No More Excuses: Automated Synthesis of Practical and Verifiable Vote-counting Programs for Complex Voting Schemes

Thomas Haines[1], Dirk Pattinson[2], and Mukesh Tiwari[2]

[1] Polyas, Denmark
[2] Research School of Computer Science, ANU, Canberra

Abstract.

1 Introduction

(leave this until the very end)

2 The Schulze Method

(this should be similar to other papers that we already have)

3 Format of Ballots

¡¡¡¡¡¡¡ HEAD Each ballot, B, is a $n\times$n matrix where n is the number of candidates participating in election. Each voter marks the entry (i,j) in ballot B as 1 if he prefers candidate i over j otherwise mark it 0. No candidate is preferred over itself, so all the diagonal entries are filled with 0. The reason for representing a ballot as a matrix because the precise

- motivate the choice of ballots and compare with plaintext version
- define encryption function that changes representation
- a ballot is valid if and only if there is evidence that proves its validity
- encrypting valid ballots leads to valid ballots, and the same for invalid ballots
- Each ballot, B, is a $n\times$n matrix where n is the number of candidates participating in election. Each voter marks the entry (i, j) as 1 and (j, i) as -1 in ballot B if he prefers candidate i over j. If the voter has no preference between candidate i and j i.e. they are ranked same then voter would mark entry (i, j) and (j, i) as 0. No candidate is preferred over itself, so all the diagonal entries are filled with 0. The reason for representing a ballot as a matrix because the precise
  nature of our algorithm involving no decryption of individual ballot to protect ballot privacy, facilitating the scrutiny for validity, and ease of computing cumulative margin over encrypted data to compute the winners.

Each entry in ballot is encrypted by additive ElGamal i.e. encrypting the message m as $(g^r, h^r g^m)$ where g is generator of cycle group G, and $h = g^x$ is private key. After performing the encryption, a plain text ballot i.e. matrix having entries of 0, and 1 changes to a matrix having pair for

– Each entry in ballot is encrypted by additive ElGamal i.e. encrypting the message m as $(g^r, h^r g^m)$ where g is generator of cycle group G of order q, private key x and randomness r are randomly chosen from $\{1, \ldots, q - 1\}$, and public key is computed as $h = g^x$. It is important to note that generator g, private key x and public key h are generated during key generation phase of Elgamal algorithm while randomness r is generated fresh for each message.

After performing the encryption, a plain text ballot i.e. matrix having entries of -1, 0, and 1 changes to a matrix having pair for each entry representing the encryption of corresponding plain text. A representation of plain text ballot

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \ldots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \ldots & x_{2n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{n1} & x_{n2} & x_{n3} & \ldots & x_{nn} \end{bmatrix}$$

and data structure used to represent this ballot in Coq is

```
Definition plaintext := Z.
Definition pballot := cand -> cand -> plaintext.
```

A representation of encrypted ballot

$$\begin{bmatrix} (y_{11}, z_{11}) & (y_{12}, z_{12}) & (y_{13}, z_{13}) & \ldots & (y_{1n}, z_{1n}) \\ (y_{21}, z_{21}) & (y_{22}, z_{22}) & (y_{23}, z_{23}) & \ldots & (y_{2n}, z_{2n}) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ (y_{n1}, z_{n1}) & (y_{n2}, z_{n2}) & (y_{n3}, z_{n3}) & \ldots & (y_{nn}, z_{nn}) \end{bmatrix}$$

and the corresponding data structure in Coq is

```
Definition ciphertext := (Z * Z)%type.
Definition eballot := cand -> cand -> ciphertext.
```

The reason for using additive ElGamal is that it's composability on ciphertext under same public key. In short, (Should I explain the algorithm here that why this scheme works or explain it encryption section ?)

$$enc_k(m_1) * enc_k(m_2) = enc_k(m_1 + m_2)$$

Our choice of representing ballot as matrix comes with price. A malicious voter could try to encode all sort of ballots including cyclic ones i.e. candidate A is preferred over candidate B, candidate B is preferred over candidate C, and candidate C is preferred over candidate A, and it is clear that voter has not

expressed his choices clearly as it can not be expressed in linear preference order and should not be considered for counting, or he could inflate his favourite candidate by any other number higher that 1, and It makes the ballot invalid because any value greater that 1 in a ballot would be similar to voter has casted more than one vote. We define a ballot is valid if all the entries in matrix is encryption of -1, 0 and 1, and there is no cycle in it. Formally, We have

```
Definition matrix_ballot_valid (p : pballot) :=
      (forall c d : cand, In (p c d) [-1; 0; 1]) /\
       valid cand p.
```

```
 Definition valid (A : Type) (P : A -> A -> Z) :=
      exists f : A -> nat,
          forall c d : A,
          (P c d = 1 <-> (f c < f d)%nat) /\
          (P c d = 0 <-> f c = f d) /\
          (P c d = -1 <-> (f c > f d)%nat)

     (* Which one should keep ? Upper one is in Code while this one is
        more intutive after expanding the definition *)
    Definition matrix_ballot_valid (p : pballot) :=
      (forall c d : cand, In (p c d) [-1; 0; 1]) /\
      (exists f : A -> nat,
          forall c d : A,
          (p c d = 1 <-> (f c < f d)%nat) /\
          (p c d = 0 <-> f c = f d) /\
          (p c d = -1 <-> (f c > f d)%nat))
```

Intuitively, it says that a ballot p is valid if we can find a function which expresses the notion of linear preference ordering for all candidates, and we prove that we can decide for each ballot if it is valid or not

```
  Lemma matrix_ballot_valid_dec :
        forall p : pballot, {matrix_ballot_valid p} +
                            {~matrix_ballot_valid p}.
```

By now, a attentive reader would note that we have claimed that we don't decrypt the ballot to preserve the privacy, so how can we decide the validity without decrypting the original ballot. We will discuss our whole cryptographic scheme in next section, but to quench the thrust of reader for now, we have proved that if ballot p was valid before encryption, then encrypting it followed by permuting each row by secret permutation σ and permuting again the each

column of resulting matrix from previous step by same permutation, σ, and decrypting it back would still lead to valid ballot. (* Write this same Lemma for $matrix_b allot_v alid$ then any permutation of it is valid and if it is invalid then any permutation of it is invalid. The proof is already lying there in other proofs so it's basically copy paste. *)

```
Lemma perm_presv_validity :
    forall (A : Type) (P : A -> A -> Z),
      (forall c d : A, {c = d} + {c <> d}) ->
      (forall c d : A, {P c d = 1} + {P c d = 0} + {P c d = -1}) ->
      forall sig : A -> A, Bijective sig -> valid A P <-> valid A (perm A P sig)
```

And similarly, for invalid ballot

```
Lemma not_perm_persv_validity :
 forall (A : Type) (P : A -> A -> Z),
      (forall c d : A, {c = d} + {c <> d}) ->
      (forall c d : A, {P c d = 1} + {P c d = 0} + {P c d = -1}) ->
      forall sig : A -> A, Bijective sig -> ~ valid A P <-> ~ valid A (perm A P sig).
```

4 Homomorphic Computation of Margin

- motivate and introduce how the margin is computed homomorphcially
- prove that encrption commutes with computation of margin

5 Homomorphic Counting

- prove that encruption commutes with computation of winners
- discuss the evidence: publish cleartext margin function
- describe certificates and how to check them

6 Experimental Results

- how long does it take, and how much memory?
- how large are the certificates?

7 Discussion and Further Work

One aspect that we have not considered here is encryption of ballots to safe-guard voter privacy which can be incorporated using protocols such as shuffle-sum [**?**] and homomorphic encryption [**?**]. The key idea here is to formalise a given voting scheme based on encrypted ballots, and then to establish a homomorphic property: the decryption of the result obtained from encrypted ballots is the same as the result obtained from the decrypted ballots. We leave this to further work.