ON THE PRIVACY THREATS OF ELECTRONIC POLL BOOKS

Stefan Popoveniuc
KT Consulting, Gaithersburg, MD, U.S.A.

John Kelsey
NIST, Gaithersburg, MD, U.S.A.

Eugen Leontie *
The George Washington University, Washington, DC, U.S.A.

Keywords: Voting, Privacy, Blind signature.

Abstract: Electronic poll books can rapidly check the eligibility of a voter due to their ability to quickly search lists. However, they also introduce a factor of concern: if the electronic poll book records the order of sign-ins and the voting machine or optical scanner records the order in which the voters cast their ballots, ballot secrecy can be compromised. Worse, if the time at which each voter signs-in and the time at which each ballot is cast are recorded, ballot secrecy is lost. It is surprisingly difficult to avoid saving such timing information, for example in event logs, and even more difficult to verify that no such information is saved. In addition, due to operational complexities, even the more efficient electronic poll books can act as a bottleneck in the voting process. We propose a simple technique to address these concerns, by allowing voters to sign-in from home, and print out a bar-coded ticket to be presented at the check-in table. Using blind signatures, this ticket need not reveal information on the voter's identity to the check-in table at the precinct. The ticket proves than the voter is authorized to vote on a particular ballot style without disclosing her identity.

1 INTRODUCTION

Most often legislative regulations (such as the Voting System Standards (vot,) and Help America Vote Act of 2002) impose strict requirements for the secrecy preserving properties of an election system. These define confidentiality as a general requirement of the voting system to "protect the secrecy of the vote such that the system cannot reveal any information about how a particular voter voted"(vot,). Protecting the confidentiality of the cast ballots has become increasingly difficult with the introduction of electronic voting equipment. Either by design, or unintentionally, computers keep time stamps of many of the operations they perform, like the creation, modification, or access of a file or a database record. Even the most diligent programmers have a difficult time writing code which is supposed to lose all traces of the order in which some operations happen. This is increasingly difficult because programmers typically reuse a large amount of previously compiled code (like the underlying operating system, the file system, etc.), and because, for debugging purposes, programmers have been taught to keep detailed logs for all the operations that their code performs.

If manual, i.e. paper-based, poll books are used, it is easy not to record the order in which the voters checked-in. The check-in judge should only make a check mark next to the voter's name and should not keep a separate log of the time when the voter arrived. Manual poll books are losing popularity in favor of electronic poll books, which have a series of advantages: no paper is used to print the poll book; the voter can go to any check-in judge (there is no A-M, N-Z division), or even to any precinct; last minute updates to voting roles are easier; and last but not least, there is a trend of doing everything electronically.

The precise order in which the voters arrive at the polling place is likely to be recorded by the electronic

*Work was made possible in part by grants NSF CNS-0831149, NSF CNS-0934725l and AFOSR FA9550-09-1-0194.

Popoveniuc S., Kelsey J. and Leontie E..
ON THE PRIVACY THREATS OF ELECTRONIC POLL BOOKS.
DOI: 10.5220/0003520202810286
In *Proceedings of the International Conference on Security and Cryptography* (SECRYPT-2011), pages 281-286
ISBN: 978-989-8425-71-3

poll books at the check-in table. This information can (but should not) be used in conjunction with the order in which ballots are cast, which may be obtained either from the optical scanner or from the DRE machines (hand counted paper ballots do not suffer from these time stamp issues). Having the two orders, every cast vote may be easily traced to a certain voter, or to a small number of voters, violating the secrecy of the ballot.

The obvious approach to preventing this linkage is to control the information stored by the electronic poll book and by the voting machine or optical scanner, and to restrict what information may be output by those devices. This is a sensible precaution, but it is hard to do well, and even harder to verify. Even if the specifications state that no machine shall record the order of check-ins or votes, and even if the standard reports generated by the machines are always in alphabetical order and contain no time or ordering information, it is all but impossible for any observer to know whether such information has been stored inside the machines, and might be accessed later to reconstruct how everyone voted. When the electronic poll books must be online, this becomes even harder. As with many situations involving privacy, it seems better not to collect the data in the first place than to collect it, and try to keep it secret.

In electronic voting systems, there are two ways to prevent the voting system from ever having enough information to link voter identities to ballots: hide the vote, or hide the voter. When hiding the vote, the voting machine does not get to see the clear-text vote that the voter wants to cast, but only an encrypted version of it. Designing such ballots is possible, as proven by systems such as Prêt à Voter (Chaum et al., 2005). Since only encrypted ballots are available to the voting machine, they cannot provide a tally at the end of the day. Moreover, hand countable paper ballots may not be available, and the only way to tally the votes may be to decrypt them by a special mechanism (e.g. a mixnet or homomorphic tallying - see Section 4).

This paper focuses on the “hide the voter” approach. The main idea is that voters check-in from home, and they get an anonymous credential that is used as an entry ticket at the polling place. Since the order in which the voters check-in from home is presumably different from the order the voters cast votes at the polling place, and since the identity of the voter is not available to the electronic machines at the check-in table, the correlation between the voter’s identity and the ballot she casts is lost.

2 PROPOSED TECHNIQUE

Inspired by the airline industry, we suggest separating the steps required for validating voters. In the first step, within a given time frame before and during the election, voters are allowed to go to a designated web site, type in their name and address (or username and password, etc), see if their voter registration status is valid, and print a check-in card that has a bar code on it, similar to an airline boarding pass. The card may contain the name of the voter and her address, along with a unique token which is digitally signed. The same party that manages electronic poll books can issue the check-in cards.

The second validation occurs when the voter gets to the polling place. She presents her check-in card, which is scanned by a bar code reader. The check-in card allows the bearer to cast one vote. The reader checks to see if the digital signature from the bar code is correct, and if the same card has been scanned before. The check-in judge may ask the voter for a photo ID, check that the information on the check-in card is consistent with the one on the ID, and check that the voter looks like the picture on the ID. The voter’s identity is not available to the electronic device at the polling station. We assume the check-in human judge has limited memory and does not keep written logs of the people it sees.

The scanners do not need to be connected to a central server (but they be if preventing of double voting is desired - as opposed to detection). The bar codes can be be explicitly linked to cast clear text ballots. At the end of the day, when all data is uploaded to a central server, double spent bar codes (at different precincts) can easily be detected and the cast ballots eliminated (however, linking double votes to a human voter is impossible). The server can cast her ballot at any polling place where her bar code can be read.

2.1 Blind Signatures

Anonymous credentials allow users to authenticate themselves in a multi-party environment. A cryptographic token issued by one party can be presented by an individual to another party as proof of her identity or recognized as an authorization to perform an action. A simple way of obtaining an anonymous credential (a token) is by the use of blind signatures (Chaum, 1982). A token consists of a random number that the voter generates, long enough such that it is unique (e.g. a 128 random bit token is unique with high probability), along with some other information (section 2.2).

The token is used in a blind signature algorithm

that does not allow the voter to derive a second signed token, similar to protocols used in electronic cash(Chaum, 1982). For example, a token t is randomly generated by the voter and a collision resistant one-way function $h(t)$ is blinded by the voter and sent to the authentication server along with the voter authentication credentials (e.g. name and address, or username and password).

The server checks the authentication credentials and, if valid, signs the blinded value and marks the voter as being checked-in. Depending on which ballot style the voter is assigned to, the server may use a different private key to sign the blinded token. The server does not have access to the value, since it is blinded. This offers information-theoretic protection. The server sends back to the voter the signed, blinded value. The voter un-blinds it, and obtains $h(t)$ which is now signed. She checks that the signed value is $h(t)$ she sent to the server and that the digital signature is correct. The voter prints the signed value and the token t, and brings them to the check-in station at a polling place during voting day. The signed token is scanned and the voter may be asked to provide some form of identification (e.g. a government-issued photo ID). The check-in judge checks that this is the first time the token has been presented (to prevent the reuse of the token), that the digital signature is valid and that it corresponds to the precinct and ballot style assigned to the voter's address.

Definitions: $\mathcal{M}$ is a large message space, the voter chooses tokens in this space with a high probability of being unique and $\mathcal{R}$ is a finite set of random strings as required by the blinding scheme:

- h is a collision resistant one-way hash function.
- $sig, valid, (pri, pub)$: a public/private key signature mechanism such that $s = sig(pri, h(m))$ does not leak knowledge of pri and $valid(pub, s, m) = true$ holds if s was derived from m and pri.
- $b : \mathcal{M}\, x\mathcal{R} \rightarrow \mathcal{M}$ is the blinding mechanism. It is used in conjunction with its reverse function unblinding $ub : \mathcal{M}\, x\mathcal{R} \rightarrow \mathcal{M}$. Given $b(m,r)$, one can not infer anything about m, but used in conjunction with the signing mechanism, one can compute $ub(s(pri, b(m,r)), r) = s(pri, m)$ and it is not possible to derive $s(pri, m)$ from $s(pri, b(m,r))$ without r.

Here are the steps taken by the voter to check-in:

1. The voter chooses 2 random values: $t \in \mathcal{M}$, and a random blinding factor $r \in \mathcal{R}$. She computes $h(t)$, then blinds the hashed token $\sigma = b(h(t), r)$; she sends σ to the check-in server.
2. If the check-in server validates the user credentials, it signs σ and, sends $sig(pri, \sigma) = sig(pri, b(h(t), r))$ back to the voter.
3. After unblinding the signature, $ub(s(pri, \sigma), r)$ the user has the signed token $\rho = s(h(t))$. Using a local application (such as a browser plugin), the user prints the check-in ticket (ρ and t) using electronically recognizable markup (like a 2D barcode) .
4. At the polling place, a scanner reads ρ and t, checks $valid(\rho, pub, t)$, checks if the token was used before, and records t.

The check-in server that signed the tokens has access to the order in which the voters check-in from home, and the electronic poll book at the polling place has access to the order of the signed tokens, but the two machines cannot match the two orders anymore. The order in which the voters checked-in from home is different from the order they come into the polling place, and the check-in server never got to see the token in clear-text (but only in blinded form). The electronic poll book at the polling place sees the token in clear-text, signed, but never gets to see the identity of the voters. The check-in judge does get to see this identity, but it is not entered into the electronic poll book.

The private key that is used by the server to sign the blinded token is unique to the ballot style belonging to the voter. A different private key is used for each ballot style. Since the server has access to the complete identity of the voter, it can easily identify the ballot style corresponding to that voter, and thus use the appropriate private key.

If a voter loses her signed token, she must provide either the token or the blinding factor to the sing-in server, so it can obtain the hash and put it on a blacklist. For situations in which the voter loses her signed token, forgets the token and the blinding factor, an additional recovery mechanism must be developed. For example, the voter may distribute the blinding factor to a number of trustees using a threshold secret sharing technique, such that only a quorum can reconstruct them. The voter may have to prove (in zero-knowledge) to the sign-in server that she distributed the same values to the trustees.

A possible attack against the simple token construction may involve a coercer that collects valid signed tokens from voters and uses them to cast multiple ballots by himself. The same person comes to various polling places multiple times and presents different authorization tokens that are validly signed. This is possible since the anonymous token is completely independent from the voter's identity, and the check-in judge that verifies the voter's identity and the validity of the token has no way to link the two. The next section presents a specially constructed token that makes this attack impractical.

2.2 Linking user Credentials with Token Generation

We present a special construction of the token that includes some attributes of the voter's identity. The token contains partial information about the voter's identity, not enough to uniquely identify a voter, but enough to have her identity validated by a poll worker.

In addition to a unique random number that the voter generates, the token also contains some incomplete information about the voter's identity. For example, the token can contain the first letter of the last name and the last letter of the first name, or it can contain the sex and an interval for the date of birth, or it can say that the last name contains at least 4 vowels and a "T". This way, the poll worker that checks the validity of the token and the ID of the voter, can also check that the identity of the voter is consistent with the partial information in the token. At the same time, the full identity of the voter is still not available to the electronic system that checks the signed token, and the information that is available is not enough to uniquely identify a voter.

To ensure that the blinded token contains partial attributes of the identity of the voter that is doing the check-in from home, a simple zero-knowledge protocol can be used: the voter is asked to create 100 tokens, each with partial information about her identity. The server receives 100 blinded tokens, and, before signing one of them, the server asks the voter to un-blind some random 99 blinded tokens, and checks that all of the opened ones contain partial information about the voter's identity (to which the check-in server has access to). The server can be fairly sure that the un-blinded token which was not opened contains partial information about the same voter. An attacker that presents one identity in clear-text to the server, but includes partial attributes about another identity in the blinded token will be detected with probability 99%. It is easy to see how this probability can be increased to a value as close to 1 as desired.

Thus, the token contains partial information about the voter's identity, and the poll books at the polling place do get access to this partial information. However, this information should be common to a number of voters, such that the check-in scanner cannot uniquely identify the voter. To be able to sell her voting credential, a voter would have to find another person in her jurisdiction for which this partial information on her token fits with the identity of the fraudulent voter. It is up to the legal framework to decide how to deal with both the voter who gave her token to somebody else, and with the person who tries to use someone else's token.

Another approach to detecting an illegitimate use of a token, is to minimally change the protocol by asking the voter to print on a second page the blinding factor used in the blind signature protocol. At random (or if suspicion arises), the check-in judge may ask the voter to provide her blinding factor. The barcode scanner would read the barcode with the blinding factor, contact the signing server, and obtain the identity of the voter from the server. The server can obtain this identity, since it has the clear-text token and the blinding factor, and the voter presented her credentials along with the blinded token. Thus the check-in judge can check the identity that was presented to the server for this token, against the identity of the voter who tries to use the token. A mismatch would trigger an alarm and both actors for this fraud may suffer consequences.

Only a small fraction of the voters would have their blinding factor scanned by the check-in judge. For these voters, the time of check-in and their full identity is available to the voting system. The order of the cast ballots does not precisely correspond to the order in which the voters come to the polling place. Rather, for one of the voters for which the identity is available to the check-in judge, a number of cast ballots in a certain time frame are possible. Thus ballot confidentiality might still be preserved.

3 METHOD ADOPTION

Like with many other security products, it may be difficult to convince voters and election officials to adopt our technique solely on the privacy properties which it offers. This section presents additional practical benefits of our proposal, properties which may be used to convince both voters and election officials that it makes their tasks faster and easier. These properties may be presented as the basic features of the new poll book system, and the privacy enhancements would be transparent to the voter and to the election officials.

Waiting in line is one of the most common complaints of voters. It is not uncommon for voters to wait in lines for up to 4-5 hours. Voters may be discouraged by the size of the line, and decide not to cast a ballot. Reducing the size of the line by expediting the check-in process is highly desirable. Checking-in from home may substantially reduce the size of the lines.

Assume we have a voting system that uses electronic poll books. The check-in process generally goes as follows. The check-in judge asks the voter for her full name and, using a touch screen and a stylus, types the first three letters of the last name and

the first letter of the first name. This usually narrows down the set of registered voters to only a handful of persons. The voter is then asked for the full name, and the check-in judge checks to see if the name is the one that came up on the electronic poll book. The judge also visually checks the sex and the age of the voter. If there is some suspicion that the voter is not who she says she is, the judge can also ask the voter for a photo ID (depending on location providing a photo ID may be optional). The electronic poll book prints a check-in ticket, which is handed to the voter. The voter has to sign the ticket and take it to the ballot issuing table. The voter surrenders the check-in ticket in exchange for a paper ballot, if optical scan is used, or an activation token, if a DRE is used.

This entire process can be time consuming. The check-in table is often the stop which causes lines to build-up. Other stops are the voting booth where the voter fills in her ballot and eventually the scanner where the voter deposits her paper ballot. In our experience, it is common that the check-in table is the only place where there is a line.

Our technique simplifies the check-in process, and, consequently, the waiting lines at the check-in table would be significantly reduced. The voters that check-in from home may have a dedicated line which would encourage more voters to use this faster option. This would be beneficial for voters, for election officials and for ballot confidentiality too.

4 PREVIOUS WORK

Ron Rivest suggested preliminary voting (Rivest, 2005) as a method of shortening the time it takes a voter to make her selection on the ballot. The voter is allowed to fill-in her ballot from home and print out a representation of her choices (e.g. a 2D barcode). Each voter has to go to a polling place and can bring with her a pre-voted ballot. The DRE scans the bar code and pre-fills the electronic ballot with the indicated choices. Even if a coercer forces the voter to bring a certain pre-vote in the booth, the voter is allowed to make any number of modifications and to select her own favorite candidates. Rivest's work is also geared towards eliminating long waiting lines at polling places and is complimentary to our technique.

While Internet voting would certainly eliminate waiting in line to vote altogether, its is hardly a wide accepted method due to its known downsides (coercion and viruses). Since polling place voting is still mainstream, voter registration and verification is still a stringent but unaddressed problem.

Controversies about using computers in the voting process has been reported as early as 1969 (Bergholz, 1969) and secure electronic voting has been proposed by Chaum as early as 1981. Both mix networks (Chaum, 1981) and blind signature (Chaum, 1982) were proposed by Chaum. Both methods suggest voting as one of the application which would benefit from such general privacy techniques, and are still today two of the three general ways of verifiably counting cast votes. The third method, homomorphic counters (Benaloh and Yung, 1986), was first proposed by Benaloh in 1986.

Over the past decade, a series of end-to-end verifiable voting systems (Popoveniuc et al., 2010) was been proposed. We briefly present some of these systems.

Andrew Neff proposed MarkPledge (VoteHere Inc., 2003), a voting system that uses a DRE with a regular printer attached. MarkPledge is based on the subtle observation that a zero knowledge proof is valid only if the prover does not know the challenge a-priori. The system commits to the ballot that the voter is about to cast, and then the voter challenges the DRE to prove that the commitment contains the correct vote. MarkPledge also produces "valid" simulated proofs for all the other candidates, such that the receipt contains valid proofs for all candidates.

Scantegrity II (Chaum et al.,) uses a regular optical scan ballot, with confirmation numbers printed in invisible ink. The voter marks the oval next to the candidate and gets the confirmation number for her vote. The voter only gets to learn the confirmation number for the selected candidates. Scantegrity II asks the voter to write down by hand this confirmation number, which may be a usability concern.

Benaloh suggested a simple way to prove that an encrypted vote is correctly constructed (Benaloh, 2008): present the voter with an encryption and allow the voter to challenge the encryption if she wishes to, and check that the encryption contains her favorite candidates. Helios (Adida, 2008) uses the Benaloh challenge principle and is geared towards Internet voting. VoteBox (Sandler et al., 2008) is also based on the Benaloh challenge, but is designed in the context of polling place voting.

The majority of proposed voting systems, including ones which were used in binding elections such as MarkPledge, Scantegrity II and Helios, but also VoteBox and BingoVoting(Bohli et al., 2007) allow the voting machine to know the clear text choices a voter makes, even if the receipt they provide is privacy preserving. Knowing the choices that the voter made, along with either the order in which the votes were cast or the time when they were cast is a real threat to ballot secrecy, as this information can be corroborated

with the order and times when the voter's checked-in.

Partial redaction of user information is often used in privacy preserving database systems and known as k-anonymity (Samarati and Sweeney, 1998). It requires that enough information is taken out of records such each combination of values occurs k or more times, such that linking personal data with any such records is difficult.

While there has been much progress on the integrity aspect of elections, to our knowledge, our current work is the first to address the secrecy problems related to electronic poll books.

5 CONCLUSIONS

We propose a technique which addresses a confidentiality problem caused by the use of electronic poll books in conjunction with any type of electronic voting machines: the identity of the voters is available to the electronic poll books along with the order in which the voters check-in; the options of the voters are available to the voting machines, along with the order in which the ballots are cast. Matching the two orders may result in binding voters' identities with their selections. To dissociate the two orders, we propose a technique based on blind signatures, with the token used in the protocol containing a small amount of information about the voters identity.

Incidentally, our technique also addresses one of the biggest practical problems for polling places: waiting in lines. Our technique reduces the amount of time a voter spends waiting in line before her credentials are checked and her ballot is issued. Also, our method does not require massive technology replacements, relying mainly on existing infrastructure.

REFERENCES

Voting system standards. Available at http://www.eac.gov/election_resources/vss.html.

Adida, B. (2008). Helios: Web-based open audit voting. In *Proceedings of the Fourteenth USENIX Security Symposium (USENIX Security 2008)*. Usenix.

Benaloh, J. (2008). Administrative and public verifiability: Can we have both? In *IAVoSS Workshop On Trustworthy Elections (WOTE 2006)*, KU Leuven, belgium.

Benaloh, J. C. and Yung, M. (1986). Distributing the power of a government to enhance the privacy of voters. In *PODC '86: Proceedings of the fifth annual ACM symposium on Principles of distributed computing*, pages 52–62, New York, NY, USA. ACM.

Bergholz, R. (1969). How election can be rigged via computers. *Los Angeles Times*.

Bohli, J.-M., Müller-Quade, J., and Röhrich, S. (2007). Bingo voting: Secure and coercion-free voting using a trusted random number generator. In Alkassar, A. and Volkamer, M., editors, *VOTE-ID*, volume 4896 of *Lecture Notes in Computer Science*, pages 111–124. Springer.

Chaum, D. (1982). Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto'82*.

Chaum, D., Carback, R., Clark, J., Essex, A., Popoveniuc, S., Rivest, R. L., Ryan, P. Y. A., Shen, E., and Sherman, A. T. Scantegrity II: End-to-End verifiability for optical scan election systems using invisible ink confirmation codes. In *EVT'07: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop*. USENIX'08.

Chaum, D., Ryan, P. Y. A., and Schneider, S. (2005). A practical voter-verifiable election scheme. In *In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, ESORICS, volume 3679 of Lecture Notes in Computer Science*, pages 118–139. Springer.

Chaum, D. L. (1981). Untraceable electronic mail, return address, and digital pseudonym. *Communication of ACM*.

Popoveniuc, S., Kelsey, J., Regenscheid, A., and Vora, P. (2010). Performance requirements for end-to-end verifiable elections. In *EVT'10: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop*. USENIX Association.

Rivest, R. L. (2005). Preliminary Voting - Prevoting. http:// people.csail.mit.edu/rivest/Rivest-PreliminaryVotingPrevoting.pdf.

Samarati, P. and Sweeney, L. (1998). Generalizing data to provide anonymity when disclosing information. In *ACM SIGACT*.

Sandler, D., Derr, K., and Wallach, D. S. (2008). Votebox: A tamper-evident, verifiable electronic voting system. In van Oorschot, P. C., editor, *USENIX Security Symposium*, pages 349–364. USENIX Association.

VoteHere Inc. (2003). Documentation. http://www.votehere.net/vhti/documentation/verifiable_e-voting.pdf.