MVP: An Efficient Anonymous E-voting Protocol

You Zhou* Yian Zhou* Shigang Chen* Samuel S. Wu†
*Department of Computer & Information Science & Engineering
†Department of Biostatistics
University of Florida, Gainesville, FL 32611, USA
Email: {youzhou, yian, sgchen}@cise.ufl.edu samwu@biostat.ufl.edu

***Abstract*—Thanks to the Internet, voters can cast their ballots over the electronic voting (E-voting) systems conveniently and efficiently without going to the polling stations. However, existing E-voting protocols suffer from anonymity issues and/or high deployment overhead. In this paper, we design a practical anonymous E-voting protocol (referred to as MVP) based on a novel data collection technique called *dual random matrix masking* (DRMM), which guarantees anonymity with low overhead of computation, and achieves the security goals of receipt-freeness, double voting detection, fairness, ballot secrecy, and integrity. Through extensive analyses on correctness, efficiency, and security properties, we demonstrate our proposed MVP protocol can be applied to E-voting in a variety of situations with accuracy and anonymity.**

I. Introduction

The practical importance of electronic voting (E-voting) technology has attracted significant research interest in recent years [1]–[4]. E-voting provides great convenience for people to participate in votings without geographical restriction and time limitation. In particular, anonymous E-voting systems incorporate an important feature of anonymity [5]–[8], which guarantees that any ballot cannot be traced back to the individual person who casts it. Anonymity is critical in encouraging voters to voice their true opinions on sensitive issues and promoting participation in voting. There are many important requirements for anonymous E-voting system design:

- **Double Voting Detection:** Only eligible and verified voters are allowed to cast ballots. In addition, each qualified voter can only cast one valid vote. If a voter casts multiple ballots, only one vote will be counted.
- **Correctness:** All votes must be counted correctly. All valid ballots must be correctly counted into the final tally. No invalid ballot can be counted into the final tally.
- **Fairness:** No intermediate results can be obtained before the voting period ends (so that such information cannot be used to affect people who have not voted).
- **Receipt-freeness:** An essential property that prevents a voter from proving to any third party that he/she has voted for a particular candidate in order to avoid ballot buying.
- **Ballot Secrecy:** Besides the voter himself/herself, others cannot know the actual ballot that the voter has cast.
- **Integrity:** Any valid ballot cast by any individual voter cannot be modified, duplicated, or removed. Any attempt to tamper with the correct final tally will be detected.
- **Efficiency:** The computation for individual voters and voting servers should be as efficient as possible.

There are two types of attack against anonymous E-voting systems: passive attack and active attack. In passive attack, the adversary tries to obtain as much original ballot data and voter identity information as possible. Active attack is more severe: the adversary attempts to sabotage the voting process by altering the final tally. Existing research mainly focuses on the passive attack. Most of the prior protocols rely on expensive cryptographic techniques [4]–[10], and they can be classified into three categories based on mix-nets, blind signatures, and homomorphic encryption. While the existing protocols are able to address many important issues through different means, they also have limitations in practicality, features and efficiency, as we will explain in Section IV. None of them meets all the above requirements. Addressing those limitations will require us to explore new approaches that differ from the past ones.

In this paper, we design a practical anonymous E-voting protocol (referred to as MVP) based on a novel data collection technique called *dual random matrix masking* (DRMM), which guarantees anonymity with low computation overhead and achieves the aforementioned requirements of receipt-freeness, double voting detection, fairness, ballot secrecy and integrity, under both passive and active attacks. In the MVP protocol, voters can cast ballots without disclosing their votes and identity information. Through extensive analyses on correctness, efficiency, and security properties, we demonstrate our proposed MVP protocol can be applied to E-voting in a variety of situations with accuracy and anonymity.

II. Preliminaries

A. System Model

We consider an anonymous electronic voting (E-voting) system as illustrated in Figure 1, which consists of three parties:

- **Voters:** The people who have been authorized to participate in the voting. Each authenticated voter can only cast one valid vote. If a voter casts multiple ballots, only one vote should be counted into the final tally.
- **Data management center (DMC):** An organization that holds the voting activities. It is the center of the voting, and responsible for authentication of voters, counting valid ballots, and the announcement of the final tally.
- **Voting Proxy Server (VPS):** A server that is responsible for collecting valid masked ballots from authenticated voters, and forwarding masked group votes to the DMC.

In our model, an anonymous E-voting system consists of voters, one or multiple VPS, a DMC with backend servers,

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

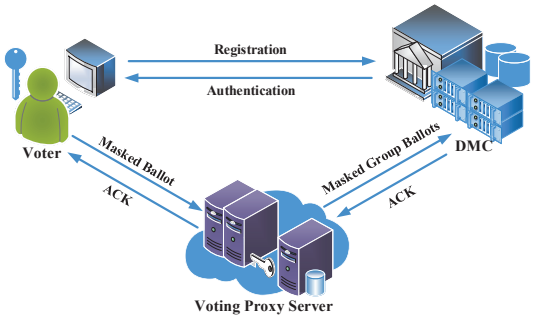

Fig. 1. Anonymous E-voting system model.

and communications between them. The DMC that holds the voting builds a server with a database of all voters' information, and it is responsible for registration, authentication and pre-voting communications with voters. Voters need to register themselves with the DMC before the voting. After obtaining the authentication from the DMC, voters can then cast their masked ballots through an E-voting website or an app on their mobile devices to the data collector VPS. For the purpose of anonymity, only masked vote data will be sent to the VPS. The VPS can be a company's web server or some other proxy server. It collects masked ballots from voters based on groups, then aggregates and masks each group of ballots, and transmits the masked group ballots to the DMC. Finally, the DMC will count the votes based on the masked group ballots, and publish the final tally result.

B. Security Model

We assume the DMC is semi-honest. On the one hand, it honestly fulfils its duty, including registration, authentication and pre-voting communications with voters, and counting each group's poll result based on the masked group ballots from the VPS. On the other hand, the DMC may be curious about and intend to obtain the original votes of individual voters.

Other potential adversary can be either a compromised VPS or some misbehaved voter(s). Misbehaved voters aim to alter the tally by double voting, but they are outnumbered by the honest voters. A sabotaged VPS may try to obtain voters' identity and their individual or group vote information, or even modify the final tally, without being detected. In spot of any adversary behavior during the voting process, the reputation of the VPS will be damaged, and the E-voting can be efficiently restarted.

More specifically, we consider an attack model with two outstanding potential attacks: the passive attack and the more severe active attack. In the passive attack, the adversary tries to obtain as much original ballot data and voter identity information as possible in the voting process. For example, a sabotaged VPS may try to mine the voting result before the DMC publishes it, or digest individual vote information from the masked ballots. In the more severe active attack, the adversary attempts to sabotage the voting process to alter the final tally result without being detected. For instance, misbehaved voters may try to cast multiple ballots, or a compromised VPS may modify or remove some collected ballots and try to create a fake tally result in favor of a particular candidate, etc.

We assume the communication channels between the three parties (voters, VPS, and DMC) in our E-voting system are secure, which can be easily realized through some known protocols (e.g., TLS or SSL), and untappable (i.e., no listening or recording device can be placed to record the voting process). With secure communication channels and proper authentication, no data in the system will be disclosed to any outsider adversary.

C. Matrix Masking

Matrix masking is the most popular technique used for data collection with disclosure limitations [11]–[14]. It refers to a class of statistical disclosure limitation (SDL) methods used to protect the confidentiality of statistical data, through some specific matrices transforming a data matrix to a masked matrix via pre- and post- multiplication and a possible addition of noise or perturbations. For example, Duncan [11] proposes to transform an $n \times p$ (cases by variables) data matrix X to the masked data of the form:

$$X \rightarrow AXB + C,$$

where matrix A is a row operator, matrix B is a column operator, and matrix C is the noise or perturbations added to the data. In our protocol, we adopt matrix masking as a building block to propose a novel *dual random matrix masking (DRMM)* technique. In particular, two matrices A and B are used to mask the original data X, where A is a random orthogonal matrix as a row operator only known to the data collector VPS, and B is a random invertible matrix as a column operator only known to the voting organization DMC. We will prove that our data collection protocols based on DRMM achieves anonymity and guarantees the same final tally result as the original data X.

1) Masked data disclosure limitation: In this paper, we use an invertible matrix B to mask the original data X, and the masked data XB will be sent to the data collector VPS:

$$X \rightarrow XB.$$

Since B is invertible, there exits a sequence of row operations $P_1, P_2..., P_s$ and column operations $Q_1, Q_2, ..., Q_t$ such that

$$P_s \dots P_2 P_1 B Q_1 Q_2 \dots Q_t = E$$
$$B = P_1^{-1} P_2^{-1} \dots P_s^{-1} E Q_t^{-1} \dots Q_2^{-1} Q_1^{-1},$$

where $P_1, P_2..., P_s$ and $Q_1, Q_2, ..., Q_t$ are elementary invertible matrices, and E is the unit matrix. In this case, the masked data can be presented as follows:

$$XB = X P_1^{-1} P_2^{-1} \dots P_s^{-1} E Q_t^{-1} \dots Q_2^{-1} Q_1^{-1}. \quad (1)$$

Since $P_1^{-1}, P_2^{-1}..., P_s^{-1}$ and $Q_1^{-1}, Q_2^{-1},..., Q_t^{-1}$ in (1) are all random column operation matrices used to do the random column switching, multiplication and adding, without knowing all these random matrices in advance, the adversary cannot get any information related to the columns of the original data X.

2) ROMM: Ting et al. propose a new perturbation method called *random orthogonal matrix masking* (ROMM) to protect confidential data [14]. ROMM is a special case of the standard matrix masking, under the conditions that B is the identity matrix, C is the zero matrix, and A is some random orthogonal matrix drawn from some known distribution D. Therefore, the transformation of ROMM is

$$X \to AX.$$

ROMM has one critical feature. It preserves the sample mean and sample covariance matrix after orthogonal pre-matrix masking. We provide a theorem that formally codifies this feature. Interested readers can refer to [14] for the formal proof.

Theorem 1. *Let X and $Y = AX$ be the original data and masked data produced by ROMM, respectively. Denote the sample mean and the sample covariance matrix of the original data X as $\bar{\boldsymbol{x}}$ and $\boldsymbol{\Sigma_x}$, and denote the sample mean and the sample covariance matrix of the masked data Y as $\bar{\boldsymbol{y}}$ and $\boldsymbol{\Sigma_y}$. Then $\bar{\boldsymbol{x}} = \bar{\boldsymbol{y}}$, and $\boldsymbol{\Sigma_x} = \boldsymbol{\Sigma_y}$, and*

$$colsum\{X\} = colsum\{Y\}, \tag{2}$$

where $colsum\{\}$ is a vector containing the sum of each column in the input matrix.

Theorem 1 states that the ROMM transformation preserves the sample mean and sample covariance of the original data, which further implies that the $colsum$ on the perturbed data is an unbiased estimate of the $colsum$ obtained from the original data. Therefore, in our protocols, the data collector VPS can simply publish the perturbed data produced by ROMM, while preserving the statistical features like $colsum$ of the original data.

D. Noise Flooding Lemma

Now, we introduce the noise flooding lemma, which will be used in our protocols.

Lemma 1. *Let c be a constant real number in $\mathbb{R}$ representing the useful information, and w be a random variable representing the noise. Denote the distribution of w as $f_w()$, which is a continuous function. Assume that $w' = aw$, and $a > 1$, then*

$$\lim_{a\to\infty}\left[f_{w'}(w'+c) - f_{w'}(w')\right] = 0,$$
$$\lim_{a\to\infty}\int_{-\infty}^{\infty}\left|f_{w'}(w'+c) - f_{w'}(w')\right| dw' = 0.$$

Proof: Since the distribution of w is $f_w(w)$ and $w' = aw$, the distribution of w' and $w'+c$ is

$$f_{w'}(w') = \frac{1}{a}f_w\big(\frac{w'}{a}\big) = \frac{1}{a}f_w(w),$$
$$f_{w'}(w'+c) = \frac{1}{a}f_w\big(\frac{w'+c}{a}\big) = \frac{1}{a}f_w\big(w+\frac{c}{a}\big).$$

Then,

$$\begin{aligned}
&\lim_{a\to\infty}\left[f_{w'}(w'+c) - f_{w'}(w')\right]\\
&= \lim_{a\to\infty}\left[\frac{1}{a}f_w\big(w+\frac{c}{a}\big) - \frac{1}{a}f_w(w)\right]\\
&= \lim_{a\to\infty}\left\{\frac{1}{a}\left[f_w\big(w+\frac{c}{a}\big) - f_w(w)\right]\right\} = 0,\\
&\lim_{a\to\infty}\int_{-\infty}^{\infty}\left|f_{w'}(w'+c) - f_{w'}(w')\right| dw'\\
&= \lim_{a\to\infty}\int_{-\infty}^{\infty}\left\{\frac{1}{a}\left|f_w\big(\frac{w'}{a}+\frac{c}{a}\big) - f_w\big(\frac{w'}{a}\big)\right|\right\} dw'\\
&= \int_{-\infty}^{\infty}\lim_{a\to\infty}\left|f_w\big(w+\frac{c}{a}\big) - f_w(w)\right| dw = 0.
\end{aligned}$$

This completes the proof. □

Lemma 1 states that, as a increases to infinity, the difference between $f_{w'}(w')$ and $f_{w'}(w'+c)$ decreases to 0. An implication is, when the noise w floods among the whole observation space, i.e., $a \to \infty$, we cannot tell the useful information c perturbed by the noise w'. Suppose there are two random variables w^1 and w^2, and they are independent and have the same distribution $f_{w'}(w')$ with a very large a. We randomly pick one variable and add a constant c to it. Then others cannot tell which variable has increased by c given a limited number of samples of the variables w^1 and w^2. In other words, the noise has overwhelmed the useful information c, so the adversary won't be able to obtain c from its perturbed format $w'+c$.

III. MVP: A NOVEL ANONYMOUS VOTING PROTOCOL

In this section, we propose a novel efficient *matrix masking anonymous voting protocol* (MVP) based on a novel technique *dual random matrix masking* (DRMM). We first introduce some basic functions to be applied in MVP, then describe the full MVP protocol in details. Finally, we analyze the correctness, efficiency, and security properties of MVP.

A. Functions

Before describing the full MVP protocol, we first introduce some basic functions in the protocol.

- **Setup:** $Setup(Reg_v) \mapsto (IB_v, Cert_v, GID_v, Index_v)$ is a function that takes the registration information Reg_v of a voter v as input, and outputs some setup parameters, including v's initial ballot noise (IB_v), certification ($Cert_v$), group ID GID_v, and group index $Index_v$. The pair (GID_v, $Index_v$) is defined as v's voting identifier.
- **Key Generation:** $KeyGen(Entity, GID) \mapsto (key)$ is a randomized algorithm that takes the entity's name and a group ID as input, and outputs a key for the entity. Through this function, the DMC obtains a random invertible masking matrix B, and the VPS obtains a random orthogonal masking matrix A, for each group.
- **Message Forwarding:** $Send(Msg, dst)$ is a reliable communication function that takes the destination dst and the message Msg as input, and accomplishes the secure message forwarding task. In our protocols, all message

forwarding between voters, the DMC, and the VPS are secured through this function.

B. MVP

Now we present our novel anonymous voting protocol MVP, which includes the following four phases: initialization, authentication, voting, and counting. Suppose there are p candidates in the voting.

1) Initialization phase: In the beginning, the DMC and the VPS generate their own pairs of asymmetric keys, i.e., (K_D^-, K_D^+) for the DMC and (K_V^-, K_V^+) for the VPS, and a shared symmetric key κ. Each voter v also generates its private key K_v^- and public key K_v^+, then uses its unique identification information (e.g., identification number, date of birth, etc.) to register with the DMC. Upon receipt of v's registration request and verifying v is a legitimate voter, the DMC generates a piece of registration information Reg_v, and sends the encrypted registration $E(K_v^+, E(K_D^+, Reg_v))$ to v. After that, the DMC uses the setup function to produce a group ID GID_v, a group index $Index_v$, an initial ballot noise IB_v, and a certification $Cert_v$ for v, and stores the information (Reg_v, GID_v, $Index_v$, IB_v, $Cert_v$) in its database. The DMC will also share the authentication information ($Cert_v$, GID_v, $Index_v$) with the VPS for it to later verify authorized voters. Finally, for each group of voters, the DMC will use the group ID GID as the seed to generate a random invertible matrix B, and store the information (GID, B) in its database. Similarly, for each group of voters, the VPS will use the group ID GID as the seed to generate a random orthogonal matrix A, and store the information (GID, A) in its database. The initialization phase is summarized in the following:

$$\begin{aligned} DMC:\ & \textbf{Setup}(Reg_v) \rightarrow (IB_v, Cert_v, GID_v, Index_v), \\ & \textbf{KeyGen}(DMC, GID) \rightarrow \text{invertible matrix } B. \\ VPS:\ & \textbf{KeyGen}(VPS, GID) \rightarrow \text{orthogonal matrix } A. \end{aligned}$$

2) Authentication phase: To guarantee only verified voters are allowed to cast ballots, voters must authenticate themselves with the DMC to obtain necessary information for voting. In the authentication phase, each voter v will decrypt its encrypted registration information obtained from the DMC to get $D(K_v^-, E(K_v^+, E(K_D^+, Reg_v))) = E(K_D^+, Reg_v)$, encrypt it with its private key, and send $E(K_v^-, E(K_D^+, Reg_v))$ to the DMC for verification. After decrypting the message with K_D^- and K_v^+, i.e., $D(K_D^-, D(K_v^+, E(K_v^-, E(K_D^+, Reg_v)))) = Reg_v$, and verifying that Reg_v is in its database, the DMC will perform the following operations:

(i) The DMC uses Reg_v to fetch v's group ID GID_v, group index $Index_v$, initial ballot noise IB_v, and certification $Cert_v$, and uses GID_v to fetch v's group masking matrix B, from its database.

(ii) The DMC generates message $M_a = (GID_v, Index_v)$, and $M_b = E(\kappa, Cert_v|GID_v|Index_v)$ by encrypting v's authentication information with the shared key κ.

(iii) The DMC computes p original ballots, $\{OB_v^1, ..., OB_v^j, ..., OB_v^p\}$, where OB_v^j represents voting for the j-th candidate, and is just a $1 \times p$ vector consisting of 0's and 1's such that $OB_v^j[j] = 1$, and $OB_v^j[i] = 0, \forall i \neq j$.

(iv) The DMC adds voter v's initial ballot noise IB_v to each original ballot to generate p perturbed ballots, $\{NB_v^1, ..., NB_v^j, ..., NB_v^p\}$, i.e., $NB_v^j = IB_v + OB_v^j, \forall j \in [1, p]$.

(v) The DMC multiplies each perturbed ballot with v's group masking matrix B to generate p masked ballots, $\{MB_v^1, ..., MB_v^j, ..., MB_v^p\}$, i.e., $MB_v^j = NB_v^j \times B, \forall j \in [1, p]$.

(vi) The DMC encrypts each masked ballot with the shared key κ to generate p encrypted masked ballots, $\{EB_v^1, ..., EB_v^j, ..., EB_v^p\}$, i.e., $EB_v^j = E(\kappa, MB_v^j), \forall j \in [1, p]$.

(vii) The DMC concatenates the p encrypted masked ballots to generate a message M_c, i.e., $M_c = (EB_v^1|...|EB_v^j|...|EB_v^p)$.

Finally, the DMC sends back a message M_1 to the voter v, which is an encrypted concatenation of M_a, M_b and M_c:

$$\begin{aligned} M_1 &= E(K_v^+, M_a|M_b|M_c) \\ &= E(K_v^+, M_a|E(\kappa, Cert_v|GID_v|Index_v)|M_c). \end{aligned} \tag{3}$$

When the voter v receives the message M_1, it will decode M_1 with its private key to get M_a, M_b and M_c: $D(K_v^-, M_1) = (M_a|M_b|M_c)$. M_a is v's voting identifier. Since the certification $M_b = E(\kappa, Cert_v|GID_v|Index_v)$ is encrypted by the key κ shared between the DMC and the VPS only, the voter v cannot decrypt M_b to access its authentication information ($Cert_v$, GID_v, $Index_v$). This serves an important purpose of assuring double voting detection, which we will explain more later.

3) Voting Phase: In the voting phase, voters cast encrypted masked ballots to the VPS. The VPS gathers the encrypted masked ballots based on voters' groups, further masks the group ballots, and sends to the DMC. There are four steps.

a) Step 1: voters cast encrypted masked ballots. From the previous authentication phase, each voter v receives its authentication information M_b, and M_c which concatenates p encrypted masked ballots. During the voting process, each voter v will decide which candidate to vote, and choose the corresponding encrypted masked ballot $EB_v = E(\kappa, MB_v) = E(\kappa, (IB_v + OB_v) \times B)$ from M_c: if v decides to vote for the j-th candidate, v will select EB_v^j from M_c, i.e., $EB_v = EB_v^j$. After that, v casts its vote by sending EB_v with the authentication information M_b in a message M_2 to the VPS:

$$\begin{aligned} M_2 &= E(K_V^+, EB_v|M_b) \\ &= E(K_V^+, EB_v|E(\kappa, Cert_v|GID_v|Index_v)). \end{aligned} \tag{4}$$

b) Step 2: VPS verifies each individual vote. Upon receiving a vote message M_2, the VPS decodes M_2 with its own private key K_V^- and the shared key κ:

$$\begin{aligned} &D(K_V^-, M_2) = (EB_v|M_b), \\ &D(\kappa, EB_v) = (MB_v), \\ &D(\kappa, M_b) = (Cert_v|GID_v|Index_v). \end{aligned} \tag{5}$$

The VPS looks up the authentication $(Cert_v|GID_v|Index_v)$ in its database, and checks whether the voting identifier $(GID_v, Index_v)$ is in the set S, which contains the $(GID, Index)$ of all already-voted voters to detect potential double voting. If $(Cert_v|GID_v|Index_v)$ exists in VPS's database, and $(GID_v, Index_v)$ does not exist in S, it means this ballot is the first

vote of an authenticated voter. So the VPS will accept the ballot, store the MB_v, and insert $(GID_v, Index_v)$ into S. Otherwise, the VPS will reject the ballot and do nothing.

c) Step 3: VPS generates masked group ballots. When the voting period ends, the VPS finishes collecting all voters' masked ballots. Next, the VPS starts to aggregate these ballots based on voters' groups: it will generate a masked group ballot GB_g for each group g of voters. Suppose in group g, there are m voters $\{v_1, v_2, ..., v_m\}$ who have cast valid ballots (i.e., their voting identifiers are in the set S), and their masked ballots are $\{MB_{v_1}, MB_{v_2}, ..., MB_{v_m}\}$. The masked group ballot GB_g is an $n \times p$ matrix satisfying the following requirement:

$$colsum\{GB_g \times B^{-1}\} = \sum\nolimits_{k=1}^{m} NB_{v_k}. \tag{6}$$

To generate GB_g based on the masked ballots $\{MB_{v_1}, MB_{v_2}, ..., MB_{v_m}\}$, we provide the following Theorem 2.

Theorem 2. *If* $colsum\{GB_g\} = colsum\{\sum_{k=1}^{m} MB_{v_k}\}$, *then* $colsum\{GB_g \times B^{-1}\} = \sum_{k=1}^{m} NB_{v_k}$.

Proof: For two arbitrary multipliable matrices T_1 and T_2,

$$colsum\{T_1 \times T_2\} = colsum\{T_1\} \times T_2.$$

Therefore,

$$\begin{aligned} &colsum\{GB_g \times B^{-1}\} = colsum\{GB_g\} \times B^{-1} \\ =&colsum\Big\{\textstyle\sum_{k=1}^{m} MB_{v_k}\Big\} \times B^{-1} = colsum\Big\{\textstyle\sum_{k=1}^{m} MB_{v_k} \times B^{-1}\Big\} \\ =&colsum\Big\{\textstyle\sum_{k=1}^{m} NB_{v_k}\Big\} = \textstyle\sum_{k=1}^{m} NB_{v_k}. \end{aligned}$$

This completes the proof. □

From Theorem 2, VPS can easily generate GB_g by producing p $n \times 1$ column vectors GB_g^1, ..., GB_g^j, ..., GB_g^p, such that $colsum\{GB_g^j\} = colsum\{\sum_{k=1}^{m} MB_{v_k}\}[j]$, $\forall j \in [1, p]$. To produce GB_g^j, VPS can employ a random number generator to create $n-1$ random numbers, $r_1, r_2, \ldots, r_{(n-1)}$, as the first $n-1$ elements of GB_g^j, and set the last element of GB_g^j to be $colsum\{\sum_{k=1}^{m} MB_{v_k}\}[j] - \sum_{i=1}^{n-1} r_i$. Repeating this process by p times, VPS will obtain all p column vectors of GB_g.

d) Step 4: VPS uploads dual-masked group ballots to the DMC. Now the VPS has generated a masked group ballot GB_g for each group g of voters. Next, it will multiply each GB_g by its corresponding group orthogonal matrix A (note that different groups have different A, and each A is a $n \times n$ random orthogonal matrix only known by the VPS), and send $A \times GB_g$ with $GID = g$ with the set S in a message M_3 to the DMC:

$$M_3 = E(\kappa, \{(A \times GB_g, GID = g)\}_{g \in \{GID\}} | S). \tag{7}$$

4) Counting Phase: Upon receiving the message M_3, the DMC decodes M_3 to obtain $A \times GB_g$ for each group $GID = g$. Then the DMC uses the $GID = g$ to fetch the group masking matrix B from its database, and computes its inverse matrix B^{-1}. To obtain the aggregate tally of all voters in the group g, the DMC first multiplies $A \times GB_g$ by B^{-1} to recover the de-masked group ballot G_g:

$$G_g = A \times GB_g \times B^{-1}. \tag{8}$$

Then, the DMC calculates $colsum\{G_g\}$, and subtracts the sum of the initial ballot noises of all voters in the group g, who have cast valid ballots according to S, i.e., $\sum_{k=1}^{m} IB_{v_k}$. The final tally result of group g is simply

$$c_g = colsum\{G_g\} - \sum\nolimits_{k=1}^{m} IB_{v_k}. \tag{9}$$

In the next subsection, we will prove $c_g = x_g$, where $x_g = \sum_{k=1}^{m} OB_{v_k}$ is the sum of the original ballots of all voters who have cast valid ballots in group g. Finally, for each group g, the DMC verifies if the number of voters in group g matches the number of votes in c_g. If they all match, the DMC will add all c_g from every group g, and publish the final tally and the set S. This completes our MVP protocol.

C. Correctness

First, we prove that for each group g, the counting result of the DMC c_g equals the actual voting result $x_g = \sum_{k=1}^{m} OB_{v_k}$.

Proof: From (9), (8), Theorem 1, and (6), we have

$$\begin{aligned} c_g &= colsum\{G_g\} - \sum\nolimits_{k=1}^{m} IB_{v_k} \\ &= colsum\{A \times GB_g \times B^{-1}\} - \sum\nolimits_{k=1}^{m} IB_{v_k} \\ &= colsum\{GB_g \times B^{-1}\} - \sum\nolimits_{k=1}^{m} IB_{v_k} \\ &= \sum\nolimits_{k=1}^{m} NB_{v_k} - \sum\nolimits_{k=1}^{m} IB_{v_k} = \sum\nolimits_{k=1}^{m} OB_{v_k} = x_g. \end{aligned}$$

This completes the proof. □

Next, the DMC calculates the c_g from (9) for each group $g \in \{GID\}$, and adds all c_g to obtains the final result R. Clearly,

$$R = \sum_{g \in \{GID\}} c_g = \sum_{g \in \{GID\}} x_g \ . \tag{10}$$

This demonstrates the universal correctness of MVP.

D. Efficiency

Suppose there are a total of N voters in M groups. The computation overhead of each entity in MVP is given as follows:

- Each voter only needs to pick one encrypted masked ballot and cast it to the VPS. So its computation cost is $O(1)$.
- The VPS first aggregates N voters' masked ballots ($1 \times p$ vectors) into M groups, whose cost is $O(N \times p)$. Then it generates M masked group ballots $\{GB_g\}_{g \in \{GID\}}$ ($n \times p$ matrices), whose cost is $O(M \times n \times p)$. Finally, it multiplies each of the M masked group ballots, GB_g, with a $n \times n$ orthogonal matrix A, to produce the dual-masked group ballots, whose cost is $O(M \times n^2 \times p)$. So the total computation overhead of the VPS is $O(N \times p + M \times n^2 \times p)$.
- The DMC needs to generate p masked ballots for each voter, whose cost is $O(N \times p^3)$. Also, for each group g

of voters, the DMC needs to multiply the dual-masked group ballot $A \times GB_g$ by a $p \times p$ matrix B^{-1} to recover the de-masked group ballot G_g ($n \times p$ matrix), whose cost is $O(M \times n \times p^2)$. Then it calculates $colsum\{G_g\}$ and subtracts $\sum_{k=1}^{m} IB_{v_k}$ to compute c_g, and add c_g for all M groups, whose cost is $O(M \times n \times p + N \times p)$. So DMC's total computation overhead is $O(N \times p^3 + M \times n \times p^2)$.

Since p and n are constant numbers far smaller than M and N, the computation complexity for voters, the VPS, and the DMC are actually $O(1)$, $O(N+M)$, and $O(N+M)$, respectively. One can see that our MVP protocol is indeed very efficient.

E. Security Analysis

1) Receipt-freeness: Generally speaking, receipt-freeness means a voter cannot provide any proof to demonstrate to a third party that he/she has voted for a particular candidate, which makes the vote-buying impossible. In our MVP protocol, receipt-freeness is achieved for the following three reasons. First, the communication channels in MVP are secure and untappable, and voters must authenticate themselves with the DMC to vote. In other words, physical recording and masquerade are not possible. Second, the MVP protocol is designed in a one-way manner. A voter v only obtains p encrypted masked ballots $\{EB_v^j\}_{j\in[1,p]}$ from the DMC, and can only cast one valid ballot. After v casts his/her encrypted masked ballot EB_v, the VPS will decrypt and obtain its masked ballot MB_v. But an honest VPS will not provide proof for v to demonstrate that MB_v is v's masked ballot. Even if the VPS is compromised to cooperate with the voter v, from its masked ballot MB_v, the voter v still cannot prove his/her vote OB_v, since neither the initial ballot noise IB_v nor the group masking matrix B are known to v or the VPS. Finally, since the DMC is semi-honest, it knows IB_v and B, but will never prove to any third party that IB_v and B are used to mask v's original ballot OB_v to allow vote-buying. Therefore, the voter v cannot provide any proof that his/her cast ballot EB_v indeed corresponds to the vote OB_v after he/she casts the ballot.

2) Double Voting Detection: In MVP, only verified voters are allowed to vote, and each qualified voter can only cast one valid vote. Voters must register and authenticate themselves with the DMC to obtain necessary information for voting. In particular, each voter obtains a unique voting identifier from the DMC, which must accompany his/her vote for authentication. The VPS keeps track of a set S of the voting identifiers of all already-voted voters. If a misbehaved voter v attempts to cast multiple ballots, the VPS will only accept v's first vote and ignore v's other votes. Even if the VPS is compromised to cooperate with the voter to cast multiple ballots, the DMC can easily detect this cheating behavior through mismatched number of voters and the final tally derived from de-masking and removing the initial ballot noises during the counting phase. Therefore, MVP achieves double voting detection.

3) Fairness: MVP also achieves fairness by employing a novel data collection technique dual random matrix masking (DRMM). Voters' ballots are protected by two matrices A and B: the orthogonal masking matrix A is only known to the VPS, while the invertible masking matrix B is only known to the DMC. In MVP, voters' individual masked ballots are collected by and stored in the VPS. Since the group masking matrix B is only known to the DMC, the VPS cannot decode the masked ballots to obtain the original ballots or the voting counts. Even if the VPS retrieves B, it still cannot derive any group ballot information without knowing the distribution of the initial ballot noise IB for this group of voters.

Suppose the VPS tries to estimate the group voting result x_g for the group g. The m original ballots, $u_1 = OB_{v_1}$, $u_2 = OB_{v_2}$, ..., $u_m = OB_{v_m}$, are modeled as realizations of m independent and identically distributed (i.i.d.) random variables (R.V.) U_i, $i \in [1,m]$, each with the same distribution as a R.V. U. The R.V. U can only have p different values $\{u^j = OB_v^j | j \in [1,p]\}$, where OB_v^j represents the original ballot voting for the j-th candidate. Similarly, the m initial ballot noises, $w_1 = IB_{v_1}$, $w_2 = IB_{v_2}$, ..., $w_m = IB_{v_m}$, are viewed as realizations of m i.i.d. R.V.s W_i, $i \in [1,m]$, each with the same distribution as a R.V. W. The original ballots are independent of the initial ballot noises, so U is independent of W. The VPS has the perturbed ballot R.V. $Z = U + W$ with m independent samples $z_1 = NB_{v_1} = u_1 + w_1$, $z_2 = NB_{v_2} = u_2 + w_2$, ..., $z_m = NB_{v_m} = u_m + w_m$, and its purpose is to estimate the posterior probability distribution of U, i.e., $P^{'}(U = u_i)$, which can reveal the group voting result, with the perturbed ballots. Using Bayes' theorem, the posterior probability distribution of U can be written in terms of the density function f_W of W as

$$P^{'}(U=u^j) = \frac{f_Z(z|U=u^j)P(U=u^j)}{f_Z(z)}$$
$$= \frac{f_Z(z|U=u^j)P(U=u^j)}{\sum_{k=1}^{p} f_Z(z|U=u^k)P(U=u^k)} = \frac{f_W(z-u^j)P(U=u^j)}{\sum_{k=1}^{p} f_W(z-u^k)P(U=u^k)}.$$

However, since the VPS doesn't know the density function f_W of W, it cannot estimate the distribution of the original ballots U from the m samples of perturbed ballots Z.

Finally, since the VPS will not upload group ballots to the DMC for counting until all voters have finished voting, the DMC can only count the final tally after the voting period ends. Therefore, no intermediate results can be obtained by any entity.

4) Ballot Secrecy: MVP achieves ballot secrecy. In MVP, the VPS and DMC each holds a key matrix: VPS keeps the orthogonal matrix A, and DMC keeps the invertible matrix B. They are both curious about the original ballot OB_v of each individual voter v, but they cannot obtain OB_v from MVP.

For the DMC, it only has access to a dual-masked group ballot $A \times GB_g$ for each group g, which is just a random $n \times p$ matrix. Since A is only known to the VPS, the DMC cannot decode the dual-masked group ballot to obtain any individual ballot. Even if the DMC retrieves A, and uses A^{-1} to recover the masked group ballot GB_g, it still cannot derive any individual ballot from the group-randomized ballot GB_g.

For the VPS, it has access to the masked ballots MB_v for each voter v. However, since the group masking matrix B is only known to the DMC, the VPS cannot decode MB_v to obtain v's original vote OB_v. Even if the VPS retrieves B, and uses B^{-1} to recover the perturbed ballot NB_v, it still cannot derive the original ballot OB_v without the initial ballot

noise IB_v. Suppose the VPS attempts to digest the original ballot OB_v from its perturbed ballot $NB_v = z$. Through Lemma 1, when the noises flood among the whole observation space, the VPS cannot tell which noise has been added by 1 (the vote), which means it cannot retrieve the original ballot OB_v from the perturbed ballot $NB_v = z = [z^1, z^2, \cdots, z^p] = [w^1, w^2, \cdots, w^p] + OB_v$. In other words, if the noises fluctuate with an amplitude higher than a certain value, the noises will overwhelm the original ballot. Therefore, the ballot secrecy is preserved.

5) Integrity: In our MVP protocol, each voter v only has access to p encrypted masked ballots, each corresponding to one vote for one particular candidate. Since the ballots are encrypted and masked by the DMC, v cannot modify these ballots to throw multiple votes for some candidate(s) in one ballot. In addition, since the masking matrix B and the initial ballot noises are only known to the DMC, if a compromised VPS attempts to duplicate, modify, or replace any individual ballot, the number of voters and the final tally result will be mismatched during the counting phase, which can be easily detected by the DMC. The VPS cannot remove any individual ballot either. When the DMC publishes the final tally result, any voter v who has voted can check whether its voting identifer $(GID_v, Index_v)$ is in the set S or not. If $(GID_v, Index_v) \notin S$, then v can detect the VPS has removed its ballot and report to the DMC. Therefore, no entity can modify, duplicate, or remove any individual ballot without being detected. The integrity of each individual ballot is achieved in our MVP protocol.

IV. Related Work

The existing anonymous E-voting protocols can be briefly summarized into three categories: mix-nets, blind signatures, and homomorphic encryption.

Mix-net is a chain of mixing servers (mixers) that provide an anonymous communication channel between the senders and the receivers by re-arranging the order of the messages from the senders [15]. The anonymous communication channel offered by Mix-net can be used to design anonymous voting protocols that prevent the sever that collects votes from knowing which vote comes from which voter [1], [5], [7], [10], [15]. However, the existing protocols based on Mix-net have their deficiency [8]: They are generally not efficient since computational effort is required for multiple mixers to prove the correctness. These mixers must be independently owned (such that they will not collude), which increases the deployment difficulty in practice.

Protocols based on blind signatures [9], [16] are more practical with simpler procedures involving an authorized authentication center that produces blind signatures and a server that collects the votes. Removing the need for a trusted center that generates blind signatures, a ring signature based protocol [8] is proposed for voters to sign their votes with linkable ring signatures [6]. The greatest drawback of these protocols is the assumption of an anonymous channel between the voters and the server.

Homomorphic encryption is another technique frequently employed by anonymous E-voting protocols [4], [17]. Most of them do not conform to the receipt-freeness requirement. For the ones that do [4], they cannot ensure the integrity requirement when the proxy server (that combines the homomorphically encrypted votes) is compromised. Moveover, the computation overhead associated with homomorphic encryption is relatively high, and each voter must provide the proof of validity of his/her ballot, which needs to be verified by the proxy server.

V. Conclusion

In this paper, we propose two novel efficient anonymous E-voting protocol MVP. Unlike existing anonymous E-voting protocols which incur high computation overhead for cryptographical operations, our protocols apply a novel efficient dual random matrix masking (DRMM) technique. Through analysis, we demonstrate their correctness, receipt-freeness, double voting detection, fairness, ballot secrecy, and integrity.

VI. Acknowledgment

This work is supported in part by the National Science Foundation under grant STC-1562485, and a grant from Florida Cybersecurity Center.

References

[1] A. Neff, "A Verifiable Secret Shuffle and its Application to E-Voting," *Proc. of ACM CCS*, pp. 116–125, 2001.

[2] D. Chaum, A. Essex, R. Carback, J. Clark, S. Popoveniuc, A. Sherman, and P. Vora, "Scantegrity: End-to-End Voter-Verifiable Optical- Scan Voting," *Proc. of IEEE Security & Privacy*, vol. 6, no. 3, pp. 40–46, 2008.

[3] B. Adida, "Helios: Web-based Open-Audit Voting," *Proc. of USENIX Security Symposium*, vol. 17, pp. 335–348, 2008.

[4] A. P. Adewole, A. S. Sodiya, and O. A. Arowolo, "A Receipt-free Multi-Authority E-Voting System," *International Journal of Computer Applications*, vol. 30, no. 6, pp. 15–23, 2011.

[5] M. Jakobsson, A. Juels, and R. L. Rivest, "Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking," *Proc. of USENIX Security Symposium*, pp. 339–353, 2002.

[6] J. Liu, V. Wei, , and D. Wong, "Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups," *Proc. of Information Security and Privacy*, pp. 325–335, 2004.

[7] D. Chaum, "Secret-Ballot Receipts: True Voter-Verifiable Elections," *Proc. of IEEE Security & Privacy*, vol. 2, no. 1, pp. 38–47, 2004.

[8] S. Chow, J. Liu, and D. Wong, "Robust Receipt-Free Election System with Ballot Secrecy and Verifiability," *Proc. of NDSS*, vol. 8, pp. 81–94, 2008.

[9] A. Fujioka, T. Okamoto, , and K. Ohta, "A Practical Secret Voting Scheme for Large Scale Elections," *Proc. of AUSCRYPT'92*, pp. 244–251, 1993.

[10] A. Riera and J. Borrell, "Practical Approach to Anonymity in Large Scale Electronic Voting Schemes," *Proc. of NDSS*, 1999.

[11] G. T. Duncan and R. W. Pearson, "Enhancing Access to Microdata While Protecting Confidentiality: Prospects for the Future," *Statistical Science*, vol. 6, no. 3, pp. 219–232, 1991.

[12] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee, "Toward Privacy in Public Databases," *Theory of Cryptography*, pp. 363–385, 2005.

[13] K. Muralidhar and R. Sarathy, "Data Shuffling-A New Masking Approach for Numerical Data," *Management Sci.*, vol. 52, no. 5, pp. 658–670, 2006.

[14] D. Ting, S. E. Fienberg, and M. Trottini, "Random Orthogonal Matrix Masking Methodology for Microdata Release," *International Journal of Information and Computer Security*, vol. 2, no. 1, pp. 86–105, 2008.

[15] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM*, vol. 24, pp. 84–90, 1981.

[16] ——, "Elections with Unconditionally-Secret Ballots and Disruption Equivalent to Breaking RSA," *Proc. of EUROCRYPT'88*, pp. 177–182, 1988.

[17] K. Peng, R. Aditya, C. Boyd, E. Dawson, and B. Lee, "Multiplicative Homomorphic E-Voting," *Proc. of INDOCRYPT 2004*, pp. 61–72, 2005.