An Efficient Weighted Voting Protocol with Secret Weights

Toru Nakanishi, Shinji Nakatake, Nobuo Funabiki and Yuji Sugiyama

Dept. of Communication Network Engineering, Okayama Univ., Japan
What’s a weighted voting?

- Conventional voting:
  All votes are tallied equally.

- Weighted voting:
  Votes are tallied in proportion to their weights.

  Weight 10
  Voter A → yes → 10 “yes”

  Weight 20
  Voter B → no → 20 “no”

E.g., weighted voting is in demand for stockholders’ meeting
Model

- Participants:
  - Voter $V_i$ with weight $w_i$ (1 ≤ $i$ ≤ N)
  - Administration server $A$
    - ensures $w_i$ of $V_i$
    - does not leak $w_i$
  - Tallying servers $T_j$ (1 ≤ $j$ ≤ L)
    - For threshold K, K or more $T_j$ do not collude.
Model (Cont.)

- **Target:** “yes/no” voting
  - Voter $V_i$ casts $v_i = 1$ (“yes”) or $-1$ (“no”).
  - Voting result:
    $$D = \sum v_i w_i$$
    (Difference between #“yes” and #“no” including the weights)
Requirements

- **Votes secrecy:**
  Vote \( v_i \) of each \( V_i \) is kept secret.

- **Verifiability:**
  The voting result can be verifiable by anyone.

- **Weights secrecy:**
  Weight \( w_i \) of each \( V_i \) is kept secret.

  *Weight is also privacy information.*
  (In stockholder’s meeting, weight is the number of stocks of user.)
Previous work [6]

- Uses homomorphic public-key encryption $E$, such as ElGamal, by $T_j$'s key.
  - $E(m_1) \times E(m_2) = E(m_1 + m_2)$
- Vote: $E(v_i) = E(1)$ or $E(-1)$
- Result:
  $$E(v_1)^{w_1} \times \cdots \times E(v_N)^{w_N}$$
  $$= E(w_1v_1) \times \cdots \times E(w_Nv_N)$$
  $$= E(w_1v_1 + \cdots + w_Nv_N)$$

$T_j$ s decrypt $D = w_1v_1 + \cdots + w_Nv_N$

Private weighting in [6]

- For weights secrecy, $E(v_i)^w_i$ from $E(v_i)$ is computed by server A using Mix protocol.

\[ \text{Shuffle} \]

$E(v_1)$  $\rightarrow$  $E(v_?)^w_?$  $\rightarrow$  $E(v_?)^w_?$  

$E(v_2)$  $\rightarrow$  $E(v_?)^w_?$  $\rightarrow$  $E(v_?)^w_?$  

\vdots \quad \vdots \quad \vdots 

$E(v_N)$  $\rightarrow$  $E(v_?)^w_?$  $\rightarrow$  $E(v_?)^w_?$  

$vi$ and $wi$ are unlinkable
Problem of [6]

- After all votes are cast, the heavy Mix protocol are performed.
  - Mix protocol needs $O(N)$ exponentiations.
  - Even the most fast Mix protocol needs $20N$ exps.
  - e.g., for $N=1,000$, it means 20,000 exps.

Contributions

- A more efficient weighted voting protocol
  - Tallying cost is reduced.
  - (O(1) exps. only)

Mix protocol is excluded.
Idea

- $V_i$ casts $E(w_i v_i) = E(w_i)$ or $E(-w_i)$.
  - Mix protocol is not needed.

How to prove that $E(w_i v_i)$ is correct?

- Using certificate of $w_i$ issued from server $A$, $V_i$ can prove that $E(w_i v_i)$ is correct.

$w_i$ ensured by certificate should be proved in zero-knowledge
Used tool: Camenisch-Lysyanskaya signature scheme[1]

- Provably secure against adaptive attacker without random oracle
- The knowledge of the signature and messages can be proved in zero-knowledge.
  i.e., ZPK of \( (m, s) \) for \( s=\text{Sig}(m) \)
  (ZPK: zero-knowledge proof of knowledge)

Our protocol: Certification issue

- In advance, $V_i$ is issued a certificate of $w_i$ from server A.
  - The certificate: Camenisch-Lysyanskaya signature $\text{Sig}(w_i)$. 

```
  V_i  Sig(w_i)  A
```
Our protocol: Casting

During a voting, $V_i$ casts $E(w_iv_i)$, i.e., $E(w_i)$ or $E(-w_i)$, to $T_i$s, together with the correctness proof.

ZPK of $(w_i, s_i)$ s.t.
- $[e = E(w_i)$ and $s_i = \text{Sig}(w_i)]$
- or $[e = E(-w_i)$ and $s_i = \text{Sig}(w_i)]$

- Encryption and ZPK □ Votes secrecy, Weights secrecy
- Correctness Proof of ZPK □ Verifiability
Our protocol: Tallying

- T_j's cooperatively decrypt

\[ E(w_1v_1) \times \cdots \times E(w_Nv_N) \]
\[ = E(w_1v_1 + \cdots + w_Nv_N) \]
\[ = E(D) \]

Exps are needed only in one threshold decryption.
(no mix protocol)

Tallying time is largely reduced!
Conclusion

- A more efficient weighted voting protocol
  - Tallying cost is reduced.
    (O(1) exps. only)

Future work

- Extension to voting for multiple candidates.