

UNIVERSITY OF CALIFORNIA

BERKELEY • DAVIS • IRVINE • LOS ANGELES • RIVERSIDE • SAN DIEGO • SAN FRANCISCO



SANTA BARBARA • SANTA CRUZ

CENTER FOR CLINICAL EDUCATION

SCHOOL OF LAW (BOALT HALL)
BERKELEY, CALIFORNIA 94720-7200
TELEPHONE (510) 643-4800 • FAX (510) 643-4625

SAMUELSON LAW, TECHNOLOGY & PUBLIC POLICY CLINIC

FACULTY

Deirdre Mulligan
Jack Lerner
Maryanne McCormick
Chris Hoofnagle
Aaron Burstein

LAW STUDENT INTERNS

Adamczyk, Sarah	Nagala, Sarala
Hanson, Eric	Rosloff, Genevieve
Marciniak, Sean	Schohn, Aaron
Massengale, Alan	Schwab, Mairead
McGhee, Heather	Simmons, Sarah
Murphy, Kristy	Snyder, David
	Tokoro, Jason

December 22, 2006

Secretary of State Mark Ritchie
180 State Office Building
100 Rev. Dr. Martin Luther King Jr. Boulevard
Saint Paul, MN 55155

RE: DISCLOSURE OF INFORMATION REGARDING
ELECTRONIC VOTING SYSTEMS

Dear Secretary Ritchie:

We write on behalf of Ben Adida and Ka-Ping Yee, computer scientists at Harvard University and UC Berkeley, respectively.

It recently came to our attention that Mr. C. Scott Cooper, a Minnesota resident, requested information from the Minnesota Secretary of State regarding the electronic voting machines used in Minnesota elections and the statewide voter registration system (SVRS). The former Secretary of State denied seven categories of requested information under Minnesota Statutes, section 13.37(a), claiming that it qualifies as "security information" and could not be released to the public without risking attack on the state's voting systems. We believe this claim to be in error and that releasing the requested information would be more likely to enhance the security of the State's voting system than compromise it.

Minnesota Statutes section 13.37(a) exempts from disclosure "government data the disclosure of which would be likely to substantially jeopardize the security of information, possessions, individuals or property against theft, tampering, improper use, attempted escape, illegal disclosure, trespass, or physical injury." Interpreting this statute involves significant discretion by the Office of the Secretary of State; we are writing to provide you with some technical background about the complex

but fundamental security principles that we hope will help guide your disclosure decisions in the future. We use Mr. Cooper's requests to illustrate how the general security principles we discuss can be applied to real world examples.

We are submitting this letter as concerned computer scientists and public advocacy lawyers and law students. We are not being compensated for our work reviewing Mr. Cooper's requests for disclosure and preparing this letter. We strongly believe that disclosure of voting system information that does not pose security risks is critical to ensuring the integrity of our voting systems. As you prepare to carry out your duties as Secretary of State, we hope that you will consider the principles we discuss and how they can be applied to future information requests your office may receive. We hope this information can assist you in establishing election processes that are as open, transparent, and secure as possible.

Based on our experience, expertise, and understanding of Mr. Cooper's requests, we believe that the kind of information Mr. Cooper requested should have been disclosed for several reasons. First, and most important, Mr. Cooper's requests largely do not seek information about the security of the state's voting system. Rather, his requests concern the reliability, availability, and usability of the system, and about how well it works operationally. To the extent that Mr. Cooper's requests implicate security issues, they do so only because a small fraction of the records might reveal information pertinent to the security of the system. Second, in any event, disclosure of the requested information and similar information will not jeopardize the security of the voting process. Third, more disclosure will generally increase the effectiveness of system security through the discovery and repair of system vulnerabilities. Finally, the information should be disclosed because the transparency provided by disclosure in response to requests such as this is an important means by which to ensure public confidence in Minnesota's voting processes. These reasons illustrate why disclosure of election system information is warranted to the extent that it does not reveal operationally sensitive data such as system passwords or encryption keys.

In this letter, we discuss relevant technical principles that we hope will help guide your decision-making process as you administer Minnesota's election system. The computer science community has reached widespread consensus that withholding information about electronic voting systems is detrimental to the long-term security of voting systems and public confidence in elections. We explain why public access to this information is essential and why the release of the requested information poses little if any security risk. We conclude, as we hope you will, that your office should strongly consider granting these types of information requests in the future.

Our Interest and Expertise in the Field of Voting Technology

The Samuelson Law, Technology, & Public Policy Clinic has worked closely with computer scientists Ben Adida and Ka-Ping Yee to prepare this letter on their behalf. Together we have relevant expertise in the technical, policy, and legal issues posed by electronic voting systems.

Ben Adida has extensive computer science experience, both in academia and industry. He has focused much of his scholarship on the security and design of electronic voting systems. While earning a Ph.D. in Computer Science from the Massachusetts Institute of Technology, he published several articles on voting system design, including his Ph.D. dissertation, "Advances in Cryptographic Voting Systems" supervised by renowned Professor Ronald L. Rivest. He has studied voting systems through his work with the Caltech/MIT Voting Technology Project and the MIT Lab for Computer Science, and he is currently a Postdoctoral Fellow with the Harvard Center for Research on Computation and Society.¹

Ka-Ping Yee is a Ph.D. candidate in Computer Science at the University of California, Berkeley. His research focuses on the design and development of a high-assurance voting machine. Yee has conducted systems security research with a number of leading experts in the field, including Professor David Wagner of the University of California, Berkeley. His relevant publications include "Prerendered User Interfaces for High-Assurance Electronic Voting Machines" and "Guidelines and Strategies for Secure Interaction Design."²

The Samuelson Law, Technology, & Public Policy Clinic is a part of Boalt Hall School of Law at the University of California, Berkeley. The Clinic operates much like a practice group in a law firm; law students, working under the supervision of attorneys, represent clients in many areas relating to technology and the public interest. The Clinic is the first of its kind in the country and has done significant work on cutting-edge technology law issues including privacy, intellectual property, and electronic voting. The Clinic specializes in informing and counseling policymakers, academics, scientists, and organizations about how to negotiate the complex interplay between law, technology, and public policy.

The Clinic's director, Deirdre Mulligan, is a Principal Investigator in ACCURATE (A Center for Correct, Useable, Reliable, Auditable, and Transparent Elections), an inter-disciplinary, multi-institution research center on electronic voting funded by the National Science Foundation under its CyberTrust program. Though this letter is not submitted on behalf of ACCURATE, the Clinic is deeply familiar with the legal, technical, and policy challenges presented by electronic voting and is at the forefront of the academic research on these issues.³

Security Through Obscurity is Not Effective

The computer science community is in near unanimous agreement that disclosure of system design, though it may seem counterintuitive, actually aids security by allowing vulnerabilities in the system to

¹ See <http://ben.adida.net/ben-adida-cv.pdf> for more information. Mr. Adida's curriculum vitae, including a list of his publications and professional activities, is attached to this letter as Appendix A.

² See <http://zesty.ca/cv.html> for more information. Mr. Yee's curriculum vitae, including a list of his publications and professional activities, is attached to this letter as Appendix B.

³ See <http://samuelsonclinic.org> for more information.

be identified and remedied by a community of experts.⁴ The idea of “security through obscurity” – keeping system designs secret to aid security – was criticized as long ago as 1883⁵ because it has the “twin faults of not providing serious security from real attackers, who can easily overcome minimal security measures, and of limiting public and general security oversight of the system, which has proven to be the best method for creating and maintaining a truly secure system.”⁶

When the security of a system relies on obscurity, or secrecy, for its effectiveness, the system designer assumes that potential vulnerabilities are unknown, and such vulnerabilities are unlikely to be discovered by hackers. Today, however, the most secure systems are developed with the assumption that attackers already know—or can easily learn—the entire design of the system, except the cryptographic key that “unlocks” the system (such as a password). In a world where hackers will inevitably identify a system’s vulnerabilities, disclosing the system design to the public allows computer scientists and others with technical skill to pinpoint the system’s flaws and fix them before hackers have a chance to attack. If the system is not subject to public oversight, malicious attacks are less likely to be forecasted and thus less likely to be fixed, and more likely to cause significant harm.

Moreover, the potential vulnerabilities of a system are never truly “secret.” Security vulnerabilities are “disclosed” to insiders through their involvement with developing and implementing the system. Failure to disclose system information, coupled with the resulting lack of public oversight, provides insiders with the opportunity to exploit a system. Those charged with overseeing the effectiveness of the system’s security may not be aware of the vulnerabilities, and thus, may not be able to detect or prevent such attacks. Information possessed by a small number of insiders may pose a greater security risk than disclosure of the system’s design to the general public.

A system is less secure when secrets other than specific operational knowledge, such as cryptographic keys, must be maintained, because the undisclosed information constitutes another point of potential compromise. There is a general consensus in the computer science community that “the fewer secrets a system has, the more secure it is. If the loss of any one secret causes the

⁴ See, e.g., Bruce Schneier, “Internet Shield: Secrecy and Security,” S.F. Chronicle, Mar. 2, 2003 (“Public scrutiny is the only reliable way to improve security – be it the nation’s roads, bridges and ports or of our critically important computer networks.”); Peter Swire, “A Model for When Disclosure Helps Security: What is Different About Computer and Network Security?”, 2 Journal on Telecommunications and High Technology Law 3 (2004) (“[R]evealing the details of the system will actually tend to improve security, notably due to peer review....[T]rying to hide the details of the system will tend to harm security because attackers will learn about vulnerabilities but defenders will not know where to patch the vulnerabilities.”); Joseph Lorenzo Hall, “Transparency and Access to Source Code in Electronic Voting,” School of Information, University of California at Berkeley, at 1 (2006), available at http://josephhall.org/papers/jhall_evt06.pdf (last visited November 12, 2006) (“Unsurprisingly, academics, activists, election officials and commentators have called for increased access to, and heightened examination of the source code that powers election systems.”).

⁵ See Auguste Kerckhoffs (January 9, 1883), “La Cryptographie Militaire”, Journal des Sciences Militaires: 5-38.

⁶ Tadayoshi Kohno, et al., “Analysis of an Electronic Voting System,” (July 23, 2003), available at <http://www.mindfully.org/Reform/2003/Electronic-Voting-System-Analysis23jul03.htm> (last visited November 14, 2006).

system to break, then the system with fewer secrets is necessarily more secure. The more secrets a system has, the more fragile it is. The fewer secrets, the more robust.”⁷

Consider the practice of hiding a spare key under the doormat in case one is locked out of one’s house. In that situation, the homeowner relies on the belief that burglars will not discover the hidden key. However, since burglars often know likely hiding places, the homeowner actually experiences greater risk of a burglary by hiding the key under the doormat, and he may think that his house is more secure than it actually is. The owner has in effect added another key to the system – the knowledge that the entry key is stored under the doormat. The key under the doormat is like a vulnerability in the voting system software, which can easily be discovered and exploited by attackers but that gives a false sense of security.⁸

The primary assumption underlying the argument against disclosing system information is that by withholding the information, the potential attackers—or even well-meaning computer enthusiasts—will not be able to figure out the system on their own. Many examples prove otherwise. In October 2006, for example, researchers at Princeton University reverse-engineered the software and hardware of Diebold AccuVote machines and discovered that malicious software could be inserted into the machine in as little as one minute.⁹ In August 2006, a nonprofit voting rights organization discovered how to tamper with a Diebold machine using only a screwdriver.¹⁰ And a recently published article entitled “How to Steal an Election by Hacking the Vote” describes the points at which specific types of voting machines are most vulnerable.¹¹ These examples show that once outsiders discover vulnerabilities, information about them spreads quickly.

Moreover, as discussed above, there will always be “insiders” who know the program’s potential flaws. As a result, these flaws cannot be considered fully secret to begin with, and there is always the risk that this information will be leaked. In 2003, source code from a Diebold voting machine found its way onto the Internet. Using that code, computer scientists at Johns Hopkins University and Rice University found numerous ways to exploit the system.¹² Because information gleaned by one person can be spread so quickly through the Internet, it is likely that releasing information about the design of the system will not provide attackers with any information they do not already know or

⁷ Bruce Schneier, “Secrecy, Security, Obscurity,” *Crypto-Gram Newsletter*, May 15, 2002, available at <http://www.schneier.com/crypto-gram-0205.html#1>.

⁸ See Security through obscurity, http://en.wikipedia.org/w/index.php?title=Security_through_obscurity&oldid=92888789 (last visited December 16, 2006). This is a common comparison made in security literature.

⁹ See Ariel J. Feldman, J. Alex Halderman, and Edward W. Felten, “Security Analysis of the Diebold AccuVote-TS Voting Machine,” available at <http://itpolicy.princeton.edu/voting> (last visited November 14, 2006).

¹⁰ Alan Dechert, “Worst Ever Security Flaw Found in Diebold TS Voting Machine,” Open Voting Foundation (August 6, 2006), available at http://openvotingfoundation.org/tiki-read_article.php?articleId=1 (last visited November 14, 2006).

¹¹ John Stokes, “How to Steal an Election by Hacking the Vote,” *arstechnica* (October 25, 2006), available at <http://arstechnica.com/articles/culture/evoting.ars> (last visited November 14, 2006).

¹² See Khono *et al.*, note 6.

cannot already discover: one must assume that, for all intents and purposes, disclosure of a security flaw will add little or nothing to the attacker's knowledge.¹³

Hiding the details of the system, on the other hand, will ultimately tend to harm security because attackers will learn about vulnerabilities but defenders will not know where to patch the vulnerabilities.¹⁴ As the examples above illustrate, there is a technical community that will monitor and help to fix problems in electronic voting machines. "Many eyes make bugs shallow,"¹⁵ argue many computer scientists, because they believe that disclosure increases the number of people who can review the details of such programs, identify flaws, and fix the flaws sooner. When the potential benefits of this public oversight are weighed against the potential costs of an attack, it becomes clear that making as much information available to the technical community as possible is the best long-term strategy.

In addition, it is important to note that attackers very often tend to be insiders. As a result, hiding the information from the outside world often hurts the security of the system: an insider attacker knows about certain weaknesses, while public overseers are not aware and thus cannot act to help fix the problem.

When "Security Through Obscurity" May Be Beneficial

There are, of course, some circumstances in which security through obscurity is reasonable as part of a "defense in depth" strategy. For example, security through obscurity may (but cannot be guaranteed to) act as a temporary "speed bump" for attackers while a resolution to a known security issue is implemented. Here, the goal is simply to reduce the short-run risk of a vulnerability exploit while a permanent solution is developed.¹⁶

In certain unique situations, "security through obscurity" can enhance the security rather than create additional vulnerabilities. One such situation is the "military paradigm," where military bases and weapon systems are cloaked in secrecy, and intelligence agencies reveal little about their capabilities, sources, and methods. In that context, secrecy may be acceptable because: (1) attackers would learn a lot from disclosure of a vulnerability; (2) disclosure will teach the designers little or nothing about how to improve their defenses; and most importantly (3) disclosure will prompt little or no improvement in the defense system by other scientists, designers of defense systems, or the general public. As we discuss below, these assumptions do not apply in the case of electronic voting machines.

¹³ Peter Swire, "A Model for When Disclosure Helps Security: What is Different About Computer and Network Security?," 2 *Journal on Telecommunications and High Technology Law* 21 (2004).

¹⁴ Swire, *supra* note 13 at 3.

¹⁵ Eric Steven Raymond, "The Cathedral and the Bazaar," (August 2, 2002), available at <http://catb.org/~esr/writings/cathedral-bazaar/> (last visited December 15, 2006).

¹⁶ Security through obscurity, http://en.wikipedia.org/w/index.php?title=Security_through_obscurity&oldid=92888789 (last visited December 16, 2006).

Why “Security Through Obscurity” Is Not Appropriate for Voting Systems

In the context of electronic voting, security through obscurity is not acceptable (except for cryptographic keys, as discussed below) because: (1) attackers will likely learn little, if any, new information about the system due to disclosure because tampering information is widely disseminated, attackers can be assumed already to have the requisite knowledge to break the software, and attackers may be insiders; (2) disclosure will allow the technical community to educate voting systems designers about how to build more secure systems; and (3) disclosure will prompt academics and other concerned citizens with technical knowledge to devise solutions to the vulnerabilities they identify. These three assumptions are critically different from those made in the military paradigm.

Furthermore, public availability of voting system design information is an important protection against vendors who deliberately try to hide vulnerabilities of weak systems from public scrutiny. “Security through obscurity” allows a vendor to guard such vulnerabilities in order to avoid adversely affecting confidence in its product and thus its marketability. When system design information is disclosed to the public, and thereby subject to scrutiny, vendors and operators have an incentive to improve the effectiveness of their security measures and avoid attempts to hide vulnerabilities that could adversely impact the consumer.

Finally, transparency (i.e., disclosure) is especially important when dealing with electronic voting systems. Voting machines are the fundamental engines that ensure the integrity of the democratic process, and citizens should be confident that they are functioning properly. Allowing the public to examine records related to electronic voting machines helps encourage trust in the system and ensures accountability on the part of officials, machine vendors, and those who designed the system. Moreover, disclosure may prompt “reforms that instill confidence in the voting public by facilitating public oversight, comprehension, access and accountability.”¹⁷

Mr. Cooper’s Specific Requests for Information

The general security principles described above may be difficult to understand in the abstract, but applying them to the specific requests made by Mr. Cooper demonstrates that these types of information pose little security risk and should therefore be disclosed. In this section, we use his requests as examples of how these general principles apply to real world requests that your office may receive in the future. Below, we analyze each of the seven requests that were denied and explain why information of this nature should be released to the public in the future.

Note that the timing of disclosure can impact what kinds of information are deemed “security information” and therefore not subject to disclosure. On the one hand, it is beneficial to disclose vulnerabilities if there is sufficient time to address and either repair or provide a work-around to them before a potential attack. On the other hand, disclosure shortly before an election can pose a serious security risk because officials may not be able to address the problem prior to the election.

¹⁷ Hall, *supra* note 4 at 1-2.

In a scenario with time constraints such as these, “security through obscurity” may be appropriate as a temporary, stop-gap measure.

After an election has taken place, however, system vulnerabilities cannot be exploited to affect election results, and the time between elections can be used for repairing or providing a work-around for known vulnerabilities. Disclosure of these vulnerabilities brings them to the attention of system designers and the computer security community, and encourages finding a solution or work-around to prevent future exploits. Therefore, now is the optimal time to disclose the information requested by Mr. Cooper, and thereby enable any flaws or vulnerabilities to be remedied or worked-around before the next election.

In assessing whether requested information should be disclosed, it is helpful to classify information about voting systems into one of three categories, each with its own associated disclosure principles. Of course, as we noted earlier, Mr. Cooper’s requests do not primarily concern security and fall into the category of system design. Furthermore, even if the requests do yield any information pertinent to the security of the system, the records produced could easily be redacted before disclosure.

- **System design** includes that information which explains how the system works and how it is used. This information should always be disclosed as it poses no security risk for a properly designed system. This includes information about the reliability, availability, and usability of the system, and information about how well it works operationally. Most of Mr. Cooper’s requests can be classified as systems design that does not directly relate to security issues.
- **Security vulnerabilities** refer to information about potential flaws in the system. Security vulnerabilities generally should be disclosed unless a fix is impossible before deployment or use of the system. For example, while disclosure of security vulnerabilities in an electronic voting system a few days before an election may be unwarranted, disclosure of these vulnerabilities may no longer pose a threat after an election.
- **Cryptographic keys and passwords**, which can be, for example, sequences of characters that enable access to a system or provide other access-like functionality, should never be disclosed when in active use because the security of the system is most effective when the keys are the only items that need to be kept secret. It is good security practice to change these access keys every election cycle; documents or records containing this type of information can be released in redacted form.

In the following pages we analyze each of the seven requests denied by the former Secretary of State and suggest why requests like these should be granted in the future.

1. *Documents reflecting detailed information about testing that has been done of Stateline Voter Registration System’s (SVRS) functionality – documents that reflect the dates that tests were performed, any problems that were discovered, when and how they were remedied. I would like*

documents related to testing of the modules of the system, as well as each of the re-builds that have been done.

This request seeks four separate types of items:

Documents that reflect the dates that tests were performed. Disclosure of the mere evidence that tests took place and the time they took place does not pose a security risk since dates and times will not provide would-be attackers with methods for exploiting the election process. Therefore, this information should be disclosed.

Any problems that were discovered. This information can be categorized as “system design” or as “security vulnerabilities,” and its release is unlikely to pose a security risk. If the software malfunctioned or exhibited errors during testing, that information should be disclosed so that the known vulnerability can be addressed before the SVRS is used again in the voting process. In any event, even if the testing revealed specific steps that could be taken to induce an error, given that the November 2006 elections have just concluded, there should be more than enough time to either provide a solution or work-around for such errors before the next election. Disclosure of these problems will assist in the development of fixes, and increase voter confidence in the security of the system as a whole by ensuring that the same problems do not occur in the next election.

When and how they were remedied. Disclosure of the fact that remedial action was taken, and the time it was taken, should not be categorized as “security information” because it merely establishes that there was a problem in the system, and that it was remedied. The information sought would not reveal methods for exploiting the solution applied, and if a specific decision has been taken to leave errors unfixed, that decision can be revealed post-election in order to prevent the vulnerability from remaining a potential risk in future elections.

Documents relating to testing of the modules and re-builds. Information about the testing of modules and re-builds, as with the other information identified in this request, most likely relates to system design and would likely not have security implications if disclosed. To the extent this information reveals flaws or vulnerabilities, it should be disclosed for the reasons discussed above. In any event, because there is presumably more than enough time to fix any potential flaws before the next election, disclosure of this information is warranted.

2. *Any “tickets” opened by the OSS helpdesk regarding problems related to the Driver License format between July 2005 and the present.*

It is likely that most of these “tickets” would not reveal any security weaknesses that would assist an attack on the system, as these are typically operational complications related to checking driver’s license formats. These tickets can very likely be revealed without fear of compromised security. If there is personal information contained in these tickets, that information should be redacted prior to disclosure so as to ensure personal privacy.

Information that reveals a weakness in the verification software or a method for compromising the registration system could be categorized as a “security vulnerability.” It therefore may be appropriate to withhold it prior to an election. However, disclosure soon after an election allows sufficient time for the development of a remedy, and assurance that it will not have any affect on future elections. If the vulnerability cannot be remedied, it may be an indication that the software or system design is not suited for use in the voting process.

Please note that the former Secretary of State had previously granted an identical request for problems regarding the driver license format that occurred between January and July 2005. It poses no more of a security risk to release the same information for the period between July 2005 and the present than to release of the information from an earlier time period.

3. *Any “tickets” opened by the OSS helpdesk regarding problems with verifying records between January 2005 and the present.*

Tickets opened by the OSS helpdesk do not necessarily relate to the security of the voting system and thus should not have been withheld. If there were information contained in a ticket that could be considered security related, it should be classified as a potential “security vulnerability,” and, as such, should still be disclosed. As we explain above, while it could potentially have been inappropriate to release some of these “tickets” right before the election, their disclosure at this juncture, immediately after an election, will likely result in the development of a solution or work-around to these problems. The security of the voting process can be increased through such disclosure by preventing the future exploitation of known vulnerabilities.

4. *From some of the other reports I have seen, it appears that the calls to the OSS helpdesk relating to problems with SVRS were transferred to the Elections Department. I would like any documents from the Elections Department about problems with verifying records in SVRS between January 2005 and the present.*

This request is similar to the previous two, except that it seeks those “tickets” that were referred to the Elections Department. As we explain above, disclosure of information related to problems with the SVRS or verifying records should no longer be withheld since there should be adequate time before the next election to fix any security vulnerabilities. Proper oversight of these problems—which is closely related to public disclosure—should lead to a quicker resolution of vulnerabilities or inconsistencies, ultimately creating a more secure system by minimizing the number of points attackers can exploit to gain access to the voting process. Finally, it is not clear that information about problems with voter registration directly pertains to security risks or vulnerabilities.

5. *Copies of all communications relating to any problems that the Automark machines have in relation to marking or reading ballots, or ejecting unmarked ballots.*

The information requested could fall under any of the three categories of voting system information, but most of it will likely fall into the “system design” category and as such, should be

disclosed. If any of the communications refer to cryptographic keys, such as passwords, such communications should not be disclosed without redaction of sensitive information since the security of the system relies on keeping that type of information secret. In addition, one would expect that, given proper security policies, these keys and passwords would change from one election to the next. Reports that the AutoMark machines malfunction (e.g. fail to mark or read the ballot, fail to display the correct ballot, etc.), might potentially be classified as “security vulnerabilities” and should thus be disclosed so that a solution or work-around to the problem can be implemented prior to the next election. Moreover, disclosure should occur if the communications describe methods by which to induce problems with the AutoMark machines because these need to be addressed prior to the next election. Again, it is best to operate under the assumption that attackers will know about these vulnerabilities whether or not they are disclosed to the public, especially given the possibility of an attack by an insider. Vulnerabilities in the voting process should not remain potential targets for attack by remaining secret if they can be feasibly repaired or worked around.

Assuming the problems do not lead to a wholesale compromise of the machine, the Secretary of State should also consider releasing this type of information well before an election. This type of disclosure informs voters about how to ensure that weaknesses in the ballot scanning system do not mistakenly discard their votes. The integrity of the voting process and the voting machines is essential to maintaining public trust in our democratic system.

6. *I expect that your office has received questions or concerns about the use and programming of the new machines from local elections officials. Please double-check to see if you have any communications or documents related to these or other issues with the Automark machines.*

It is appropriate to disclose the way the machines are deployed and used because such information is already available to poll workers and local elections officials. Thus, unless these communications between elections officials and the Secretary’s office contain information related to passwords or other cryptographic keys, they should be released in complete form.

7. *I am requesting copies of all communications related to any planned changes (or re-builds) yet to be made to SVRS or other election procedures during the balance of 2006.*

Insofar as the request relates to how the machine is programmed, this is a system design issue and the information should be disclosed immediately so that public participation and robust oversight can occur. Furthermore, changes to the software’s feature list or the way it is used are not security risks. Changes to security mechanisms in SVRS should be public information since the system will become more secure if the technical community and the public are allowed to examine it. If the design of the SVRS relies solely on secrecy to prevent exploitation, the system is more susceptible to attack than it if it were revealed to the public and allowed to be tested by outside experts. As we note above, however, any cryptographic keys, passwords, and the like should be changed and/or redacted from documents and records before they are disclosed.

Conclusion

Thank you for taking the time to review our letter and consider the issues we raise. We hope that the information we have provided assists you in designing the optimal voting system and in developing policies for handling election information requests received by the Secretary of State's office. If you should be in need of further assistance either with this matter or with future electronic voting issues, please do not hesitate to contact Samuelson Clinic Fellow Jack Lerner at (510) 642-7515 or at jlerner@law.berkeley.edu. We would welcome the opportunity to work with you.

Sincerely,

Jason H. Tokoro
Student Intern
Samuelson Law, Technology
& Public Policy Clinic

Sarala V. Nagala
Student Intern
Samuelson Law, Technology
& Public Policy Clinic

Jack I. Lerner
Samuelson Clinic Fellow
Samuelson Law, Technology
& Public Policy Clinic

cc: C. Scott Cooper

APPENDIX A

Ben Adida

ben@mit.edu - ben@eecs.harvard.edu - <http://ben.adida.net>

PROFILE

Specialized in cryptography applied to public policy, including voting systems, authentication infrastructures, and secure health records. Extensive experience in all aspects of web software, including database design and semantic web. Extensive industry experience in software development and the management of software development teams.

EDUCATION

Massachusetts Institute of Technology 2003-2006

PhD in Computer Science, *August 2006*.

Thesis: Advances in Cryptographic Voting Systems

Advisor: Ronald L. Rivest

Massachusetts Institute of Technology 1998-1999

MEng in Computer Science, June 1999.

Thesis: Self-Describing Cryptography Through Certified Universal Code

Advisor: Ronald L. Rivest

Massachusetts Institute of Technology 1994-1998

SB in Computer Science, June 1998.

RESEARCH EXPERIENCE

Harvard Center for Research on Computation and Society - Postdoctoral Fellow 2006-present

Research on security and privacy in voting, health records, and web applications.

Cambridge-MIT Institute - Research Assistant 2004-2006

Research on the security of cryptographic APIs

Caltech/MIT Voting Technology Project - Research Assistant 2004-2006

Research on universally-verifiable voting.

MIT Lab for Computer Science - Research Assistant 1998-1999

Research on voting systems and self-describing cryptography.

INDUSTRY EXPERIENCE

Children's Hospital Informatics Program (Boston) - Consultant *May - Aug 2005*

Performed security review of source code for Personal Health Record management system, designed and implemented fixes at cryptographic and application security levels. Designed and developed a new secure and efficient storage mechanism for genomic data.

OpenForce - co-founder, CEO & CTO 2000-2003

Defined and implemented the company business plan: providing enterprise internet software services. Often hired as acting Chief Technology Officer by customers (GreenOrder, Creative Commons). Responsible for signing on, architecting, and leading the software implementation of client projects, including the MIT Sloan School of Management, the LA Unified School District, GreenPeace, Creative Commons, GreenOrder, the Berklee School of Music.

OpenACS & dotLRN open-source projects - co-founder and director 1999-2002

Led design and implementation of major open-source enterprise web software endeavors: the OpenACS web application toolkit and dotLRN course management system. More than 15 companies now service this software and hundreds of web sites run it, including more than 100 universities.

ArsDigita - founding member

1998-1999

Led design, implementation, and deployment of major software projects with Levi Strauss and GreenTravel.com (now Away.com). Helped launch the first open-source web application toolkit in 1998: the ArsDigita Community System.

RSA Data Security - developer

summer 1997

Co-developed the first Java cryptography toolkit (JSAFE 1.0).

Sun Labs - developer

summer 1996

Implemented `java.lang.Math` in pure Java for Java-only platforms.

Instrumented the Java Virtual Machine for collecting statistics about garbage collection.

REFEREED PUBLICATIONS

“SCRATCH & VOTE: SELF-CONTAINED PAPER-BASED CRYPTOGRAPHIC VOTING,” with Ronald L. Rivest in *Proceedings of the Workshop on Privacy in the Electronic Society (WPES) 2006*, October 2006.

“LIGHTWEIGHT EMAIL SIGNATURES,” with David Chau, Susan Hohenberger, and Ronald L. Rivest in *Proceedings of the Fifth Conference on Security and Cryptography in Networks (SCN) 2006*, September 2006.

“BALLOT CASTING ASSURANCE,” with C. Andrew Neff in *Proceedings of the First USENIX/ACCURATE Electronic Voting Technology Workshop (EVT) 2006*, August 2006.

“GENEPING: SECURE, SCALABLE MANAGEMENT OF PERSONAL GENOMIC DATA,” with Isaac S. Kohane, in *BioMed Central Genomics 2006 7:93*, April 2006.

“LIGHTWEIGHT ENCRYPTION FOR EMAIL,” with Susan Hohenberger and Ronald L. Rivest in *Proceedings of USENIX's First Steps to Reducing Unwanted Traffic on the Internet (SRUTI) 2005*, pages 93-99, July 2005.

“EVALUATION OF VOTING SYSTEMS,” with P. Vora, R. Bucholz, D. Chaum, D.L. Dill, D. Jefferson, D.W. Jones, W. Lattin, A.D. Rubin, M. I. Shamos, M. Yung in *Communications of the ACM*, page 144, November 2004.

FORTHCOMING PUBLICATIONS

“AD-HOC GROUP SIGNATURES FROM HIJACKED KEYPAIRS,” with Susan Hohenberger and Ronald L. Rivest, preliminary version in *Proceedings of DIMACS Workshop on Theft in Electronic Commerce*, April 2005.

“HOW TO SHUFFLE IN PUBLIC,” with Douglas Wikström, *in submission*.

Preliminary version on eprint: <http://eprint.iacr.org/2005/394>

“A FAST APPROXIMATION OF REENCRYPTION MIXNET PROOFS,” with Ronald L. Rivest, *in preparation*.

“ON THE SECURITY OF THE EMV SECURITY API,” with Mike Bond, Jolyon Clulow, Amerson Lin, Ross Anderson and Ronald L. Rivest, *in submission*.

“(ALMOST) ROBBING THE BANK WITH A THEOREM PROVER,” with Paul Youn, Mike Bond, Jolyon Clulow, Jonathan Herzog, Amerson Lin, Ross Anderson and Ronald L. Rivest, *in submission*.

“BUILDING INTEROPERABLE METADATA,” with Hal Abelson, *in preparation*.

INVITED PRESENTATIONS

PRIVACY IN AN ALWAYS-ONLINE WORLD

Simplicity 2006, MIT Media Laboratory

July 2006

INTRODUCTION TO CRYPTOGRAPHY

Lecture in MIT's 6.976 - Quantitative Foundations of Engineering Systems

May 2006

WEB SECURITY

Lecture in MIT's 6.171 - Software Engineering for Internet Applications

May 2006

DIRECT VERIFICATION OF ELECTIONS WITH CRYPTOGRAPHY

Lecture at ARIA, University of Massachusetts at Amherst

April 2006

LIGHTWEIGHT SIGNATURES FOR EMAIL

Lecture in MIT's Network and Computer Security Class (6.857)

December 2005

MIT/Cisco Security Summit

December 2005

Harvard's Center for Research in Computation and Society

November 2005

Google, Palo Alto

August 2005

MIT's Decentralized Information Group

May 2005

MIT's Cryptography and Information Security Seminar

May 2005

LIGHTWEIGHT ENCRYPTION FOR EMAIL

Steps to Reducing Unwanted Traffic on the Internet, Cambridge, MA.

July 2005

CRYPTOGRAPHIC VOTING TUTORIAL

Radcliffe Institute for Advanced Study, Harvard University, Cambridge, MA.

February 2005

ROBUST MIXNETS IN ELECTRONIC VOTING

Cambridge University, Cambridge, England.

January 2005

TRUSTING THE VOTE

Internet & Society Conference, Harvard Law School

December 2004

SECURE AND FAIR ELECTIONS

Digital Democracy, joint class of the Harvard Law School and MIT

November 2004

WEB SECURITY

Lecture in MIT's Software Engineering for Internet Applications (6.171)

November 2003

TEACHING AND ADVISING EXPERIENCE

Advisor to Undergraduates and Master's Students

2004-present

Provided guidance to Undergraduate and Master's students in security and cryptography:

David Chau, Amerson Lin, Joy Forsythe.

Software Engineering for Internet Applications (6.171) - Teaching Assistant

Fall 2003

Helped design the course and problem sets, coordinated 10 student software development teams, graded all problem sets and midterm. Rated *6.5/7.0* by student-led course guide.

Structure and Interpretation of Computer Programs (6.001) - Teaching Assistant *Spring 1998*
Taught 8 weekly hours of tutorials. Graded problem sets, quizzes, and exams.

Software Engineering for Web Applications (6.916) - Teaching Assistant *Fall 1999*
Helped design the first version of this course. Developed and maintained the software platform for students to use.

Introduction to Interactive Programming (6.096) - Lab/Teaching Assistant *Fall 1996*
Helped design the first version of this course. Developed software for problem sets.

PROFESSIONAL ACTIVITIES

Workshop On Trustworthy Elections (WOTE) *July 2006*
Member of Program Committee.

External Conference and Journal Reviewer *2004-present*
PKC 2005, ACM CCR Journal, PKC 2006, IEEE Security & Privacy 2006.

Harvard Law School, Berkman Center StopBadware - Working Group Member *2006-present*
Advisor on spyware research working group in association with Google, Sun, and Lenovo.

Harvard Medical School, Countway Library - Member, Technology Advisory Board *2005-present*
Advisor on web development and semantic web issues.

W3C - Chair, RDF-in-XHTML Task Force, Semantic Web Best Practices Working Group *2004-present*
Leading a team of industry experts drawn from the W3C's Semantic Web and HTML working groups to specify mechanisms for including semantic web statements (RDF) in HTML. Developing the RDFa standard. Primary author of the RDFa syntax and primer documents.

Harvard Law School Berkman Center - Associate *2003-present*
Advisor to the Berkman Center on technology issues.

Creative Commons - Member, Technology Advisory Board *2003-present*
Advisor on web development issues, particular focus on semantic web. Representative to the W3C.

Center for Strategic & International Studies - Member, Authentication Working Group *2002-2003*
Developed recommendations on Federated Authentication systems.

APPENDIX B

Ka-Ping Yee

ping@zesty.ca
Berkeley, California, 94709

Education

- present [UC Berkeley](#): pursuing a Ph. D. in Computer Science (human-computer interaction)
- 1998 [U of Waterloo](#): B. A. Sc. (Hons.) in Computer Engineering (top student in graduating class)

Employment

- present [UC Berkeley](#): graduate research in security and usability with [Marti Hearst](#) and [David Wagner](#)
- 2004 [HP Labs](#): design and development of a virus-safe environment based on the Principle of Least Authority
- 2003 [IBM Research](#): design and development of improvements to a next-generation e-mail client
- 2001 [Opera Software](#): design and implementation of user interface enhancements to the Opera Web browser
- 2000 [UC Berkeley](#): teaching assistant for [CS 61A](#)
- 1998-2000 [Industrial Light and Magic](#): development and support of interactive graphics software used daily in feature film production by a community of expert artists
- 1997 [Xerox PARC](#): development of a Web-based document services platform
- 1996 [Alias|Wavefront](#) Tokyo: development of 3-D design, animation, and rendering tools (shipped in Alias 8)
- 1995 [U of Waterloo Math Faculty Computing Facility](#): network and robotics lab hardware maintenance
- 1995 [Canadian Space Agency](#): development of the Agency's first Web services and design of a haptic device control protocol
- 1994 [Alias Research](#): development of new polygonal modelling and 2-D animation tools (shipped in Alias 6)

Achievements

- 2002-2004 Recipient, [IBM Ph. D. Fellowship](#)
- 2000 Category Winner, [Software Carpentry Open-Source Design Competition](#)
- 1998 Gold Medal for Academic Achievement, [Professional Engineers Ontario Governor General's Silver Medal](#)
- 1998 [Governor General's Silver Medal](#)
- 1996 Runner-up, [Outstanding Undergraduate Award](#), [Computing Research Association](#)
- 1994 Member of World Champion Team, [ACM International Programming Contest](#)
- 1993 [Gold Medal](#), [International Mathematical Olympiad](#)

Publications

In Print

- 2006 Marc Stiegler, Alan H. Karp, Ka-Ping Yee, Tyler Close, Mark S. Miller. [Polaris: Virus-Safe Computing for Windows XP](#). In *Communications of the ACM*, September 2006.
- 2006 Ka-Ping Yee, David Wagner, Marti Hearst, Steven Bellovin. [Prerendered User Interfaces for High-Assurance Electronic Voting](#). In Proceedings of the USENIX/ACCURATE Electronic Voting Technology Workshop, 2006.
- 2006 Ka-Ping Yee, Kragen Sitaker. [Passpet: Convenient Password Management and Phishing Protection](#). In Proceedings of the Symposium on Usable Privacy and Security, 2006.
- 2006 Ka-Ping Yee. [Firefighters and Engineers](#). In *ACM Interactions*, May–June 2006.
- 2005 Ka-Ping Yee. Guidelines and Strategies for Secure Interaction Design (Chapter 13). In [Security and Usability: Designing Secure Systems that People Can Use](#), edited by Lorrie Faith Cranor, Simson Garfinkel. O'Reilly, 2005.
- 2004 Marc Stiegler, Alan H. Karp, Ka-Ping Yee, Mark S. Miller. [Polaris: Virus Safe Computing for Windows XP](#). HP Labs Technical Report HPL-2004-221.
- 2004 Ka-Ping Yee. [Aligning Usability and Security](#). In *IEEE Security & Privacy*, September 2004.

- 2004 M. Markstein, R. Zinzen, P. Markstein, Ka-Ping Yee, A. Erives, A. Stathopoulos, M. Levine. [A regulatory code for neurogenic gene expression in the *Drosophila* embryo](#). In *Development* 131, 2387–2394, 2004.
- 2004 Ka-Ping Yee. [Two-Handed Interaction on a Tablet Display](#). In *Extended Abstracts of the ACM Conference on Computer-Human Interaction*, 2004.
- 2003 L. Jean Camp and Ka-Ping Yee. Human implications of technology. In [Practical Handbook of Internet Computing](#), edited by M. P. Singh, CRC Press, 2003.
- 2003 Ka-Ping Yee, Kirsten Swearingen, Kevin Li, Marti Hearst. [Faceted Metadata for Image Search and Browsing](#). In *Proceedings of the ACM Conference on Computer-Human Interaction*, 2003.
- 2003 Ka-Ping Yee. [Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers](#). In *Proceedings of the ACM Conference on Computer-Human Interaction*, 2003.
- 2003 Mark McKelvin, Ragnhild Nestande, Leticia Valdez, Ka-Ping Yee, Maribeth Back, Steve Harrison. [SeismoSpin: a Physical Instrument for Digital Data](#). In *Extended Abstracts of the ACM Conference on Computer-Human Interaction*, 2003.
- 2003 Mark Miller, Ka-Ping Yee, Jonathan Shapiro. [Capability Myths Demolished](#). Technical Report SRL2003-02, Systems Research Laboratory, Johns Hopkins University.
- 2002 Ka-Ping Yee. [CritLink: Advanced Hyperlinks Enable Public Annotation on the Web](#). Demonstration abstract. ACM Conference on Computer-Supported Co-operative Work, 2002.
- 2002 Ka-Ping Yee. [Zest: Discussion Mapping for Mailing Lists](#). Demonstration abstract. ACM Conference on Computer-Supported Co-operative Work, 2002.
- 2002 Ka-Ping Yee. [User Interaction Design for Secure Systems \(ACM\)](#). In *Proceedings of the 4th International Conference on Information and Communications Security* (Lecture Notes in Computer Science 2513), 278–290, Springer-Verlag, 2002. An extended version of this paper is also available as [UC Berkeley CS Technical Report CSD-02-1184](#).
- 2002 J. English, M. Hearst, R. Sinha, K. Swearingen, and Ka-Ping Yee. [Finding the Flow in Web Site Search](#). In *Communications of the ACM*, September 2002.

- 2002 J. English, M. Hearst, R. Sinha, K. Swearingen, and Ka-Ping Yee. [Hierarchical Faceted Metadata in Site Search Interfaces](#). In *Proceedings of the ACM Conference on Computer-Human Interaction*, 2002.
- 2001 Ka-Ping Yee. [Operating an Emergency Information Service](#). In *Communications of the ACM*, Dec 2001.
- 2001 Ka-Ping Yee, D. Fisher, R. Dhamija, and M. Hearst. [Animated Exploration of Dynamic Graphs with Radial Layout](#). In *Proceedings of the IEEE Symposium on Information Visualization*, 2001.

Online

- 2005 Ka-Ping Yee, Marti Hearst. [A Visualization to Facilitate Productive Discussions](#). Position paper accepted to Beyond Threaded Conversation Workshop at the ACM Conference on Computer-Human Interaction, 2005.
- 2003 Ka-Ping Yee. [Secure Interaction Design and the Principle of Least Authority](#). Position paper accepted to HCI and Security Workshop at the ACM Conference on Computer-Human Interaction, 2003.
- 2002 Jennifer English, Marti Hearst, Rashmi Sinha, Kirsten Swearingen, Ka-Ping Yee. [Flexible Search and Navigation Using Faceted Metadata](#).
- 2000 Ka-Ping Yee. [Roundup: An Issue-Tracking System for Knowledge Workers](#). Category Winner, Software Carpentry Open Source Design Competition.

Projects

I was the primary designer or implementor of the following:

- 2005 [PFIF](#), a data model and format for locating people displaced by natural disasters (the prevailing interchange format for Hurricane Katrina survivor data)
- 2003 the [Peephole Display](#), an interactive, spatially aware handheld display
- 2003 [SeismoSpin](#), an interactive visualization of earthquake data in time and space
- 2001 [Fly Enhancer](#), a public search engine for [clusters of binding sites in cis-regulatory DNA in the fly genome](#)
- 2001 [national survivor registry](#) for victims of September 11 terrorist attacks

- 2001 [Flamenco](#), a series of user interfaces for browsing large collections using faceted metadata
- 2000 [pydoc](#), the documentation generator and interactive help system for [Python](#)
- 1999 [pyxi](#), the graphical browser released with the [Xanadu hypertext system](#)
- 1998 [Roundup](#), the issue-tracking system now in daily production use at ILM and other sites
- 1997 [crit.org](#), a zero-install, browser-independent, public, fine-grained annotation system for the Web (won [Engelbart Hypertext Achievement Scholarship](#))
- 1996 [MINSE](#), a simple, extensible notation and zero-install, browser-independent display system for mathematics in Web documents
- 1995 [Shodouka](#), a transformation engine that displays Japanese pages on browsers and systems without font support (won [ACM Webbie Prize](#))

Activities

- 2007 [FC Usable Security Workshop](#): Program Committee Member
- 2007 [Symposium on Usable Privacy and Security](#): Program Committee Member
- 2002 [10th Python Conference](#): Program Committee Member
- 2001 [9th Python Conference](#): Program Committee Member
- 2000 [Alternative Computer Expo](#): Special Guest Speaker
- 2000 [Shad Valley Summit](#): Keynote Speaker
- 1999 [Shad Valley](#): Program Assistant (Acadia campus)
- 1992 [Shad Valley](#): Participant (Waterloo campus)
- Senior Associate, [Foresight Institute](#)
- Member, [Electronic Frontier Foundation](#)
- Member, [Free Software Foundation](#)
- Member, [Association for Computing Machinery](#)
- Member, [Python Software Foundation](#)
- Member, [Alcor Life Extension Foundation](#)

This document lives at <<http://zesty.ca/cv.html>>.