University of Iowa Student Elections: An After–Action Report

Robert J. Hansen, Tristan D. Thiede, Douglas W. Jones
{rjhansen, tthiede, jones}@cs.uiowa.edu*

March 31, 2006

Contents

*NSF grant CNS–052431 provided partial support for this work.

Abstract

Instant runoff voting was introduced to the University of Iowa's student government presidential elections in 2006. The ACCURATE group at the University of Iowa was asked to write software for this endeavor, as well as serving as election consultants. The election had critical troubles which cast the results into serious jeopardy. In this paper, we detail the failures in the hopes that a careful analysis of them will be useful to future elections.

1 History

The University of Iowa Student Government (UISG) decided to bring instant runoff voting (IRV) to certain student elections in 2006. The political considerations motivating this decision will not be discussed here, since the purpose of this document is technical and not political.

Several years previously, UISG determined their elections would be conducted

entirely via electronic balloting over the internet, using the Iowa Student Information System (ISIS).[1] At about that time UISG solicited the help of salaried programmers working for the University of Iowa (UI) to create the web and database environments necessary for voting.

After UISG decided to move to IRV, the ISIS team was unable to commit the resources necessary to support IRV ballot counting. This led UISG to solicit the help of the Department of Computer Science for the task. The Department referred UISG to various student computing groups, but attempts to enlist the help of these groups were unsuccessful.

During the discussions with UISG representatives, Jones suggested several different formats the ISIS team could use to record ballot data which would ease the task of IRV vote counting. UISG made inquiries to the ISIS team and the ISIS team confirmed they could easily deliver the data in such a format. Jones concluded that a semi–manual count could be completed quickly and volunteered the services of the ACCURATE group at UI (UI–ACCURATE) to implement a vote–counting system for UISG.

Two weeks before the election, a Ph.D. candidate (Hansen) and an undergraduate (Thiede) were given the job of writing this vote–counting software. Development began under the working name CHAD, a humorous acronym meaning "CHAD Handles Auditable Data". The project was originally set for completion on March 3, 2006, with a deliverable on March 7.

On February 27 we were informed there would be a tech preview on March 1. CHAD was functional but not finished at that time. We had been given no opportunity to cast test ballots and we were working from an incomplete specification of the ballot file. This meant our first exposure to real data was at the tech preview. This led to some embarrassment when the data format CHAD was expecting turned out not to be identical to the format provided by the ISIS team. Thankfully, this problem proved easy to surmount.

[1] `http://isis.uiowa.edu/`

1.1 Areas of Concern

Our concerns can be broadly broken into three categories: political, administrative, and technical. We are a nonpolitical research group, and for that reason the remarks made here will be confined largely to the latter two categories.

1.1.1 Political

At the last minute before the tech preview, we were asked to extend our presentation to cover the basics of IRV and electronic balloting: round tallies, election controls, tie–breaking policies and other mechanisms. It is clear that no–one on the UISG electoral board understood IRV prior to the tech preview. The electorate might understandably be concerned about UISG's election board not understanding the concepts of IRV as late as a week before an IRV election. This must be characterized as a serious failure.

1.1.2 Administrative

1. The administrative organization of UISG was never clearly communicated to us. This is counter to electoral best practices, wherein the chain of authority should be clearly documented, from legislative mandate down through those responsible for policy implementation. We still don't know who ultimately held responsibility for the safety and accuracy of the election.

2. We were never given a reasonable chance to meet with the ISIS team to discuss technical matters. Instead, almost all communication was mediated through UISG representatives. This made it difficult for us to ensure CHAD would interoperate correctly with ISIS–supplied vote data. This inability to collaborate was counter to the best practices of software engineering, which holds that cross–communication between development teams is essential to building reliable and software.[2,3] As a consequence of this lack of cross–communication, embarrassing flaws occurred in both the tech preview and the actual election.

3. We were given an incredibly short time frame for this project. To make matters worse, the requirements we were asked to meet were constantly changing. Even for full–time software developers, two weeks is simply not enough time to have great confidence in the reliability or security of any piece of software, no matter how trivial. Two weeks is perhaps adequate for a single generation of rapid prototyping, but not for a final deliverable and especially not if the requirements are still in rapid flux.

1.1.3 Technical

1. While we give cautious assurance that CHAD is without obvious flaws or deficiencies, we can give no assurances for the ISIS–hosted vote recording software. To the best of our knowledge that system was not made available, and is still not available, for public scrutiny.

   The information security community defines a *trusted system* as one which can violate the larger system's security policies.[4] By this definition, the ISIS vote recording system is a very trusted system, yet we are aware of no compelling reason why it should be so.[5]

   We note that ISIS is strongly constrained by the Family Educational Rights and Privacy Act of 1974 (FERPA).[6] This mandates a level of security that is comparable to the level mandated for some trusted systems, although without the rigor usually associated with trusted system administration.

   Please note that we are *not* suggesting the programmers who implemented the ISIS–based vote recording system are in any way corrupt or incompetent. We are only pointing out that when it comes to elections, trust is like ha-

[2]Steve McConnell, *Rapid Development: Taming Wild Software Schedules*. Microsoft Press, 1996. "Information must be readily available. . . . Team members need informal opportunities to raise issues in an environment where titles, positions, office sizes and power ties are not part of the equation."

[3]Roger S. Pressman, *Software Engineering: A Practitioner's Approach*. McGraw–Hill, 2005. "Effective communication (among technical peers, with the customer and other stakeholders, and with project managers) is among the most challenging activities that confront a software engineer. . . . Collaboration and consensus occur when the collective knowledge of members of the team is combined to describe product or system functions or features. Each small collaboration serves to . . . creat[e] a common goal for the team."

[4]Interested readers are referred to the Common Criteria (ISO/IEC 15408) and its predecessor, the Trusted Computer System Evaluation Criteria (DOD 5200.28–STD).

[5]It is certainly possible that by exposing this part of ISIS for public inspection, it would weaken the security of the nonexposed parts of ISIS. If this is the case, the solution would appear to be to redesign ISIS: anything else is an appeal to security through obscurity.

[6]20 USC §1232g; 34 CFR Part 99

banero peppers: best in extraordinarily small quantities.

2. The incredibly tight time frame for this project forced us to disregard a number of important rules of software engineering. While our practices can be understood in light of the development schedule, they should not be overlooked. We recommend that the current version of CHAD be viewed as a prototype, and as with all prototypes, it should be completely overhauled and subjected to rigorous reimplementation.[7]

2 Polling

To conduct a fair election, it is necessary to accurately capture the intent of the voters, transport a record of that intent to the place of counting, and accurately count the votes. The purpose of developing polling protocols is to ensure that each of these steps in the election is properly carried out.

We have essentially no information about UISG's polling protocols. This is a matter of considerable concern to us, because transparency is essential for public confidence in an election. It might also be of considerable concern to the electorate: if UISG's own election consultants are in the dark about the polling protocols, how much confidence should the electorate have in those protocols?

It is possible the protocols were well–developed but poorly–distributed. However, we believe it is far more likely that the protocols were underspecified and largely ad–hoc. There is evidence to support this proposition. For instance, UISG forbade cell phones from the counting room on the reasonable grounds of keeping the counters sequestered, but explicitly approved making use of wireless internet access. The accumulated weight of decisions such as these lead us to believe most of the polling protocol was ad–hoc.

In the event it was not ad–hoc, we would like to raise the following questions about UISG's polling protocol:

1. Was a security evaluation made of the ISIS vote recording system prior to its deployment?
2. What controls were in place to prevent election fraud from voters?
3. What controls were in place to prevent election fraud from the ISIS team?
4. What self–checks were in place to detect errors in the system?
5. What software was used to record votes?
6. What engineering methodology was used in developing and/or certifying this software?
7. Will the ISIS software be made available for public inspection?
8. What evidence can be offered to the public that the vote recording software used in the election was indeed the software certified for this purpose?
9. What controls were in place to guarantee the confidentiality of each ballot?

[7] Frederick P. Brooks, Jr. *The Mythical Man–Month: Essays on Software Engineering.* Addison Wesley Longman, 1995. "The management question, therefore, is not *whether* to build a pilot system and throw it away. You *will* do that. The only question is whether to plan in advance to build a throwaway or to promise to deliver the throwaway to customers... Delivering that throwaway to customers buys time, but it does so only at the cost of agony for the user, distraction for the bulders as they do the redesign, and a bad reputation for the product that the best redesign will find hard to live down."

10. What controls were in place to guarantee the confidentiality of each voter?

We have evidence indicating no security evaluation was made; that no self–checks were in place to detect errors in the system; and that the vote recording system was an ad–hoc construction without much in the way of formal software engineering process behind it.

We also note that FERPA strongly constrains ISIS in ways which affect many of the other questions, particularly with respect to confidentiality and fraud. Given the requirements of FERPA, it's very possible that many of these questions have answers grounded in the best practices of voting. It's also very possible that they will not. In the realm of voting, skepticism is a virtue.

3 Ballot Transport

Prior to election day, we wrote a detailed how–to manual for using CHAD in elections. Much of this manual was concerned with procedures for ensuring the accurate transport of votes.

We saw no evidence that anyone other than ourselves read the manual.

3.1 Prior to Receipt

Two representatives of UI–ACCURATE (Hansen and Thiede) arrived at the Iowa Memorial Union (IMU) just past six o'clock on election night, well ahead of the agreed–upon six–thirty meeting time for election officials. One UI–ACCURATE member (Jones) remained in his office in another building, so that he could conduct an independent hand count of the data without concern for corrupting the UISG counting process.

Upon entering the counting room in the IMU, we attempted to set up laptops and make secure connections back to the machine which was acting as the central election server. This server is located on the third floor of MacLean Hall in a locked computer lab. Relatively few people have access and their comings and goings can be monitored via the passcard entry system. Our group and UISG agreed that conducting the vote tally on this machine was preferable due to its increased level of physical security.

However, this plan relied upon internet connectivity in the place of counting. Internet connectivity turned out to be unreliable. Future elections should ensure the availability of internet access prior to election night.

3.2 Receipt

3.2.1 Failure to Authenticate

Within minutes of the close of polls, electronic copies of the ballots were sent via email to Jones and the UISG election committee. However, neither UISG nor the ISIS team followed the authentication procedure specified in the CHAD documentation, despite our clear warnings that this was required for voter confidence.

Email is neither secret nor inviolable. It can be read, even tampered with, in transit. Let us be unequivocal: without authentication it cannot be known whether the ballots the UISG election committee received were the same ones the ISIS team sent.

We attempted to explain this to the ISIS team. After considerable delay, the ISIS team sent authentication for the vote data—but did so via email. This is a textbook example of a chicken–and–egg problem. In the absence of authentication there is no assurance that email is received in an unadulterated form. Attempting to authenticate email *via email* just compounds the problem!

The CHAD manual covers this subject in detail and presents a protocol which avoids this problem altogether. Unfortunately, it was quite clear to us that the only people who read the CHAD manual were its authors.

We must be frank: we have absolutely no confidence in the integrity of the ballots. They might have been correct, or they might not have been. There is absolutely no evidence to suggest one way or another.

As things stand, there is no evidence the ballots we received are authentic. There is also no evidence they are not. It is improper to draw any conclusions about the integrity of the ballots in the absence of any evidence. However, we would be remiss if we were to neglect the obvious point that a lack of credibility in the ballots must necessarily lead to a lack of credibility in the election itself.

3.2.2 Buggy Ballots

While Hansen and Thiede were wrestling with issues of ballot integrity, Jones was discovering problems with the ballots.

Jones was neither part of UISG's election committee nor present in the counting room, and thus could look at his copy of the ballots without jeopardizing the impartiality or correctness of the official count. He quickly discovered irregularities in the reported votes. It appeared that many people spoiled their ballot by voting for the same party more than once.

We sent emails back and forth between us for a while considering the possible effects this might have on the outcome.[8] We determined this was unlikely to affect the vote, and further, that if it did affect the vote, it would be trivial to detect.

[8]As mentioned, internet connectivity was very spotty. Fortunately, email can be sent and received in only a few seconds, so it was sufficient for this purpose.

At this point, there were three major options for how to proceed with the election count:

- Throw out all the ballots sight–unseen and call for a new election
- Inspect the ballots and decide whether the count should take place at all
- Count the ballots, then inspect them and decide whether the count should be certified

The first option is unreasonable because it is unnecessarily rash. Ballots should enjoy a presumption of validity, even given that this presumption had already been jeopardized by the lack of authentication.

The second option is unsafe because it allows an election worker the opportunity to open up the ballot file prior to the count. It is not necessary to assume the existence of a malicious election worker; the assumption of human fallibility is more than enough. Mistakes get made. Things are accidentally deleted. A finger might accidentally tap a key and introduce extraneous data. The possibilities for data corruption are virtually limitless.

The third option is the only choice in keeping with the best practices of voting. A count, by itself, is meaningless until certified by the election authority. It does no harm to make the count, and mitigates the potential for inadvertent damage during the inspection process.

We strongly recommended the third option. UISG elected for the second. A UISG election worker inspected the ballots to see if Jones' report was correct. It appeared that it was.

According to the CHAD manual, in the event of an incorrect or corrupted transmission of ballots the data would be copied directly to electronic media and couriered over to the counting room in the IMU. UISG

instead elected to have the ISIS team look into the ballots and send a new copy of the data. The ISIS team lead wasn't in his office, and it took him roughly a half–hour to get back there.

3.2.3 Data Overexposure

For purposes of security, CHAD tallies votes for symbols and not tickets. Each ticket is replaced with a randomly–chosen letter. This helps prevent CHAD's programmers from putting in a "back door" which could surreptitiously throw votes to a preferred ticket. If CHAD never gets the information, CHAD can't cheat with the information.

At the end of the election, CHAD announces which letter won. Election officials use that information to declare a winner. We emphatically recommended to UISG that no–one with knowledge of the letter–ticket pairs should be present for the counting. If no–one who is doing the counting knows what letters correspond to which tickets, no–one in the counting process can throw votes towards a preferred ticket.

While waiting for the ISIS administrator to return to his office, a UISG election official set down a pad of paper on the table in front of both UI–ACCURATE representatives. Written at the top of this pad was the mapping between letters and tickets!

UISG was not the only offender. The ISIS team sent Jones the raw vote data and the ticket–symbol pairs. This cavalier neglect by both UISG and the ISIS team needs to be viewed as a critical failure on the parts of both parties to carry out their electoral responsibilities.

We should note that we must now be added to the list of people who could have potentially tampered with the election, since we had been given the knowledge of which candidate was represented by which letter.[9]

[9]For those who are keeping track, this list now contains everyone involved in the election process.

3.2.4 Buggy Ballots II

After arriving at his office, the ISIS team leader was left alone to study the ballots as they sat in a database and to re–extract them into a corrected file for our use in CHAD. This confused us. In an election, ballot integrity must be protected. It was unclear to us how this was meant to be achieved by leaving someone alone with unrestricted access to the ballot database.

Ultimately, a corrected file was extracted from the database and sent on to UISG after an hour–long delay. Just as with the first (damaged) set of ballots, this set of ballots was sent without authentication.

In light of these failures, we can only conclude the ballot transport step was plagued by critical and unrecoverable failures. We once more repeat that we have no evidence of deliberate misconduct at any step; we further repeat that this lack of evidence does not warrant confidence in the results.

3.3 Failure Analysis

The ISIS team has not adequately explained the source of the corruption. The explanations we have been given are superficial at best. According to the ISIS team, the source of corruption in the first set of data turned out to be a trivial bug.[10] According to ISIS, the program which extracted ballots from the ISIS database assumed that there were no undervotes: that each voter had given a preference to each candidate. This was a faulty assumption. Many voters gave incomplete orderings, which is not only allowed in IRV but is expected in any real–world election. However, the damaged ballot file possesses some properties which indicate that at least some of the time, undervotes were being properly extracted. This

[10]This raises the question of how a "trivial bug" survived even basic testing.

leads us to question the sufficiency of an explanation as simple as that provided by the ISIS team.

Not only that, but our own failure analysis showed the same data corruption was present in the data given to us at the tech preview on March 1. CHAD has inherent to it an assumption common in computer science—that the inputs to it will be meaningful and properly formed. This is a dangerous assumption when dealing with an untrusted system, but the entire point of a trusted system is that these sorts of assumptions can be made. Given that ISIS was clearly a trusted system, we must question the appropriateness of that trust.

The malformed input file was immediately obvious upon studying it. Had the ISIS team reviewed their own outputs with any rigor, they clearly would have discovered the presence of this corrupting bug. That they did not strongly indicates to us that the ISIS team's engineering methodology lacked either rigor or a commitment to the best practices of the field.

Similarly, the insufficiency of the ISIS team's explanation of the bug strongly suggests an ad–hoc approach to problem–solving ("shotgun debugging", as some of us would call it) and a failure to live up to the standards of software engineering.

We have several ideas for what might have happened, many of which run along similar lines to the official explanation given by the ISIS team, but none completely describe the observed phenomena within the damaged ballot file. Statistical comparison between the damaged ballot file and the corrected ballot file appears to indicate the damaged file was substantially similar to the corrected file, with the introduction of statistical noise. The same winner would have been selected using either set of data. This evidence leads us to believe that the corrected file was not tampered with after the transfer of the initial (damaged) file.

All of our uncertainties aside, we can point to problems in the ballot file and answer in the negative our earlier question of whether the ISIS vote–recording software had sufficient self–checks. Some ballots were clearly mangled, and the number of ballots in the data file did not correspond to the number of ballots cast. Even the most rudimentary checking of the data, such as comparing the ballots cast to the number of records in the database, would have revealed these problems.[11] We consider this to be evidence that the ISIS vote–recording software was put together in an ad–hoc manner, without much (if anything) in the way of software engineering process.

4 Counting

Once the corrected (but unauthenticated) ballots were received, UISG inspected them to ensure their correctness. We note again that this is precisely backwards from the best practices of voting.

UISG gave us the go–ahead to run CHAD on the ballots, which took somewhat longer than expected due to the poor network connectivity in the counting room. However, once a connection was made to the central election server and the ballots transferred,[12] counting was swift and straightforward.

CHAD operated within its parameters, running to completion in essentially no time. After CHAD finished, Hansen and Thiede contacted Jones in order to get his independent hand–count. CHAD was in

[11] The ISIS–hosted DRE system, as it turns out, was not even capable of this trivial self–check. The number of cast ballots was determined by counting the records in the resulting file, which rather misses the point.

[12] This transfer was authenticated, although it was a case of locking the barn door after the cattle had fled.

perfect agreement with the hand–count results.

We and a UISG election official selected and reviewed a portion of CHAD's audit logs. The audit logs showed no unexpected behaviors. CHAD's math checked out, and CHAD's selection of which candidate to drop was correct.

5 Post–Election

At the tech preview, we reached an agreement with UISG that the CHAD audit logs, input files, and all other data associated with the UISG presidential election would be made available on the web for public review. On election night, when UISG's spokesperson announced results she also announced that this data would be made publically available.

We took custody of the data and are currently storing it. No–one has asked for certified copies of the election results. The data have not been published on UISG's website; they have not been published in the *Daily Iowan;* they have not been asked for by any of the parties involved in the election.

Aside from our independent hand–count and examination of the CHAD audit logs, there has been no review (public or private) of any step of the vote count.

6 Lessons Learned

6.1 Applicability

The UISG election process is an interesting microcosm for the study of larger, higher–stakes elections. Several traits make it so, among them being:

1. Roughly half of the American electorate has at least some of college education, and many election officials lack college education. In the UISG student elections, the entire electorate (from which, clearly, election officials are drawn) has at least some college education. Thus, we conclude the UISG electorate and election officials are far better–educated than the American public.

2. UISG election workers are overwhelmingly young and technically aware. All election workers we encountered were familiar with concepts like electronic mail, instant messaging, files on a storage device, and other common technologies. No election worker was surprised that we could count the ballots on a remote computer over the internet.

3. UISG's election workers are future politicians, poll workers, county clerks, lawyers, civil–rights activists, and more. Their failure to execute a credible election should be viewed only as a lack of understanding, not as a lack of drive or passion. To the contrary, we can only recommend that election workers in real–world elections demonstrate the same degree of volunteerism and civic zeal.

4. The student body of UI is, speaking generally, politically active and aware. Rallies, protest marches, impassioned letters to the *Daily Iowan* and issue–awareness days are part of daily life on campus, far more so than in the community at large.

6.2 Suggestions

In short, voting researchers could not ask for a better environment in which to study election failures. Any error made by UISG is an error likely to occur in a real–world election. Thus, we come to the following

suggestions for real–world election boards and voting researchers:

6.2.1 Technical Skill

We may have radically underestimated the level of technical skill required to administer an electronic election. The voting research community has noted the graying of election workers and has raised the understandable question of whether it's reasonable to expect senior citizens to have the technical skills required to administer DRE–based elections.

Implicit in this question is the assumption that a younger, more technically savvy force of election workers would be more capable. Based on our experience with the UISG student elections, we must sharply question this assumption.

6.2.2 Misestimation of Risk

Despite our warnings about the unsuitability of CHAD, UISG officials were quite pleased with its performance and made no inquiries about sending it back to the drawing board for re–engineering. It appears likely that the perception of its correct operation led to an inappropriate level of confidence in its continued correct operation.

Two things are deserving of particular note:

1. CHAD failed at the tech preview due to our working from a different ballot file specification than the one the ISIS team was providing. Despite this failure, UISG continued to have great confidence in the system rather than ask pointed (and deserved) questions about CHAD's reliability in a real election.

2. Every opportunity for systems integration (both at the tech preview and on election night) failed. By any reasonable software engineering metric, the complete system (CHAD plus ISIS–hosted DRE) had a zero percent success rate. "Do–overs" and second attempts in the wake of failure ought not be considered as successes. Yet, UISG's confidence in the complete system and its subcomponents appears unflagging.

This disturbing tendency towards the normalization of component failure is, of course, not unique to election committees. In the aftermath of the 1986 launch failure of the Space Shuttle *Challenger*, the Rogers Commission accused NASA of using past success in the face of out–of–specification components as evidence that those components being out–of–specification presented no flight risk.[13] The institutional inertia within the NASA bureaucracy was such that these misestimations continued and ultimately contributed to the re–entry failure of the Space Shuttle *Columbia* seventeen years later.[14]

[13]Richard P. Feynman, "The Rogers Commission Report on the Space Shuttle Challenger Accident", Appendix F. "We have . . . found that certification criteria used in Flight Readiness Reviews often develop a gradually decreasing strictness. The argument that the same risk was flown before without failure is often accepted as an argument for the safety of accepting it again. Because of this, obvious weaknesses are accepted again and again, sometimes without a sufficiently serious attempt to remedy them. . . . [W]hy do we find such an enormous disparity between the management [reliability] estimate and the judgment of the engineers? It would appear that . . . the management of NASA exaggerates the reliability of its product, to the point of fantasy.

[14]The Columbia Accident Investigation Board, "The CAIB Report, Vol. I", pg. 101: "By the eve of the *Columbia* accident, institutional practices that were in effect at the time of the *Challenger* accident—such as inadequate concern over deviations from expected performance, a silent safety program, and schedule pressure—had returned to NASA."

If NASA—a world–class engineering outfit by any measure, whose bureaucracy is filled with Ph.D.s in extraordinarily technical fields—is so prone to misestimate risk, it would be foolish of us to expect an election board to be any better.

6.2.3 Inadequate Oversight

UISG's oversight for the implementation of electronic IRV was sadly lacking. It appears that once the responsibility for its correct implementation was delegated to outside agencies, UISG considered it to be a solved problem and moved on to the next entry on their lengthy list of pre–election tasks.

Delegating certain election–related tasks to contractors and/or nonpartisan agencies can be appropriate, even recommended, but those tasks must be supervised and audited to ensure confidence in the results. We suspect that real–world electoral boards, which have even lengthier lists of pre–election tasks, will be under even more implicit pressure to ignore their oversight responsibilities where contractors are concerned.

6.2.4 Inadequate Briefing

As evidenced by the last–minute request for a presentation on IRV balloting, a week before the election UISG's election board still had no idea how the votes would be counted. We cannot comment on the political motives behind the move to IRV. Yet, our lack of ability to comment on the political motives may reflect on how well the legislative body publicized the changes and the reasons underlying them.

However, whether the legislative body understands the issues involved in a new election technology is irrelevant to whether the election boards understand those same issues. Legislative bodies must ensure the election boards are properly briefed.

6.2.5 Inadequate Rehearsal

UISG had no dress rehearsal for the election. No reason was offered (or, for that matter, asked). Students have busy schedules, juggling classes and work and family commitments. So, too, do professors and programmers and everyone else involved. This is shared with real–world elections in which most of the workers are volunteers, all of whom already have their own busy lives.

UISG's only nod towards a dress rehearsal came in the form of the tech preview they requested on March 1. We asked that this preview, which was slated to be a demonstration of CHAD and nothing more, be expanded to a full end–to–end test of the system and all its linkages. Our request was not enacted, and as a result UISG had no opportunity to familiarize itself with the link–level authentication protocols mentioned in the CHAD manual.

It's a great temptation to write off rehearsals in the name of there not being enough hours in the day. This temptation must be resisted. The alternative is for the election itself to be a dress rehearsal of the next year's election.

6.2.6 Flawed Protocol Execution

Election–night procedures, especially ballot transportation procedures, were largely ad–hoc even when formal processes were available. It appears that UISG felt the presence of election–security experts was sufficient to ensure the security of the election, regardless of whether the counsel of those experts was heeded.

This sense of sufficiency is tempting despite its flawed logical premises: so much so that we suspect real–world elections will have to wrestle with this same issue. Those tasked with election security must be mindful of the possibility that their presence will be viewed as talismanic rather than as a

knowledge resource.

6.2.7 Inadequate Post–Election Protocol

In the aftermath of an election, the winner may ride on such a wave of popular acclaim as to make post–election protocol seem like a superfluous gesture. This is what appears to have happened in the UISG election. The logic seems to be that the post–election protocol is necessary only if the election is in dispute and otherwise is extraneous.

We understand and sympathize with UISG's feelings. However, post–election protocol is necessary even when the outcome is in absolutely no doubt. It is important to create a culture of openness and transparency, in order to stymie future attempts to subvert the system. If the post–election protocols are ignored, then over time the electorate will become accustomed to a lack of transparency in the results. That creates an environment in which electoral misconduct thrives.

We must anticipate that real–world election boards will have a similar reluctance to execute post–election protocol as diligently as perhaps they ought.

7 Conclusion

We found major problems with the administration of the most recent UISG elections, most especially in the areas of election software and ballot transport.

These flaws lead quite directly to a crisis of confidence in the electoral results. We have no evidence the ballots were at any time tampered with, but neither do we have evidence the ballots were *not* tampered with.

We cannot conclude whether UISG elections were fair. Due to a lack of evidence, the 2006 UISG elections exist beyond the reach of such questions.

We conclude UISG has been far too willing to trust nonpolitical groups, such as ourselves and the ISIS team.

We make no accusations of malfeasance or deceit. All parties we encountered were helpful and straightforward. Some members of the ISIS team are known personally by us, and we know them to be of the highest integrity. However, this should carry no weight in a public election review.

We strongly recommend that next year UISG empanel its own voting systems group, and give it the authority to administer the elections in accordance with voting protocols UISG will approve prior to an election. If UISG must contract out the election process, they must monitor their contractors and audit their work.