Stolen elections, naked votes

Eradicating auditor nightmares, extinguishing coercer delight

Ben Smyth

December 13, 2024

"Stop the steal," rallied President Donald Trump, moments before the Capitol was stormed. "This is not just an attack on me and my campaign. This is an attack against our country. This is about the integrity of our democracy and the security of our nation." The House is in agreement. "Serious questions have arisen about the accuracy and reliability of new electronic voting machines," raged Hillary Clinton, "including concerns that they can be susceptible to fraud and computer hacking." Academics have long taken this position. Yet 'merica, the so called leader of the free world, is awash with voting technology that was never fit for purpose.

"The whole election is...rigged," cried Donald Trump, four years earlier. "The whole thing is one big fix. One big ugly lie." Promptly rebranding the election "beautiful," as he clamoured for victory. Rival Hillary blaming her defeat on Vladimir Putin and his Russian hackers. The truth:

Our election systems are vulnerable to mass fraud.

Dominion surely cannot win their legal battle. Shockingly, a patch has existed for aeons: Administrators can prove that votes expressed in recorded ballots match election results. We can take an election's digital transcript and determine whether the right people take office—we have the means to deliver fair elections, we implore democracy builders to adopt them.

Disclaimer. Although I personally believe no U.S. election has been electronically thrown, I concede I'm simply not devious enough to pull-off such a heist.

We must demand free and fair elections to preclude demonstrations of evil.

Roadmap. We formulate standards for voters voting freely and for detecting malicious behaviour—provide means to certify elections free and fair, enable expression of free-will, eradicate cheats (Part I). Evaluate and certify the world's state-of-the-art election systems (Part II). Dig into further aspects of security

that must be considered, including authentication, the fight against evil corp, and strong notions of privacy (Part III). Present the world's most secure election systems (Part IV). Gape, bemusedly, at voting technology that was never fit for purpose, question whether rotten tech has any value, and make the case for smartphone voting (Part V). Delve into the mysteries that political parties, politicians, and their electioneers exploit to maximise their advantage (Part VI), and look to remove foul play (Part VII). Finally, we speculate on the extinction of democracy and state our mission to save it (Part VIII).

Grand challenge; call for collaboration: We must eradicate rotten voting technology, equip democracy builders with certifiably free and fair election systems. I invite you to contribute. First, we must reach consensus on the suitability of standards for certifying elections free and fair. Second, we must gather the world's state-of-the-art election systems, trial them, report our findings, make recommendations for future use. Finally, we must work with voting equipment providers to deliver certifiably free and fair elections.

Want to contribute? Get in touch!

I Foundation for secret, verifiable elections 5

Elections: Look and feel 5

Security: Exceptionally brief, single paragraph primer 6

Fair elections: Verifiability 6

Balancing fairness and freedom 8

Free elections: Ballot secrecy 8

Secret verifiable elections for prime time 9

II Secret verifiable voting systems 10

Helios: From homomorphic combinations 10

Helios Mixnet: Throwing in a shuffle 13

III Further security aspects 15

Mind the gap: We're at the mercy of machines 15

Authentication: Trent, unforgeability, and audits 16

Stronger privacy notions: Asshole protection 18

Lost, destroyed, lied about, stolen, ... keys 19

Sacrifice freedom? Hold politicians accountable? 19

IV Auditor's dream, coercer's nightmare 20

Design heuristics 20

Athena: Coercion-free, verifiable voting 21

That's not what I meant: Formal definitions 25

Denial of Service is inevitable: Is coercion resistance viable? 26

Scale, security, and robustness: Decentralisation 27

V Rotten tech, broken machines, smartphone voting 27

Bemused gape at rotten voting technology **27**

Irrecoverably broken voting machines **27**

The case for smartphone voting **28**

VI Life is unfair 28

Defective, unsafe at any speed: Plurality voting **28**

Some voters are more equal: Ranked voting **29**

Arrow's impossibility theorem **30**

VII Life could be fairer: We're being screwed 35

Range voting: Hot or Not? **35**

Democracy is the worst, except for other forms **35**

Gerrymandering **36**

Popular vote; the electorate's choice **36**

President vs. CEO; delegate representation **36**

Democracy, the other form: Universal suffrage **36**

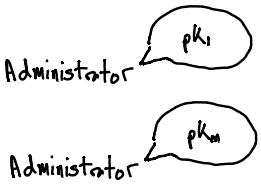
Secret verifiable voting illegal in UK! **37**

VIII Mission 37

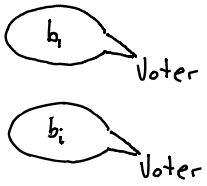
Part I—Foundation for secret, verifiable elections

Elections: Look and feel

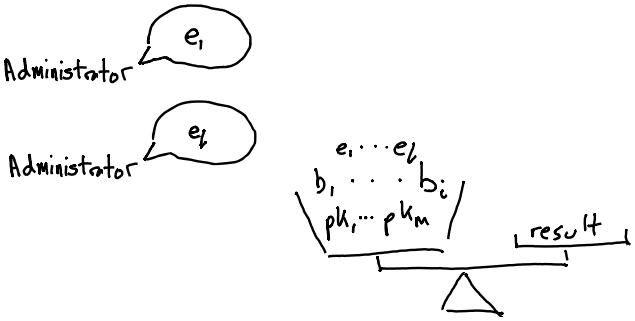
Elections are commonly managed by some centralised authority. For electronic elections, public key cryptography can be used to encrypt votes, with administrators necessarily engaging in some initialisation or setup procedure. Those same administrators will later decrypt votes in some way that reveals the election result whilst keeping each individual voter's vote secret. The administrators should also provide evidence proving that voters' votes match the election result. We consider the most efficient formalisation, whereby each administrator (respectively, each voter) runs the same algorithm, with different bits chosen uniformly at random during execution: Administrators announce public key shares $pk_1, \ldots, pk_m$



voters construct ballots $b_1, \ldots, b_i$ encrypting their votes



and administrators produce evidence $e_1, \ldots, e_\ell$ mathematically proving that election results match votes expressed in those ballots



We demand multiple administrators (each with a public key share), since a lone administrator could otherwise decrypt individual votes.

Security: Exceptionally brief, single-para. primer

Formal mathematical notions of security, such as fairness and freedom, guarantee that certain properties are upheld when voters, administrators, and other stakeholders fulfil certain duties. We can settle for probabilistic, computational guarantees, rather than absolute, information-theoretic guarantees, because we needn't be concerned with attacks that cannot be executed in practice (i.e., attacks in non-polynomial time or with negligible probability of success). Such guarantees can be established by considering voting system executions in which an adversary attempts to subvert the system, whilst honest participants fulfil their roles.

Fair elections: Verifiability

Elections are big business; history teaches us that malice is rife: To detect misbehaviour, voting systems can produce evidence (demonstrating that a result corresponds to votes expressed in collected ballots) of correct operation, which can be verified before any result announced—fairness can be ensured. We challenge an adversary to orchestrate a system execution in which verification is unable to detect a result that doesn't correspond to votes expressed in collected ballots. In the absence of such an adversary, we infer the voting system always announces valid results.

Definition. *A voting system guarantees* Soundness*, if election results are supported by evidence proving results match votes expressed in ballots: For all public key shares, ballots, evidence, and results (computed by an adversary in probabilistic polynomial time), such that evidence supports results, we have votes expressed in ballots correspond to results (with overwhelming probability).*

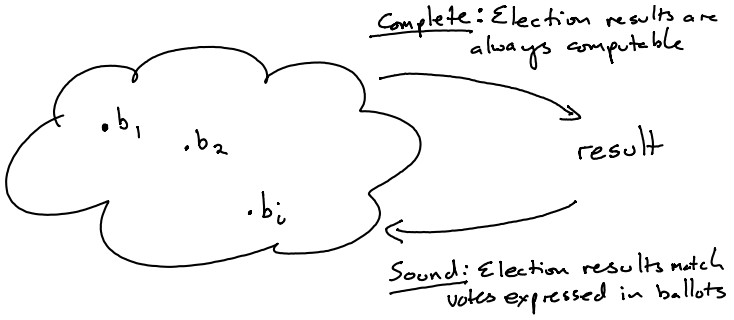
Soundness precludes announcement of invalid results. Voting systems must also preclude system executions in which an adversary forces an error state—from which no result can be announced—to avoid suppression of "unfavourable" results. Voting systems must be capable of producing valid evidence. We challenge an adversary to orchestrate a system execution from which no result can be announced.

Definition. *A voting system guarantees* Completeness*, if election results and supporting evidence are always computable: For administrator generated public key shares and adversary (probabilistic polynomial time) computed ballots, election results and supporting evidence are always computable (with overwhelming probability).*

Taken together, Soundness and Completeness provide universal verifiability, as illustrated in Figure 1.

Definition. *A voting system guarantees* Universal-Verifiability*, if* Soundness *and* Completeness *are guaranteed.*

Figure 1 Universal verifiability demands Soundness and Completeness

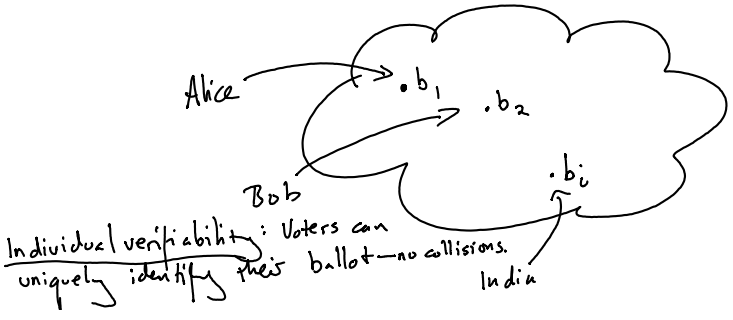


Soundness guarantees detection of results that do not correspond to votes expressed in collected ballots. Voting systems must also guarantee that voters can detect omission of their ballot. Since ballots should be collected and recorded on a (public) bulletin board, it suffices for voters to detect omission of their ballot from the board. Merely observing a ballot identical to the ballot they constructed is insufficient, since an identical ballot may have been constructed by another voter. By comparison, it suffices for voters to be able to uniquely identify their ballot, i.e., ballots do not collide, as illustrated in Figure 2.

Definition (Individual-Verifiability)**.** *A voting system guarantees* Individual-Verifiability*, if independently computed ballots are distinct (with overwhelming probability, for all public key shares computed by an adversary in probabilistic polynomial time), or an error is raised.*

To demonstrate application of our definitions, we introduce a voting system wherein ballots are computed by sampling a nonce (from $\mathbb{Z}_{2^\kappa}$) and coupling that nonce with the voter's vote, election results are computed to match votes expressed in ballots. Individual verifiability is satisfied because nonces (hence, ballots) collide with negligible probability. Universal verifiability is satisfied too, because, even without supporting evidence, results match votes expressed in ballots (Soundness) and such results are always computable (Completeness).

Figure 2 Individual verifiability demands ballots don't collide



Balancing fairness and freedom

Daily decisions are tiresome, delegation ubiquitous; elections are the bedrock of society. Small groups, boardrooms, and the like, can run their own (decentralised) elections: Participants jointly compute results from individual desires. Vocally expressing preferences, *yea* or *nay*. Physically raising hands. Or privately marking ballots, affording freedom.

Freedom to vote one's conscience, morals, and will, without fear of others' preference, pleasure, nor patronage, is a recent innovation: With the notable exception of Swiss voters—who are still entitled to vote by raising their bayonets in public—the nineteenth century closed with the introduction of voting in private, following the realisation that votes belonged to some opulent man (assuming you had one, universal suffrage *still* hasn't been delivered), when votes are cast in public. Whilst private voting is free, public voting is fair, everyone present can verify the election result themselves. Unfortunately, the push for freedom seemingly disregards fairness, voting systems are routinely broken, undue influence expressed; simultaneously achieving fair and free elections is proving elusive.

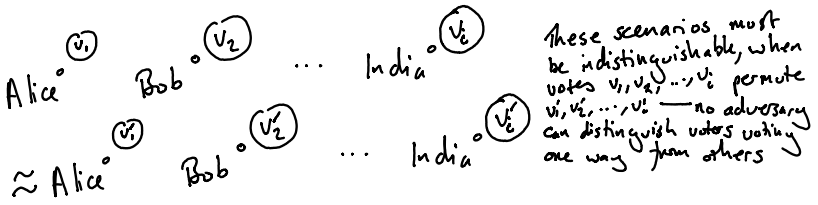
Decentralised elections do not scale. Administered, centralised procedures do, whilst risking freedom: Ultimately, results must be computed, and curious administrators may compute results for individual ballots, violating freedom. Such circumvention can be eradicated when computation is distributed amongst all participants, as per decentralised elections, and can be limited (at a cost to scalability) by distributing computation between several administrators. For fairness, we've seen we can do better: Rather than blindly trusting administrators, their actions can be verified, malicious behaviour detected.

Free elections: Ballot secrecy

Freedom cannot be absolute: Elections have winners; dictators know when they've been "wronged"—elections necessarily leak information. Freedom cannot be formalised by challenging an adversary to orchestrate a system execution in which voters' votes are revealed. A more subtle formalisation is necessary. We challenge an adversary to orchestrate a distinction between a system execution in which voters cast some votes, from another instance in which the voters cast a permutation of those votes. Intuitively, if those scenarios are indistinguishable, then no adversary can distinguish between a voter voting one way, from the voter voting another. A seemingly sufficient notion of freedom.

For centralised election procedures, administrators can compute results for individual ballots, violating freedom. Accordingly, we must assume at least one administrator is trusted to uphold freedom. We also assume voters are willing and able to act in a manner that ensures their freedom: We assume devices used by voters to construct ballots are functioning correctly. That's a tall order—we're writing-off the freedom of those unwilling or unable. We'll relax the assumption later. With those assumptions in place, our formalisation pits

Figure 3 Voting one way is indistinguishabile from the other



an adversary against a (lone, trusted) administrator that guards their private key share, whilst announcing their public key share and evidence. That evidence being computed with respect to a bulletin board (constructed by the adversary) that may include ballots of voters willing and able to seek privacy. Our formalisation challenges the adversary to violate the freedom of such a voter, as illustrated in Figure 3.

Definition (Ballot secrecy)**.** *A voting system guarantees* Ballot Secrecy*, if an adversary corrupting all but one administrator cannot determine the value of bit β from a bulletin board they compute using voters' ballots—each ballot expressing vote v_β, computed upon the adversary's demand for a ballot expressing vote v_0 or v_1—along with the non-corrupt administrator's public key share and evidence, wherein for bulletin-board ballots computed on demand with inputs $(v_{1,0}, v_{1,1}), \ldots, (v_{n,0}, v_{n,1})$, we have $v_{1,0}, \ldots, v_{n,0}$ is a permutation of $v_{1,1}, \ldots, v_{n,1}$.*

The definition captures a scenario in which voters cast some votes, rather than some permutation of those votes, and challenges the adversary to distinguish which scenario they've observed. In the absence of distinction, no adversary can distinguish between a voter voting one way, from the voter voting another.

Ballot secrecy proofs are long, arduous, time-consuming: It took me six and half pages to prove that a toy voting system satisfies an early version of the above definition. Rather than burden ourselves with tedium, we prove non-malleable ballots—i.e., ballots which cannot be transformed into other ballots (such that a relation exists between votes expressed in original ballots and votes expressed in transformed ballots)—suffice for ballot secrecy, allowing straightforward proofs.

Secret verifiable elections for prime time

Voting machines have bad press. Rightly so. The US is awash with technology that was never fit for purpose. Yet, voting machines aren't inherently bad: Their algorithms can be upgraded. Voting machines can be made secure. We present one such upgrade, based upon our earlier sketch:

A voting system comprises at least two administrators and a setup procedure, voters and a ballot construction procedure, and an evidence production procedure (used by administrators), along with a procedure to evaluate (weigh-

up) the evidence against an election result, wherein the election result may have been extracted from the evidence with some procedure.

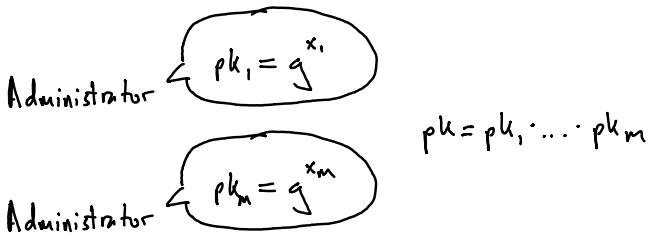
The at least two administrators execute the setup procedure, optionally on input including a security parameter, to each produce a public key share, wherein those key shares can be combined to derive a public key, which may simply be a vector over the key shares. To generate ballots, voters each execute the ballot construction procedure, on input including their vote and optionally the public key, the security parameter, or both. Ballots are collected and, after some deadline, administrators each execute the evidence production procedure on inputs including those collected ballots. Evidence optionally being used by some procedure to extract an election result. And, finally, a procedure being executed on input of the evidence and a purported election result. That procedure indicating success if votes expressed in ballots correspond to results, and failure otherwise.

We have means to deliver free and fair elections: Voting machines can be upgraded to include the algorithms hereon in. These algorithms aren't new, they started emerging in the eighties. A young David Chaum toiling away—presumably dressed in one of his trademark Hawaiian shirts—developing cryptographic fundamentals, long before they'd go mainstream. Most notably known for inventing cryptocurrencies, widely rumoured to be Satoshi Nakamoto, David also invented the first cryptographic-based voting system. Ballot secrecy was the era's focus, until cryptographers realised just how powerful our adversaries could be. Verifiability took centre stage. Balancing privacy and verifiability became the drive.

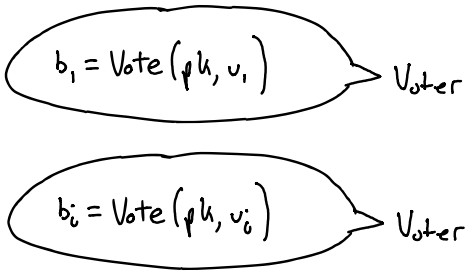
Part II—Secret verifiable voting systems

Helios: From homomorphic combinations

Helios (Adida, Marneffe, Pereira, and Quisquater, 2009) is built from asymmetric encryption scheme El Gamal, with a sprinkling of proof systems used to demonstrate correct operation (construction of public key(share)s, ciphertexts, and partial decryptions), using some administrator-agreed parameters, including a group generator g (of the multiplicative group $\mathbb{Z}_p^*$ of order q, where p and q are large primes such that $q \mid p - 1$). Administrators each announce their public key shares (computed using El Gamal's key generation procedure):



keeping private key shares $x_1, \ldots, x_m$ (chosen uniformly at random) to themselves. (Public key shares are coupled with proofs of correct construction, omitted for brevity.) Public key shares are combined to derive a public key pk, and the ratified candidate list is announced. Each voter uses the public key to encrypt their vote v (selected from the candidate list):



wherein $\texttt{Vote}(pk, v)$ indicates constructing a ciphertext for all but the last candidate, each ciphertext encrypts 0, except for the vth ciphertext which encrypts 1. Ciphertexts are constructed using El Gamal, which is homomorphic, meaning two ciphertexts can be homomorphically combined to derive a third ciphertext encrypting the sum of the original plaintexts. E.g., given a ciphertext encrypting seven and another encrypting one, a ciphertext encrypting eight can be derived by homomorphic combination.

Tallying exploits the homomorphic property to compute election results: Collected ballots are homomorphically combined to derive the sum of votes for each candidate. For example, given Alice's ballot, comprising ciphertexts $c_1, \ldots, c_{nc-1}$, Bob's $c'_1, \ldots, c'_{nc-1}$, and Charlie's $c''_1, \ldots, c''_{nc-1}$ (wherein nc is the number of candidates), we can homomorphically combine Alice's first ciphertext (c_1) with Bob's (c'_1) and then the result with Charlie's (c''_1), giving us a ciphertext encrypting the sum of votes for the first candidate, and similarly for all but the last candidate. (Combining homomorphic ciphertexts to tally votes originates from Cohen and Fischer, 1985, an extension from referendums to elections with arbitrarily many candidates is due to Hirt, 2001, from which Helios is derived by generalisation and optimisation.) Decrypting reveals the number of votes for each candidate, the number of votes for the last being the number of ballots less the number of votes for other candidates. The election result comprises the number of votes for each candidate.

To enable detection of malicious behaviour, ciphertexts are proven to encrypt bits, and ballots prove the homomorphic combination of ciphertexts encrypts a bit too, enabling detection of voters that attempt to unduly influence results. (For instance, by encrypting integers greater than one—casting multiple votes for their candidate.) Ballots of cheating voters, cast with invalid proofs, are discarded. Malicious administrator behaviour is also detectable: Public key shares along with parameters including the group generator are proven to be well-formed and decryption is proven to be well-computed, enabling detection of administrators attempting to subvert elections. (The latter is proven by

Crypto 1 El Gamal, homomorphic combinations, and partial decryptions.

For the mathematically inclined, we delve into the cryptographic details of El Gamal ciphertexts, homomorphic combinations, and partial decryptions, in the context of our earlier example, wherein the first ciphertext of Alice, Bob, and Charlie, respectively, are defined as:

$$g^r,\ pk^r \cdot g^a \qquad g^s,\ pk^s \cdot g^b \qquad g^t,\ pk^t \cdot g^c$$

where a, b, and c are 0 or 1, depending on whether Alice, Bob, and Charlie, respectively, voted for the first candidate, and r, s, and t are nonces. The homomorphic combination of those ciphertexts is:

$$g^r \cdot g^s \cdot g^t,\ pk^r \cdot g^a \cdot pk^s \cdot g^b \cdot pk^t \cdot g^c \equiv g^{r+s+t},\ pk^{r+s+t} \cdot g^{a+b+c}$$

namely, an encrypted sum of votes for the first candidate. To decrypt, administrators compute partial decryptions $(g^{r+s+t})^{-x_1}$, ..., $(g^{r+s+t})^{-x_m}$, which are combined (by multiplication) to derive:

$$(g^{r+s+t})^{-x_1} \cdot \ldots \cdot (g^{r+s+t})^{-x_m} \equiv (pk^{r+s+t})^{-1}$$

From which the sum of votes for the first candidate can revealed:

$$\log_g (pk^{r+s+t})^{-1} \cdot pk^{r+s+t} \cdot g^{a+b+c} \equiv \log_g g^{a+b+c}$$

Although solving discrete logarithms is inefficient in general, in this instance, we can compute $g^0, g^1, \ldots, g^{nv}$ (wherein nv is the number of voters) in advance, hence, solving the above simply requires a table look-up.

announcing evidence comprising partial decryptions of homomorphic combinations, along with proofs that those partial decryptions are well-computed.)

We formalise the above description by insisting proof systems are zero-knowledge and satisfy simulation sound extractability (which can be achieved (in the random oracle model) when proof systems are built by applying the Fiat-Shamir transformation to sigma protocols satisfying special soundness and honest verifier zero-knowledge, as shown by Bernhard et al., 2012), and mathematically prove that we deliver a secret, verifiable voting system:

Theorem. *We have a verifiable voting system,* Universal-Verifiability *and* Individual-Verifiability *are satisfied.*

Our verifiability proof follows from good design: Public keys are proven to be well-formed, allowing ciphertexts to be well-constructed. Moreover, since ciphertexts are proven to represent voter's votes, it follows that homomorphic combinations encrypt election results, hence, proving correct decrypt suffices to ensure votes expressed in ballots match election results. We sketch our proof below. (Smyth et al., 2021, present a more laborious, detailed proof.)

Corollary. Ballot Secrecy *is satisfied too.*

Our secrecy proof follows from non-malleable ballots coinciding with ballot secrecy coincide.

Proof sketch of (verifiability) theorem. Ballots comprise El Gamal ciphertexts, each including a group element chosen uniformly at random (generated by raising the group generator by some nonce, after verifying it is indeed a group generator, Chang-Fong and Essex, 2016), hence, ballots do not collide, Individual-Verifiability is satisfied. Tallying discards ill-formed ballots, ensuring the remaining ballots comprise well-formed ciphertexts (regardless of whether they were generated by a malicious adversary), which can be homomorphically combined and partially decrypted, since keys are correctly computed, thereby enabling election results and supporting evidence to be computed, Completeness is satisfied too. Proving Soundness is a little more involved.

Ill-formed ballots are discarded, the remaining ballots encrypt voters' votes: Those remaining ballots must have been computed as $\texttt{Vote}(pk, v_1), \texttt{Vote}(pk, v_2), \ldots, \texttt{Vote}(pk, v_i)$, wherein $\texttt{Vote}(pk, v)$ indicates that each ballot comprises $nc-1$ ciphertexts (wherein nc is the number of candidates), the vth of which encrypts 1, the others 0. It follows that homomorphically combining ciphertexts allows us to compute the encrypted sum of votes for each candidate. (El Gamal is absolutely, perfectly homomorphic—rather than probabilistically, computationally homomorphic—which is necessary to ensure we derive an encrypted sum of votes for each candidate, given that Soundness quantifies over adversarial-constructed public keys, which may not have been constructed using coins chosen uniformly at random.) The election result comprises the number of votes for each candidate, with partial decryptions (of homomorphic combinations) serving as evidence. (El Gamal is perfectly correct, which is necessary to ensure decryption recovers the number of votes for each candidate, given that the adversary-constructed public key may not have been computed using coins chosen uniformly at random.) Votes expressed in ballots correspond to results; Soundness, hence, Universal-Verifiability, is satisfied. □

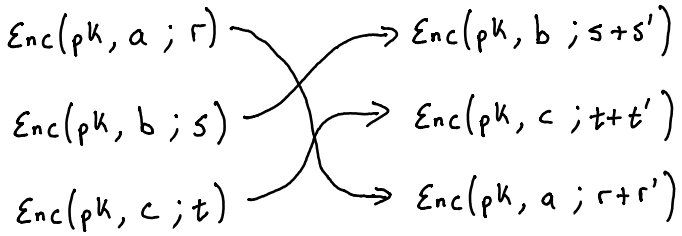
Proof of (secrecy) corollary. Smyth and Hanatani (2019) prove that coupling (malleable) ciphertexts with proofs of correct construction ensures Non-Mealleable Ballots, which suffices for Ballot Secrecy (when proof systems are suitably built, Bernhard et al., 2012). By contrast, tallying meaningfully related ballots violates freedom (Cortier and Smyth, 2013), when election results comprise the number of votes for each candidate (a functional requirement of many nations, that comes at a cost to freedom; revealing the number of votes for each candidate leaks more information than, for instance, revealing only the winning candidate). □

Helios Mixnet: Throwing in a shuffle

Helios Mixnet (Adida, 2008; Bulens, Giry, and Pereira, 2011) is rather similar to Helios: Both schemes are built from El Gamal and a sprinkling of proof systems,

using some administrator-agreed parameters, from which administrators compute their public key shares, keeping private key shares to themselves. (Public key shares are again coupled with proofs of correct construction.) Public key shares are combined to derive a public key, and the ratified candidate list is announced. Thereafter, the voting systems diverge. In Helios Mixnet, each voter (directly) encrypts their vote for the vth candidate, casting the resulting ciphertext coupled with a proof that v is amongst the candidates. (By comparison, Helios encrypts votes indirectly, as a series of ciphertexts.)

Tallying again exploits the homomorphic property to compute election results, but rather differently—collected ballots are processed by a mixnet, which inputs ciphertexts encrypting voters' votes, and outputs distinct ciphertexts encrypting the same votes. For instance, on input of ciphertexts from Alice, Bob, and Charlie, those ciphertexts could be shuffled into a new order, and each homomorphically combined with a ciphertext encrypting the identity element, before outputting those homomorphic combinations. Those outputs are distinct from the originals, yet encrypt the same votes:



where r, s, and t are nonces used to encrypt votes a, b, and c of Alice, Bob, and Charlie, respectively, and nonces r', s', and t' are each used to encrypt identity element 0, the resulting ciphertexts being homomorphically combined with voters' ciphertexts, before outputting shuffled homomorphic combinations. Mixed ciphertexts are decrypted to reveal voters' votes. (Mixnets can operate independently of administrators, since private key shares aren't required.)

To enable detection of malicious behaviour, public key shares (and parameters) are proven to be well-formed, mixing is proven to output ciphertexts encrypting voters' votes, and decryptions are proven to be well-computed, enabling detection of administrators, or mixnet operators, attempting to subvert elections. (Decryption is proven by announcing evidence comprising partial decryptions of mixed ciphertexts, along with proofs that those partial decryptions are well-computed.) Additionally, voters' ciphertexts are proven to encrypt valid candidate choices, to ensure Completeness.

We formalise the above description by insisting proof systems are zero-knowledge and satisfy simulation sound extractability, and mathematically prove that we deliver a secret, verifiable voting system:

Theorem. *Helios Mixnet is a secret, verifiable voting system.*

Our proof is similar to those for Helios. (Smyth presents detailed proofs of secrecy, 2021, and verifiability, 2018.)

Part III—Further security aspects

Mind the gap: We're at the mercy of machines

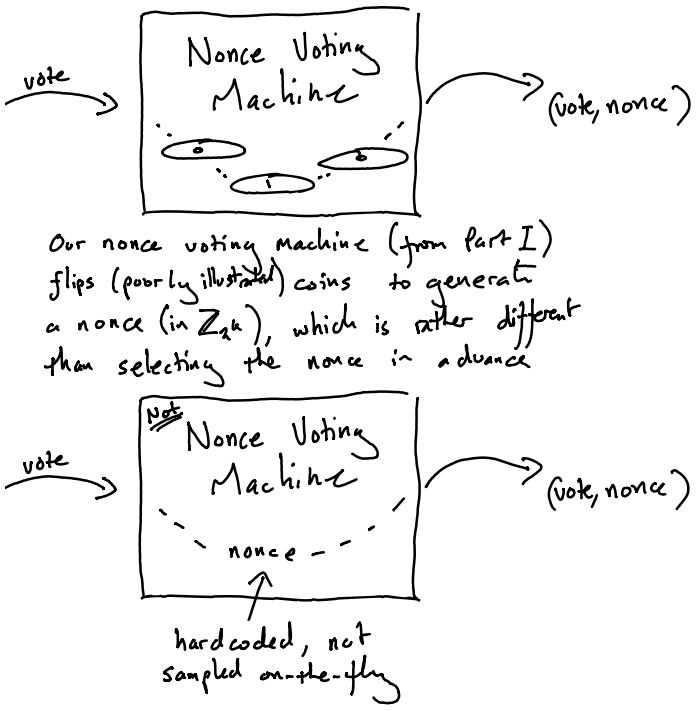
Publicly recording ballots allows each voter to check the presence of their ballot (individual verifiability) and coupling results with evidence allows anyone to check whether an election result represents votes expressed in collected ballots (universal verifiability). Taken together, these facets enable voters to check whether tallying includes the vote expressed by their ballot. Yet, digital ballots—one hopes—are constructed by cryptographic means; we mere mortals, even the most studious, can't compute ballots. Voters are at the mercy of machines, which mightn't even compute ballots, let alone correctly compute ballots expressing voters' votes. Individual- and universal-verifiability only suffice for verifiable voting systems when voters can compute their own ballots, which is atypical of digital ballots.

In an attempt to bridge this gap, some systems define ways for a voter to check whether machines produce a ballot expressing their vote (cast-as-intended). For instance, a voter may input a vote to some machine and receive a purported ciphertext, encrypting that vote. Rather than blindly trusting a machine, the voter can demand evidence, which a trusted system or party can check to determine whether the received value represents such a ciphertext. (Checks are typically cryptographic, beyond comprehension of us mere mortals.) Relying on a trusted system or party may seem disingenuous: Trust is contrary to verifiability. However, multiple systems or parties can perform checks to limit trust. (Albeit, scalability takes a hit.) Evidence proves whether a ciphertext encrypts a particular vote and such ciphertexts shouldn't be cast to avoid compromising privacy. So, voters repeat the process until they're convinced a machine functions correctly, then they cast the next value received from the machine (without demanding evidence), providing probabilistic assurance that the value is a ciphertext encrypting their vote.

Digital ballot construction typically mandates sampling bits, generating nonces. If a machine abandons the prescribed sampling procedure, computation delivers something resembling a ballot, rather than a correctly computed ballot: A ballot constructed in disregard for the prescribed procedure is not correctly computed. It may resemble a ballot. A ballot may even be computable in that way. However, ignoring the construction mandate means the result cannot a priori be considered a correctly computed ballot. Herein lies the rub—voters cannot determine whether machines compute ballots or things resembling ballots. Cast-as-intended and individual verifiability don't compose to enable determination of whether collected ballots express voters' votes, since voters cannot determine whether machines even compute ballots correctly, as illustrated in Figure 4.

Consequently, voting systems believed to be verifiable, might not be. Clash

Figure 4 Voters cannot determine whether machines compute ballots correctly



attacks are one example of insecurities that may arise. For example, rather than sampling bits correctly, a machine may sample bits in advance and use those bits when encrypting votes. Two voters inputting the same vote (having amassed equal volumes of evidence) will receive the same ciphertext. They'll rightly be assured the ciphertext encrypts their vote and rightly detect the ciphertext's presence, but neither can determine whether the ciphertext is theirs, since both voters received identical values. There's a mismatch between assumptions underpinning cast-as-intended and individual verifiability: Cast-as-intended assures a ballot expresses a vote, not whether the ballot is correctly computed, whilst individual verifiability assures correctly computed ballots are collected. Cast-as-intended and individual verifiability don't compose in the expected way. Authenticating ballots side-steps the problem of clash attacks.

Authentication: Trent, unforgeability, and audits

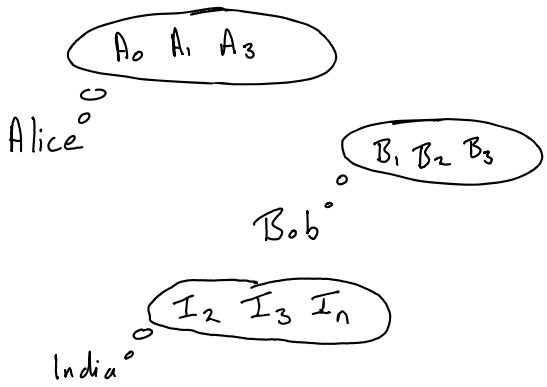
One citizen, one vote: Elections must enforce eligibility requirements. Otherwise, mass fraud. Especially for systems connected to the Internet. One way to achieve authentication is to physically verify voter credentials in polling stations. Another is to rely on a trusted third party (sometimes known as Trent), tallying only ballots they authenticate. For example, the Helios system sup-

ports authentication via Facebook, Google, and Twitter, using OAuth. Other voting systems use cryptography to ensure only voters can construct authorised ballots.

Universal verifiability ensures voting systems announce election results corresponding to votes expressed in collected ballots. For the first two classes, universal verifiability assumes collected ballots are authorised ballots. That's the best such systems can do, ballots are not themselves authorised nor unauthorised, they're merely branded as such, either by virtue of being cast in a polling station (hopefully after voter credentials are physically verified) or by accepting the claims of a trusted third party (hoping their claims are legit). Whether collected ballots are indeed authorised is never known, and cannot be checked. We aren't assured that only voters vote and at most once. We can do better with the third class of voting systems: Voters can use cryptography to authenticate ballots.

Voters can be issued with credentials, computed in a distributed manner, by at least two parties (to avoid a lone individual issuing credentials to non-voters or issuing voters with multiple credentials), optionally including the voter performing part of that distributed computation. Enfranchisement teaches us that voters should be free to select their registrar(s), and society should agree upon a threshold. Minimally, the voter and a single registrar, perhaps increasing to multiple registrars (to avoid abuse by cohorts of registrars), and possibly omitting computation by the voter themselves.

Registrars and voters can engage in a registration procedure, optionally on input including the public key or public key shares, to each produce a public and private credential share, wherein those credential shares can be combined to derive, respectively, a public credential and private credential, which may simply be a vector over the (respective) credential shares. For instance, Alice may generate part of her own credential and request two further parts from the first and third registrar, Bob might rely on the first three registrars, and India may request credential parts from the second, third and nth registrar. (Voter credentials needn't all consist of the same number of parts, that's just a possibility.) Voter public credentials must be ratified and published.



To generate ballots, voters each execute the ballot construction procedure, on

input including their vote and optionally the public key, their credential (public, private, or both), the security parameter, or a combination. Ballots are collected and, after some deadline, administrators each execute the evidence production procedure on inputs including those collected ballots and ratified credentials. That evidence can be verified with respect to a purported election result and ratified credentials.

Credentials allow us to refine our universal verifiability definition as follows, namely, *election results are supported by evidence proving results match votes expressed in* authorised *ballots: For all public key shares, public credential shares, ballots, evidence, and results (computed by an adversary in probabilistic polynomial time), such that evidence supports results, we have votes expressed in authorised ballots (constructed using a private credential corresponding to a ratified public credential) correspond to results (with overwhelming probability),* which complements our definition of Unforgeability, namely, *authorised ballots can only be constructed by voters in possession of a private credential corresponding to a ratified public credential.*

Cryptography cannot ensure credentials are only ever issued to voters—groups of registrars can issue credentials to non-voters or issue multiple credentials to an individual. Audits can be used to establish whether (or roughly how many) credentials have been issued in bad faith. For instance, an auditor could ask voters to prove whether they posses the private credential corresponding to the ratified public credential they are purported to control. (That process isn't perfect, voters could refuse to engage or otherwise disrupt auditing.)

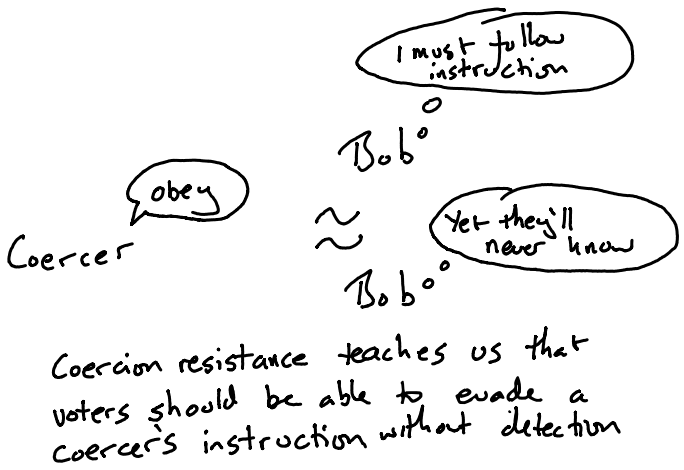
Stronger privacy notions: Asshole protection

Freedom is fundamental. Yet, ballot secrecy writes-off the freedom of voters unwilling or unable to protect their fundamental right. Voters may be at the mercy of abusers, voter buyers, coercers, dictators, those that must not be 'wronged.' Ballot secrecy only suffices for freedom, when those powers are kept at bay—surely an impossibility. We must do better; enfranchise all voters, not just the fearless, those under threat too.

Coercion-resistance takes the position that voters cannot act in a manner that ensures their freedom, voters are assumed to be under a coercer's instruction. Freedom is formalised by challenging the coercer to distinguish between a voter following instruction and evading coercion to vote freely. If those scenarios are indistinguishable, then no coercer can distinguish between a voter doing as they are told, from a voter doing as they please, expressing their free-will, as illustrated in Figure 5. (Evasion may require guile or trickery; psychologists should examine coercion-resistance's practicality.)

None of the voting systems we've seen (so far) achieve coercion-resistance. It's actually impossible in the context of presented systems: A coercer can simply instruct a voter to cast a pre-constructed ballot. We need a new perspective, the introduction of registrars suffices: For coercion resistance, ballots must include private input(s) that cannot be simulated by voters. Private credentials

Figure 5 Voters should be able to evade coercers



suffice, assuming they can be distributed in private.

As we'll see in Part IV, one way to achieve coercion resistance is to ensure voters cannot prove whether they even voted, let alone who they voted for.

Lost, destroyed, lied about, stolen, ... keys

The government, the opposition, and some minority party could be the administrators, the sole custodians of an election. They'll likely loose, destroy, or lie-about keys. That's a bit of a wrinkle: Completeness only guarantees computability of election results using private keys. Rip out keys—election results can't be guaranteed. A voting system must surely be Robust, namely, *election results and supporting evidence are computable by j (out of m) cooperating administrators.* Yet, that's a spanner in the works.

Sacrifice freedom? Hold politicians accountable?

Allowing election results to be computable by a group of administrators. Well, that means, groups of administrators can reveal the vote of each (and every) individual voter—stripping their freedom. Yet, not doing so introduces the possibility that a politician, acting alone, could deny our right to election results, be it by incompetence or malice. Now, should we sacrifice freedom? Or hold politicians accountable? (Disqualify an opponent with no key; they surely cannot be trusted with state secrets.) That's not for us to decide. Our definition of Soundness can be tweaked to capture Robustness, and our definition of Ballot Secrecy can be weakened to tolerate violations of secrecy by groups of administrators in cohorts. Those definitions can be satisfied by voting systems that

switch-out El Gamal for the (j, m)-threshold variant. Whether such systems are desirable is for society to decide.

Part IV—Auditor's dream, coercer's nightmare

Design heuristics

We have administrator public keys and voter credentials, from which voters construct ballots expressing their votes. An election result is extracted and administrators produce evidence proving that the result matches those votes. Our first design heuristic follows intuitively:

1. Evidence must map votes expressed in authorised ballots to the election result. (Except for any early votes.)

(Voting systems have traditionally permitted each voter to cast at most one vote. More recent systems permit multiple votes, with only a voter's last having influence: Voters can change their vote, providing flexibility and allowing voters to 'test drive' voting systems, Meyer and Smyth, 2019.) Our second heuristic follows from the need to identify authorised ballots, and the next two follow from the need for secrecy:

2. Voters must authenticate ballots.
3. Ballots must ensure votes cannot be revealed.
4. Evidence must not reveal any (meaningful partial) mapping between votes expressed in ballots and the election result.

Forgoing coercion resistance (in favour of ballot secrecy), the previous design heuristic can be weakened: Evidence must not reveal any (meaningful partial) mapping between votes expressed in *authorised* ballots and the result. But this may allow *simulation attacks* (Juels et al., 2010, §1.1), whereby a coercer instructs a voter to reveal their private credential, uses that private credential to cast a ballot, and determines whether the voter followed instruction by evaluating evidence to determine whether the cast ballot is amongst those mapped to the election result, i.e., whether the cast ballot is authorised. By comparison, coercion resistance gives way to the following design heuristic:

5. Evidence must not reveal authorised ballots.

Similarly, authorised ballots must not be revealed during casting and collection:

6. Ballots must not reveal whether they are authorised.

Since ballots must be authenticated (1) without revealing authorised ballots (5 & 6), the following heuristics emerge:

7. Ballots should be anonymised prior to authentication, and

8. Evidence may reveal authorised anonymised ballots.

Given that ballots should be anonymised and that bulletin boards may contain more than just ballots, the following heuristics are proposed:

9. Ballots should prove correct construction, and

10. Garbage is discarded, i.e., disregard non-ballots.

Revealing early votes after anonymisation can be problematic, for instance, a voter that casts a specific number of ballots can de-anonymise their anonymised ballots, giving way to our final design heuristic:

11. Ballots representing early votes are discarded prior to anonymisation.

For compatibility between (11) and (4), our notion of *meaningful* should exclude early votes, i.e., evidence is permitted to reveal (unauthenticated) ballots representing early votes (that don't contribute to election results).

By combining our design heuristics, we observe that the bulletin board must be filtered to remove garbage (10) and ballots representing early votes (11). After anonymising any remaining ballots, anonymised ballots should be authenticated and any unauthorised anonymous ballots removed (7). Finally, votes expressed in any authenticated anonymous ballots should be mapped to the election result corresponding to those votes (1), as illustrated in Figure 6.

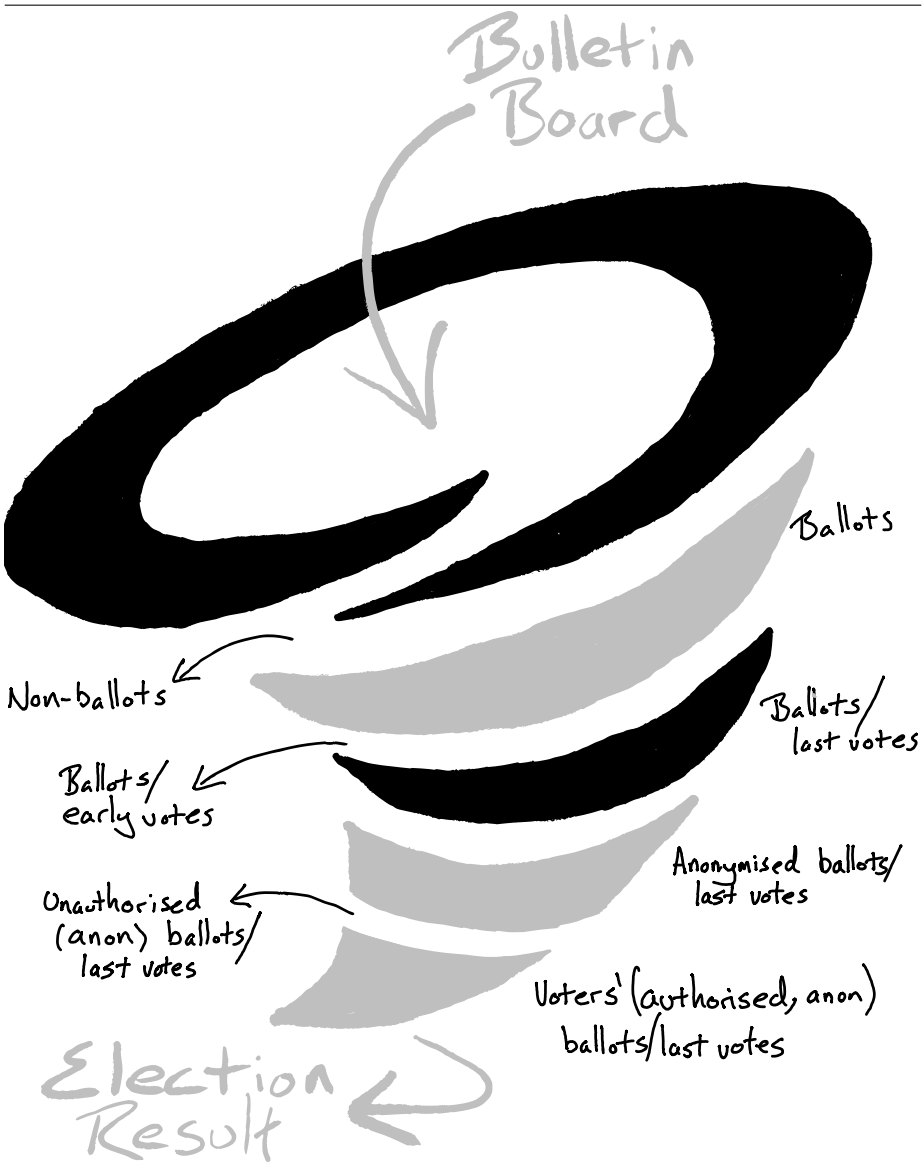
Athena: Coercion-free, verifiable voting

Athena arose from many hours of thought, conducted long after dark, curled-up on cold winter's nights, under the eaves: Our design heuristics make goals clear, I was left pondering how to deliver? Windows were scrawled upon, notes were scribbled, many early attempts failed. I wasn't the first to pursue this quest: Athena is inspired by the voting system due to Juels et al. (2010), and motivated by the need for linear complexity.

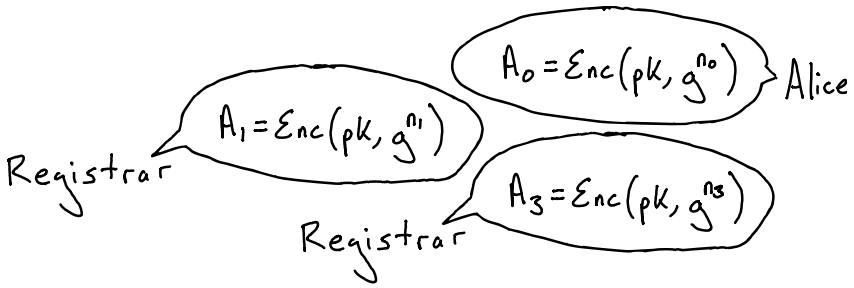
Athena uses a homomorphic asymmetric encryption scheme such as El Gamal, various proof systems to demonstrate correct operation, and a mixnet. As per Helios and Helios Mixnet, administrators agree on parameters, compute public key shares (keeping private shares to themselves), combine shares to derive a public key, and announce the ratified candidate list. (Public key shares are again coupled with proofs of correct construction.) Unlike Helios and Helios Mixnet which rely on a trusted third party to authenticate ballots, Athena uses cryptography.

Every voter is issued with credential pair shares, wherein each private credential share is a nonce and each public credential share is an encryption (under the administrators' public key) of that nonce, the credential pair shares being

Figure 6 A bulletin board should be filtered to discard garbage (non-ballots) and early votes, the remaining ballots should be anonymised and unauthorised ballots discarded, finally, any authenticated (anonymous) ballots—i.e., ballots expressing voters' last votes—should be mapped to the election result



computed in a distributed manner, by at least two parties, optionally including the voter themselves. (Correct construction of public credential shares is proven.) Private and public credential shares are privately transmitted to the voter who homomorphically combines public shares to derive a public credential, namely, an encryption of nonces. E.g.,



The underlying nonces serve as the private credential. (Alternatively, a private and public credential can be computed by a lone registrar, assuming protection against a malicious registrar isn't necessary.) Voter public credentials are ratified and published, private credentials are known only to voters.

Each voter encrypts (under the administrator public key) the negation of their private credential and their vote, and publishes (to the bulletin board) the two resulting ciphertexts, prepended with their public credential and appended with a counter. E.g., Alice casts

$$\mathcal{E}nc(pk, g^{n_0+n_1+n_3}) \quad \mathcal{E}nc(pk, g^{-n_0-n_1-n_3}) \quad \mathcal{E}nc(pk, v) \quad \text{timestamp}$$

(Ballots also prove correct construction.) A voter computes ballots for any later votes similarly (only the last of which will have influence), using an incremented counter.

Voters use private credentials to authenticate ballots, without revealing whether ballots are authorised: No voter can prove a ballot contains their private credential (since no relation can be proved without access to coins used to generation public credential shares); well-formed ballots that encrypt the negation of a voter's private credential is indistinguishable from an ill-formed ballot that encrypts some other value, rather than such a negation—a voter cannot prove whether they cast a well-formed ballot (that will be counted, as opposed to an ill-formed ballot that will not), let alone prove how they voted. Heuristics (2) & (6) are satisfied, as are (3) and (9), because votes are encrypted and correct ciphertext construction proven.

Coercion-resistant voting systems, such as Athena, depend on 'noise': A coercer demanding no votes for a particular candidate, can trivially determine they've been 'wronged,' should that candidate receive any votes. Coercion-resistance depends on large noisy crowds [of voters] providing cover, casting well-formed ballots for an array of candidates, casting ill-formed ballots too. The former hides who 'wronged' the coercer, the latter creates uncertainty over which

ballots will be counted, enabling anonymisation prior to authentication. For Athena, noise includes ill-formed ballots for a variety of public voter credentials, cast by administrators, privacy advocate groups, talliers, voters, etc.

Tallying discards garbage (any bulletin board entry without a valid proof of correct construction is discarded, as is any entry not containing a ratified public credential), achieving (10). The second ciphertext of each remaining ballot is homomorphically combined with itself n-times (for some nonce n). (Evidence includes proof of correct combination.) Administrators partially decrypt the resulting combination to reveal an anonymised credential. Entries that share an anonymised credential are discarded, except for the one with the highest counter, hence, only the last vote associated with each anonymised credential is retained, achieving (11). The first two ciphertexts of each retained ballot are homomorphically combined, deriving either: a) the combination of a private credential and the negation of that credential, e.g.,

$$\mathcal{E}_{nc}\left(pk, g^{n_0+n_1+n_3} \cdot g^{-n_0-n_1-n_3}\right) \equiv \mathcal{E}_{nc}\left(pk, g^0\right)$$

or b) the combination of a private credential and some other message (excluding the credential's negation). The resulting homomorphic combinations and corresponding encrypted votes are processed (pairwise) by a mixnet. Administrators perform plaintext equality tests on each of the mixed homomorphic combinations to determine whether they contain $g^0 \equiv 1$, partially decrypting the corresponding mixed encrypted votes when the test holds. E.g., for Alice's mixed encrypted vote, but not for any mixed encrypted vote corresponding to a mixed combination of a private credential and some other message. The mixnet anonymises ballots and reveals authorised anonymised ballots, without unveiling whether voters' ballots are authorised, in compliance with heuristics (5), (7), & (8).

Malicious behaviour is detectable throughout: Public key and credential shares, and underlying parameters, are proven to be well-formed, as are ballots. Non-ballots (without proof of correct construction) are discarded, evidence proves anonymised credentials are correctly generation and decrypted. Mixing is proven to output pairs of ciphertexts and plaintext equality tests assure that the election result maps to votes expressed in authorised ballots, satisfying heuristic (1), whilst zero-knowledge proofs ensure no partial mappings (4).

Smyth (2019) formalises the details and proves that Athena is a verifiable, coercion-resistant voting system:

Theorem. *Athena satisfies* Universal-Verifiability, Individual-Verifiability, Unforgeability, *and coercion resistance.*

We omit a formal definition of coercion resistance, for reasons we'll now explain.

That's not what I meant: Formal definitions

In their seminal introductory text, Katz and Lindell (2007) exude, "the first step in solving any cryptographic problem is the foundation of a rigorous and precise definition of security." As Koblitz and Menezes (2013) explain, "definitions allow analysts to know exactly what they're dealing with. When a protocol designer makes a security claim and a cryptanalyst then devises a break, the designer cannot simply say 'that's not what I meant by security.' If it weren't for precise definitions, security claims would be a moving target for analysts." (The theory is all well-and-good, yet formal definitions of secrecy and verifiability came long after many voting systems, and definitions of coercion resistance are fiendishly complex, incomprehensible even to specialists, there's more to be done.)

Ballot secrecy. Consideration of ballot secrecy in cryptographic-based voting systems originates from Chaum (1981) and the earliest definitions are due to Benaloh *et al.* (1996; 1994; 1986). More recently, Bernhard *et al.* (2012; 2015; 2013) proposed a series of definitions. Smyth (2021) found these definitions not to detect vulnerabilities arising when an adversary controls ballot collection, and proposes a new definition that does. Herein, we simplify Smyth's definition.

Ballot secrecy has been criticised: "You can sell your vote," they say—the idea being a voter can present the nonce(s) they used to encrypt their vote. (*Caveat emptor*. Even voter buyers are aware.) Yet, we mere mortals can't even generate a nonce, let alone compute an encryption. We're reliant on opaque machines, which don't typically hand-out nonces. Extracting the things is rather difficult; specialist knowledge and software required. Voters cannot generally prove how they voted. And, as we'll discuss, they surely ain't gonna install some rogue, virus-ridden software to do the job. Even if they did, the notion that voters can extract nonces, whilst being unable to execute their chosen instructions, is rather confounded; strengthening systems to protect against this narrow class of voting-buying attacks neglects the bigger picture: Although vote buying is a cause for concern, coercers are a more significant, more powerful threat. Coercion resistance is the gold standard.

Coercion resistance. Consideration of coercion resistance originates from Okamoto (1998) and the first definition is due to Juels et al. (2002, 2010), with a plethora of coercion-resistant voting systems now in existence. Further definitions have since been proposed (Gardner et al., 2009; Küsters et al., 2012; Unruh and Müller-Quade, 2010). Haines and Smyth (2020) explored each definition, finding one too strong, another burdensome to apply, and the remaining pair too weak. They went on to conclude:

> After two decades of study, coercion-resistance remains poorly understood[.] There is no widely accepted definition. Several are unsuitable[, only one] can capture a range of [evasion] techniques. Yet,

> that definition has only been applied by its authors and only to a few in-person, poll-station voting systems. Consensus on an understandable, general definition that captures a diverse collection of coercion-resistant voting systems remains an open problem.

Formalising coercion resistance remains an open problem, which is why we omit a definition.

With Donald Trump and Hillary Clinton ranting about stolen elections, verifiability belongs centre stage.

Verifiability. Consideration of verifiability originates from Cohen and Fischer (1985) and is advanced by Fujioka et al. (1992), Benaloh and Tuinstra (1994) and Sako and Kilian (1995). More recently, definitions of universal verifiability have been proposed by Juels et al. (2010), Cortier et al. (2014) and Kiayias et al. (2015), with complimentary work by Küsters et al. (2010) proposing global verifiability, a holistic notion of verifiability. Smyth and Clarkson (2020) found that definitions of universal verifiability don't capture Completeness and two don't capture Soundness, similarly, Smyth (2020) found that global verifiability doesn't capture Soundness either. These findings motivated definitions by Smyth et al. (2021), which we've simplified herein, and applied to certify voting systems verifiable.

We implore democracy builders to mandate certified free and fair voting systems, to hold system makers accountable for security claims, preferably before a superpower's election is thrown—a plausible democracy-extinction event.

Denial of Service is inevitable: Viable?

Ballots must not reveal whether they are authorised (6), only anonymised ballots may be authenticated (7). Consequently, coercion-resistant voting systems are vulnerable to Denial of Service (DoS) attacks: An adversary may flood a system with ballots, which must be anonymised prior to authentication; a deluge of unauthorised ballots may overload administrators, render them unable to compute evidence. Further research is required to establish whether coercion resistance is viable—can coercion-resistant systems withstand denial of service attacks?

Coercion resistance demands following instruction be indistinguishable from evading coercion and voting freely. A coerced voter instructed to abstain must be able to evade coercion and vote. That's why ballots must not reveal whether they are authorised (6). If coercion-resistant systems cannot withstand denial of service attacks, they're useless. We'd have to tolerate forced-abstention attacks, simulation attacks too. Further research is required to establish the degree of privacy that's practical.

Scale, security, and robustness: Decentralisation

We've focused on scalable, centralised elections, with administrators. At smaller scale, voting amongst board members, shareholders, and other small groups, an administrator isn't required, enabling secrecy without trust. For instance, every Helios voter could also be an administrator, empowering each voter with the responsibility of protecting their own privacy, without trusting any centralised administrator nor fellow voter. This doesn't scale: Lost or destroyed keys are troublesome for Robustness, ultimately demanding sacrifices in privacy, which is contrary to the purpose of seeking decentralisation. Whether decentralised voting systems can be engineered to offer an advantage remains an active area of study.

Part V—Rotten tech, broken machines, smartphone voting

Bemused gape at rotten voting technology

Researchers from Johns Hopkins and Rice analysed source code believed to have been inadvertently published by voting machine manufacturer Diebold, finding the system "far below even the most minimal security standards...[voters] can cast unlimited votes without being detected...and the threats posed by insiders such as poll workers, software developers, and even janitors, is even greater" (Kohno et al., 2004; Rubin, 2006). Dutch activists found their nation's voting machines leak radio waves that reveal voters' votes (Gonggrijp and Hengeveld, 2007). One of those activists teamed up with researchers from the University of Michigan and Netindia to analyse source code for India's voting machines, finding them "vulnerable to serious attacks that can alter election results and violate the secrecy of the ballot" (Wolchok et al., 2010). Researchers from the University of Michigan went on to play in DC. "Within 48 hours of the [mock election] system going live, we had gained near-complete control of the election server...changed every vote and revealed almost every secret...Election officials did not detect our intrusion for nearly two business days—and might have remained unaware for far longer had we not deliberately left a prominent clue" (Wolchok et al., 2012). (That clue: Voters were doused in the university's fight song upon casting their ballots.) Not quite done, the lead researcher went on to study Estonia's voting system, finding "serious procedural and architectural weaknesses that expose Estonia to the risk that attackers could undetectably alter the outcome of the election" (Springall et al., 2014).

Irrecoverably broken voting machines

Voting machines can be made secure: They just need to be upgraded↪install the algorithms described herein. However, voting machines must still be protected; 'secure' machines can be wiped, insecure software installed. Unfortu-

nately, we are incapable of protecting voting machines. "Anybody can walk up to [unattended] voting machines [in poll stations, just before election day] anybody can manipulate them...nobody will know," explains Professor Venetis (2019), on the John Oliver show: `https://youtu.be/svEuG_ekNT0?t=633` (from 10 min 33 secs). Voting machines must surely be considered harmful to democracy.

The case for smartphone voting

Smartphones can be infected with viruses; that surely makes them unsuitable for voting? Well, maybe, but maybe not: Smartphones are an essential business tool, they are relied upon for vast sums of money—surely they can be relied upon to cast ballots. We've learnt that smartphones can prove they're behaving, voters can be assured that their vote has influence. We concede a virus may reveal a voter's vote. Nonetheless, perhaps smartphones are good enough. (It's not as if voters are actively seeking infection; maybe we should tolerate the occasional leak, no system is perfect.) Smartphones are certainly easier to protect than voting machines. (Just try taking one from a teen!) Maybe smartphones are the best option.

Part VI—Life is unfair

Defective, unsafe at any speed: Plurality voting

Borda (1781) denounced plurality voting systems "seriously defective" (Grazia, 1953) upon discovering *vote splitting*, whereby like-minded voters cast votes for similar candidates, rather than rallying behind a single candidate, creating an advantage for any dissimilar candidate. For example, when two left-leaning candidates run against a right-leaning candidate, the right-leaning candidate has an advantage, because left-leaning voters must pick between two candidates. This is known as the *spoiler effect* when it causes an opponent of similar candidates to win and as a *spoiler* when caused by a minor candidate. Poundstone proclaims "[f]ive presidential elections were probably decided by spoilers...At least two others...are questionable cases. In still another race...four-way vote splitting and the electoral college created such ambiguity that it was a factor in precipitating civil war" (2008, p91). He concludes, "[w]ere the plurality vote a car or an airliner, it would be recognized for what it is—a defective consumer product, unsafe at any speed."

Borda offered a solution: A voting system, now known as *Borda count*, which instructs voters to rank candidates, e.g., $A>B>C$. Tallying involves assigning a descending number of points to each position of each ranking, e.g., a voter's first preference might receive 3 points, the second might receive 2, and the last might receive just 1. The candidate with the most points wins. For instance, given preferences $A>B>C$, $B>A>C$, and $C>B>A$, candidate A receives 6 points, B

receives 7, and C receives 5, candidate B wins. Equivalently, voters can rank candidates by putting numbers next to candidate names and the candidate with the least points wins. Henceforth, we'll consider this version of Borda count.

Helios is tailored to plurality voting rather than ranked voting. Although rankings can be represented as candidates, with voters casting ballots for the candidate that represents their ranking, and tallying revealing the frequency of each ranking, from which an election result can be determined, this approach does not scale. For instance, rankings $A>B>C$, $A>C>B$, $B>A>C$, $B>C>A$, $C>A>B$ and $C>B>A$ can be represented as candidates 1–6. Moreover, given two ballots for candidate 1, one ballot for candidate 4, and one ballot for candidate 6, tallying produces frequency distribution $A>B>C$:2, $A>C>B$:0, $B>A>C$:0, $B>C>A$:1, $C>A>B$:0, $C>B>A$:1, from which candidate A can be determined as the winner. However, ballots include $nc! - 1$ ciphertexts and nc proofs, where nc is the number of candidates. Moreover, tallying requires $(nb-1)\cdot(nc!-1)$ homomorphic combinations and $nc!-1$ proofs demonstrating correct decryption of combinations, where nb is the number of ballots with valid proofs. Such complexity is avoided when votes are encrypted differently, as per Helios Mixnet, for instance.

Some voters are more equal: Ranked voting

Condorcet (1785) claims Borda count is "bound to lead to error...because it takes into account elements which should be ignored" (McLean, 2003, p126). Let's consider this claim with an example: Three voters prefer $A>B>C$ and two voters prefer $B>C>A$. Candidate A is awarded 9 points (one point is awarded by each of the three voters that favour candidate A and three points are awarded by each of the remaining two voters) and B is awarded 8 (one point from each of the two voters favouring candidate B and two points from each of the remaining three voters). Candidate B wins. Borda would reason that B appears one place behind A on three ballots and two places ahead of A on two, hence, B is preferred (McLean, 2003, p11). By comparison, Condorcet reasons that candidate A is favoured over candidate B by 3 out of 5 voters, hence, A is preferred. Or, more generally, any winning candidate should beat every other candidate in a two-way race. Such a winner is known as a *Condorcet winner* and a voting system that ensures this criterion is known as a *Condorcet method*.

Condorcet went on to identify a general issue with ranked voting systems, which we consider with another example. Suppose two voters prefer candidate A to candidate B, and only one voter prefers B to A. A fallacy of composition allows one to infer that the collective preference favours candidate A. However, a truth about partial preferences is insufficient to derive a collective preference. Indeed, amongst the two voters that prefer candidate A to B, one may prefer C to both A and B, and the other may prefer C the least. Moreover, the voter that prefers B to A may prefer B the most and A the least. That is, the voters' preferences might be $C>A>B$, $A>B>C$, and $B>C>A$. Hence, the collective preference is intransitive. The majority prefer A to B, B to C, and C to A.

Despite transitivity of individual preferences. This is known as the *Condorcet paradox*, which demonstrates ranked voting systems can fail to select a winner.

Black (1948) postulated positioning voter preferences along a linear spectrum: Starting with the voter's favourite, who is closest to the voter's ideology, and continuing through the voter's preferences, becoming increasingly distant from that ideology. Hence, a voter that prefers a left-wing candidate A will always favour a centre-left candidate B over a right-wing candidate C. Moreover, a voter that prefers B, will favour A over C. Similarly, a voter that prefers candidate C, will always favour B over A. It follows that only three of the six possible preferences will arise. This can be verified by positioning candidates A, B and C along a line, and attempting to draw a line between candidates. (Candidate A should be closer to B than B is to C.) It is rational to draw lines through A, B, C; B, A, C; and C, B, A; but illogical to draw lines through A, C, B; B, C, A; and C, A, B. (This is presumably somewhat akin to Black's observation: "I wrote down a single diagram and saw...that the diagram...could be interpreted as referring to a committee using a simple-majority procedure," 1996, pp43.) This suffices to preclude individual preferences that lead to intransitive collective preferences, i.e., the Condorcet paradox cannot occur, and leads to *the median voter theorem*, which asserts majority rule voting systems select the candidate favoured by the median voter. (Black's results are theoretical, rather than practical; "political landscapes...are inherently multidimensional and cannot be reduced to a single left-right dimension, or even to a two-dimensional space," Alós-Ferrer and Granić 2015.)

Having rediscovered the Condorcet paradox and that linear ideologies avoid it (Poundstone, 2008, pp38–39), Arrow discovered far more sinister defects of voting systems (1950).

Arrow's impossibility theorem

Arrow states two fairness conditions (Pareto efficient and independence of irrelevant alternatives) and proves that any ranked voting system (including plurality voting, which simply ignores all but the first-ranked candidate) satisfying those two conditions fails to achieve a third (non-dictatorial), in the context of any mathematical function that inputs a set of voter preferences and outputs the corresponding election result. (Our Soundness definition could incorporate that function, in particular, the notion of evidence supporting an election result can be interpreted to mean the result is the function's output.)The consequence: Ranked voting is unfair. (Or we tolerate dictators.) For any ranked voting system that satisfies our seemingly innocuous fairness conditions, a single voter controls the election result!

Theorem. *Given a social welfare function that is Pareto efficient and independent of irrelevant alternatives, the social welfare function is dictorial, assuming at least three candidates.*

The theorem shows there exists scenarios resulting in a dictatorial voter. Let's

work through the details. (What follows is rather technical and you may prefer to skip this section, perhaps watching—`https://youtu.be/Q60ZXoXP6Hg`—an explanation of the key ideas instead, maybe watching regardless, as a primer.)

We start from a situation in which all voters rank a candidate last, e.g.,

$$[\succ]_0 = \begin{Bmatrix} a \succ_1 c \succ_1 \mathbf{b} \\ a \succ_2 c \succ_2 \mathbf{b} \\ c \succ_3 a \succ_3 \mathbf{b} \end{Bmatrix}$$

(In matrices we highlight in bold any candidate about whom we make an assumption.) Next, we start flipping voter preferences for that candidate from last to first, whilst preserving all other relative rankings, e.g.,

$$[\succ]_1 = \begin{Bmatrix} \mathbf{b} \succ_1 a \succ_1 c \\ a \succ_2 c \succ_2 \mathbf{b} \\ c \succ_3 a \succ_3 \mathbf{b} \end{Bmatrix} \quad [\succ]_2 = \begin{Bmatrix} \mathbf{b} \succ_1 a \succ_1 c \\ \mathbf{b} \succ_2 a \succ_2 c \\ c \succ_3 a \succ_3 \mathbf{b} \end{Bmatrix} \quad [\succ]_3 = \begin{Bmatrix} \mathbf{b} \succ_1 a \succ_1 c \\ \mathbf{b} \succ_2 a \succ_2 c \\ \mathbf{b} \succ_3 c \succ_3 a \end{Bmatrix}$$

We flip voter preferences until we discover a situation with a pivotal preference such that flipping the preference causes the social welfare function's rank of the candidate to increase from last to first, i.e., the function ranks the candidate last for set of preferences $[\succ]_{i-1}$ and first for $[\succ]_i$. That search cannot be in vain, because (PE) requires the function to rank the candidate last for $[\succ]_0$ and first when all voters rank the candidate first.

> A social welfare function W is *Pareto efficient*, if the function ranks a candidate above another candidate when all voters rank the former candidate above the latter:
>
> $$\forall a, b \in O \,.\, \bigwedge_{\succ \in [\succ]} a \succ b \implies a \succ_{W([\succ])} b \qquad \text{(PE)}$$
>
> That is, given a pair of candidates a and b from the set of candidates O and given a set of voter preferences $[\succ]$, such that $a \succ b$ for every voter preference $\succ$ in $[\succ]$, it follows that $a \succ_{W([\succ])} b$.

Pareto efficiency teaches us that a social welfare function must rank a candidate last when voters unanimously do so (e.g., as per $[\succ]_0$), but is insufficient to teach us the social welfare function rank of the candidate when ranked either first or last by voters (e.g., as per $[\succ]_1$ and $[\succ]_2$). Coupling Pareto efficiency with independence of irrelevant alternatives allows us to proceed.

> A social welfare function W is *independent of irrelevant alternatives*, if the function computes relative rankings between candidates based only on the voters' relative rankings of those candidates:
>
> $$\forall a, b \in O \,.\, \left(\bigwedge_{\succ \in [\succ]} a \succ b \text{ iff } \bigwedge_{\succ' \in [\succ']} a \succ' b \right) \implies \left(a \succ_{W([\succ])} b \text{ iff } a \succ_{W([\succ'])} b \right) \qquad \text{(IIA)}$$

> That is, given a pair of candidates a and b from the set of candidates O and given sets of voter preferences $[\succ]$ and $[\succ']$, such that $a \succ b$ for every voter preference $\succ$ in $[\succ]$ iff $a \succ' b$ for every voter preference $\succ'$ in $[\succ']$, we have $a \succ_{W([\succ])} b$ iff $a \succ_{W([\succ'])} b$.

Independence of irrelevant alternatives teaches us that a social welfare function must rank a candidate above another when voters unanimously prefer that ranking. A voter changing their rank of a further candidate doesn't alter that fact, e.g., if you favour candidate a over b, then your preference for candidate c is irrelevant to the fact that you prefer a to b. Taken together, Pareto efficiency and independence of irrelevant alternatives teach us that a social welfare function must rank a candidate first or last, when voters rank the candidate either first or last, giving way to the following lemma.

Lemma. *Social welfare function W ranks a candidate first or last when every voter preference $\succ \in [\succ]$ ranks the candidate first or last.*

Proof. Suppose every voter ranks a candidate first or last, e.g.,

$$[\succ] = \begin{Bmatrix} \mathbf{b} \succ_1 a \succ_1 c \\ a \succ_2 c \succ_2 \mathbf{b} \\ c \succ_3 a \succ_3 \mathbf{b} \end{Bmatrix}$$

Now suppose to the contrary that the social welfare function ranks the candidate neither first nor last, hence, there exists higher- and lower-ranked candidates, e.g., $a \succ_{W([\succ])} b \succ_{W([\succ])} c$. Let set of voter preferences $[\succ']$ be derived from $[\succ]$ by moving one such lower-ranked candidate immediately above one such higher-ranked candidate in every voter's preference, e.g., by moving candidate c immediately above a in every preference, i.e.,

$$[\succ'] = \begin{Bmatrix} \mathbf{b} \succ_1 \mathbf{c} \succ_1 a \\ \mathbf{c} \succ_2 a \succ_2 \mathbf{b} \\ \mathbf{c} \succ_3 a \succ_3 \mathbf{b} \end{Bmatrix}$$

Only the relative ranking between those two candidates are modified, so by (IIA), the social welfare function ranking must not change, e.g., $a \succ_{W([\succ'])} b \succ_{W([\succ'])} c$, hence, $a \succ_{W([\succ'])} c$ by transitivity. Yet, that contradicts (PE), since all voters rank one of those two candidates higher than the other, e.g., each voter ranks c above a. □

It follows naturally that a pivotal preference can be identified.

Corollary. *Given a social welfare function and a candidate, there exists a set of voter preferences for which the function's rank of the candidate increases from last to first when a single voter preference is modified.*

Proof. By (PE), the social welfare function ranks a candidate last when every voter preference does so. Moreover, from our lemma, it follows that there exists a set of voter preferences such that flipping a single voter's preference for that candidate from last to first increase the social welfare function's ranking of that candidate to first. □

Since a pivotal preference can be identified, we learn that there exist scenarios in which a single voter preference determines a candidate's fate. To demonstrate that the voter controlling that preference is a dictator, we will show they can pick the relative ranking of all other pairs of candidates, that is, the voter's preference coincides with the social choice function's rank.

Having identified a candidate that the social welfare function ranks last for set of voter preferences $[\succ]_{i-1}$ and first for set $[\succ]_i$, we know the function must rank every other candidate higher in the former instance. The function must also rank one of those other candidates higher when we modify a pivotal preference to favour one such candidate, because the relative rankings between the two candidates remain the same. Indeed, one of the candidates is ranked last for voters $1, \ldots, i-1$ and first for voters $i+1, \ldots, |[\succ]|$, moreover, that candidate is below the other candidate in the modified pivotal preference, hence, by (IIA), the social welfare function must rank the other candidate higher. E.g., suppose we modify $[\succ]_2$ to derive $[\succ]_2^a$ by replacing pivotal preference $b \succ_2 a \succ_2 c$ in $[\succ]_2$ with $a \succ_2 b \succ_2 c$, i.e.,

$$[\succ]_2^a = \begin{Bmatrix} b \succ_1 a \succ_1 c \\ \mathbf{a} \succ_2 \mathbf{b} \succ_2 c \\ c \succ_3 a \succ_3 b \end{Bmatrix}$$

Since the social welfare function ranks a higher than b in $[\succ]_1$, and since the relative rankings between a and b remain the same in both $[\succ]_1$ and $[\succ]_2^a$, the social welfare function must rank a higher than b in $[\succ]_2^a$. Moreover, the function must continue to rank that other candidate higher when we modify all other voter preferences (excluding the aforementioned pivotal preference) by switching that other candidate's rank with the rank of some other candidate, because the relative ranking between our initial two candidates are unchanged. E.g.,

$$[\succ]_2^{\hat{a}} = \begin{Bmatrix} b \succ_1 \mathbf{c} \succ_1 \mathbf{a} \\ \mathbf{a} \succ_2 \mathbf{b} \succ_2 c \\ \mathbf{a} \succ_3 \mathbf{c} \succ_3 b \end{Bmatrix}$$

Similarly, the candidate that ranks first for $[\succ]_i$ must rank higher than a further additional candidate, moreover, the first-ranked candidate must still be favoured after applying both of the aforementioned modifications to $[\succ]_i$, e.g., in $[\succ]_2^{\hat{a}}$. It follows by transitivity that the first additional candidate must rank higher than the second additional candidate.

Our reasoning is rather contrived—we've orchestrated a scenario in which the voter controlling the pivotal preference controls the fate of a pair of candidates. Let's eliminate the scaffolding, generalise to show the controlling voter is a dictator: Suppose we arbitrarily modify ranking of b in $[\succ]_2^{\hat{a}}$ for all preferences, and by arbitrarily modifying the rank of a in the pivotal preference under the constraint that a remains higher than c. E.g.,

$$[\succ]_2^{\bar{a}} = \begin{Bmatrix} c \succ_1 b \succ_1 a \\ b \succ_2 \mathbf{a} \succ_2 \mathbf{c} \\ a \succ_3 c \succ_3 b \end{Bmatrix}$$

It follows that voter preferences are all arbitrary, except the pivotal preference which ranks a higher than c, hence, the voter controlling the pivotal preference controls the fate of that pair of candidates. By (IIA), we can further generalise to all pairs, i.e., the voter controlling the pivotal preference is a dictator.

> A social welfare function W is *dictatorial*, if the function computes the result from a single voter's preference:
>
> $$\exists \succ \in [\succ] \,.\, \forall a, b \in O \,.\, a \succ b \implies a \succ_{W([\succ])} b \tag{D}$$
>
> That is, there exists a set of voter prefences $[\succ]$ containing voter preference $\succ$, such that for each pair of candidates a and b from the set of candidates O, if $a \succ b$, then $a \succ_{W([\succ])} b$.

Lemma. *The voter that modifies their preference in our corollary—to increase the social welfare function's rank of a candidate from last to first—is a dictator over all other pairs of candidates.*

Proof. Let $[\succ]_{i-1}$ be the set of voter preferences that exists by our corollary, and let b be the candidate whose rank increases from last to first. Suppose $x, y \in O \setminus \{b\}$, e.g., when $x = a$ and $y = c$. Let $[\succ]_i^x$ be derived from $[\succ]_i$ by modifying a pivotal preference to rank x first, whilst preserving all other relative rankings, e.g., $[\succ]_2^a$. Moreover, let $[\succ]_i^{\hat{x}}$ be derived from $[\succ]_i^x$ by modifying the relative rankings of x and y for all preferences except the pivotal preference, whilst preserving b in its extremal position, e.g., $[\succ]_2^{\hat{a}}$. From our corollary, we know the social welfare function ranks b last, hence, $x \succ_{W([\succ]_{i-1})} b$ and, by (IIA), we have $x \succ_{W([\succ]_i^{\hat{x}})} b$, because the relative rankings between x and b are the same for all voters. Indeed, b is ranked first for voters $1, \ldots, i-1$ and last for voters $i+1, \ldots, |[\succ]|$, moreover, b is ranked last in the unmodified pivotal preference and below x after modification. Similarly, we also know $b \succ_{W([\succ]_i)} y$ and, by (IIA), we have $b \succ_{W([\succ]_i^{\hat{x}})} y$, because relative rankings between b and y are the same in $[\succ]_i$ and $[\succ]_i^{\hat{x}}$. It follows by transitivity that $x \succ_{W([\succ]_i^{\hat{x}})} y$.

Let $[\succ]_i^{\bar{x}}$ be derived from $[\succ]_i^{\hat{x}}$ by arbitrarily modifying the rank of b for all preferences, and by arbitrarily modifying the rank of x in the pivotal preference under the constraint that x remains higher than y, e.g., $[\succ]_2^{\bar{a}}$. We started from the set of voter preferences $[\succ]_i$, which only made assumptions on the position of candidate b; we derived $[\succ]_i^x$ by ranking x first in the pivotal preference and $[\succ]_i^{\hat{x}}$ by modifying the relative rankings of x and y for all preferences except the pivotal preference; finally, we derived $[\succ]_2^{\bar{x}}$ by arbitrarily modifying the rank of b for all preferences, and by arbitrarily modifying the rank of x in the pivotal preference under the constraint that x remains higher than y. It follows that voter preferences are all arbitrary, except the pivotal preference which ranks x higher than y, e.g., a beats c.

We have $x \succ_{W([\succ]_i^{\hat{x}})} y$ and, since the relative rankings between x and y are unchanged between $[\succ]_i^{\hat{x}}$ and $[\succ]_i^{\bar{x}}$, it follows by (IIA) that $x \succ_{W([\succ]_i^{\bar{x}})} y$. Yet, we have only assumed that x is ranked higher than y in the pivotal preference. Thus, the voter that modifies their preference in the above corollary is a dictator over x and y, and for that matter all pairs of candidates that exclude b. □

Proof of Arrow's impossibility Theorem. By our corollary, given a candidate z, there exists a set of voter preferences for which the social welfare function's rank of candidate z increases from last to first when a single voter's preference is modified. It follows by the above lemma that the single voter is a dictator over all other pairs of candidates $x, y \in O \setminus \{z\}$. □

Our proof is inspired by Kevin Leyton-Brown's excellent lecture: `https://youtu.be/QLi_5LCwJ20`.

Arrow's impossibility theorem teaches us that ranked voting is fallible, an ideal ranked-voting system simply cannot exist; there is a possibility that a single voter may yield dictorial control over the result. That's not to say such scenarios are probably, only that such scenarios can exist.

Part VII—Life could be fairer: We're being screwed

Range voting: Hot or Not?

At least one of the authors was (unknowingly) inducted to *Hot or Not* by his bestie, at uni, c. 2003—she wanted to establish her superior attractiveness. Receiving an almost two-decade later reminder, from Poundstone, was unexpected! *Hot or Not* (aka range voting), wherein voters score a candidate (or photo) on a scale (e.g., ten for hot, one for not), provides a means around Arrow's impossibility result (which only applies to ranked voting systems, rather than rating-based systems such as Hot or Not). Vote splitting and spoilers are avoided; similar candidates can be scored the same: "Arrow's Nobel-winning 1951 'impossibility theorem' misdirected the entire field of voting systems for 50 years," bashed Smith (2007), "Arrow's theorem is not nearly as important as it at first seems" (2005). Having previously argued the superiority of range voting over ranked voting, on the basis that range voting creates less "human unhappiness" than ranked voting (2000; c. 2008), Smith concludes "switching to range voting would be a larger improvement in 'democracy' than the entire invention of democracy" (2004)—Churchill, we have a solution, range voting, which perhaps pre-dates the human race, possibly originating from the nesting behaviour of honeybees (Smith, 2007).

Democracy is the worst, except for other forms

Voting system deployment is rigged; not mentioning so would be indiligent. That's not a problem for the technologist that authored this manuscript, the problem is independent of voting systems themselves. The problem is for you, us, society. The failings are blindingly obvious:

One person, one vote—is a fallacy

Cornell Law School's Legal Dictionary and Encyclopedia places limitations on the above, restricting voter equality only to the state: Alice has more power than Bob to elect the President of the United States of America, when they live in different states! That's not the worst of it:

Gerrymandering

Our legislators manipulate voting districts to facilitate their re-election or advance another of their agendas, including discrimination on the basis of ethnicity, race, and religion. By rearranging voting districts, our legislators pack opponents into districts, wasting votes of those packed in, whilst spreading favourable voters to achieve sufficient winning advantage. (Moon Duchin gives an excellent lecture on the topic: `https://math.uni.lu/mast/duchin.html`.) Perhaps we should try some other forms known to be beneficial:

Popular vote; the electorate's choice

One person, one vote is trivial to achieve—just take the presidential candidate with the most votes, nationwide. This would have been unheard of; without mechanisation, counting a nation's votes is foolhardy. The industrial age has come and gone, the popular vote is computable. Of course, candidates would then need to appeal to the electorate, rather than key Gerrymandered constituents, which is hardly appealing to politicians …

President vs. CEO; delegate representation

Shareholders elect CEOs, not their team—the same could be true for presidents: Representative (s)election can be delegated to parties (in ratios defined by the popular vote) and representatives randomly assigned to constituencies.

We should dump defective plurality voting, perhaps opt for ranked voting, maybe even range voting, surely that's what we deserve. We could even abandon governments, favour direct democracy, we have powerful communication systems, it is the information age.

Democracy, the other form: Universal suffrage

Our mission isn't to prescribe range voting (evidence already indicates it's the best), our mission is to provide democracy builders with the best tools for their stakeholders, in alignment with their constitution or bylaws, our mission is to enable universal suffrage; democracy builders are ultimately responsible for granting stakeholders a free and fair vote—we're merely here to provide tools to do so.

Secret verifiable voting illegal in UK!

Shockingly, UK law prohibits secret national elections: By royal decree, voters' votes can be revealed by matching numbered ballot papers with numbered counterfoils (bearing voter identities). Ballot papers and counterfoils are huddled up in warehouses long after election day, only accessible by order of the High Court or Parliament. Theoretically, at least. Are we to truly believe secrecy has never been compromised by the explicitly-incorporated backdoor?

Part VIII—Mission

Doubt over election results erode democracy, descend civilised society into chaos: On 6 January 2021, a mob stormed the Capitol, attempting to overthrow those that had 'stolen' the election from Donald Trump. Their mission: To restore democracy. "If you don't fight like hell, you're not going to have a country anymore."

Our mission: Eradicate rotten voting technology, equip democracy builders with certifiably free and fair systems, preferably before a U.S. election is thrown by hackers, a plausible extinction catalyst.

References

Ben Adida. Helios: Web-based Open-Audit Voting. In *USENIX Security'08*, pages 335–348. USENIX Association, 2008.

Ben Adida, Olivier de Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. Electing a University President Using Open-Audit Voting: Analysis of Real-World Use of Helios. In *EVT/WOTE'09*. USENIX Association, 2009.

Carlos Alós-Ferrer and Đura-Georg Granić. Political space representations with approval data. *Electoral Studies*, 39:56–71, 2015.

Kenneth Joseph Arrow. A Difficulty in the Concept of Social Welfare. *Journal of Political Economy*, 58(4):328–346, 1950.

Josh Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Department of Computer Science, Yale University, 1996.

Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections. In *STOC'94*, pages 544–553. ACM Press, 1994.

Josh Benaloh and Moti Yung. Distributing the Power of a Government to Enhance the Privacy of Voters. In *PODC'86*, pages 52–62. ACM Press, 1986.

David Bernhard, Olivier Pereira, and Bogdan Warinschi. How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. In *ASIACRYPT'12*, volume 7658 of *LNCS*, pages 626–643. Springer, 2012.

David Bernhard, Véronique Cortier, David Galindo, Olivier Pereira, and Bogdan Warinschi. SoK: A comprehensive analysis of game-based ballot privacy definitions. In *S&P'15*, pages 499–516. IEEE Computer Society, 2015.

Duncan Black. On the Rationale of Group Decision-Making. *Journal of Political Economy*, 56(1):23–32, 1948.

Duncan Black. Some notes on the development and structure of the theory of committees. In Gordon L Brady and Gordon Tullock, editors, *Formal Contributions to the Theory of Public Choice: The Unpublished Works of Duncan Black*, chapter 5. Springer, 1996.

Jean-Charles de Borda. Mémoire sur les Élections au Scrutin. *Mémoires de l'Académie Royale des Sciences*, pages 657–665, 1781.

Philippe Bulens, Damien Giry, and Olivier Pereira. Running Mixnet-Based Elections with Helios. In *EVT/WOTE'11*. USENIX Association, 2011.

Nicholas Chang-Fong and Aleksander Essex. The Cloudier Side of Cryptographic End-to-end Verifiable Voting: A Security Analysis of Helios. In *ACSAC'16*, pages 324–335. ACM Press, 2016.

David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24:84–90, 1981.

Josh Daniel Cohen and Michael J. Fischer. A Robust and Verifiable Cryptographically Secure Election Scheme. In *FOCS'85*, pages 372–382. IEEE Computer Society, 1985.

Nicolas de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Cambridge University Press, 2014. First published in 1785.

Véronique Cortier and Ben Smyth. Attacking and fixing Helios: An analysis of ballot secrecy. *Journal of Computer Security*, 21(1):89–148, 2013.

Véronique Cortier, David Galindo, Stephane Glondu, and Malika Izabachène. Election Verifiability for Helios under Weaker Trust Assumptions. In *ESORICS'14*, volume 8713 of *LNCS*, pages 327–344. Springer, 2014.

Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A Practical Secret Voting Scheme for Large Scale Elections. In *AUSCRYPT'92*, volume 718 of *LNCS*, pages 244–251. Springer, 1992.

Ryan W. Gardner, Sujata Garera, and Aviel D. Rubin. Coercion Resistant End-to-end Voting. In *FC'09*, volume 5628 of *LNCS*, pages 344–361. Springer, 2009.

Rop Gonggrijp and Willem-Jan Hengeveld. Studying the Nedap/Groenendaal ES3B Voting Computer: A Computer Security Perspective. In *EVT'07: Electronic Voting Technology Workshop*. USENIX Association, 2007.

Alfred de Grazia. Mathematical Derivation of an Election System. *Isis*, 44: 42–51, 1953. Translation of Borda (1781).

Thomas Haines and Ben Smyth. Surveying definitions of coercion resistance. IACR ePrint 2019/822, 2020.

Martin Hirt. *Multi-Party Computation: Efficient Protocols, General Adversaries, and Voting*. PhD thesis, ETH Zurich, 2001.

Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections, 2002. IACR ePrint 2002/165.

Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. volume 6000 of *LNCS*, pages 37–63. Springer, 2010.

Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2007.

Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. End-to-end verifiable elections in the standard model. In *EUROCRYPT'15*, volume 9057 of *LNCS*, pages 468–498. Springer, 2015.

Neal Koblitz and Alfred Menezes. Another look at security definitions. *Advances in Mathematics of Communications*, 7:1–38, 2013.

Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin, and Dan S. Wallach. Analysis of an Electronic Voting System. In *S&P'04: 25th Security and Privacy Symposium*, pages 27–40. IEEE Computer Society, 2004.

Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Accountability: Definition and relationship to verifiability. In *CCS'10*, pages 526–535. ACM Press, 2010.

Ralf Küsters, Tomasz Truderung, and Andreas Vogt. A Game-Based Definition of Coercion-Resistance and its Applications. *Journal of Computer Security*, 20(6):709–764, 2012.

Iain McLean. The Reasonableness of Independence: A Conversation from Condorcet and Borda to the present day. Technical Report 2003-W6, Nuffield College, University of Oxford, March 2003.

Maxime Meyer and Ben Smyth. Exploiting re-voting in the helios election system. *Information Processing Letters*, (143):14–19, 2019.

Tatsuaki Okamoto. Receipt-Free Electronic Voting Schemes for Large Scale Elections. In *SP'97*, volume 1361 of *LNCS*, pages 25–35. Springer, 1998.

William Poundstone. *Gaming the vote: Why elections aren't fair (and what we can do about it)*. Macmillan, 2008.

Aviel D. Rubin. *Brave New Ballot: The Checkered Past And Frightening Future of Electronic Voting in America*. Broadway Books, 2006.

Kazue Sako and Joe Kilian. Receipt-Free Mix-Type Voting Scheme - A Practical Solution to the Implementation of a Voting Booth. In *EUROCRYPT'95: 12th International Conference on the Theory and Applications of Cryptographic Techniques*, volume 921 of *LNCS*, pages 393–403. Springer, 1995.

Warren D. Smith. Range Voting, 2000.

Warren D. Smith. Smith's replies to attempts to criticize Range Voting, 2004.

Warren D. Smith. The voting impossibilities of Arrow, Gibbard & Satterthwaite, and Young, 2005.

Warren D. Smith. Ants, Bees, and Computers agree Range Voting is best single-winner system, 2007.

Warren D. Smith. Bayesian regret for dummies, c. 2008.

Ben Smyth. Verifiability of Helios Mixnet. In *Voting'18: 3rd Workshop on Advances in Secure Electronic Voting*, LNCS. Springer, 2018.

Ben Smyth. Athena: A verifiable, coercion-resistant voting system with linear complexity. IACR ePrint 2019/761, 2019.

Ben Smyth. Surveying global verifiability. *Information Processing Letters*, 2020.

Ben Smyth. Ballot secrecy: Security definition, sufficient conditions, and analysis of Helios, 2021. IACR ePrint 2015/942.

Ben Smyth and David Bernhard. Ballot secrecy and ballot independence coincide. In *ESORICS'13*, volume 8134 of *LNCS*, pages 463–480. Springer, 2013.

Ben Smyth and Michael R. Clarkson. Surveying definitions of election verifiability, 2020. Draft.

Ben Smyth and Yoshikazu Hanatani. Non-malleable encryption with proofs of plaintext knowledge and applications to voting. *International Journal of Security and Networks*, 14(4):191–204, 2019.

Ben Smyth, Steven Frink, and Michael R. Clarkson. Election Verifiability: Cryptographic Definitions and an Analysis of Helios, Helios-C, and JCJ, 2021. IACR ePrint 2015/233.

Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine, and J. Alex Halderman. Security Analysis of the Estonian Internet Voting System. In *CCS'14: 21st ACM Conference on Computer and Communications Security*, pages 703–715. ACM Press, 2014.

Dominique Unruh and Jörn Müller-Quade. Universally Composable Incoercibility. In *CRYPTO'10*, volume 6223 of *LNCS*, pages 411–428. Springer, 2010.

Penny Venetis. Voting Machines: Last Week Tonight with John Oliver, 2019.

Scott Wolchok, Eric Wustrow, J. Alex Halderman, Hari K. Prasad, Arun Kankipati, Sai Krishna Sakhamuri, Vasavya Yagati, and Rop Gonggrijp. Security Analysis of India's Electronic Voting Machines. In *CCS'10: 17th ACM Conference on Computer and Communications Security*, pages 1–14. ACM Press, 2010.

Scott Wolchok, Eric Wustrow, Dawn Isabel, and J. Alex Halderman. Attacking the Washington, D.C. Internet Voting System. In *FC'12: 16th International Conference on Financial Cryptography and Data Security*, volume 7397 of *LNCS*, pages 114–128. Springer, 2012.