

E-Voting: a new approach using Double-Blind Identity-Based Encryption

ABSTRACT

We present a novel approach to e-voting for elections, based on a new cryptographic construction, double-blind identity based encryption (IBE). In standard IBE, the identity string that is passed to a key generation centre (\mathcal{KGC}) during the key extraction phase is visible to the \mathcal{KGC} . Recent work introduced the notion of blinding the identity string, resulting in a private key generation protocol which obscures the identity string from a \mathcal{KGC} . A double-blind IBE scheme is an extension of blind IBE schemes. It allows a user to construct an identity string that is partially obscured, with certain elements visible to the \mathcal{KGC} while enabling the \mathcal{KGC} to add elements to the user constructed string which are obscured from the user.

The contribution of this paper is a new protocol for e-voting using double-blind IBE that adheres to conventional security criteria for e-voting schemes. Our protocol provides end-to-end verifiability. We also contribute a plaintext voter receipt and an audit trail, using receipts.

1. INTRODUCTION

Democratic governments face the challenge of holding elections to appoint successors. Voting is a complex task, made more so by the conflict between the requirements of verifiability and secrecy. Voters and independent observers want to verify the election process occurred correctly, and that individual votes were recorded accurately. However, an individual must not be able to convince a third party that she voted in a particular way.

A cryptographic voting scheme is an election system that provides mathematical proofs of the election results, with the aim of producing a secure and verifiable election. Electronic voting refers to both on-line voting, such as remote internet voting, and physical voting. Physical voting incorporates the use of direct recording electronic machines (DREs) in polling stations [10] and paper-based systems such as Scratch & Vote and ThreeBallot [1, 13]. E-voting has the potential to offer voters the ability to cast votes from a convenient location, increase efficiency and accuracy in vote processing, provide transparency in the process and maximise voter turn out [2].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

We present a novel e-voting scheme that uses a new cryptographic construction, *double-blind IBE*, to generate voter ballots in a privacy preserving manner. Our scheme allows a voter to verify her vote has been correctly recorded in an intuitive and human readable manner. The scheme facilitates auditing in all stages of the voting process. We examine required properties of e-voting schemes and evaluate our scheme accordingly.

1.1 Identity-Based Encryption

IBE was introduced by Shamir [14] in 1984 to allow a party to encrypt a message using some publicly known identity string of a recipient as their public key. The corresponding private key is generated by a trusted third party, who is given the identity string during the key generation phase.

The property of *blinding* is most commonly associated with digital signatures. In a blind digital signature, a signature on a message can be obtained without revealing the contents of the message. Recent work [5, 9] has focused on blinding the key extraction protocol of IBE schemes. This allows a user to obtain a secret key for an identity without revealing the identity string.

We propose using double-blind IBE as a means of achieving key privacy for the user, while allowing the trusted third party generating the key enforce certain restrictions on the public key. In a blind IBE scheme [5, 9] a user \mathcal{U} presents a key generation centre \mathcal{KGC} with a blinded public key id for which it wants a corresponding private key. At the end of the key extraction process, the user obtains a valid private key for id from the \mathcal{KGC} , while the \mathcal{KGC} learns nothing about the id the private key was generated for. The \mathcal{KGC} cannot link this instance of the key extraction protocol with the private key obtained by the user.

In our scheme, we use double-blind IBE to achieve the desirable property of the \mathcal{KGC} entity adding an element to the identity string without revealing it to the user. We are unaware of any other schemes that allow this.

2. PRELIMINARIES

2.1 Electronic voting requirements

Elections are bound by criteria that reflect the characteristics required by legal principles. They must be universal and equal. Universal elections provide equal suffrage, which is tantamount to access to voting for all. Equal elections count all ballot as having commensurate influence on the final election outcome. An e-voting scheme is also expected to satisfy a number of commonly accepted security criteria [1, 2, 7, 6].

Privacy (secrecy) The way in which a voter casts her vote is not revealed to the election authorities or any other party.

Uniqueness Every eligible voter can only vote once.

Coercion resistance No voter can prove to a third party how she cast her vote.

Individual verifiability (ballot casting assurance) Each voter can verify that her vote is accurately recorded.

Universal verifiability Any observer can verify that the recorded tally of an election matches the published tally.

Fairness No partial tally of results may be leaked during the election so as not to bias the election outcome.

In addition to these requirements, we make the following assumptions. A voter will always find a buyer for her vote, should she attempt to sell it. A voter will want the opportunity to verify her vote. Fraudulent votes (such as a voter attempting to vote multiple times) will be detected. Voting is to be carried out in a ‘vote and go’ manner. It is accepted that vote manipulation [8] is part of the election process, and no attempt to prevent it is made.

2.2 Bilinear Pairings and Identity-Based Encryption

Definition 1. Let G and G_T be groups of prime order p . A bilinear map $e : G \times G \rightarrow G_T$ satisfies the following properties:

- (a) Bilinearity: The map $e : G \times G \rightarrow G_T$ is bilinear if $e(a^x, b^y) = e(a, b)^{xy}$
- (b) Non-degeneracy: for all generators $g \in G$, e is non degenerate, that is $e(g, g) \neq 1$
- (c) Efficiency: e is efficiently computable

The security of our scheme is based on the following number-theoretic assumption.

Definition 2 (Decisional BDH Problem (DBDH)). Consider $g, g^a, g^b, g^c \in G$, where $a, b, c \in \mathbb{Z}_p$ and a value β is selected at random from $\{0, 1\}$. Let $z = e(g, h)^{abc}$ if $\beta = 1$ and let z be a random element from G_T otherwise. Given g, g^a, g^b, g^c, z output a guess β' of β . An algorithm has an advantage ϵ in solving DBDH if

$$|\Pr[\beta' = \beta] - \frac{1}{2}| \geq \epsilon$$

Identity-Based Encryption

An IBE scheme consists of four polynomial time algorithms: $\Pi = \text{Setup, Extract, Encrypt and Decrypt}$. Setup takes a security parameter k and generates the \mathcal{KGC} public system parameters $params$ and a master secret key msk ; Extract takes the msk and an identity id and generates the private key sk_{id} corresponding to id ; Encrypt takes id , $params$ and a message M , and generates the ciphertext ct for the M ; Decrypt takes the private key sk_{id} and a ciphertext ct , and decrypts ct using sk_{id} to retrieve the M .

3. DOUBLE-BLIND IDENTITY BASED ENCRYPTION

A double-blind IBE scheme consists of an IBE scheme Π , where the Extract algorithm is replaced with an interactive protocol $\text{DoubleBlindExtract}$. In $\text{DoubleBlindExtract}$ the identity string id consists of non-blinded elements \hat{id} known to both \mathcal{U} and \mathcal{KGC} , blinded elements \overline{id} known only to \mathcal{U} , and double-blind elements $\overline{\overline{id}}$ known only to the \mathcal{KGC} , such that $id = \hat{id}|\overline{id}|\overline{\overline{id}}$. The interactive key issuing protocol between \mathcal{U} and \mathcal{KGC} is described as follows:

$\text{DoubleBlindExtract}(\mathcal{U}(params, \hat{id}|\overline{id}), \mathcal{KGC}(params, msk, \overline{\overline{id}})) \rightarrow (sk_{\hat{id}|\overline{id}|\overline{\overline{id}}}, \hat{id})$ returns a private key $sk_{\hat{id}|\overline{id}|\overline{\overline{id}}}$ to \mathcal{U} that corresponds to the identity string $\hat{id}|\overline{id}|\overline{\overline{id}}$ provided; the non-blinded elements \hat{id} of the identity are returned to \mathcal{KGC} .

Our double-blind IBE scheme is based on an IBE scheme due to Naccache [11]. This scheme is a variant of the Waters’ IBE scheme [15] with a smaller public key size. Due to space restrictions we do not present it here.

$\text{DoubleBlindExtract}$

1. Given a vector $\gamma = (\gamma_1, \dots, \gamma_n) \in (\{0, 1\})^n$, if $\gamma_i = 1$ then v_i is to be blinded by \mathcal{U} . We define a vector \vec{v} such that

$$\vec{v} = \begin{cases} (u_i^\beta, v_i) & \text{if } i \in \{0, m\} \wedge \gamma_i = 0 \\ \perp & \text{if } i \in \{0, m\} \wedge \gamma_i = 1 \\ u_i^\beta & \text{if } i > m \end{cases}$$

\mathcal{U} computes $X \leftarrow (g^{\beta y} u'^{\beta} \prod_{i=1, \gamma_i=1}^m u_i^{\beta v_i})$ and sends $(X, \vec{v}, g^\beta, u'^{\beta})$ to \mathcal{KGC} . \mathcal{U} can prove to \mathcal{KGC} that it knows y, β and v_i where $\gamma_i = 1$ using zero-knowledge proofs as outlined in [12].

2. \mathcal{KGC} chooses random $r \in \mathbb{Z}_p$ and constructs $d'_v = (d'_1, d'_2)$ as

$$\begin{aligned} d'_v &= (g_2^\alpha (\prod_{i=1, \gamma_i \neq 1}^m u_i^{\beta v_i})^r X^r (\prod_{i=m+1}^n u_i^{\beta v_i})^r, g^{\beta r}) \\ &= (g_2^\alpha g^{\beta y r} (u' \prod_{i=1}^n u_i^{v_i})^{\beta r}, g^{\beta r}) \end{aligned}$$

computes $f = \prod_{i=m+1}^n u_i^{v_i}$ and passes d'_v, f to \mathcal{U} .

\mathcal{U} then tests that

$$e(g_1, g_2) \cdot e(d'_2, g^y u' \prod_{i=1}^n u_i^{v_i}) = e(g, d'_1)$$

3. If the test passes, \mathcal{U} chooses random $z \in \mathbb{Z}_p$ and computes

$$\begin{aligned} d_v &= (d'_1 / (d'_2)^y \cdot (u' \prod_{i=1}^n u_i^{v_i})^z, d'_1 \cdot g^z) \\ &= (g_2^\alpha (u' \prod_{i=1}^n u_i^{v_i})^{\beta r + z}, g^{\beta r + z}) \end{aligned}$$

If the test fails, \mathcal{U} outputs \perp and aborts. Note that the \mathcal{KGC} does not know d_1 or d_2 , where $d_v = (d_1, d_2)$.

3.1 The security of the scheme

Unusually, double-blind IBE schemes require that the user applying for a private key does not know the full corresponding public key $id = \hat{id}|\overline{id}|\overline{\overline{id}}$. We capture this requirement by introducing the concept of *Ciphertext Awareness* (CTA). Informally, CTA is a requirement that a decrypting entity can only produce a valid plaintext by applying the decryption algorithm to his private key and a ciphertext encrypted using the corresponding public key. This concept mirrors that of *Plaintext Awareness* (PA) [4, 3] which models an adversary’s inability to produce a ciphertext without knowledge of the underlying plaintext. That is, if a scheme is PA, then the only way an adversary can produce a valid ciphertext is to apply the encryption algorithm to the public

key. Similarly, CTA models an adversary’s inability to produce a plaintext x without knowing the corresponding ciphertext y . Consider an adversary \mathcal{A} for ciphertext awareness, given the secret key sk and access to a random oracle \mathcal{O} . \mathcal{A} is also given access to a second oracle $\mathcal{D}_{sk}^{\mathcal{O}}$. This second oracle is used to model the ability of an oracle to access valid plaintexts without the corresponding ciphertexts that it would get using queries to \mathcal{O} , $R[\mathcal{A}]$. By querying \mathcal{O} , \mathcal{A} has access to a decryption oracle that will, on input ciphertext C , extract the corresponding plaintext P , add P and C to a list of queried plaintexts \mathcal{O}_{list} and return P . In the real game, \mathcal{A} can query an encryption oracle on any plaintext $P \notin \mathcal{O}_{list}$, and the oracle will return $\text{Encrypt}(pk, P)$. In the ideal game, \mathcal{A} can query an encryption oracle on any plaintext $P \notin \mathcal{O}_{list}$ and the oracle will execute $\mathcal{A}^*(pk, P, R[\mathcal{A}], \mathcal{O}_{list})$ and return the result.

Real Game

$(pk, sk) \leftarrow \text{KeyGen}(params, msk, pk)$
 $x_{Real} \leftarrow \mathcal{A}^{\text{Encrypt}(pk, \cdot), \text{Decrypt}(sk, C(\cdot))}(sk)$

Ideal Game

$(pk, sk) \leftarrow \text{KeyGen}(params, msk, pk)$
 $x_{Ideal} \leftarrow \mathcal{A}^{*\text{Encrypt}(pk, \cdot, R, \mathcal{O}_{list}), \text{Decrypt}(sk, C(\cdot))}(sk)$

Definition 3 (Ciphertext Awareness). *A double-blind IBE scheme $\Pi = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ is ciphertext aware if for all polynomial-time plaintext extractors \mathcal{A} , there exists a polynomial-time ciphertext creator \mathcal{A}^* such that for an efficient distinguisher \mathcal{D} , the advantage*

$$\text{Adv}_{\mathcal{A}, \mathcal{A}^*, C, \text{Encrypt}} = |\Pr[\mathcal{D}(x_{Real})] - \Pr[\mathcal{D}(x_{Ideal})]|$$

The definitions of security for blind IBE, as per Green [9], focus on two aims: leak-free and selective-failure blindness. They can be extended for double-blind IBE schemes by constructing the pair of identities $\hat{id}|\overline{id}_0|\overline{id}$, $\hat{id}|\overline{id}_1|\overline{id}$. It is necessary to restrict the non-blinded information for each identity to be a common value for \hat{id} and the double-blind information to be a common value \overline{id} also. Due to space considerations we do not present the definitions and proofs here.

Definition 4 (Secure Double-Blind IBE).

An IBE scheme Π is secure if and only if: (1) the underlying Π is a secure IBE scheme, (2) DoubleBlindExtract is leak-free and selective-failure double-blind and (3) the resulting scheme is ciphertext aware.

4. AN E-VOTING SCHEME USING DOUBLE-BLIND IBE

The e-voting scheme we present has the same steps that voters are familiar with from secret ballot voting. Secret ballot schemes were introduced in Australia in the 19th century, and constituted a radical change in how elections were carried out: a voter retained the right to vote in secret, uninfluenced by external parties. Secrecy is now a required feature of most democratic elections, which can lead to issues with auditing an election. Auditing an election is equivalent to verifying that votes are counted as cast, and that all votes are included. We aim to provide a verifiable, secret e-voting protocol.

The secrecy of a ballot is achieved by ensuring an individual voter cannot be associated with the vote she cast. The vote is tallied along with all others but the voter’s choice remains private.

If all votes are kept secret, how do we prevent a fraudulent election result being announced as valid? We provide an individual voter with the ability to ensure her vote has been correctly recorded. We also provide all observers, including voters, with the ability to ensure election integrity by examining the tally.

4.1 Overview

Our e-voting scheme consists of two stages: *Ballot Generation* and *Ballot Casting*. As with existing physical voting schemes using paper ballot cards, the voter is required to register and present her credentials in order to prove her right to vote. Once the election is over, in the *Count and Publish* process the ballots are tallied and an official election result is published. In traditional voting schemes there is no way to verify that this is an accurate result.

Ballot Generation. At this point in traditional elections, a voter is allocated a single ballot card, and directed to a physically private voting booth. The ballot paper she fills in is unlinkable to her registration details. This unlinkability is crucial if the vote is to remain secret.

Our scheme is similar in structure. The voter presents her credentials but instead of receiving a ballot card, she receives access to a physically private voting booth containing a Ballot Generation Machine (\mathcal{BGM}). The \mathcal{BGM} takes voter input of her vote and outputs to the voter a ballot card containing details of her vote and information that will act as a receipt. The voter in any \mathcal{BGM} is unlinkable to the credentials she used to register. The resulting ballot card is unlinkable to the voter.

Ballot Casting. Once a voter has constructed her ballot, she casts her vote by placing the ballot into the ballot box. This completes the ‘vote and go’ process. The voter has no receipt of her vote and has no means to verify her vote is recorded and tallied correctly.

We also require the voter to cast her vote. She presents her ballot to a machine acting as a Ballot Box (\mathcal{BBOX}). Once the \mathcal{BBOX} accepts the vote as valid, it appends the vote to the Bulletin Board \mathcal{BB} . At this point the \mathcal{BBOX} separates the ballot into two pieces. The piece containing the vote details is kept by the \mathcal{BBOX} which returns the voter her receipt. This receipt can be used by the voter to check her vote appears correctly on the \mathcal{BB} . It can not be used to convince any third party that she voted in a particular way.

Count and Publish. Once the election is over, the ballots are counted and the result is published. The \mathcal{BB} containing a list of the votes cast and their corresponding random numbers is made public. This allows any interested party to confirm that the votes cast were correctly counted, and also allows an individual voter to confirm her vote was correctly recorded.

4.2 The scheme

Ballot generation

The trusted \mathcal{BGM} receives the vote V_A from the voter. As each vote must have an associated unique random number R_A , the \mathcal{BGM} generates such a number. The \mathcal{BGM} has read-only access to the Bulletin Board (\mathcal{BB}), from which it retrieves a valid random number R_i associated with a vote V_i for each of the candidates running in the election. In the early stages of the election, this requires the \mathcal{BB} to generate a nominal number of such votes V_j , a count of which is published without the corresponding R_j at the count and publish stage. The election tally is adjusted accordingly. The \mathcal{BGM} then constructs the vote string corresponding to the voters choice of candidate V_A for election E as $E| \overline{V_A} | \overline{R_A}$. The \mathcal{BGM} passes the partially-blind vote string to the \mathcal{BBOX} , who acts as the \mathcal{KGC} . It accepts the vote is for the current election,

as the election E is visible in the string.

The $BBO\mathcal{X}$ appends a secret value R_M to the string, using the double-blind key extraction protocol and extracts the key. This R_M value acts as a validity token, ensuring only keys generated by the $BBO\mathcal{X}$ can be validated. It is required to remain a secret for the duration of the election. Thereafter, it is only of value to a party who wants to audit the election.

The blinded private key for the vote $K'_A = E | \overline{V_A} | \overline{R_A} | \overline{R_M}$ is returned to the BGM , who unblinds it and constructs the voters ballot. As a result of this unblinding by the BGM , the $BBO\mathcal{X}$ cannot associate the resulting key, K_A with the instance of the DoubleBlindExtract protocol it uses to generate the key.

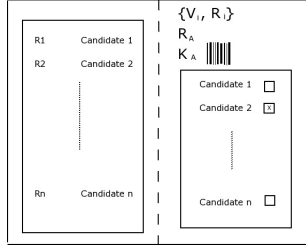


Figure 1: Voter's Ballot

Ballot structure. The voters ballot is a paper ballot. It is perforated along the centre axis, allowing the separation of the ballot in two. The right hand side contains the vote V_A , the random number R_A associated with the vote and, the key associated with this vote K_A . The vote V_A is presented as a list of candidates with the voters choice marked. This provides the voter with a visual representation of her vote, which she can easily check is correct. The right hand side of the ballot provides an auditor with sufficient information to ensure the vote chosen in the visual representation corresponds to the vote recorded. It is retained by the $BBO\mathcal{X}$ to facilitate auditing if the election result is queried.

The left hand side of the ballot is the voters receipt. She is free to carry this out of the election hall with her, and as such it must meet certain requirements:

1. The voter must be able to use this receipt to verify her vote has been correctly recorded
2. The voter must not be able to use the receipt to prove how she voted to any observer of the election

The construction of the ballot is key to achieving these requirements. Firstly, the voter must be able to use the receipt to verify her vote is recorded accurately. The receipt contains a list of all candidates in the election. The voter can simply check that the random number R_A and her vote V_A appear as a valid pair on the BB once it is published.

The voter's receipt is constructed by associating with each possible candidate a unique valid random number such that an observer checking the bulletin board finds each pair is recorded as a vote on the bulletin board. Thus a voter can claim any of the votes on her receipt as her own. She cannot prove conclusively to an observer which of them is hers. This fulfils the second requirement.

Ballot generation over an authenticated connection:

$$\begin{aligned} BGM \rightarrow \mathcal{KGC} & : E | \overline{V_A} | \overline{R_A} \\ \mathcal{KGC} \rightarrow BGM & : K'_A \\ & E | \overline{V_A} | \overline{R_A} | \overline{R_M} \end{aligned}$$

Ballot Casting

The voter presents her perforated ballot to the ballot box ($BBO\mathcal{X}$), which validates the vote using secret random value R_M and public parameters for the election.

The $BBO\mathcal{X}$ constructs the encryption of a nonce value, \mathcal{N} , under the vote string $E | V_A | R_A | R_M$, where the V_A, R_A values are those presented in the ballot. Using the key barcode K_A , it attempts to decrypt the ciphertext. If the $\text{Decrypt}(\text{Encrypt}(\mathcal{N}))$ returns \mathcal{N} , $BBO\mathcal{X}$ accepts the vote is valid. The challenge can only be decrypted if the voter has performed the ballot generation step correctly, as it is not possible for her to construct K_A otherwise. Once the vote is accepted as valid, the $BBO\mathcal{X}$ dissects the ballot. It retains the right hand side as a means to audit the election result should there be a query, and returns the left hand side, the receipt, to the voter. The voter appends it to the BB by the $BBO\mathcal{X}$, to which it has write-only access.

Ballot casting over an authenticated connection:

$$BBO\mathcal{X} \rightarrow BB : E | V_A | R_A$$

4.3 Discussion

The main trust requirement of the scheme is that the parties involved execute the algorithms and protocols correctly. As such, the BGM is required to generate a unique random number for each vote. An ill-formed ballot (malicious or otherwise) that does not accurately record the voters preference will be detected at one of the following stages: by the voter when she is presented with an incorrect ballot, by the $BBO\mathcal{X}$ when check the vote is valid or by the voter when she checks the BB at the end of the election. If the $BBO\mathcal{X}$ attempts to record a vote different to that of the voter, it can be caught by the voter when she checks the BB . As protocol maintains a paper trail of votes, an audit will capture this malicious behaviour.

The scheme provides *end-to-end verifiability*, as illustrated in figure 2.

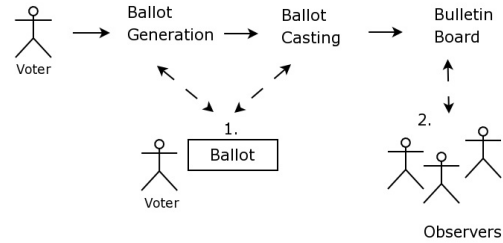


Figure 2: End-to-end Verifiability

Two checkpoints are required to achieve end-to-end verifiability.

1. The voter is required to carefully examine the ballot she receives before casting it. If the ballot does not reflect her vote, she should appeal to an election official at this point.
2. Any observer to the election can check the bulletin board and tally the human readable votes it contains. This can then be compared to the published election result. As each vote has associated with it a unique random number, each voter can ascertain if their vote was accurately recorded on the bulletin board using her receipt.

Auditing the election is possible by checking the right side of each ballot, as collected by the $BBO\mathcal{X}$. There is sufficient information

on each one to verify or disprove the validity of each vote recorded on the bulletin board. Validity is examined using encryption and decryption to verify K_A , as per ballot casting.

4.4 Security

Theorem 1 (Privacy / Secrecy). *Even if the BGM and $BBOX$ collude, the relationship between a voter and her vote V_A cannot be detected.*

PROOF. The relation between the voter's credentials and her vote $E|V_A|R_A$ remains obscured as the voter credentials are checked prior to entering the voting booth. The vote is cast anonymously.

Theorem 2 (Uniqueness). *No voter can vote twice.*

PROOF. Each voter has their credentials checked before accessing the BGM and cannot generate a valid ballot without it. There exists a one-to-one relationship between the validity of the credentials a user presents and the number of votes she can cast. This can be enforced by a requirement to reset the BGM for single use prior to each voter being admitted.

Theorem 3 (Coercion Resistance). *No voter can prove to a third party how she voted.*

PROOF. A voter's receipt contains a valid vote and associated random number for each candidate that will appear on the published bulletin board. A voter can claim to have voted for any of the candidates in the election but it is not possible for a voter convince a third party a specific vote on her receipt is the one she cast.

Theorem 4 (Individual Verifiability). *A voter can be assured her vote is accurately recorded.*

PROOF. A voter can check her vote has been accurately recorded in two ways. Firstly, by examining both sides of her ballot before casting. Secondly, by searching the bulletin board for her unique random number R_A . Beside the R_A value, the voter will find her vote, as it was recorded.

Theorem 5 (Universal Verifiability). *Any third party can check that the tallying actions that produced the official election result are valid.*

PROOF. A third party can compare the recorded votes on the bulletin board, which are plaintext, with the official election result.

Theorem 6 (Fairness). *No partial tally of results may be released until the official count is complete.*

PROOF. The bulletin board is not published until an election is complete. While it is possible for a malicious observer to try to reconstruct the contents of the bulletin board using the voter receipts during an election, this is not considered a feasible attack.

5. FUTURE WORK

We have presented a new scheme that provides a familiar experience for voters. It adheres to commonly accepted security criteria of electronic elections. A novel contribution is the recording of votes in on a public bulletin board in plaintext which provides a simple method of verification for voters and independent observers alike. The scheme facilitates a voter receipt as well as a paper based auditing system.

The scheme requires further examination. Currently, the scheme is only functional for majority rule and plurality 'first past the post' style elections for n candidates. More work is required to discover if proportional representation is possible under this scheme.

6. REFERENCES

- [1] B. Adida and R. Rivest. Scratch & vote: self-contained paper-based cryptographic voting. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 29–40. ACM New York, NY, USA, 2006.
- [2] R. Anane, R. Freeland, and G. Theodoropoulos. e-Voting Requirements and Implementation. In *E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, 2007. CEC/EEE 2007. The 9th IEEE International Conference on*, pages 382–392, 2007.
- [3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Plaintext Awareness, Non-Malleability, and Chosen Ciphertext Security: Implications and Separations. In *Crypto*, volume 98, pages 26–45. Springer.
- [4] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In *Advances in Cryptology: Proceedings*, page 26. Springer, 1998.
- [5] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy. Blind and Anonymous Identity-Based Encryption and Authorised Private Searches on Public Key Encrypted Data. In *Public Key Cryptography-Pkc 2009: 12th International Conference on Practice and Theory in Public Key Cryptography Irvine, Ca, USA, March 18-20, 2009 Proceedings*, page 196. Springer, 2009.
- [6] S. Delaune, S. Kremer, and M. Ryan. Verifying Properties of Electronic-Voting Protocols. In *Proceedings of the IAVoSS Workshop On Trustworthy Elections (WOTE06)*, pages 45–52, 2006.
- [7] G. Dini. A secure and available electronic voting service for a large-scale distributed system. *Future Generation Computer Systems*, 19(1):69–85, 2003.
- [8] A. Gibbard. Manipulation of Voting Schemes: A General Result. *Econometrica*, 41(4):587–601, 1973.
- [9] M. Green and S. Hohenberger. Blind identity-based encryption and simulatable oblivious transfer. In *ASIACRYPT*, pages 265–282, 2007.
- [10] D. Moynihan. Building Secure Elections: E-Voting, Security, and Systems Theory. *Public Administration Review*, 64(5):515–528, 2004.
- [11] D. Naccache. Secure and practical identity-based encryption. *Information Security, IET*, 1(2):59–64, 2007.
- [12] T. Okamoto. Efficient blind and partially blind signatures without random oracles. *TCC*, 3876:80–99, 2006.
- [13] R. Rivest. The ThreeBallot voting system. *Unpublished draft*, <http://theory.lcs.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf>, 2006.
- [14] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO '84*, volume 196 of *LNCS*, pages 47–53. Springer Verlag, 1985.
- [15] B. Waters. Efficient Identity Based Encryption Without Random Oracles. *Advances in Cryptology-Eurocrypt 2005: 24th Annual International Conference on the Theory And Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, 2005.