GEMSPACE VOTING RECORDATION SYSTEM

Inventors: Daniel Wolf, Esq., 1339 West Pennsylvania Avenue, San Diego, CA,
Glenna Dawn Parker, 416 West San Ysidro Blvd., #L217, San Ysidro, CA
Alec Hahn, Republic of Germany
Steven Morrey, Arizona, USA

PROVISIONAL PATENT APPLI CATION

I. FIELD OF THE INVENTION

A method of electronically recording, submitting, securing, tabulating and verifying voting results in real time, and providing for real-time 3rd-party oversight and documentation of the same.

II. GLOSSARY

Authority - Any party interested in the outcome of a particular election whether a political party, government agency or NGO. Specifically the authority is the one concerned enough with the outcome of the election to take action to ensure it remains free and fair.

Bitcoin - A specific implementation of blockchain technology which is widely known and referenced here for example purposes.

Address - The result of a specific hash function as applied to the "public" part of a public/private keypair (see ECDSA)

Bitmap - A collection of bytes arranged (mapped) to a specific format. In general a bitmap will usually represent image data. For our purposes bitmap refers to any image data when we do not need to reference a specific format or standard such as JPEG, PNG, SVG etc.

Block - A block is a specific arrangement of transactions. The specific arrangements and requirements in order to be accepted as a valid block is dependent upon the underlying blockchain utilized. For our purposes it is merely any arrangement of transactions that the blockchain records.

Blockchain - A distributed, trustless time-stamping service with an open and public ledger. Messages to the blockchain are called transactions. For our purposes all transactions contain at least a time-stamp, a hash of the bytes of the transaction not counting the signature and a cryptographic signature of the hash of the transactional bytes. They can reference a previous transaction or not depending on the class of transaction and the specific implementation of the blockchain.

These transactions are then broadcast to the network and placed into the blockchain ledger and shared amongst all participants.

Note: The ledger itself is what is actually termed "the blockchain", but colloquially the entire network is also called a blockchain. More properly the network should be termed a "blockchain network", but that is not the common parlance. Someone skilled in the art of implementing blockchain networks and their tools should be easily able to identify the specific term simply from the context, so that will not be called out specifically here as it is considered common knowledge.

A blockchain network serves as an indisputable witness to the events that are submitted to it, because all nodes that participate are agreeing to a certain set of rules called "consensus rules". As long as the majority of nodes agree on the same set of rules, then the data in the blockchain ledger will remain intact and cannot be disputed by any party.

The predominate implementations of blockchain technology at this point in time are so called "cryptocurrencies" such as bitcoin, litecoin and thousands of others, which use this shared ledger technology predominately as a ledger of accounts. Our primary innovation is to adapt this ledger to serve a purpose more inline with that of a "journal or ledger of notorial acts", while still maintaining the original usage of the blockchain as a financial ledger or whatever the implementer originally intended. This is because when all else is stripped away every blockchain network is really nothing more than a distributed time-stamped ledger of witnessed events.

Collision - In mathematics a collision is when a given hash function produces the same output given 2 different inputs. Collisions are generally considered undesirable.

Cryptocoin - Blockchain technology when used as a general ledger of account balances. Bitcoin is the most well known example of this.

ECDSA - Cryptographic standard as defined in NIST FIPS 186-4

Hash - A hash is a one-way mathematical function. This means that a given input will always produce a particular output, but the output cannot be used to "reverse" the computation and arrive at the input. The most common use of hashes is to verify that a file which was downloaded is exactly the same file that was originally uploaded. Hashes are useful in cryptography because they can be used to ensure that nothing in a given document (or image) has changed.

High quality hashing algorithm feature a cascading mechanism, meaning that a minor change to the input will always result in a major change to the output. If a single pixel were changed in a photograph, for instance, a completely different hash will be generated. With this cascading mechanism, assuming you have all the original bytes that the hash was computed against, you should always get the same resulting hash as anyone else who also had those same bytes.

In this regard a hash can be thought of as a tamper proof seal. If you compute a hash and it's different from the posted hash then you know for a fact that someone tampered with whatever input you are checking against. There are a myriad of hashing functions. In general we are speaking of SHA256 when we say hash, unless the underlaying blockchain network we are piggybacking on top of has requirements for a different algorithm.

Keypair - In public key encryption a keypair is used to refer to the combination of private and public keys.

Mining - Mining is a reward based system intended to encourage independent individuals and companies to commit their computing resources to the blockchain network and aide the network in the process of arranging transactional data into blocks.

Anyone who runs equipment engaged in the act of mining is termed a "miner".

Mining Pool - As a blockchain grows in capacity it becomes less and less likely for any one miner to find a block. Mining pools are collectives of miners. Pools are more likely to find a block than any individual miner. Most blockchain networks possess several large mining pools.

P2KH - The most common of several different types of transactions. The person creating the next transaction in the chain need only reference the P2KH transaction and sign it with the correct private key.

P2SH - One of several transactional types that is widely supported by most blockchain networks. In a P2SH transaction, write authority for the next transaction in a chain of transactions is confered by a hash of a script. A script is a small application.

This enables the composition of advanced and complex transactions such as time locked transactions that also require multiple signatures to be valid.

Parallel Count - A separate tally of votes kept by a party that is distinct from the body officially tasked with counting the votes.

PKE - Public Key Encryption (Asymmetric Key Encryption) - With this form of encryption there exists both a public key and a private key. The public key can be derived from the private key, but the private key cannot be derived from the public key. Anyone who has the recipient's public key can use that key to encrypt a message that only someone possessing the corresponding private key can decrypt. Furthermore the private key can be used to sign a message and if the signers public key is known then authenticity of the message can be assured. When we speak of PKE we are mostly speaking of Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Integrated Encryption Scheme (ECIES).

Poll Watcher - A group or person tasked watching the polling location(s) and reporting on any abnormal or suspicious behavior.

Poll Worker - A person employed to conduct the business of the vote. This may include but is not limited to, signing in voters, verifying voter franchise, handing out ballots, directing voters to specific booths and collecting ballots. The specifics of the role will depend upon a number of factors including the type of voting equipment as well as local laws and regulations.

Signature - A signature is a set of bytes produced by a cryptographic "signing" function. Anyone who has the public key of the sender can later verify that the sender created the message by applying the DSA function to the bytes of the message along with the public key and the signature.

Surveyor - A person or group tasked with compelling individuals to answer questions about a topic of interest such as whom specifically the person being queried decided to vote for.

Wallet - A colloquial term which implies a collection of 1 or more keypairs. The correct cryptographic term for this is a keyring. The term wallet was chosen for much of this document because it is the parlance of most of the specific blockchain networks. For our purposes wallet will always equal a keyring.

III. CURRENT STATE OF THE ART OF VOTING SYSTEMS

Democracy relies on 2 principles, franchisement also known as the right to vote, and authenticity which is the assurance of accuracy of the count. In ancient times, people could watch the ballot and the actual act of counting. In modern times computer equipment has replaced the counters of old and this introduces a myriad of new problems. The weaknesses and vulnerabilities in American voting and counting systems in particular are well documented. With all current systems and methods, it is a relatively straightforward matter to steal, switch or disappear opponents' votes or to augment one's own count and this is a danger to the democratic process.

Guaranteeing accurate counting of authentic citizens' votes is as important as guaranteeing their right to vote.

Verified Voting Foundation proposes the following Principles for New Voting Systems1:

1. It should use human-readable marks on paper as the official record of voter preferences and as the official medium to store votes.

2. It should be usable by all voters; accessible to all voters, including those with disabilities; and available in all mandated languages.

3. It should provide voters the means and opportunity to verify that the human-readable marks correctly represent their intended selections, before casting the ballot.

4. It should preserve vote anonymity: it should not be possible to link any voter to his or her selections, when the system is used appropriately. It should be difficult or impossible to compromise or waive voter anonymity accidentally or deliberately.

4 No voter should be able to prove how he or she voted.

5. It should export contest results in a standard, open, machine-readable format.

6. It should be easily and transparently auditable at the ballot level. It should:

export a cast vote record (CVR) for every ballot,

in a standard, open, machine-readable format,

in a way that the original paper ballot corresponding to any CVR can be quickly and unambiguously identified, and vice versa.

7. It should use commercial off-the-shelf (COTS) hardware components and open-source software (OSS) in preference to proprietary hardware and proprietary software, especially when doing so will reduce costs, facilitate maintenance and customization, facilitate replacing failed or obsolete equipment, improve security or reliability, or facilitate adopting technological improvements quickly and affordably.

8. It should be able to create CVRs from ballots designed for currently deployed systems and it should be readily configurable to create CVRs for new ballot designs.

9. It should be sufficiently open to allow a competitive market for support, including configuration, maintenance, integration, and customization.

10. It should be usable by election officials: they should be able to configure, operate, and maintain the system, create ballots, tabulate votes, and audit the accuracy of the results without relying on external expertise or labor, even in small jurisdictions with limited staff.

The existence of these proposed standards constitutes circumstantial evidence that most systems in use fall short of their level of rigor.

Voting machine vulnerability "is a problem that few government officials are able or willing to deal with", according to a new study by the Brennan Center for Justice called America's Voting Machines at Risk.2 Among the key findings in the report:

* Nearly every state is using some machines that are no longer manufactured, and many election officials struggle to find replacement parts;

* Election jurisdictions in at least 31 states want to purchase new voting machines in the next five years. Officials from 22 of these states said they did not know where they would get the money to pay for them;

* Without federal or state funding, wealthier counties will replace aging machines, while poorer counties will be forced to use outdated machines for far longer than they should."3

2. "Some Princeton researchers made a demonstration video of how it's possible to steal an election with a Diebold voting machine in under a minute. Anyone with physical access to the machine can put in malicious software to steal votes; such as election workers who have unsupervised access to the machines before elections. All they have to do is open up the machine with a key (or pick the lock), remove the old memory card, stick in your own memory card, boot the machine, and it automatically installs any software that was on the memory card.

At the end of the demonstration election, the poll machine prints out the incorrect "stolen election" result. The internal memory card also stores the incorrect result. Every piece of evidence of how the election actually went reflects the "wrong" result. And, after the election is over, the vote stealing software can delete itself. There's no evidence left that the vote has been conducted incorrectly."4

3. Diebold and Sequoia voting machines "used by as many as a quarter of American voters heading to the polls in 2012 could be hacked with just $10.50 in parts and an 8th-grade science education, according to computer science and security experts at the Vulnerability Assessment Team at Argonne National Laboratory in Illinois. The experts said the newly developed hack could change voting results while leaving absolutely no trace of the manipulation behind.5

"We believe these man-in-the-middle attacks are potentially possible on a wide variety of electronic voting machines," said Roger Johnston, leader of the assessment team "We think we can do similar things on pretty much every electronic voting machine."

"The really nice thing about this attack, the man-in-the-middle, is that there's no soldering or destruction of the circuit board of any kind," Warner says. "You can remove this attack and leave no forensic evidence that we've been there."6

4. Many voting machines can be hacked quite easily.7,8 Finnish computer expert Harri Hursti, according to UC Berkeley experts, was "able to change the election results by doing nothing more than modifying the contents of a memory card.

He needed no passwords, no cryptographic keys, and no access to any other part of the voting system, including the GEMS election management server."9 Many use proprietary software that cannot be examined by independent computer experts, and that therefore cannot be independently certified as secure or free of vote-altering bugs. Independent analysis in California demonstrated that the software examined was rife with problems.10 AVS machines in Virginia, with passwords "admin" and "abcde", could be hacked from a distance by persons with very low sophistication; they used an obsolete version of Windows that would allow malicious code to be run, were never tested for penetrability, never kept auditable logs, and were still being used until recently.11

The problem, of course, is not restricted to presidential elections: Vote switching and vote losing are common phenomena, and have probably caused the wrong candidates to win - one particularly egregious example is the 13th Congressional District in Florida, where in 2006 a Republican was elected in a Democratic-leaning district by 369 votes after 18,000 votes mysteriously disappeared.12,13

Many of these suspect machines are still in use around the country, in spite of the widespread knowledge of their weaknesses.14

One of the foremost experts on voting technology, Dr. David Dill of Stanford University, wrote that "Computerized voting equipment is inherently subject to programming error, equipment malfunction, and malicious tampering. It is therefore crucial that voting equipment provide a voter-verifiable audit trail, by which we mean a permanent record of each vote that can be checked for accuracy by the voter before the vote is submitted, and is difficult or impossible to alter after it has been checked. Many of the electronic voting machines being purchased do not satisfy this requirement.

Voting machines should not be purchased or used unless they provide a voter-verifiable audit trail; when such machines are already in use, they should be replaced or modified to provide a voter-verifiable audit trail. Providing a voter-verifiable audit trail should be one of the essential requirements for certification of new voting systems."15

IV. SUMMARY

Our invention is a voting and vote recordation system that uses blockchain technology and solves various challenges by using secure storage and recordation of the vote.

The system elaborated upon herein is for all practical purposes, transparent, permanent, and invulnerable to mis-recording and miscounting, and that produces direct evidence of the validity of the individual votes and other data by creating a digital record in real-time of the ballots as cast by the voter.

Furthermore it also preserves, in applications where it is called for, photographic evidence, all of which is encrypted and uploaded onto publicly accessible storage networks, where it is preserved permanently and made available for inspection in real time or with delayed release when appropriate to allow polling stations to close.

The voting system is Embodiment One. We anticipate at least six (6) total preferred embodiments. All use the same software and data handling methods of Embodiment One, but run in separate instantiations to preserve independence, create redundancy, and provide an accountability check against official data.

Results from all embodiments are reported instantly on the blockchain, which makes it impossible for anyone to claim that tabulators of the data have changed it retrospectively. This also allows real-time independent verification of results. And because the public keys can be uploaded to another service where people can view the results of their own voting while still maintaining anonymity.

Embodiment One

A method of electronically checking in voters, recording, submitting, securing, tabulating and verifying their votes, and allowing third-party verification of the voters' choices, in near-real time.

Embodiment Two

An official method for precincts to record and submit tallies of precinct-level data, including vote totals, voided ballots, voter check-ins and no-shows, etc.

Embodiment Three

A civil-society-owned method for third-party poll watchers to independently record and submit the same data as provided for in Embodiment Two, but running in a separate instantiation to preserve independence and create redundancy and an accountability check.

Embodiment Four

A method for third-party surveyors to directly record, submit, secure, tabulate and verify the voting choices of cooperating voters departing from their polling places, using essentially the same voter-facing GUI as described in Embodiment One. (Embodiment One does not have to be in use for this embodiment to be useful.) It functions identically to Embodiment One, but the authority in this context is an independent party instead of "Secretary of State" or whatever official would be tasked with collecting this information for official purposes.

Embodiment Five

A method for third-party observers to photograph and notate tangible and visible events, e.g., ballot-box stuffing, voter intimidation, human rights violations etc., and to secure and upload them in real time without disclosing their own identity.

Embodiment Six

A method very similar to Embodiment One that allows voters themselves to digitally and, photographically record their votes while preserving their anonymity, such that a separate voter count is provided. It functions identically to Embodiment One, but the authority in this context is an independent party instead of "Secretary of State" or whatever official would be tasked with collecting this information for official purposes.

V. DRAWINGS

Per 37 CFR 1.81 a drawing does not appear necessary to understand the subject matter being patented. However, if the patent examiner requests one, it will be duly provided.

VI. DETAILED DESCRIPTION

This invention ensures a true, correct and complete record of voting via submission of said votes to a globally distributed time-stamping service called a blockchain.

Many blockchain implementations exist. The most widely known implementation of blockchain technology is bitcoin, there are thousands of others but for our purposes they are all the same. A t present the primary use of blockchain technology is as a ledger of accounts showing the flow of currency from one party to another. But the technology which powers it has applicability to any situation in which a transfer of information between two parties must occur, needs to be witnessed and trust cannot be assured.

Blockchains are constructed by design so that neither party needs to trust the other nor find a trusted intermediary.

Voting on the blockchain is such an obvious application of the technology that it was discussed by its inventor Satoshi Nakamoto in 2009 as a possible use and this is common knowledge to the point that no citation has to be made. There are dozens of implemented and half implemented voting solutions out there. Ours is unique in several distinct ways. This patent does not claim to have invented blockchain voting as that would be an obvious use of the technology and therefore non-patentable subject matter. We do claim to secure election results using the blockchain as one tool in a kit of tools. In this scenario the blockchain serves as a database, and a source of ultimate final authority on data transmission between a voter or other user and an election arbiter.

The apparatus provides a record of the vote that is undeniable to any party using the invention. It does so by storing the record of voter intent directly on the blockchain in real time and this record is then distributed globally. There is no direct link between the voter and the vote, however, thus preserving voter anonymity.

How Data is Secured on the Blockchain

Every blockchain works on an idea of "distributed consensus", which means that all parties involved are playing by the same set of rules. If suddenly more than fifty percent of the hashing capacity for a network started playing by a different set of rules, then the network would become unreliable and vulnerable to "blockchain forking", i.e., multiple, competing versions of the "correct" data blocks.

It could be a tempting target for bad actors in positions of power to direct funds at acquiring enough capacity so that they could disrupt the blockchain and change the data record. The three largest blockchains, Bitcoin, Litecoin & Dogecoin, make it cost-prohibitive to disrupt the network in any meaningful way, because they have so much hashing capacity distributed across so many different nodes that only a large nation state would be able to muster the requisite resources and even that grows more and more unlikely with each passing day.

This feature is especially useful in jurisdictions allowing absentee and distance voting. There is no chance of ballots getting lost in the mail. Overseas service members who wish to vote at home can pre-register and receive a customized voting application that is downloaded directly to their personal electronic device. Elderly and disabled people can benefit via the same method. Voter identity would be assured by the same means it is today, no change to the voter registration system is necessary.

This system could easily reduce or replace polling stations. The user could simply show up at the polls with their own personal electronic device, scan a QR code to check in, download the app to their device, and cast their vote all without standing in line waiting for a booth to open.

The most complex and comprehensive embodiment, a full voting system intended to replace existing paper ballot and eVoting solutions is described below. **This invention is end-to-end verifiable and the only system in the world to date be created as such.**

Verifying Voter Identity

Ensuring that Only Properly Registered Voters Can Vote:

The voter registration and check-in process is usually dictated by state and local law, we do not seek to supplant that. Although specific implementations of the invention may enhance it. The current process appears to work fine for the most part as is in the USA.

However, not every implementation will be in the USA and a significant portion of our system does expect something along the lines of American style for this and so it is important to call it out as an expectation; this is why it is elaborated further.

A voter's franchise, i.e. their right to vote is determined according to law and custom. Usually a voter presents identifying documents, fills out a form and receives a voter registration card. This card contains a unique identifying number called a "voter id". At this point the voter is now added to the official roles of the precinct & location in which they are to vote. This is called pre-registration. Some locales also allow registration at the polls, this of course complicates the system. Therefore, we will address the complications as we go along by highlighting the term "poll registered", when it only applies to registration at the polls, otherwise the elaboration will include both pre & poll registration.

It is important to note the following terms.

State, is the organization tasked with creating and tabulating the ballots.

District, is the largest sub organization below state, concerned with creating, distributing and tabulating ballots.

Precinct is the smallest sub organization below state to have it's own unique ballot even if the difference is only a single item.

Polling location is the physical place where the poll is being conducted.

For our purposes, we map a flow of State -> District -> Precinct -> Poll.

In practice there may be any number of intermediary steps and in practice many of these roles are combined. We call them out as separate roles in order to differentiate the systemic expectations.

The actual act of performing the tasks involved may be handled by anyone with proper authority as given by local, state, federal laws etc. These terms have no legal meaning and are here for convenience purposes only.

Ensuring Accurate and Transparent Tabulation of Results

Note: Actor for our purposes is anyone in the chain of State, District, Precinct, Poll.
Tabulation of the results is a common location for difficulties and, in some cases, wholesale fraud, by both governmental and non-governmental actors.

This invention records votes on the blockchain, thereby preventing ballot stuffing.

There are a number of ways to do this, but the simplest is to permanently record a message by utilizing a system of reconstructed destination addresses and sending a very tiny amount of data to them which indicates the choice made by the user. This method has been discussed since 2009 and is neither new nor novel.

What is novel is the method of accounting and tabulation.

Under the commonly known method, people literally vote with their blockchain wallets, allowing anyone with sufficient resources to rig an election by sending multiple transactions to the destination address for the preferred candidate or option.

The Invention prevents this by assigning each actor a master wallet. Prior to the start of the election the funds flow from State to District to Precinct to Poll and finally to Machine. Every wallet before Machine is called Actors wallet for our purposes since it doesn't matter so much who funds the machine, just that we are able to know the source of all funds.

The machine has it's own distinct wallet address. Ideally this address will be constructed according to m of n PS2H semantics. One key will be the machine's own public-key and a secondary key should be the public-key of the authority.

Note: This could be exploited to fake votes, so we address this with a filter at tabulation time. The reason for the semantics of the key construction here is to allow for a device to be replaced and voting to continue.

The device can be easily replaced, but the act of doing so becomes very immediate and public knowledge and thus it serves as a check against machine tampering while still allowing the machine to be serviced if necessary.

Note: "Funds" for our purposes mean specifically whatever is required by the underlaying blockchain technology to enable recordation of the transaction on the ledger. In Bitcoin parlance this would be bitcoins, Litecoin parlance would be Litecoins, Dogecoin would be Doge. Whatever it is called is irrelevant, it is merely an transfer of authority to write an entry in the shared public ledger.

As stated, prior to election start, the authority broadcasts sufficient funds to the voting machine's wallet to cover the expected number of voters.

This provides us with a chain of authority proving that the machines performing the task of voting are known by the authority and authorized to do so, since it is trivial to trace this back to it's original source.

When the voter begins interaction with the voting application an internal identifier termed an nonce (number used once) is incremented.

This nonce is encoded into each transaction and used to differentiate between voters who are interacting with the machine while still maintaining a strong guarantee of voter anonymity.

The actual transaction created reads approximately

Where...OPNULL = BYTES[0..MAX]

VIN:REQUIREMENTS(whatever is needed to spend VIN with VIN being one or more source transactions)

VOUT:ADDRESS:AMOUNT(VIN.AMOUNT - FEE)

OPNULL:[n..,i,o...]

n.. = nonce an unsigned integer (ideally 32 or 64bit, however size is irrelevant as long as the voting machine and the tabulator agree to the same number).

This is used to distinguish which voter is using a machine. It also serves as a check against an external compromise. This is because the machine is only aware of which nonce it used last. Should the machine become compromised then n.. will either be non-unique or non-contiguous.

This is a direct indication of a malfunctioning or machine or voting fraud and so a repeated or non-contiguous n should be rejected and investigated.

i = item, The specific ballot item being voted on in that transaction. n+i should always be a machine unique number, As the machine iterates through i and completes a ballot, n should increment only when a new voter is using the machine.

o... Options. Typically this will be a single byte since a single byte can represent 256 options 0..255.

Option 0 should ideally be reserved for a "declined" option in order to properly encode "false", but this can be left to the specific implementation without impacting the overall design.

There are two cases where option being a single byte will not be true.

The first would be n of m votes. (e.g. Which 3 of these 5 candidates do you want to elect to the school board).

The other would be write in candidates in which case the candidate name would be encoded as text.

An implementation specific detail which makes this easier is if the tabulator understands that bytes following o[0] = 0, represent a text field that is bound to the specific limits of the underlying blockchain and the locale of the authority.}

e.g. If the ballot is in the USA the encoding would be en-us which corresponds to UTF8 or ASCII. If the ballot is in China then the character-set encoding zh or whatever. The specifics of how this encoding occurs can be divined, but ideally should be spelled out specifically in the meta-data of the ballot itself. To facilitate this and numerous other options such as accessibility it is strongly recommended although not required that the ballot be encoded according to the proposed votescript standard which is covered in a later section of this.

The key thing to remember is that the o bytes represent the voter's actual input in some form or another.

Vote tabulation occurs by following the flow of funds as follows:

* Step 1, the voting authority releases funds to the district.

* Step 2, the district releases funds to the precinct.

* Step 3, the precinct notifies either the district or the authority as to the machine's address by releasing said funds to the voting machines.

* Step 4, the voter votes and, upon clicking submit, the Invention sends a continous chain of transactions representing the voter's choices. The VIN of the next transaction should be the VOUT of the previous transaction. If the machine runs out of funds it should be taken out of service for inspection. If it fails, then it should be drained using the secondary key. No matter how it is taken out of service, the nonce should be reset as part of the "restore to service" procedure.

While it is not strictly necessary for the purposes of our patent, ideally any maintenance should be logged and if possible, the fact that the machine is to be reset, should be broadcast publicly. At a minimum the tabulator needs to be informed of the service so that it can expect the reset nonce and account for it accordingly.

At this point we have almost everything we need.

A tabulation of results can be computed live in real time simply by calculating the flow of transactions.

We propose a *de facto* standard that specifies...

"Initial funding flows from Authority to machine. This may or may not involve any number of intermediaries including 0, but at some point the public address of each machine is to be known and published publicly along with the ultimate funding authority transaction. From this point the voting machine reacts to each voter interaction by creation of specifically structured transactions in which funds from the previous transaction are referenced and sent back to it's publicly known address.

The voter's individual selections are encoded as meta-data into the transactional space using OPNULL or other similar structure designed to hold data."

Now we have something that anyone can independently implement and use to tabulate and verify the election results in real time.

Securing the Inviolability and Permanence of Tabulations and Defeating Vote-Counting Fraud.

It would require a concerted effort on the part of the voting authority, the district and the precinct in order to ballot stuff. If, in the event that this occurred, it would be uncovered quickly because the machine has no way to know which nonce has been used, and would create a non-continuous with public record(s) nonce. It is further secured by the fact that for a vote to count, the funds from the previous transaction be spent in the subsequent transaction, thus forming a transactional chain that would break if for some reason it's key were compromised and thus it would immediately exit service because the voter's choice would be rejected by the rules of the blockchain network.

Though it is highly unlikely that any poll worker would have the necessary access or skill to subvert the processes described, an additional check can be put in place to prevent any well-positioned election officer, or a determined attacker such as a hacker, from tampering with the process.

Each precinct contains a known number of registered voters. This means that the amount of funds representing that unique number of voters can be provisioned ahead of time. The implementor can utilize "time locked" transactions that do not release funds until after the polls are opened. This prevents before and after hours tampering of the machines, leaving only "voting hours" tampering which would ideally be caught by any independent observers.

Because funds would be time-locked and not released, any precinct that attempted to ballot stuff, intentionally or otherwise, would quickly find those transactions rejected by the network because not enough time had passed.

One possible option might be segregation of the voting machines. For example a number of voting machines are kept out of circulation and stored in a warehouse or something, where someone sits and casts votes on them. This is the reason for having the polling location broadcast publicly the wallet address of each machine as well as it's booth number. With this anyone can see that all machines are accounted for and only accounted for machines are voting.

This obviously does not cover absentee ballots, however absentee ballots can be covered with a similar system. The primary difference being that there will always be a limited number of absentee ballots permitted and by definition an absentee cannot register at the polls. Therefore the absentee is given their own copy of the voting application on a personal device. They effectively become their own polling booth.

This works well and maintains strict anonymity as long as their device is marked as being under an "*absentia*" list and only allowed to vote a single time, by the consensus rules. This means that it is pre-registered to the official device list to the precinct for which it is assigned.

The only drawback might be a compromise of "strict" anonymity if the absentee voter is the sole absentee in their area. This is statistically unlikely and as long as there are two or more absentees in a single precinct, there is no way to match a single voter back to a single vote as long as both machines are marked as absentee.

The absentee's wallet is then downloaded only at the time they actually check into the device to make their vote. At this point, the device would broadcast a special "*absentia* checkin" message, structured like the voting message except that all bytes including the nonce are set to 0. In areas where the voter id is unique but not a matter of public record, the voter id could be used here except that this has the potential to violate anonymity principles should there be any point at which the voter id leaks or otherwise becomes public.

Governments' changes of practices and hardware greatly lag the pace of change of technology. Current voting technology is hardly cutting edge, and in many cases is so easy to tamper with that hackers have inserted Mickey Mouse and Donald Duck as "official" candidates in some compromised election systems. In hotly contested elections where the vote may be decided by a small number of voters, and where an incumbent may have undue influence on the election process, it is important for citizens to have an independent check and verification process, while maintaining an assurance of strict anonymity.

Invention A: Votescript standard for ballot encoding.

We've discussed at great lengths the way our invention may secure a vote and by what methods. However validation of the vote, both by the voter and by the public at large is a key piece of the puzzle. In order for that step to function, the voting machine, the tabulator and the auditor (anyone who wishes to do an independent count) must speak the same language.

What is described below is one possible language that we have created to accomplish task.

It is called Votescript and it is a structured subset of the well known Javascript Object Notation (JSON) encoding used for data transference using human readable syntax.

A votescript ballot is created by the precinct or other authority who is tasked with the act of doing ballot creation generally.

In other words the person tasked with ballot creation would not change, just the tool that they use to do it.

Votescript specifies that a ballot contains

(order is not important and unlikely to be enforced, they are numbered here for illustrative purposes only).

#1 id : a globally unique identifier for the ballot

#2 rev : a revision number

#3 encoding : the encoding to be used for the character set of text fields.

#4 lang[] : the languages supported by this ballot, en, fr, es. This is an ISO standard already, so we just call it out specifically so the device knows what to show the user when they select language. At a minimum 1 language has to be selected.

#5 items[] : an array containing ballot item objects which follow the format shown in votescript.items

#6 replaces : the id of the ballot it is replacing (allows for change control)

#7 time-stamp : a numerical time-stamp in unix time-stamp format

#8 hash : a unique hash of all the bytes of the entire ballot values (sans structural bytes).

#9 pubkey : public-key of the authority who created the ballot

#10 sig : a digital signature of the signed hash, signed with the private-key that matches pubkey

In JSON for an election in 2016 in California district 1, this looks like:

```
{id : "us.ca.1.2016.11.12",//this can be anything unique it is for internal tracking purposes only
rev : 1.0,encoding : "UTF8",lang : ["en.us, es.es"],
items : [], //see votescript.items
replaces : null, //would be id & rev of previous, again this is only for tracking purposes only
time-stamp : 1234567890, //time created in unix format
hash : (SHA256 hash of a concatenated string containing
encoding+lang+items+replaces+time-stamp)
pubkey : authority's public key
sig : an ECDSA signature of the hash bytes}
```

It is important to note that hash can and should, replace id wherever possible once publicized, hence id & rev are subject to flexibility for construction purposes. This is because the process of creating a ballot usually involves several people in disparate locations and a ballot is completely unique per precinct if there is even a single item of difference.

Therefore replaces should only be used if there was an error in the published ballot (similar to a printers error) and a change was made that would effect the underlaying hash value. However id and rev would change a lot during the construction purpose and there is no need for final sign-off until the very end.

It is also important to note that once published, individual items are incorporated by their reference hash.

If everything has gone correctly, this produces a smaller ballot, the item reference hashes should correspond to publicly known objects and since the hash is computed directly from the bytes of the object itself, it allows distributed distribution of the ballot.

For example, news media & public interest groups could easily republish the ballot and items directly on their own websites.

If a school were used as a polling location that school's website could be used to serve the ballot to machines placed within it's halls.

This effectively neuters any Denial of Service Attacks since the device could Google the ballotś hash and each item's individual hash as a final fallback and once the device has this information it would just keep it local because if it changes the hash changes and it should never change once published, if it does there is a procedure to address that.

It's trivial to compute the hash, but only if you have all of the exact bytes.

As mentioned earlier, it forms a type of tamper proof seal.

Votscript.items are formatted according to the following rules.

#1 id : a globally unique identifier for the ballot (pre-publish tracking purposes only)

#2 rev : a revision number (see above)

#3 image : an image which is optional, but an example be a something which denotes the scope (federal, state, local) of the decision such as a seal. base64 hex encoded image aka dataurl (this should not be an href url since there is no guarantee that a web-server could handle the traffic)

#4.. title.lang : the title as given in the languages supported by the ballot, each language would have a unique title.lang

#5.. desc.lang : the description of the ballot item, as above each language would have a unique desc.lang.
#6.. extra.lang : a link to additional information, this is optional but included so that the device may present the information to the voter if it is relevant. Ideally each language supported would have a link of it's own. Again this is a base64 encoded dataurl, because we cannot guarantee any web-server could withstand the load.
#7 options[] : an array of options encoded as follows in Votescript.items.options
#8 min_max[] : Minimum and Maximum selections, usually this will be 1,1 but a school board might be 3,5 if set to [1, options.length] then write ins are allowed. If you want any or all, so called m of m then [0, options.length -1].
#9 hash : this serves the same purpose as hash in the overall ballot, but allows each precinct to use the master ballot from the state as a template while maintaining local control of the particulars of local items. The individual composing the master ballot may then incorporate the item by referencing the hash. Unlike the master ballot, a pubkey and sig line are allowed but not required on a per item basis.
This means that generally speaking an individual item looks like

```
{id : "us.ca.1.2016.11.12.1",rev : 1.0,image : BASE64DATA,
title.en_us : "President of the United States of America",
title.es_es : "Presidente de Estados Unidos",
desc.en_us : "Whom do you select for President of the USA?",
desc.es_es : "¿Elige tu persona por El Presidente de Estados Unidos?",
extra.en_us : BASE64DATA,
extra.es_es : BASE64DATA,
options : [], //see votescript.items.options for more details on this field
min_max : [1, options.length], // Allows for any single option, or a write in candidate
memo_allowed : [true | false],
hash : (SHA256 hash of a concatenated string containing
image+title+desc+extra+options)}
```

A major advantage of this approach is that it allows small communities to easily add, edit and approve local items to local ballots via whatever process they choose and the only thing the voting authority needs to know is the final hash in order to add the item to the ballot.

votescript.items.options

#7 in votescript.items is options which is an array containing 1 or more options.

Options are by far the most complex item to encode, as with items the only thing actually incorporated into the ballot is the hash of the relevant bytes. However options can present themselves in a number of ways and this presents a challenge for the author of any encoding package.

To simplify this an option presents with the following fields.

```
{id : "us.ca.1.2016.11.12.1",rev : 1.0,
image : BASE64DATA,
title.en_us : "Candidate #1,
title.es_es : "Candidate #1",
desc.en_us : "Candidate #1 is the Republican candidate, formerly governor of... ",
desc.es_es : "Candidate #1 es Republicrate, prev el gobierno de...",
extra.en_us : BASE64DATA,
extra.es_es : BASE64DATA,
hash : (SHA256 hash of a concatenated string containing
image+title+desc+extra+options)}
```

In the case of a write in option, the title, desc & extra field are simply something along the lines of "write in" and we add an "is_memo" Boolean field to the object.

It is up to the voting machine to encode all fields including "write in" correctly, and it is up to the tabulator to decode all fields including "write in" correctly. This is why every option, item and ballot has a hash value and they are incorporated by reference. It prevents anyone from asserting that the script is encoded incorrectly.

Incorrect encoding would effect the encoding and decoding of the votes submitted to the blockchain. Part of the hash calculation should include a pre-flight check that ensures that if an option field has "is_memo" then the parent "item" object also has a memo_allowed field and that said field is set to true. Also the reverse should be validated, if an item has memo_allowed then it needs to have at least a single "is_memo" field.

Keep in mind that most blockchains will severely limit the number of bytes allowed by OPNULL or MEMO class transactions. Bitcoin only allows 32 bytes and between 2 & 4 of these bytes is already taken by the nonce, leaving about 28 bytes. This is still longer than most people's First Middle and Last Names combined. However many countries specify titles and additional names.

If the implementor is in one of these countries and tries to encode the full name it could easily run more than the max allowed bytes. Also some countries rely upon UTF16 and other 2 or more byte encodings. This is the same problem, the byte field could be insufficient for encoding the memo. Therefore if that is a possibility, truncation is not a valid option.

Instead an object corresponding to the following should be created by the voting application

```
{ballot_id : ballot.hash,
item_id : item.hash,
option_id : option.hash,
nonce : (machine's voter nonce value),
option_mimetype : (mimetype),
option_value : (user specified data),
option_hash : (SHA256 hash of .._id+nonce+mimetype + value),
pubkey : (public key of the machine)
sig : (signature bytes of hash)}
```

At this point, ballot_id+item_id+option_id are concatenated in order to produce a public/private key pair.

The object is encrypted using the public part of the key and the message is submitted to an out of band messaging service such as bit-message. From there it circulates encrypted until it drops out, this is anywhere from 2 days to 2 weeks depending on the service selected. Ideally it should circulate for at least the number of days the polls are open.

In the meantime the tabulator should know to watch for a transaction where the OPNULL or MEMO corresponds to as many bytes of the option_hash as can be reasonably accommodated given the current n..,i,o semantics.

In this case option_hash is what will fill in the o bytes up to the lesser of hash.length or field.length which will mostly be up to individual blockchain selected.

Once the tabulator sees that, it will provision a write in field and scan the out of band message queue for a message that decrypts given the private key that can be derived directly from the process above and for which the message contents match the expected format. It should validate that the pubkey in the object, matches the pubkey from the blockchain transaction and that the signature also correctly matches. If that is all true then option_value will represent the user's choice and it should be tallied accordingly.

This also allows an independent implementor to watch without any ability to interfere since the private key is derived and it is only used for decryption purposes and the message hash is stored on the blockchain.

It becomes quite a simple matter for anyone, anywhere, skilled in the art of programming to implement, so long as the authority publicizes all the relevant parameters such as which out of band network is selected so this should be part of the final votescript ballot.

It also prevents anyone from stuffing the ballot or otherwise altering it no matter how much information the authority discloses about the internal operations of the vote creation and tabulation process.

For the both the tabulator and the voting machine designer, the important fields to pay attention to are ballot.item.min_max since that directly effects how the vote is encoded and to a lesser extent where to find the actual vote data.

The memo_allowed and is_memo fields are also very important as well. Without them there is no reason to expect memo data to exist.

If min_max is [1,1] then only 1 option may be encoded and there is no chance that it will be a "write in" option.

If min_max is ANYTHING else, then o will extend greater than 1 byte past the n,i bytes and the implementor should pay close attention to this. Testing for this condition should be checked in order to certify to the standard.

[1,options.length] the voter can choose ANY & ALL option(s) but a minimum of 1 item (remember options.length + is_memo must be present in order for the field to constitute a memo, otherwise this is a an array of 0 & 1 true, false, no,yes for each option and the option selected is set by offset].

[m, n = up to options.length - 1] the voter can choose ANY OR ALL options, but no write in is allowed.

Again m is the MIN and n is the MAX, so "any 3 of 5" would be [3,3] with 5 options given, not [3,5]

Ideally this will be encoded as 0 or 1 per byte with the byte offset corresponding to the option offset.

Now one final note about this standard. It allows for more than text to be stored in the memo field. In truth photographic evidence, audio evidence, even video can all be encoded this way. The hash is stored on the blockchain, there is a 1:1 mapping between authorized device and option data but no direct link to any particular person. The data is public yet encrypted unless someone is specifically looking for it and if the key (which can be derived easily enough) is published, by for instance a watchdog group then the whole encoding system functions as a publicly visible Drop-Box that cannot be disputed. Thus making fields compliant with the memo aspect herein is all that is needed to change from any previous embodiment to Embodiment 6.

It can be taken a step further as well. Any party can easily record 100% of traffic going across the network during the polling period. They could then upload that record to freenet or bittorrent and publish the relevant keys & transaction from the blockchain, thus providing an unassailable record of the vote, one that cannot be censored period.

***

Preparing ballots becomes a task that can easily be performed with a simple text editor, or custom tools can be created by any web developer.

This allows exit surveys and the other embodiments to serve as a genuine check against election returns.

Pollsters can give their clients, such as news agencies, links to a live portal where the clients can see the polling results in real time.

Final official tabulation:

Final tabulation is straight forward. All vote data is encoded the blockchain, the exact shape of which is defined by the standard selected. We use our proposed votescript as an example. However the tabulator really only needs to know which devices are allowed to vote. Everything else is encoded in the transaction or can be derived from data that should be checked anyways.

From there it just watches the blockchain for any transactions involving the already published machine's wallet.

Since a given wallet, will only be voting a single ballot and a ballot is defined as any set of items and options wherein there is at least a single unique item, and because each item and option has a specific hash tabulation is dead simple.

It means that every possible option has an absolutely unique identifier composed of ballot.hash+item.hash+option.hash

we call this number the bio.hash value.

Therefore to fit this patent a tabulation machine need only...

Watch blockchain for wallets (w..),

```
When ANY wallet w receives a transaction, examine transaction source.
If original source of funds is the authoritative master wallet continue, else ignore
Examine transaction
If nonce is a contiguous integer increment from previous nonce for w, and w+nonce+item is unique, continue, else alert BAD DEVICE ERROR & break
if bio.is_memo
extract hash
compute key
decrypt memo
create new option space if necessary
increment vote count on bio.memo.hash by 1
else increment vote count associated with bio.hash by 1
```

That's really all that's needed on the tabulation end. Votescript is a subject claim of this patent and we would love to see it become a global standard. However the usage of votescript or a variant thereof is not strictly necessary for this patent. So long as the ballot creation system, voting machine and tabulation service or devices all speak the same language, the specific semantics of said language are irrelevant and are only called out here for example purposes. Our invention relies on the cryptographic techniques called out herein and the features inherent in public cryptographic networks as well as human factors to secure a true, free and fair vote for all.

NIST FIPS 186-4 ECDSA standard - http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

[1] https://www.verifiedvoting.org/wp-content/uploads/2015/02/Principles.pdf

[2] https://www.brennancenter.org/publication/americas-voting-machines-risk

[3] http://www.citylab.com/design/2015/09/mapping-the-nations-failing-voting-machines/405724/

[4] http://gizmodo.com/200693/how-to-steal-an-election-with-a-diebold-machine

[5] http://www.ne.anl.gov/capabilities/vat/

[6] http://www.salon.com/2011/09/27/votinghack/

[7] http://avirubin.com/vote.pdf#search=%22electronic%20voting%20tampering%22

[8] http://www.votetrustusa.org/index.php?option=com_content&task=view&id=1958&Itemid=162

[9] https://en.wikipedia.org/wiki/Hacking_Democracy

[10] http://elections.cdn.sos.ca.gov/taskforce_report_1.pdf

[11] http://www.theguardian.com/us-news/2015/apr/15/virginia-hacking-voting-machines-security

[12] https://en.wikipedia.org/wiki/Florida%27s_13th_congressional_district#2006

[13] http://www.heraldtribune.com/article/20061109/NEWS/611090343

[14] http://www.verifiedvoting.org/verifier2014/

[15] http://www.verifiedvotingfoundation.org/projects/electronic-voting-resolution/

[16] The formula for calculating the fee is located here: https://en.bitcoin.it/wiki/Transaction_fees

[17] https://en.wikipedia.org/wiki/Circumstantial_evidence