Static Validation of Voting Protocols

Christoffer Rosenkilde Nielsen* Esben Heltoft Andersen†
Hanne Riis Nielson‡

January 18, 2006

Due to the rapid growth in computer networks, most people nowadays have access to the internet. This makes electronic voting a viable alternative to the classic paper vote for governmental elections as well as small scale elections and surveys. However, the use of electronic voting systems introduces new ways to systematically disrupt the vote or falsify the result. If these systems are to replace the classical way of voting, the communities that hold the elections, should be convinced of their correctness.

In present work we shall present a framework for validating the security properties of electronic voting protocols and describe how this framework has been used to validate four of the five security properties for three of the most well-known voting protocols; FOO92, Sensus and E-Vox.

The work presented here was done as part of the work for achieving the master's degree as presented in the thesis by Esben Heltoft Andersen and Christoffer Rosenkilde Nielsen [1]. Parts of work also appeared in the article *Static Validation of a Voting Protocol* [6] which was recently presented in Lisbon, Portugal, at *The Second Workshop on Automated Reasoning for Security Protocol Analysis* (ARSPA'05), one of the satellite workshops of ICALP'05.

1 Design Goals of Electronic Voting Systems

In order to validate electronic voting protocols we need reasoning on the properties they have to satisfy. In our work we focus on the security properties, hence we need to identify what security properties a voting protocol should satisfy. It is important to note that the security properties for electronic voting systems differ from those of ordinary key-distribution protocols, and in [3] these security properties have been formalised into four main properties:

- **Verifiability**: A system is verifiable if the voters independently can verify that their votes have been counted correctly.
- **Accuracy**: The accuracy of a voting system is divided into three parts: (1) it is not possible for a vote to be altered, (2) a validated vote cannot be eliminated from the final tally and (3) an invalid vote cannot be counted in the final tally.
- **Democracy**: A system ensures democracy if (1) only eligible voters can vote and (2) eligible voters can only vote once.

*Email: crn@imm.dtu.dk
†Email:esben@heltoft.dk
‡Email: riis@imm.dtu.dk

- **Privacy**: In a voting system the privacy is obtained if nobody can link any vote to the voter who cast it.

Often a fifth property is added [2, 5]:

- **Fairness**: No early results from the vote can be obtained.

The validation technique presented in our work covers four of these five properties; verifiability, accuracy, democracy and fairness.

The study of proposed electronic voting protocols has revealed that security properties are informally proved and argued to be satisfied. However, as security protocols are notoriously difficult to design and analyse, techniques for formally validating the security properties are particular important, which stresses the importance of present work.

2 Framework for Validation

To automatically validate voting protocols we need some sort of strategy and our strategy is illustrated by the framework in Figure 1.

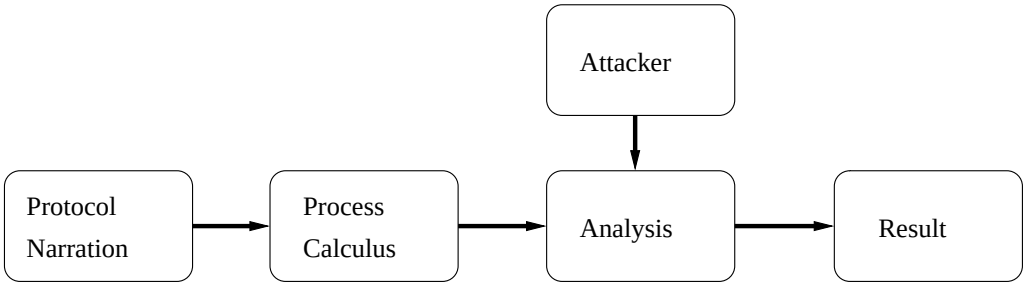

Figure 1: Framework

In order to analyse a protocol, we need a mathematical model of the protocol instead of the informal protocol narration. Hence we use process calculi to model the protocol, and in particular we have chosen the process calculus LySA.

In addition to a model of the protocol, we shall also model the environment in which the protocol is evaluated. In the scenario we consider all communication is done on a public channel; the ether. This model of the environment must be realistic with respect to the actual behavior that could occur in such a network, and therefore is all malicious behavior of the environment represented by an *attacker* model. This attacker is modeled following the Dolev-Yao approach [4].

The validation of the protocol is then performed by applying an analysis to the mathematical model of the protocol and the attacker. The technique used for this validation is known as program analysis [7].

The aim of program analysis is to approximate answers and in our case we look for conservative answers; ie. over-approximations. An over-approximation possibly includes behaviors of the program that can never occur in real-life, so if we study one specific behavior in the analysis result, we must be aware that it might be a *false-positive*. However, the over-approximation will at least hold all behaviors of the program possible, so if the analysis finds that the approximation does not violate any properties we can conclude that these properties can never be violated.

That the analysis ensures conservative answers is a result of it being *semantics based*; ie. the information obtained from the analysis can be proven sound with respect to the semantics of the process calculi analysed.

3 Results

The contribution of our work is to the area of automated analysis of networking systems. More specifically, we present a framework for automatically validating the security properties of voting protocols. Our analysis covers four of the five security properties for voting protocols, and as the analysis calculates an over-approximation of the possible behavior of the protocols, we can be certain that the validation is *sound*.

We have used this framework to validate three voting protocols; the FOO92 voting protocol, one of the most established voting protocols in the literature, the Sensus voting protocol and the E-Vox voting protocol. Of these have, to the best of our knowledge, only the FOO92 protocol been validated previously in the literature, and that was only for some of the security properties.

Our initial analysis results for verifiability and accuracy pinpoint flaws in the FOO92 and the Sensus protocols: We have identified a denial of service attack which could force the counter to repeatedly disqualify the voting process and we have identified a flaw which allows the attacker to forge the publishing of votes. However for both of these protocols we proposed an amendment, and the four security properties were subsequently validated for the amended protocols.

The E-Vox protocol had a bit different design, and the above mentioned flaws are not present in this protocol. However because of the protocol design, E-Vox cannot satisfy fairness.

References

[1] E. H. Andersen and C. R. Nielsen. Static Analysis of Voting Protocols. M.Sc. thesis, IMM-2005-1218, Technical University of Denmark, Aug. 2005.

[2] T. Asano, T. Matsumoto, and H. Imai. A study on some schemes for fair electronic secret voting. In *The Proceedings of the 1991 Symposium on Cryptography and Information Security*, SCIS91-12A, Feb. 1991. (in japanese).

[3] L. F. Cranor and R. K. Cytron. Design and Implementation of a Practical Security-Conscious Electronic Polling System. Research Report WUCS-96-02, Department of Computer Science, Washington University, 1996.

[4] D. Dolev and A. Yao. On the Security of Public Key Protocols. *Proc. 22th IEEE Symposium on Foundations of Computer Science*, pages 350–357, 1981.

[5] A. Fujioka, T. Okamoto, and K. Ohta. A Practical Secret Voting Scheme for Large Scale Elections. *Lecture Notes in Computer Science: Advances in Cryptology - AUSCRYPT '92*, 718:244–251, 1992.

[6] C. R. Nielsen, E. H. Andersen, and H. R. Nielson. Static validation of a voting protocol. In P. Degano and L. Viganó, editors, *Proc. ARSPA 2005, proceedings of the 2nd workshop on Automated Reasoning for Security Protocol Analysis*, pages 115–134. ENTCS 135(1), 2005.

[7] F. Nielson, H. R. Nielson, and C. Hankin. *Principles of Program Analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.