Published Encrypted Rosters: Technical Summary

David Chaum, Jeremy Clark, Neal McBurnett, Nan Yang

Abstract

We propose a practical technique for effectively auditing lists of voters without the authority in possession of the actual voter identities on the list releasing any individually identifiable data. Such auditability is novel and important, since rosters of registered voters are recognized as fundamental to election integrity but are sometimes partially withheld or made difficult or expensive to obtain.

Our Goals

An authority in possession of a list of voter identities can keep them confidential, using the techniques presented here, even while in some sense making them public in the form of a "Published Encrypted Roster." The authority can then demonstrate a number of useful properties of the roster, while it remains public in encrypted form.

One such property, responsive to an inquiry by an individual in this case, is the presence or absence of that individual voter on the published roster. The identity of the particular individual making the request is authenticated, so as to inhibit the possibility of illicit probing.

A second property that can be shown about a published encrypted roster is that no voter appears in the roster more than once.

A third property the authority can prove about a published encrypted roster is how many of the persons listed in a challenge, such as that supplied by a third-party auditor, are in the roster.

A published encrypted roster can also be used to randomly but verifiably sample voters for "Random-Sample Voting." The techniques presented here can establish the maximum percentage of disagreement between a published encrypted roster and one or more challenges.

We are concerned with efficiency. In principle, any question can be answered in zero-knowledge. In practice, rosters can be millions of entries long, and the complexity of textbook zero-knowledge proofs becomes impractical. Thus, we only study those questions which we think or know have practical solutions.

Our Solutions

The election authority (EA) makes public an additively homomorphic public-key encryption scheme $\mathcal{E}$ along with its public keys.

The plaintext of the roster consists of k 128-bit integers $x_1 < \ldots < x_k$. This is known only to the EA.

The main encrypted roster (call it L_1) consists of a list of ciphertext triples. This is a sorted "linked list" of encryptions of the form

$$L_1 = \langle v_1 = (\mathcal{E}(x_1), \mathcal{E}(\Delta_1), \mathcal{E}(x_2)), v_2 = (\mathcal{E}(x_2), \mathcal{E}(\Delta_2), \mathcal{E}(x_3)), \ldots \rangle$$

where $\Delta_i = x_{i+1} - x_i$. That is, in the sense of

$$(a, b, c), (c, d, e), (e, f, g), \ldots$$

Associated with the main roster is a "disposable" list L_2. This second list is a permutation of the the first list's v_i's and re-randomizations of their ciphertexts. That is if π is some permutation on k then

$$L_2 = \left\langle v_{\pi(1)}, v_{\pi(2)}, \ldots \right\rangle$$

along with a zero-knowledge proof that it was permuted and re-randomized correctly. This is done by the EA and we will call this a "re-mixing" of the roster.

The second, permuted list may need to be discarded after each use. This is to ensure that the auditor does not learn any additional information about the roster except for the answer to his query. However, these permuted rosters can be pre-computed before any interactions with auditors.

An auditor can verify the following offline, without interacting with the EA.

1. Is L_2 a permutation of L_1?
2. Is L_1 sorted?

An auditor can interactively verify the following with the EA.

1. Is some 128-bit integer y on the roster? That is, is there an index j such that the first element of v_j decrypts to y?
2. I have a set $Y = \{y_1 < \ldots < y_n\}$. Exactly which ones are on the roster, and which are not?
3. Are there *at least/at most/exactly* m/n elements of Y on the roster (without learning their identities)?

Open Problems

1. (Spelling/error detection) How to check whether encodings of names are spelled correctly, or check for alternate spellings?
2. (Disitributed roster/audit) Is it possible to have the roster split among different, mutually-untrusting EAs; or have different auditors combine their lists without revealing their content?
3. Defining the security of PERs in more rigorous terms. Having it UC-secure?