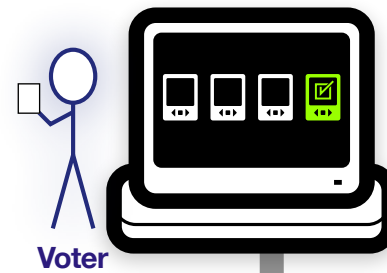


Workflow: Authorization

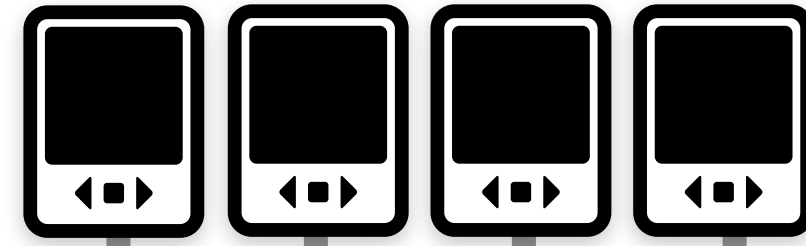
Registration



Controller



Voting terminals



Auth: 52794

Similar to Hart InterCivic eSlate

Workflow: Voting

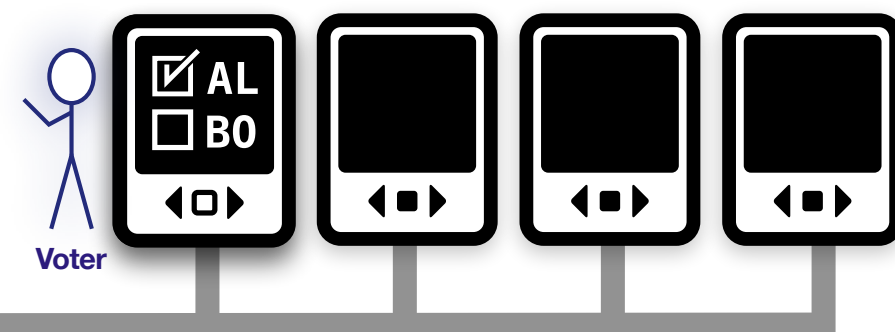
Registration



Controller



Voting terminals



Workflow: Ciphertext copies distributed

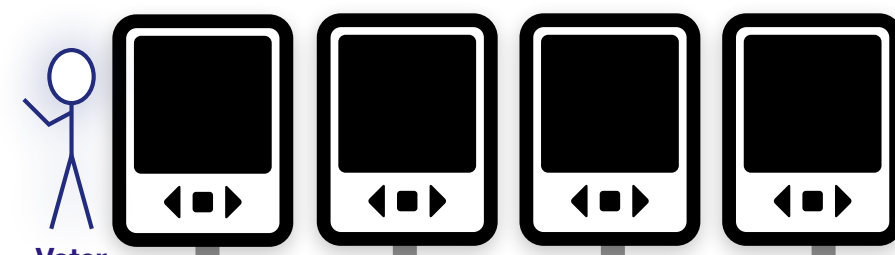
Registration



Controller



Voting terminals



Workflow: Ciphertext copies distributed

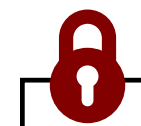
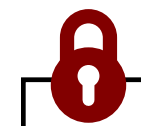
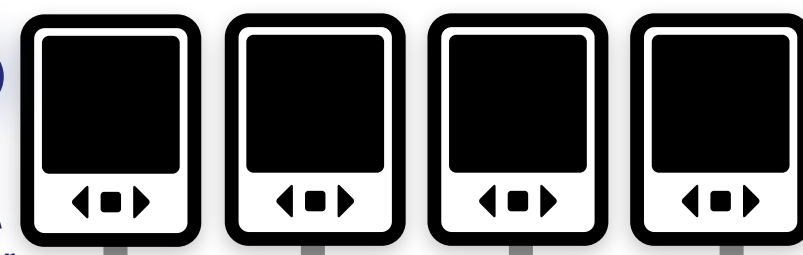
Registration



Controller



Voting terminals

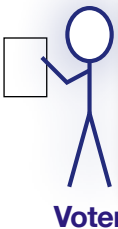


Workflow: Casting

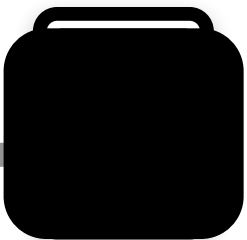
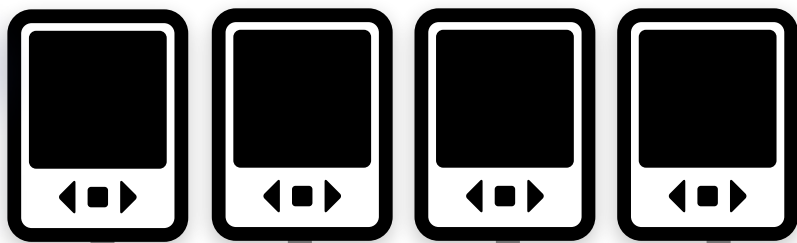
Registration



Controller



Voting terminals



Ballot box

Official Ballot November 4, 2012
Joint General and Special Elections
Travis County, Texas Precinct 101A

11042012 Pct 101A BID11042012 BCID3457894 Pg 1 of 2

Travis County General Election continued
Precinct 145, Justice of the Peace

Travis County General Election	
Straight Party	PURP
District 210, United States Representative	PURP
Governor	PURP
Lieutenant Governor	PURP
Attorney General	PURP
State Senator	PURP
Comptroller of Public Accounts	GLD
Attorney General	PURP
State Senator	PURP
Comptroller of Public Accounts	GLD
Commissioner of the General Land Office	PURP
Commissioner of Agriculture	PURP
Railroad Commissioner	PURP
Place 334, Justice, Supreme Court	NO SELECTION
Place 445, Justice, Supreme Court	NO SELECTION
Place 549, Justice, Supreme Court	NO SELECTION
Place 223, Judge, Court of Criminal Appeals	NO SELECTION
Place 552, Judge, Court of Criminal Appeals	NO SELECTION
Railroad Commissioner	PURP
Place 334, Justice, Supreme Court	NO SELECTION
Place 667, Judge, Court of Criminal Appeals	NO SELECTION
District 589, Member State Board of Education	PURP
District 257, State Senator	NO SELECTION
District 147, State Representative	PURP
County Judge	PURP
County Court at Law 677, Judge	PURP
County Probate Court Judge	PURP
District Clerk	PURP
County Clerk	GLD
County Treasurer	PURP
District Clerk	PURP
County Clerk	GLD
County Treasurer	PURP
County Commissioner	PURP
Railroad Commissioner	PURP
Place 332, Justice, Supreme Court	NO SELECTION
Place 554, Justice, Supreme Court	NO SELECTION
Place 998, Justice, Supreme Court	NO SELECTION
Place 221, Judge, Court of Criminal Appeals	NO SELECTION
Place 155, Judge, Court of Criminal Appeals	NO SELECTION
Place 166, Judge, Court of Criminal Appeals	NO SELECTION
Place 332, Justice, Supreme Court	NO SELECTION
Place 554, Justice, Supreme Court	NO SELECTION
District 245, Member State Board of Education	PURP
Place 442, Justice, 33rd Court of Appeals District	PURP

Workflow: Casting

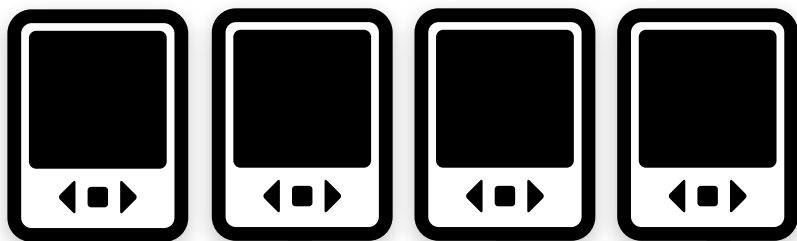
Registration



Controller



Voting terminals



Official Ballot November 4, 2012
Joint General and Special Elections
Travis County, Texas Precinct 101A

11042012 Pct 101A BID11042012 BCID3457894 Pg 1 of 2

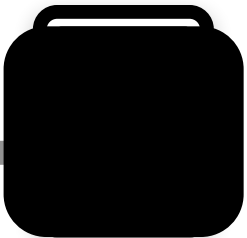
Travis County General Election continued

Precinct 145, Justice of the Peace

PURP	Nancy Nu	<input checked="" type="radio"/>
District 147, State Representative	Xena Xi	<input checked="" type="radio"/>
County Judge	Oscar Omicron	<input checked="" type="radio"/>
County Court at Law 677, Judge	Peggy Pi	<input checked="" type="radio"/>
County Probate Court Judge	Rhoda Rho	<input checked="" type="radio"/>
District Clerk	Samuel Sigma	<input checked="" type="radio"/>
County Clerk	Teresa Tau	<input checked="" type="radio"/>
County Treasurer	Uma Upsilon	<input checked="" type="radio"/>
District Clerk	Selena Sigma	<input checked="" type="radio"/>
County Clerk	Thomas Tau	<input checked="" type="radio"/>
County Treasurer	Ulysses Upsilon	<input checked="" type="radio"/>
County Commissioner	Phillip Phi	<input checked="" type="radio"/>
Railroad Commissioner	Charles Chi	<input checked="" type="radio"/>
Place 332, Justice, Supreme Court	NO SELECTION	<input type="radio"/>
Place 554, Justice, Supreme Court	NO SELECTION	<input type="radio"/>
Place 998, Justice, Supreme Court	NO SELECTION	<input type="radio"/>
Place 221, Judge, Court of Criminal Appeals	NO SELECTION	<input type="radio"/>
Place 155, Judge, Court of Criminal Appeals	NO SELECTION	<input type="radio"/>
Place 166, Judge, Court of Criminal Appeals	NO SELECTION	<input type="radio"/>
Place 332, Justice, Supreme Court	NO SELECTION	<input type="radio"/>
Place 554, Justice, Supreme Court	NO SELECTION	<input type="radio"/>
District 245, Member State Board of Education	Patrice Psi	<input checked="" type="radio"/>
Place 442, Justice, 33rd Court of Appeals District	Orlando Omega	<input checked="" type="radio"/>



Voter



Ballot box

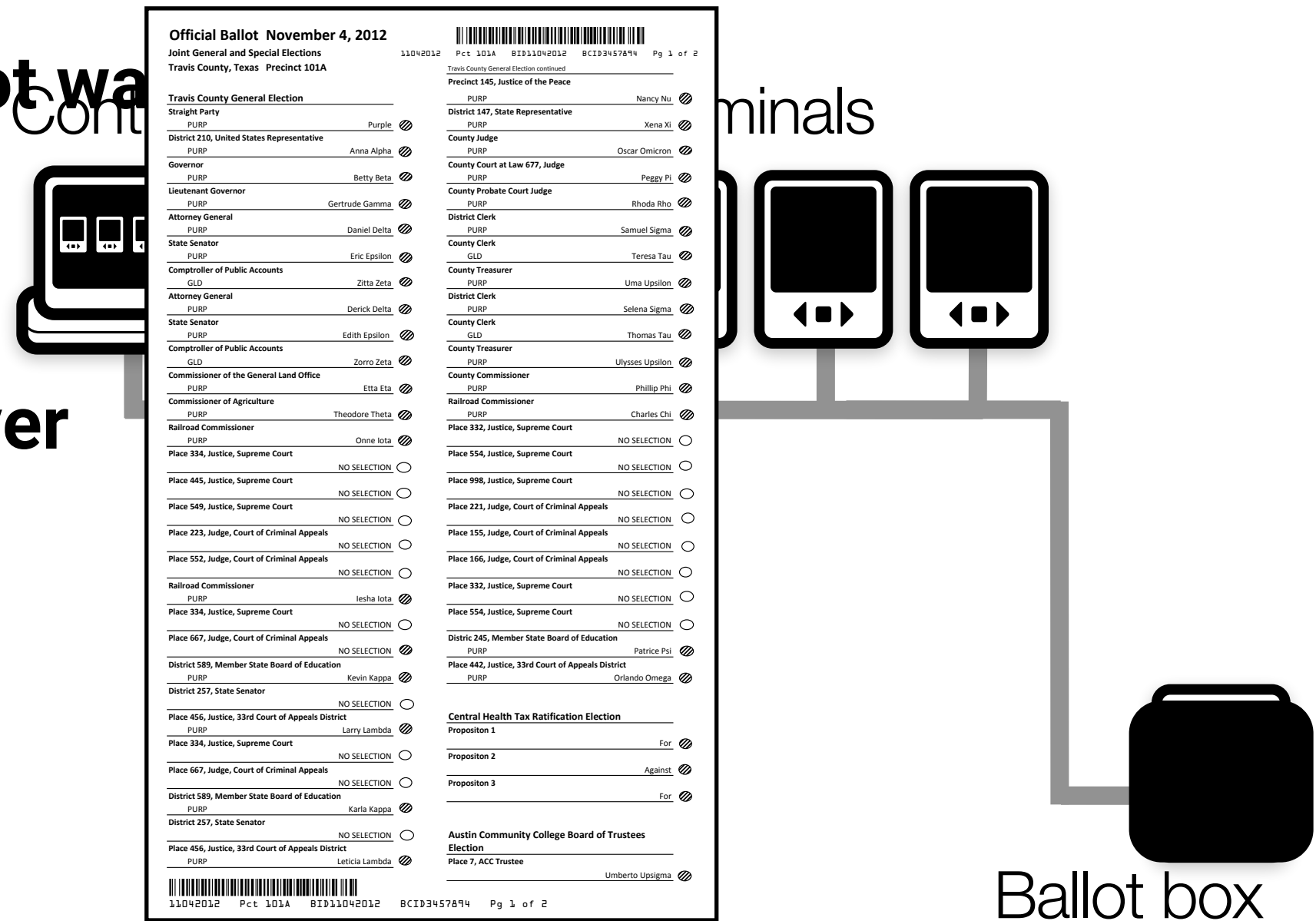
Networked ballot box

Notifies other machines that ballot was
Ballot has random ID

Voter can spoil ballot and start over
Usability win!

Ballot box has no UI

Deposit and done (just need a scanner for the ballot ID)



Networked ballot box

Notifies other machines that ballot was
Ballot has random ID

Voter can spoil ballot and start over
Usability win!

Ballot box has no UI

Deposit and done (just need a scanner for the ballot ID)

Official Ballot November 4, 2012

Joint General and Special Elections

Travis County, Texas Precinct 101A

11042012

Pct 101A

BID11042012

BCID3457894

Pg 1 of 2

Travis County General Election

Straight Party

PURP

Purple

District 210, United States Representative

PURP

Anna Alpha

Governor

PURP

Betty Beta

Lieutenant Governor

PURP

Gertrude Gamma

Attorney General

PURP

Daniel Delta

State Senator

PURP

Eric Epsilon

Comptroller of Public Accounts

GLD

Zitta Zeta

Attorney General

PURP

Derick Delta

State Senator

PURP

Edith Epsilon

Comptroller of Public Accounts

GLD

Zorro Zeta

Commissioner of the General Land Office

PURP

Etta Eta

Commissioner of Agriculture

PURP

Theodore Theta

Railroad Commissioner

PURP

Onne Iota

Place 334, Justice, Supreme Court

NO SELECTION

Place 445, Justice, Supreme Court

NO SELECTION

Place 549, Justice, Supreme Court

NO SELECTION

Place 223, Judge, Court of Criminal Appeals

NO SELECTION

Place 552, Judge, Court of Criminal Appeals

NO SELECTION

Railroad Commissioner

PURP

Iesha Iota

Place 334, Justice, Supreme Court

NO SELECTION

Place 667, Judge, Court of Criminal Appeals

NO SELECTION

District 589, Member State Board of Education

PURP

Kevin Kappa

District 257, State Senator

NO SELECTION

Place 456, Justice, 33rd Court of Appeals District

PURP

Larry Lambda

Place 334, Justice, Supreme Court

NO SELECTION

Place 667, Judge, Court of Criminal Appeals

NO SELECTION

District 589, Member State Board of Education

PURP

Karla Kappa

District 257, State Senator

NO SELECTION

Place 456, Justice, 33rd Court of Appeals District

PURP

Leticia Lambda

11042012

Pct 101A

BID11042012

BCID3457894

Pg 1 of 2

Travis County General Election continued

Precinct 145, Justice of the Peace

PURP

Nancy Nu

District 147, State Representative

PURP

Xena Xi

County Judge

PURP

Oscar Omicron

County Court at Law 677, Judge

PURP

Peggy Pi

County Probate Court Judge

PURP

Rhoda Rho

District Clerk

PURP

Samuel Sigma

County Clerk

GLD

Teresa Tau

County Treasurer

PURP

Uma Upsilon

District Clerk

PURP

Selena Sigma

County Clerk

GLD

Thomas Tau

County Treasurer

PURP

Ulysses Upsilon

County Commissioner

PURP

Phillip Phi

Railroad Commissioner

PURP

Charles Chi

Place 332, Justice, Supreme Court

NO SELECTION

Place 554, Justice, Supreme Court

NO SELECTION

Place 998, Justice, Supreme Court

NO SELECTION

Place 221, Judge, Court of Criminal Appeals

NO SELECTION

Place 155, Judge, Court of Criminal Appeals

NO SELECTION

Place 166, Judge, Court of Criminal Appeals

NO SELECTION

Place 332, Justice, Supreme Court

NO SELECTION

Place 554, Justice, Supreme Court

NO SELECTION

District 245, Member State Board of Education

PURP

Patrice Psi

Place 442, Justice, 33rd Court of Appeals District

PURP

Orlando Omega

Central Health Tax Ratification Election

Propositon 1

For

Against

Propositon 2

For

Against

Propositon 3

For

Against

Austin Community College Board of Trustees Election

Place 7, ACC Trustee

Umberto Upsilon

The diagram illustrates a networked ballot box system. A central ballot box is connected via a network to two voting machines. The machines are labeled 'Criminal' and 'Ballot box'. The ballot box is shown as a black bag-like container. The voting machines are represented by icons with screens and buttons. The network is depicted as a series of lines connecting the machines to the ballot box.

Post-election verification

Separate page to take home

Ballot hash for lookup on public bulletin board

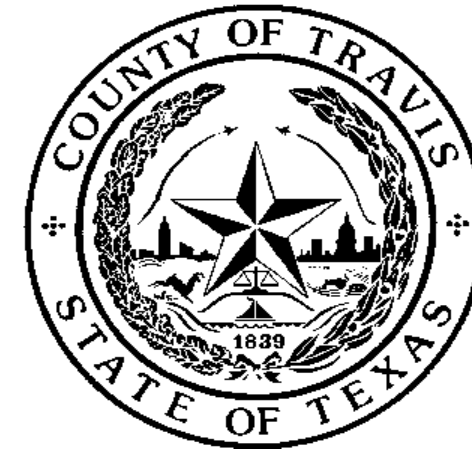
Cast ballot: ciphertext will match

Challenge ballot: plaintext also present, verifiable

Homomorphic tallying: provably includes all ballots on the bulletin board

Hash chains: also publicly verifiable

YOUR VOTE COUNTS



Thank you for voting!

Take this confirmation of voting with you

Verify your ballot at:

www.star-vote.org/ballot/HV1235Z7568RK84

Or, scan this code with your phone:



Find your code on the STAR-Vote website to ensure that your vote was recorded correctly.

Look for Election results and other tools for confirming the election at:
www.traviscountyelections.org

Voting Date: October 30, 2012
Voting Terminal: UI12345

Location: Randall's South Mopac
Time: 18:45:56

Risk limiting audits

Random sampling of individual paper ballots

Each should exactly match up with electronic records

Required in many states, growing in popularity

- *Requires touching tens of ballots, rarely hundreds, unlikely more*

Mitigates malicious software in the ballot scanners

Discrepancies require more samples, degenerates to a full manual recount of the ballots



Voting + Cryptography

Microsoft's open 'ElectionGuard' SDK aims to secure democratic elections

ElectionGuard is open source and built to make sure votes are verifiable, secure, and auditable.

DAN THORP-LANCASTER 6 May 2019

14



How *not* to do e2e cryptography

Ceci n'est pas une preuve

The use of trapdoor commitments in Bayer-Groth proofs
and the implications for the verifiability of the
Scytl-SwissPost Internet voting system*

Sarah Jamie Lewis¹, Olivier Pereira², and Vanessa Teague³

¹Open Privacy Research Society, sarah@openprivacy.ca

²UCLouvain – ICTeam, B-1348 Louvain-la-Neuve, Belgium,
olivier.pereira@uclouvain.be

³The University of Melbourne, Parkville, Australia,
vjteague@unimelb.edu.au

March 12, 2019

The implementation of the commitment scheme in the SwissPost-Scytl mixnet uses a trapdoor commitment scheme, which allows an authority who knows the trapdoor values to generate a shuffle proof transcript that passes verification but actually alters votes. We give two examples of details of how this could be used. The first example allows the first mix to use the trapdoors to substitute votes for which it knows the randomness used to generate the encrypted vote. The second example does not even require knowledge of the random factors used to generate the votes, and could be used by the last mix in the sequence.

*Since making this work public, we have learned that the same issue was identified independently by Thomas Haines of NTNU, and also by Rolf Haenni of the Bern University of Applied Sciences, <https://e-voting.bfh.ch/publications/2019/>

Swiss Post Public Intrusion Test

Undetectable Attack Against Vote Integrity
and Secrecy

ROLF HAENNI

Bern University of Applied Sciences

1 Theoretical Background

Commitment schemes are usually *perfectly hiding* and *computationally binding*. This means that no information about the message can be derived from the commitment and that the commitment can not be opened to a message other than the original one. The Pedersen commitment scheme achieves these properties by computing $c = G^m H^r$ in a multiplicative cyclic group, for which the discrete logarithm assumption (DL) is believed to hold. This scheme is perfectly binding, because a randomization $r' \neq r$ exists for any other message $m' \neq m$ such that $c = G^{m'} H^{r'}$. This means that c could potentially be opened to all q messages from \mathbb{Z}_q , but this requires the computation of the discrete logarithm. The Swiss Post voting protocol works with a subgroup $G_q \subseteq \mathbb{Z}_p^*$ of integers modulo p , where $q = |G_q|$ denotes the prime order of the subgroup, $m \in \mathbb{Z}_q$ the message, and $r \in \mathbb{Z}_q$ the randomization. Both G and H are elements of G_q .

A pre-condition for the scheme to be computationally binding is the *independence* of the two values $G, H \in G_q \setminus \{1\}$ (in a group of prime order q , both values are generators of G_q). Independence means that respective discrete logarithms $h = \log_G H$ and $g = \log_H G$ are unknown to everyone. Otherwise, for example if $h = \log_G H$ is known to the person who created c , then c can be rewritten as

$$c = G^m H^r = G^m (G^h)^r = G^{m+hr \bmod q}.$$

Therefore, to open c to a different message $m' \neq m$, the adversary can easily solve

$$m + hr \equiv m' + hr' \pmod{q}$$

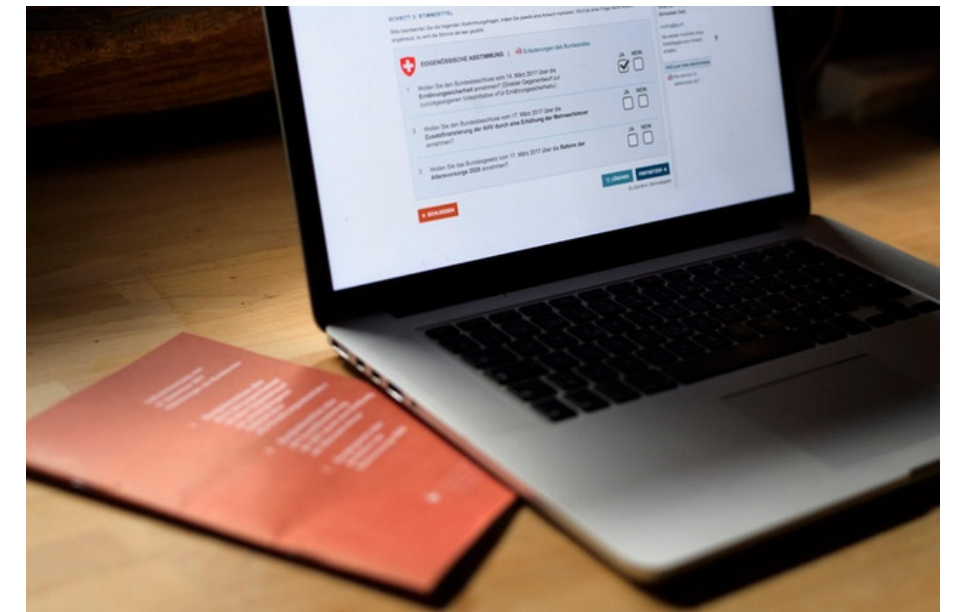
to find the matching randomization $r' = (m - m')h^{-1} + r \bmod q$. As a consequence, the binding property of the commitment scheme is completely broken in that case.

SWI swissinfo.ch

ONLINE DEMOCRACY

Swiss Post's e-voting system pulled for May votes

MAR 29, 2019 - 13:46



Swiss Post's e-voting system had been in use in four cantons: Basel City, Fribourg, Neuchâtel and Thurgau

(Keystone)

The e-voting system operated by Swiss Post will not be available for nationwide votes on May 19. This is the consequence of “critical errors” found during a public intrusion test, the Federal Chancellery and Swiss Post announced on Friday.

The Federal Chancellery [said in a statement](#) it would review the licensing and certification procedures for e-voting systems. It added that it had no indication that these flaws had resulted in votes being manipulated in previous ballots.

Swiss Post's e-voting system had been in use in four cantons: Basel City, Fribourg, Neuchâtel and Thurgau.

Will ElectionGuard be better?

Talented researchers from Microsoft Research, others (including Wallach)

Multiple current implementations (open source, on GitHub today)

Your project 1 (going online tomorrow!) is based on ElectionGuard

The ElectionGuard model: plugins for existing voting machines

Targeted outreach to election equipment vendors

Still requires in-precinct voting

Sophisticated cryptography doesn't solve problems with internet voting