Online voting: measure of privacy and verifiability

boire.sebastien

November 2020

1 First scheme: public storage of a subset of possible vote values

1.1 Description

We consider an election with candidates $C_1, ..., C_K$. Voter V votes for C_i, and the value of the vote is stored encrypted so that it can be counted, verified by V, and secret. In addition, (k-1) other candidates are chosen at random, and a public ticket is produced: "Voter V voted for C_{i_1} or C_{i_2} or ... or C_{i_k}.", with one of the C_{i_j} being C_i, the correct value.

1.2 Evaluation of verifiability

We note X_j^i the binary variable corresponding to "Does voter j voted for candidate i?". Then:

$Y_\alpha^i = X_1^i + ... + X_\alpha^i$ is a binomial variable following $\mathcal{B}(\alpha, p_i)$, with p_i probability of vote for candidate i.

In a similar way, we note $\tilde{X}_j^i$ the binary variable corresponding to "Does the public vote of voter j contains candidate i as a possible vote?".

$\tilde{Y}_\alpha^i = \tilde{X}_1^i + ... + \tilde{X}_\alpha^i$ is a binomial variable following $\mathbb{B}(\alpha, \tilde{p}_i)$, with $\tilde{p}_i$ probability that candidate i is in a public vote.

We have:

$\tilde{p}_i = p_i + (1 - p_i) * \frac{k-1}{K-1}$

We use Bienayme-Tchebychev inequality on $\tilde{Y}_\alpha^i$:

$\mathbb{P}(|\dfrac{\tilde{Y}_\alpha^i - \alpha * \tilde{p}_i}{\alpha}| > x) \leqslant \dfrac{\tilde{p}_i(1 - \tilde{p}_i)}{\alpha x^2}$

Using the public votes of α voters, we can compare the scores we obtain for each candidate to the global results of the election, and measure whether the difference is likely or not. This provides a measure of verifiability (we just need to choose a level of verification with the term $\dfrac{\tilde{p}_i(1 - \tilde{p}_i)}{\alpha x^2}$).

1.3 Evaluation of privacy

Privacy can be defined in multiple ways. Here, we will measure the entropy on one vote, as the amount of unknown information to an attacker to know one vote.

$Privacy = H(vote) = -\sum_{c \in candidates} p(c) log_2(p(c)) = log_2(k)$

In this, we do not take into account that each candidate has a probability p_i of being the real value of the vote (to simplify calculations). This corresponds to the situation where each candidate obtains as much vote as any other.

2 Second scheme: public storage of the votes under probabilistic value

2.1 Description

The election is in the same configuration as in the previous section. However, before the the voter chooses who he votes for, he sees a set of random variables appearing on the screen: $R_1, ..., R_K$. They all follow a Gaussian law, with R_1 following a $\mathcal{N}(\mu, \sigma^2)$, and all the other R_i following a $\mathcal{N}(\frac{1-\mu}{K}, \sigma^2)$. Then, the voter chooses who he votes for, and the variables R_i are rotated so that R_1 is associated to the candidate chosen by the voter. The values of the R_i are then made public.

μ is chosen such that $\mu > \frac{1-\mu}{K}$, so that on average we can distinguish the correct candidate. The bigger the difference, the easier it is to identify the real value of the vote. In addition, σ quantifies the randomness of the values of R_i, and also takes part in how easy it is to identify the real value of a vote.

This structure allows the voter to be sure that he is not duped on the value of his public vote, since the random variables R_i are evaluated before the voter enters his vote.

2.2 Verifiability evaluation

2.3 Privacy evaluation

3 Third scheme: Privacy of the votes depending on subgroups of the voting population

References