VoteVerifier: Independent Verifier for UniVote

Subject: Computer Science
Thesis advisors: Prof. Dr. Eric Dubuis, Prof. Dr. Rolf Haenni
Expert: Han van der Kleij (SBB)

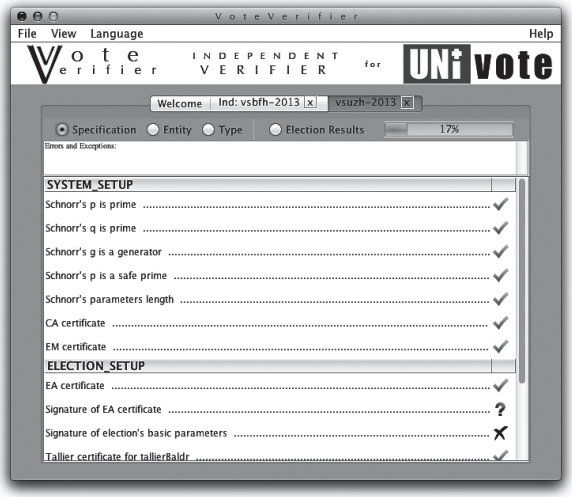
VoteVerifier is a verification software implemented to confirm the veracity and accuracy of elections held with the UniVote e-voting system. The software verifies the cryptographic data created in the course of an election, which assures voters that their vote was accurately recorded, kept confidential, and that no cheating or tinkering occured with election results. The VoteVerifier software provides participants in the elections of Swiss universities an extra boost of confidence in the integrity of the university student-governmental system.

Introduction

The Internet plays an ever-greater role in our lives and it is therefore no small wonder that the lengthy and arduous process of holding an election be made more convenient by creating a safe and secure service to hold elections electronically. Electronic voting deserves great emphasis on security. The UniVote e-voting system creates a high level of security with proven cryptographic methods. But should blind trust be placed on the UniVote team to have honestly and diligently implemented the system?

VoteVerifier

This software provides a quick and simple tool to accompany and reinforce the validity of the UniVote e-voting system. The graphical user interface provides a user-friendly application to input parameters to start a verification process and in which the results are clearly displayed.



Verification Results for the UniVote Election in Zurich

The verification process verifies the various cryptographic data published by UniVote, which can include, for example, RSA signatures or X509 certificates. There are two types of possible verifications, universal and individual. For a universal verification our software verifies all the data relative to a specific election. An end-user can provide an election ID and at the end of the verification process he can scrutinize the verification results.

An individual verification performs a similar task, except that only a single ballot is verified. After having successfully voted in a UniVote election, the user receives an election receipt in the form of a quick response code (QR code), which must be provided to the program for the individual verification process to begin.

The program then provides the user with an overview of the verification results, as well as the election results for the political parties and candidate who took part in the election.

Conclusion

In the course of this project we have deepened our knowledge of cryptography; especially of signatures, encryption, and zero-knowledge proofs. Above all it was interesting to see how these different aspects of cryptography and security interact to create a secure system. Up until now most of our knowledge had regarded individual cases of each technology, but after this project it is much more clear how different security features compliment each other and can be used together.

Giuseppe Scalzi

Justin Springer