Security of Programs and Data for an Electronic Voting System.

Jonathan A. Enokela, Ph.D.

Department of Electrical Engineering, University of Agriculture, PMB 2373, Makurdi, Nigeria.

E-mail: jenokela@yahoo.com

ABSTRACT

An electronic voting machine was designed around a microcontroller for which the code was written in assembly language. Various code protection schemes specified by the manufacturer of the microcontroller were used to prevent inadvertent or deliberate reading and reproduction of the code contained in the microcontroller. The election data contained in the EEPROM of the microcontroller was downloaded into a central computer for tabulations. The security of data in this computer was enforced by generating digital signatures for each data file created. This process makes it impossible for anyone to substitute wrong or deliberately altered data files at any intermediate stage between the capturing of voter's intent by the machine and the final results tabulations.

(Keywords: electronic voting, microcontroller, data security)

INTRODUCTION

Elections measure the feelings of a people about a government and its policies. It is through this process that leaders are chosen in democratic governments. The transparency of elections and the accuracy of their results are of great importance as the quality of governments is related to the quality of elections that brought that government into power. The results of elections accord political power to the winner but can also cause chaos if they are improperly handled [1].

Issues about elections are deservedly given high priority and security attention. In elections conducted using the paper ballots and ballot box method these devices must be highly protected to prevent them from falling into wrong hands before and during elections [1].

The electronic voting system is becoming increasingly popular as a means of conducting elections in many democratic countries [2], [3], [4], [14], [15]. The application of electronic voting systems in elections can solve many problems associated with the traditional method of conducting elections. The problems of multiple registrations of voters, multiple voting by individuals, stuffing of ballot boxes, inflation of votes cast, and bandwagon effect can be solved [5]. It is also, however, necessary to enforce security policies at various stages of the design of an electronic voting system by proper integration of various components used in the design [13].

The inadequacy or complete absence of security policies in the implementation of many electronic voting systems has contributed to their vilification by many researchers [13], [16], [18], [19]. This has also led to the public perception that the electronic voting systems are smart devices designed to cheat them of their votes [17]. Security policies must firstly protect the code that runs in the machine. The sanctity of data generated by the machine and at various stages of results tabulation must also be assured.

MATERIALS AND METHODS

The electronic voting machine shown in Figure 1 was built around the PIC18F2685 microcontroller [6]. The firmware for the machine was written in assembly language by using the instruction set of the microcontroller. The use of the assembly language makes it virtually impossible to hack into the code of the machine. The need for voting machines to have their code written at a low level or, in fact, with their own separate language is advocated by [18]. Although the code was written in assembly language, it can still be used by another person if the person removes the microcontroller from the application circuit and downloads the code by placing it in a programmer such as the Microchip's MPLAB PM3 [3].

Figure 1: The Voter Interface of the Electronic Voting Machine.

In order to make this impossible code protection was enforced to protect various blocks of the flash program memory and the EEPROM memory of the microcontroller from external read and write operations. This was achieved by appropriately setting the configuration bits of the microcontroller [8].

The voter's preferences are captured and stored in the data EEPROM memory of the microcontroller. These results must be collated and tabulated by a host computer for fast results declarations. The Electronic Voting Machine (EVM) does this by using the built-in Enhanced Universal Synchronous/Asynchronous Receiver Transmitter (EUSART) of the microcontroller to communicate with the computer through the RS-232 serial port [9]. The results in the memory of each EVM which represent the results of a particular polling station are collated by physically connecting the EVM to a host computer. The transfer of data deliberately avoids the use of any network since network based systems are vulnerable to a number of attacks that are difficult to solve [14], [16], [20].

Once the results of the elections are in the files of the host computer these data must be protected. The protection aims at making it impossible for anyone to modify the results contained in the data files as well as to make it impossible for anyone to generate his own data and to substitute this data file in place of a genuine one. The software that retrieves data from the EVM was written using the Java programming language (JDK 1.6_10) [10].

Each data file created was signed with a digital signature using the Java security API. The file can then be read for various purposes only by using the corresponding public key [11]. This public key is known only by the program which has been archived by using the Java's jar tool [12].

RESULTS AND DISCUSSIONS

Typical data generated at various stages of compilation of results are shown in the screen shots of Figures 2 and 3. When attempts were made to read files that had no digital signatures or files that were signed with different digital signatures the "security clearance failure" screen shot shown in Figure 4 resulted.

The program halted afterwards showing that only files created by the tabulation software can be used. The code protection scheme used with the machine firmware also prevented external access to the code contained in the flash memory of the microcontroller.

CONCLUSION

Various schemes were used to protect the code of the EVM and also the software and data of the high level language program used to retrieve data from the EVM during the tabulation of results. These protection schemes proved to be successful.

The question of what could happen if the code of the machine at the assembly language level and the high level code were to be stolen and modified necessitates that codes for EVM be proprietary. This security through obscurity and the factor of trade secrets are principally why most manufacturers of Electronic Voting Machines have traditionally refused to divulge their source code [13], [21]. The necessity for EVMs to run on open source code, however, also has its many advocates [22], [23].

REFERENCES

1. Kurfi, A. 1983. *The Nigerian General Elections 1959 and 1979 and the Aftermath*". Macmillan Nigerian Publishers Ltd.: Ibadan, Nigeria.

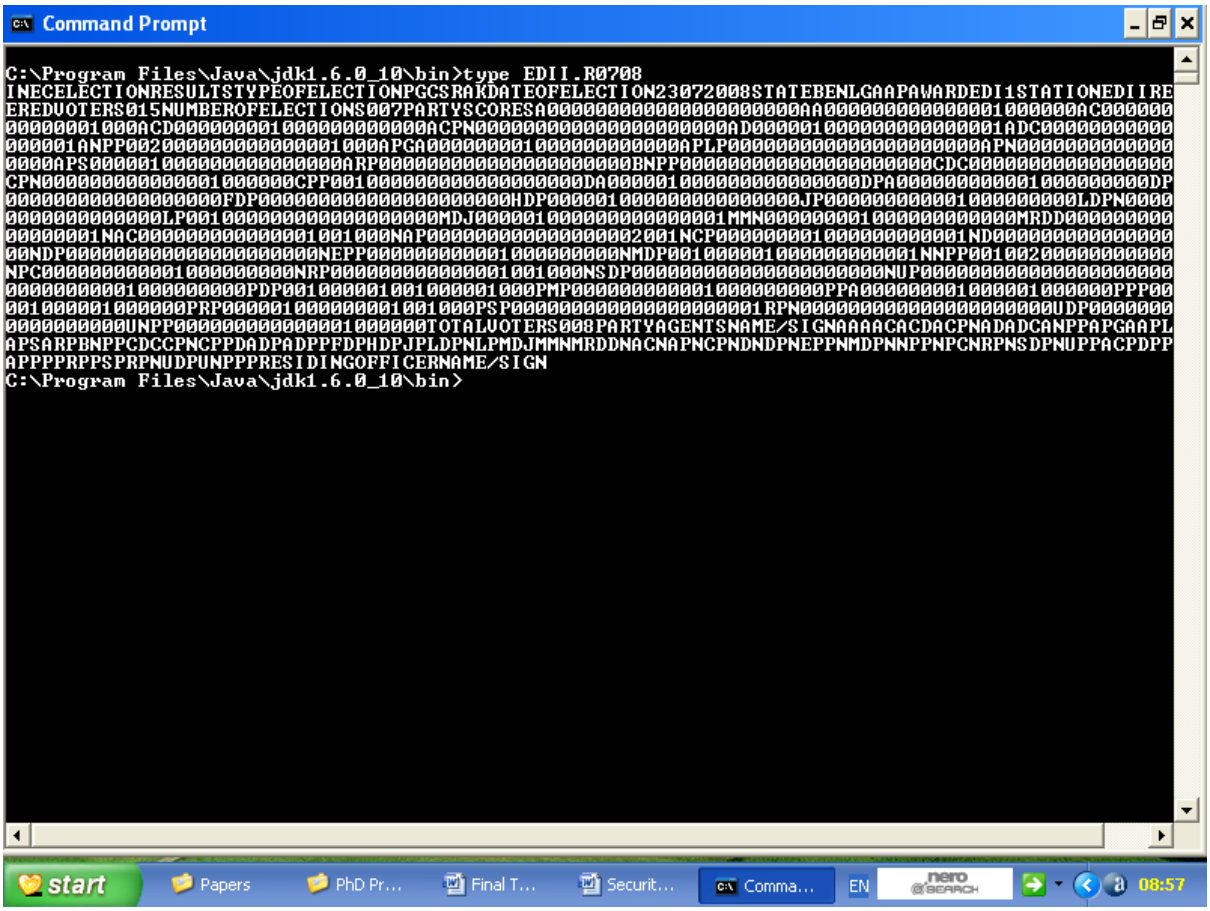

Figure 2: Compilation of Results using the EVM.

2. Election Commission of India. 2006. “Frequently Asked Questions – Electronic Voting Machines”. http://www.eci.gov.in/faq/elecvtmach.htm

3. Beroggi, G.E.G. 2008. “Secure and Easy Internet Voting”. *Computer.* IEEE Computer Society Publications. 52-56.

4. He, M., Almeida, R., and Gissoni, E. 2002. “National Semiconductor and Unisys Equip Brazil with New Voting Machines for Fast and Accurate Election Results in the Fall”. http://www.national.com/news/item/o,1735,757,00.html

5. Enokela, J.A. 2009. “The Design and Implementation of an Electronic Voting Machine”. Ph.D. Dissertation. Department of Elect. Eng., University of Nigeria: Nsukka, Nigeria.

6. Microchip Technology, Inc. 2004. “PIC18F2585/4585/2680/4680 Data sheet”. http://ww1.microchip.com/downloads/en/DeviceDoc/39625C.pdf

7. Microchip Technology, Inc. 2006. “MPLAB PM3 User’s Guide”. http://ww1.microchip.com/downloads/en/DeviceDoc/51464C.pdf

8. Microchip Technology, Inc. 2005. “MPLAB ASM30, MPLAB LINK30, and Utilities User’s Guide”. http://ww1.microchip.com/downloads/en/DeviceDoc/Asm_link_util_51317e.pdf

9. Garbutt, M. 2005. “AN774: Asynchronous Communications with PICmicro USART”. Microchip Technology, Inc.: Chandler, AZ. http://ww1.microchip.com/downloads/en/AppNotes/00774a.pdf

10. Sun Microsystems. 2009. “Java 6 Development Environment”. http:// java.sun.com/javase/downloads/

11. Sun Microsystems. 2009. “Java 6 Tutorial”. http://java.sun.com/docs/books/tutorial/security/

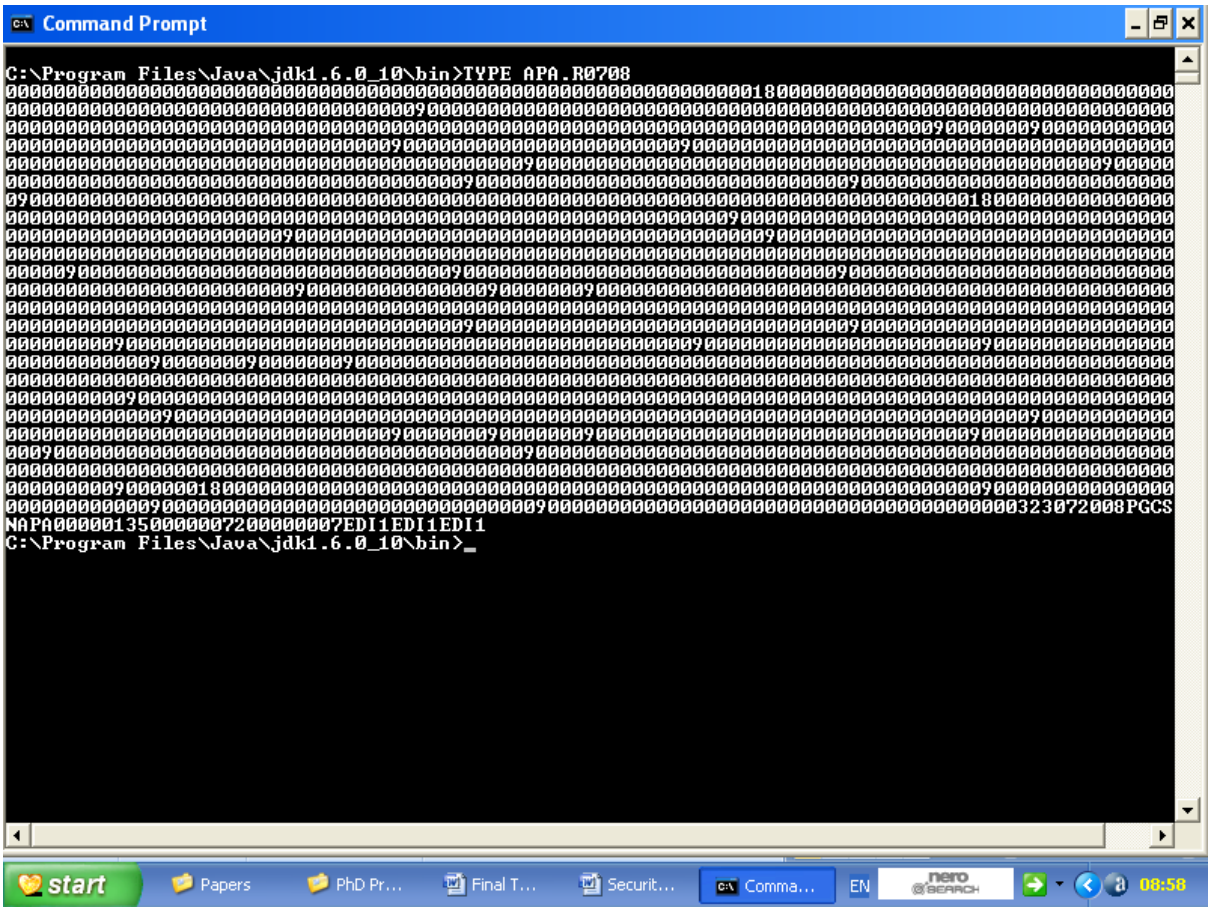

Figure 3: Further Compilation of Results using the EVM.

12. Sun Microsystems. 2009. “Java 6 Tutorial”. http://java.sun.com/docs/books/tutorial/deployment/jar/

13. Balzarotti, D., et al. 2008. “Are Your Votes Really Counted? Testing the Security of Real-World Electronic Voting Systems”. ISSTA’08. http://www.cs.ucsb.edu/~seclab/projects/voting/issta08_voting.pdf

14. Andrieu, R.J., Jose, A.O.R., and Brown, P. 2003. ”Advanced Security to Enable Trustworthy Electronic Voting”. Scytl Online World Security, S.A.: Barcelona, Spain. http://www.scytl.com

15. Degregorio, P.S. 2007. “New Voting Technology: Problem or Solution”. *e-Journal USA*. U.S. Department of State. 12(10):8-11. http://www.usinfo.state.gov/pub/ejournalusa.html

16. Jefferson, D., Rubin, A.D., Simon, B., and Wagner, D. 2004. “A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE)”. http://www.servesecurityreport.org/paper.pdf

17. Theisen, E. 2006. *Myth Breakers: Facts about Electronic Elections (2nd ed.)*. Voters Unite: Washington, D.C. http://www.votersunite.org/MB2.pdf

18. Gaines, J. 2006. “Democracy’s Downfall: Is the Computing Technology for Electronic Voting Secure and Reliable Enough for National Use?”. *ACM SIGCAS Computers and Society*. 36(4).

19. Armen, C. and Morelli, R. 2005. “Teaching about the Risks of Electronic Voting Technology”. *Proceedings of 10th Annual SIGCSE Conf. Innovation and Technology in Computer Science Education*. Monte de Caparica, Portugal. 227-231.

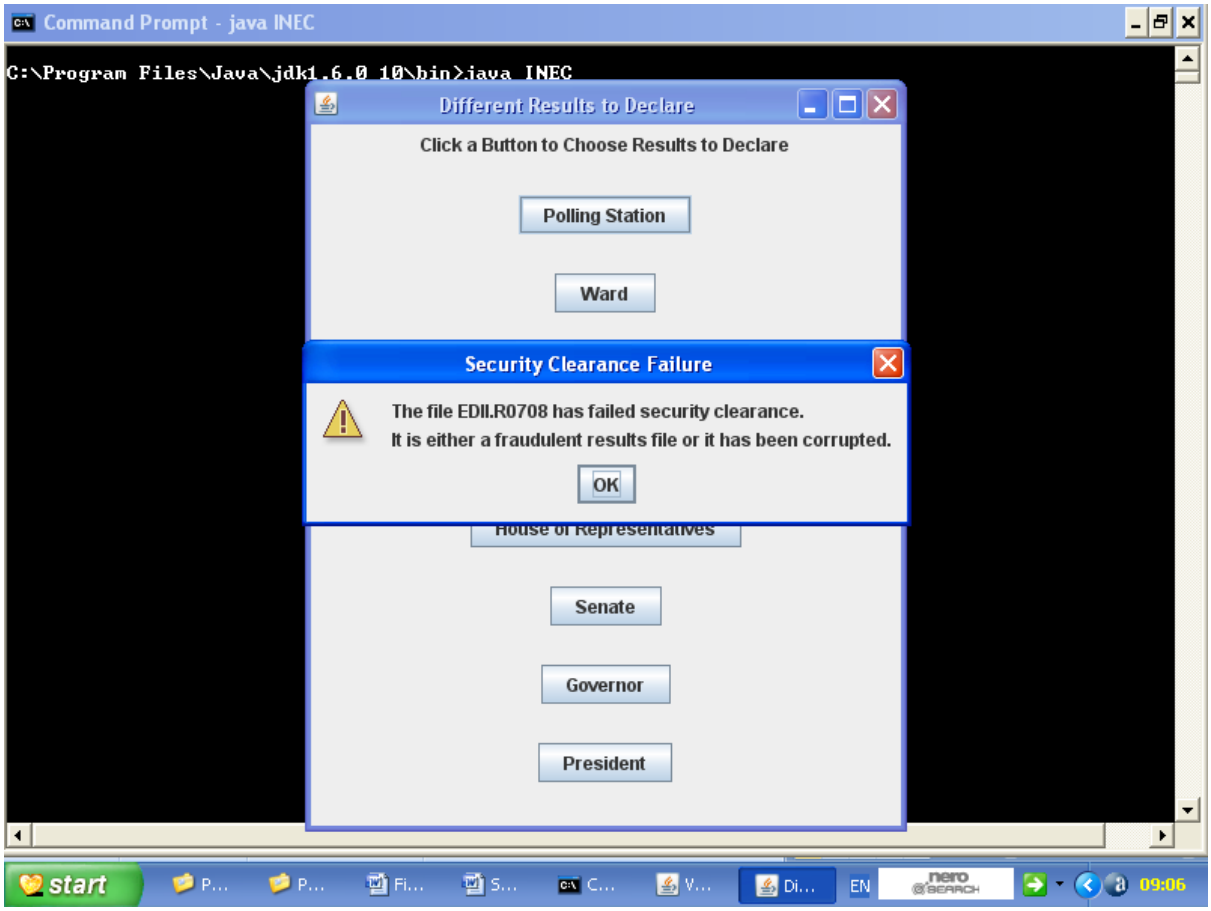

Figure 4: Results of Using a File without a Digital Signature.

20. Neumann, P., Mercuri, R., and Weinstein, L. 2000. "Internet and Electronic Voting". *The Risks Digest, ACM Committee on Computers and Public Policy*. 21(14). http://catless.ncl.ac.uk/Risks/21.14.html

21. Selker, T. and Goler, J. 2004. "The SAVE System-Secure Architecture for Voting Electronically". *BT Technology Journal*. (22): 89-95.

22. Penha-Lopes, J.M. 2005. "Why Use an Open-Source E-Voting System?". *Proceedings of. 10th Annual SIGCSE Conf. Innovation and Technology in Computer Science Education*. Monte de Caparica, Portugal. 412.

23. Parakh, A. and Kak, S. 2007. "How to Improve Security in Electronic Voting?. *ACM Ubiquity*. 8(6)

ABOUT THE AUTHOR

Engr. Dr. Jonathan A. Enokela is a Lecturer in the Department of Electrical/Electronic Engineering at the Federal University of Agriculture, Makurdi, Nigeria. He has taught various aspects of analog and digital systems design to both the undergraduate and postgraduate students. He is a registered engineer with the Council for Regulation of Engineering in Nigeria (COREN) and has a wide range of practical experiences. His research interests include embedded systems design and applications.

SUGGESTED CITATION

Enokela, J.A. 2010. "Security of Programs and Data for an Electronic Voting System". *Pacific Journal of Science and Technology*. 11(2):283-287.

Pacific Journal of Science and Technology