Poster/Paper Title: Verifiable Homomorphic Tallying for the Schulze Vote Counting Scheme

Bibliographic Reference: Haines T., Pattinson D., Tiwari M. (2020) Verifiable Homomorphic Tallying for the Schulze Vote Counting Scheme. In: Chakraborty S., Navas J. (eds) Verified Software. Theories, Tools, and Experiments. VSTTE 2019. Lecture Notes in Computer Science, vol 12031. Springer, Cham. https://doi.org/10.1007/978-3-030-41600-3_4

Abstract: The encryption of ballots is crucial to maintaining integrity and anonymity in electronic voting schemes. It enables, amongst other things, each voter to verify that their encrypted ballot has been recorded as cast, by checking their ballot against a bulletin board.

We present a verifiable homomorphic tallying scheme for the Schulze method that allows verification of the correctness of the count—on the basis of encrypted ballots—that only reveals the final tally. We achieve verifiability by using zero knowledge proofs for ballot validity and honest decryption of the final tally. Our formalisation takes places inside the Coq theorem prover and is based on an axiomatisation of cryptogtaphic primitives, and our main result is the correctness of homomorphic tallying. We then instantiate these primitives using an external library and show the feasibility of our approach by means of case studies.

Keyword: Electronic Voting, Theorem Proving, Cryptography, Zero-Knowledge-Proof

In preferential voting schemes, universal verifiablity can reveal your ballot if there is a large number of candidates. How can we solve this?

Verifiable Homomorphic Tallying for the Schulze Vote Counting Scheme

Mukesh Tiwari, Dirk Pattinson, Thomas Haines

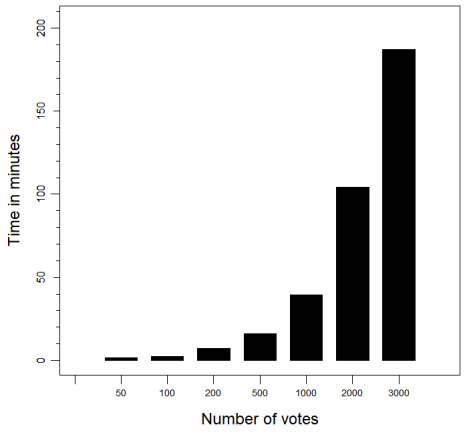
1 BACKGROUND AND PROBLEM

Universal verifiablity allows anyone to check that the announced result is correct. However, it may lead to coercion and vote selling.

2 METHODS

1. Compute the final tally homomorphically from encrypted ballots
2. Decrypt the final tally to compute winners and losers
3. Augment the scrutiny sheet with Zero-Knowledge-Proofs about various claims

3 RESULTS



4 SOFTWARE INDEPENDENCE

- Scrutiny Sheet for independent verification
- Implementation is formally verified in Coq

DETAILS

Attack: In a election, a coercer would ask a voter to mark her first and the rest of the candidates in certain order (a unique permutation which would serve as an identifier for the voter).

Feasibility of Attack: Dr Kevin Bonham, a political reporter from Tasmania, was able to link 15 similar ballots posted on bulletin board to a particular family on Facebook.

Additive ElGamal Encryption: $(g^r, h^r g^m)$

Homomorphic Property: $(g^{r_1}, h^{r_1} g^{m_1}) * g^{r_2}, h^{r_2} g^{m_2}) = (g^{r_1+r_2}, h^{r_1+r_2} g^{m_1+m_2})$

Zero-Knowledge-Proof: sigma protocols are efficient way to achieve zero-knowledge-proof. A concrete example of sigma protocol is Schnorr protocol, where the goal of a prover P is to prove the knowledge of discrete log in a Group of order q (q is prime) to a verifier V. Furthermore, g is the generator of group G, x is the public input, and w is private input with relation $x = g^w$. The protocol follows:

1. Prover P randomly selects an element r from [0 . . . q), computes $a = g^r$ and sends a to verifier V
2. Verifier V randomly selects an element c from [0 . . . q) and sends it to P
3. Prover P sends $z = r + c * w$ to V. V checks $g^z = a * x^c$

Schulze Method is a preferential voting scheme, which rests on relative margins between two candidates, i.e. the number of voters that prefer one candidate over another.

Australian National University

Download the paper →