An Electronic Voting Protocol for General Elections over the Internet

Dimitrios Poulakis, Department of Mathematics, Aristotle University of Thessaloniki

Electronic voting schemes are a natural alternative to traditional voting. Electronic voting can solve the cost problem that occurs in the traditional elections as well several other problems as non-mobility, inconvenience, and so forth. Thus, the design of an efficient electronic voting system is one of the most important research issues in contemporary Internet applications. In this paper, we present a quite simple electronic voting scheme that is suitable for large scale voting over Internet and satisfies the following requirements: Fairness, Eligibility, Uniqueness, Uncoercibility, Anonymity, Accuracy, Efficiency, Robustness, Mobility, Convenience, General Election, Vigorousness of Authorities, and Verifiability.

Our electronic voting scheme includes the following parties:

- *Voters:* $v_1, \ldots, v_M$. The people who have the right to vote.
- *Two Voting Centers (VC_1, VC_2):* Two web sites responsible to collect the ballots, to check their validity and to tally the votes.
- *Authentication Center (AC):* A web site responsible to certify the legitimate voters.
- *Supervision Center (SC):* A web site responsible to supervising the whole procedure and announcing the final result.
- *Public Proxy Server (PPS):* The proxy server takes the responsibility to forward anonymous votes without an IP address that identifies the voter.

We assume that the VC_1 and VC_2 will not conspire simultaneously. Every voter can chooses m among k options ($m \leq k$). A valid ballot is a vector $(\epsilon_1, \ldots, \epsilon_k) \in \{0,1\}^k$ with at most m nonzero coordinates. If a voter votes for the j-option, then he sets $\epsilon_j = 1$. Otherwise, he sets $\epsilon_j = 0$. Thus, the number of possible ballots is $\sum_{i=1}^{m} C(k,i) \leq 2^k$.

Initialization Phase

1. *AC* selects two large prime $p < q$ such that $\gcd((p-1)/2, q-1) = 1$, the factorization of $n = pq$ is infeasible and an integer $N > M$ with $N^k < \sqrt{n}$. Next, it selects $g_p, g_q \in \{1, \ldots, n-1\}$ with $\mathrm{ord}_n(g_p) = p-1$ and $\mathrm{ord}_n(g_q) = q-1$, an one-way function $h : \mathbb{Z}_n \to \mathbb{Z}_n$ and $e, d \in \{1, \ldots, \phi(n)-1\}$ with $ed \equiv 1 \pmod{\phi(n)}$. *AC* keeps secret (p, q, d) and makes public (n, g_p, g_q, N, e, h).
2. *AC* sends to every authorized voter v_i a different even number $a_p^i \in \{1, \ldots, p-1\}$.
3. VC_1 selects $x_p \in \{1, \ldots, [\sqrt{n}]\}$, computes $y_p = g_p^{x_p} \pmod{n}$ and publishes it.
4. VC_2 selects $x_q \in \{1, \ldots, [\sqrt{n}]\}$, computes $y_q = g_q^{x_q} \pmod{n}$ and publishes it.
5. VC_1 (respectively VC_2) computes all the integers $\epsilon = \epsilon_1 N^{k-1} + \cdots + \epsilon_k$, where $(\epsilon_1, \ldots, \epsilon_k) \in \{0,1\}^k$ and has at most m nonzero coordinates. Let E be the set of these integers. Next, for every $\epsilon \in E$, VC_1 (respectively VC_2) computes $G(\epsilon) = g_q^{\epsilon} \pmod{n}$ and constructs a table T containing the couples $(G(\epsilon), \epsilon)$ ($\epsilon \in E$) sorted by the increasing order of numbers $G(\epsilon)$. Thus, using binary search, one can easily decides in time $O(k)$, if a given integer is one of $G(\epsilon)$ and in this case to find ϵ.

Authentication Phase

1. The voter v_i chooses randomly an even integer $a_q^i \in \{1, \ldots, [\sqrt{n}]\}$ and computes $A_i = g_p^{a_p^i} g_q^{a_q^i} \pmod{n}$. Next, v_i computes $\epsilon(i) = \epsilon_1^i N^{k-1} + \cdots + \epsilon_k^i$ and $B_i = y_p^{a_p^i} y_q^{a_q^i} g_q^{\epsilon(i)} \pmod{n}$. Note that, since the exponents a_p^i are different, the numbers A_i are also different.
2. The voter v_i selects $r_i, s_i \in \{1, \ldots, n-1\}$ with $\gcd(r_i, n) = \gcd(s_i, n) = 1$, computes $Y_i = r_i^e h(A_i) \pmod{n}$, $Z_i = s_i^e h(B_i) \pmod{n}$ and sends (Y_i, Z_i) to *AC*.
3. *AC* computes $\tilde{Y}_i = Y_i^d \pmod{n}$, $\tilde{Z}_i = Z_i^d \pmod{n}$ and sends $(\tilde{Y}_i, \tilde{Z}_i)$ to v_i. *AC* performs this step only one time for every voter.
4. v_i computes $U_i = r_i^{-1} \tilde{Y}_i \pmod{n}$, $W_i = s_i^{-1} \tilde{Z}_i \pmod{n}$. Then U_i and W_i are the signatures of A_i and B_i, respectively.

Voting Phase

1. v_i sends (A_i, B_i, U_i, W_i) to VC_1, VC_2 and SC through PPS.

2. VC_1, VC_2 and SC receive (A_i, B_i, U_i, W_i) and check the signatures U_i and W_i of A_i and B_i, respectively. If the two signatures are valid, they accept the quadruple.

Counting Phase

1. After the voting deadline, AC sends p, q to VC_1 and VC_2. Moreover, VC_1 and VC_2 exchange x_p and x_q.

2. Since $\gcd((p-1)/2, q-1) = 1$, we have $\gcd((p-1)/2, (q-1)/2) = 1$. So, VC_i $(i = 1, 2)$ computes $r_p, r_q \in \mathbb{Z}$ such that $r_p(p-1) + r_q(q-1) = 2$. For the decryption of (A_j, B_j), VC_i computes $M_p^j = (A_j^{-x_p} B_j)^{(q-1)/2} \pmod n$, $M_q^j = (A_j^{-x_q} B_j)^{(p-1)/2} \pmod n$ and $E_j = (M_p^j)^{r_q} (M_q^j)^{r_p} \pmod n$. We have $g_q^{\epsilon(j)} = F_j \pmod n$. Next, VC_i computes $\log_{g_q} F_j = \epsilon(j)$ $(j = 1, \ldots, M)$ using the table T. If $F_j \notin T$, then the ballot is not valid.

3. Suppose that VC_i finds that (A_j, B_j, U_j, W_j) $(j = 1, \ldots, L(i))$ are all the quadruples containing the valid ballots. For $j = 1, \ldots, L(i)$, VC_i computes $\epsilon(j) = \epsilon_1^j N^{k-1} + \cdots + \epsilon_k^j$ and next $z(i) = \sum_{j=1}^{L(i)} \epsilon(j)$. VC_i sends $z(i)$ to SC and the quadruples (A_j, B_j, U_j, W_j) $(j = 1, \ldots, \Lambda(i))$ containing the no valid votes.

Publication Phase

1. SC compares the integers $z(1)$ and $z(2)$ sent by VC_1 and VC_2, respectively. If $z(1) = z(2) = z$, then SC accepts the result of the tally. Next, SC computes the representation of z in base N: $z = z_1 N^{k-1} + \cdots + z_k$. The voting outcome is the vector $(z_1, \ldots, z_k)$.

2. SC announces the voting outcome $(z_1, \ldots, z_k)$.

3. SC selects $b_p^j \in \{0, \ldots, [\sqrt{n}]\}$, computes $C_j = g_p^{b_p^j} \pmod n$, $D_i = y_p^{b_p^j} B_j \pmod n$ $(j = 1, \ldots, M)$ and publishes the list Λ_1 of triples (A_j, C_j, D_j) $(j = 1, \ldots, L = L(i))$ containing the valid ballots sorting by increasing order of integers A_i and constructs another list Λ_2 with the triples (A_j, C_j, D_j) $(j = L+1, \ldots, M)$ containing the no valid ballots.

SC can convince a person X that the quadruples of the list Λ_1 give the voting outcome $(z_1, \ldots, z_k)$. First, SC makes public p and q. Since (A_j, C_j, D_j) $(j = 1, \ldots, L)$ and z are public, everyone can compute the quantities $A = \prod_{i=1}^{L} A_i$, $C = \prod_{i=j}^{L} C_j \pmod n$, $D = \prod_{i=j}^{L} D_j \pmod n$, $\Delta = D g_q^{-z} \pmod n$. On the other hand, the computation of $\Gamma = \prod_{i=1}^{L} y_p^{a_p^i + b_p^i} y_q^{a_q^i} \pmod n$ cannot be done public because a_p^i and a_q^i are not public. Using a version of Chaum-Pederson protocol, SC can prove to X that $\Delta^{(q-1)/2} \equiv \Gamma^{(q-1)/2} \pmod n$ and $\Delta^{(p-1)/2} \equiv \Gamma^{(p-1)/2} \pmod n$ which is equivalent to $\Gamma = \Delta$. Next, it is shown that the validity of the congruence $D \equiv g_q^z \Delta \pmod n$ can convince X that the voting outcome is correct. Finally, we show that our electronic voting protocol verifies all the aforementionned properties of security.