E-voting: Dependability Requirements and Design for Dependability

J W Bryans
School of Computing Science,
University of Newcastle upon Tyne, UK
Jeremy.Bryans@newcastle.ac.uk

B Littlewood
Centre for Software Reliability,
City University, London, UK
B.Littlewood@csr.city.ac.uk

P Y A Ryan
School of Computing Science,
University of Newcastle upon Tyne, UK
Peter.Ryan@newcastle.ac.uk

L Strigini
Centre for Software Reliability,
City University, London, UK
L.Strigini@csr.city.ac.uk

Abstract

Elections are increasingly dependent on computers and telecommunication systems. Such "E-voting" schemes create socio-technical systems (combinations of technology and human organisations) that are complex and critical, as the future of nations depends on their proper operation. Thus heated debate surrounds their adoption and the possible methods for making them demonstrably dependable. We discuss the dependability requirements for such systems, and the design issues in ensuring their satisfaction, with reference to a recent proposal that uses cryptography for fault tolerance, in order to avoid some of the perceived dangers of electronic voting. Our treatment highlights the need for considering the whole socio-technical system, and for integrating security and fault tolerance viewpoints.

1 Introduction

Many electronic voting methods are currently being investigated in many countries. Brazil held its first fully electronic national election in 2002. State-level electronic elections have been held in the US. Across Europe many countries have also trialled electronic voting systems.

"E-voting" has both potential advantages and risks. These systems can make the casting of a vote more convenient and may therefore lead to improved turnout. Electronic recording and counting of votes could be faster, more accurate and less labour intensive. Digital technology could also provide greater anonymity than conventional approaches.

On the side of risks, the scenario of integrated computer and communication systems performing all functions from collecting the voter's opinion (without paper records) through transmitting and counting them raises the possibility of large-scale vote-forging and/or spying on voters (and thus coercion or vote-buying). The possibility of these systems being implemented on current off-the-shelf computing platforms and with the low assurance standards common in much of the software industry has caused many experts to voice a concern that huge risks are being taken and spurred demands for voter verifiability of the functioning of voting machines, and guaranteed possibility of recounts, usually via paper records.

Any national or state election is a large, complex and important system, involving millions of voters and thousands of officials, as well as being increasingly reliant on IT. It is of immense importance that these systems be dependable. In this paper, we discuss the conventional dependability attributes of an e-voting system, including accuracy, secrecy, availability, and discuss the design concerns involved in guaranteeing them. Our focus is on proposed methods for reducing the current hazards in elections using voting machines in secret booths, not on the more extreme proposals for internet or mobile-phoned based voting. However, our discussion of requirements has a very general scope. We also consider the issue of reputation, and show that attacks on the reputation of a electronic voting system may potentially be more damaging than attacks on the system itself.

The paper is organised as follows. Section 2 discusses the balance between design time and run time measures for achieving and assessing the dependability of such a complex system. Section 3 describes the high-level dependability requirements for such a system. Section 4 introduces the Prêt à Voter scheme [5] for ensuring tamper-resistant, secret, paper-less transmission of ballots. Section 5 proceeds to show the fault-tolerant design implied by the proposed scheme and discuss the allocation of dependability requirements to subsystems, for defence against both accidental fault and malicious attacks. Sections 6 and 7 further discuss

some peculiarities of this socio-technical system: the composite, human and technical recovery mechanisms required for detected failures, and the risk of "loss of reputation". Our conclusions follow in Section 8.

2 Assurance

In security there is a common tendency to look for provable correctness – impossibility of successful attacks – but the community's attitude has been evolving towards a more substantial role for fault tolerance and for probabilistic assessment (as highlighted by the increasing visibility of security in recent dependability conferences). It is considered essential that mechanisms be provided to detect, contain and recover from failures. These mechanisms need to be robust in the face of malicious as well as accidental threats.

There are a number of ways in which assurance in a system's correct behaviour with respect to a specification can be achieved. These can be thought of as lying along a spectrum with pure verification at one end and run time monitoring at the other, with testing lying somewhere in between. It is clear that we should use all of these techniques in combination to achieve increased levels of assurance.

In order to verify a system we assume some model of its behaviour and subject this to various forms of mathematical analysis to prove that it will satisfy certain (formally stated) requirements. This is fine up to a point but suffers from a number of deficiencies:

- Our analysis will only be as good as the models on which it is based. Unless we have succeeded in ensuring that our models are entirely faithful with respect to the properties of interest, proofs about the models will not necessarily carry over to the real system.
- It is difficult to ensure that the verified system will correspond exactly to the fielded system. Even supposing that our models start off as being faithful representations of the real system and its environment, systems and their environments evolve. This evolution, which could include degradation, patches etc, can invalidate the original analysis.

On the other hand, run time monitoring (error detection) also improves assurance. It directly improves dependability, because detecting unwanted states ("errors") within the system can be made to trigger containment and recovery mechanisms; and it will deter some possible attacks[1]. But it also suffers various drawbacks:

- Monitoring can be difficult: you need to know exactly what to monitor, monitoring has to be accurate and, in a hostile environment, robust against subversion.
- Good response and recovery strategies may be difficult to devise and execute.
- Detecting a violation at run time may be too late. For example, once a secret is out, it cannot be recovered.
- Inappropriate or compromised behaviour may not be easy to detect. Some properties, by their very nature, are not amenable to run time monitoring within the system that needs to satisfy them. Information flow properties (e.g. an attacker cannot read information transmitted along a wire) are a case in point. It is often the case that no monitorable event occurs when an information flow is violated.

We see these issues very clearly illustrated in the context of digital and electronic voting systems. Many of the proposed and even deployed electronic voting systems depend heavily on claims for verified and tested code, but fail to provide even the possibility of run time monitoring.

The Prêt à Voter Scheme [5], which we discuss in Section 4, is almost at the other extreme: for the accuracy property, virtually no reliance need be placed in the deployed components and all the assurance comes from close monitoring of the behaviour of the components. For the secrecy and availability requirements some verification will be necessary. It is therefore more amenable than many other voting systems to a dependability or fault tolerance style analysis.

3 Electronic Voting Dependability Requirements

There are a large number of types of election, common examples being "First Past The Post" and "Proportional Representation". Although the requirements we consider here will apply to all elections, the type of election itself will introduce its own complexities. For concreteness, in this paper we will consider a very simple scenario. We will assume several candidates, with each voter selecting only one, and the winner being the candidate with the most votes.

We will refer to the dependability of an "election system", by which we will mean the socio-technical system – people, as well as physical, computing and communication resources – performing all functions from collecting the vote through to ultimately producing the final tally.

We outline in this section a template of attributes and dependability requirements, that one would need to elicit in order to design and assess an election system. The details

[1] We will follow the convention of calling *error* an undesired state inside the system, *failure* an undesired output of the system to the external world, irrespective of whether they are caused accidentally or by malice [1]. Errors may or may not cause failures, and the goal of fault-tolerant design is to reduce the probability of their causing failures.

to fit into the template (down to numerical values of parameters) should be informed by the special type of election used in that society, the preferences of the society using the system and the specific threats that elections face in that society; politicians required to state these requirements would obviously need the assistance of disciplines outside computing and engineering.

The main requirements of an election system belong to two categories: accuracy and ballot secrecy. Ballot information should be transmitted and counted correctly, and the link between voters and the votes they have cast should be secret. Neither property is fully guaranteed in current elections, so a reasonable requirement for any improved election system is for counting errors and violations of secrecy to be within acceptable limits. Dependability requirements will be about sufficiently low probabilities of these limits being exceeded, given the environment in which the system is deployed and, in particular, the threat profile, e.g., the technical ability, aggressiveness, and willingness to take risks of the potential attackers. We will discuss these acceptable limits and dependability requirements in the next subsections. The primary dependability requirements about accuracy and ballot secrecy imply secondary requirements: the system must be robust and resilient in the face of accidental and malicious threats. The balance between imperviousness to attack (or avoidance of accidental faults) and ability to survive them (or tolerance of accidental faults) is a matter of design trade-offs; but in practice a large amount of the latter seems necessary, since the system needs to survive highly competent and motivated attacks, and to be trusted by the general population. If the system does fail with respect to the primary requirements (i.e. accuracy or secrecy requirements are violated) in an election, it is essential that the failure be detected and flagged, so that it can be properly dealt with.

It is also essential for such a system to gain public trust in its accuracy and secrecy. One way to help engender confidence in the accuracy of the system is to provide voter verifiability: some way for the voter to assure themselves that their vote has been accurately included in the tally.

An important difficulty with these requirements is that naive implementations of verifiability would immediately violate secrecy, by requiring that votes be traceable back to the voters, or that a voter declare his/her vote when claiming that it has been miscounted. So, "end-to-end" checks, comparing the output of the system against its input, are not feasible without jeopardising secrecy. Assurance of accuracy must thus rest on assurance (by prior verification and/or run time monitoring) of the proper functioning of the mechanisms meant to protect it.

3.1 Accuracy

At the most abstract level, we would like the outcome of an election to accurately reflect the "intentions" of the eligible electorate. At this level we would need to consider social and psychological issues that might, for example, favour certain sectors of society, bias voter choices or encourage voter error.

In this paper we will restrict ourselves to the purely technical question of ensuring that votes counted in the final tally accurately reflect votes cast. We will assume that issues of authentication and prevention of double voting have been addressed.

Scientific studies of dependability, particularly of software-based systems, have long exhibited some tension between "perfection" and "good enough". It has sometimes been said that a computer program can be made fault-free so that it will never fail during its life: from a dependability perspective it is "perfectly reliable". Claims for complete perfection of this kind are now rarely, if ever, made. Instead, it is generally recognised that programs of even modest complexity may contain faults that will show themselves as failures at some time during the operational life of the system. Once this view is taken, questions about the acceptability for use of a system will involve dependability attributes such as reliability and safety: e.g. they will address questions such as "will it fail sufficiently infrequently?"

These considerations also apply to voting systems. Whilst we would *like* to have perfect accuracy, i.e. a complete *guarantee* that the result of an election reflects in all respects the voting intentions of the electorate, this seems an unrealistic goal in practice[2]. Rather, we need to know that a result is *sufficiently* accurate, or (expressing it slightly differently) that sufficient confidence can be placed in the result.

In order to be able to talk about the "dependability" of a voting scheme we need to put some flesh on the bones of informal concepts like "sufficient". The key here is uncertainty. There will be uncertainty about the nature and number of faults in the voting system, there will be uncertainty about the kinds and frequency of malicious threats it might meet. The result will be uncertainty in the relationship between the reported result of an election and the "true" result. As usual in dependability, the appropriate calculus for uncertainty is probability.

There are three potential sources of uncertainty: collection, transmission, and counting of votes. Even if we knew the intentions of all voters, there would be some uncertainty as to what is collected, and in how that is transmitted; even

[2]This remark is not intended to imply that approaches that seek perfection are invalid. On the contrary they may be plausible means of *achieving* dependability. For example, it may be possible to prove the absence of a particular class of faults: a kind of conditional perfection.

if we knew what is transmitted, there is some uncertainty as to how the outcome of the election is decided (for an example of this in paper-based elections, consider the variations in recorded totals between successive recounts in closely contested elections).

In any calculation, all the variables involved (votes as cast, votes as transmitted, etc.) would be *random* variables. So any measure of the discrepancies must be a stochastic one. For example, we might say that a voting procedure is "sufficiently accurate" so long as the *expected* proportion of votes reported for every candidate does not differ from the actual proportion by more than 1%. Alternatively, we might require that there is less than a 5% chance that any difference between a reported proportion and the actual one exceeds 1%.

Clearly, other formulations of what we mean (i.e. require) by *sufficiently* accurate are possible, even in this extremely simple example. In more complex voting schemes this issue of deciding what we mean by "sufficiency" may turn out to be quite difficult, but obviously necessary for a rational choice of system design.

Another difficulty in voting schemes is that the input for a particular election will generally be unknown: the confidentiality requirement will see to that. So, directly checking whether a particular election result is sufficiently accurate will be difficult or impossible. Of course, there are many systems for which we cannot know the exact value of the input data, but here we are specifically forbidden to find it out, by the requirement of confidentiality.

On the other hand, we can *test* a voting scheme to see how it behaves when presented with known inputs, and even with selected types of attacks or attackers ("tiger teams"). This will give some information – deterministic or probabilistic – about its dependability with respect to accuracy requirements, either conditional on the scenario adopted on testing, or unconditional, assuming a distribution of inputs and threats. Techniques akin to the various forms of software testing and fault injection have their place here.

3.2 Ballot Secrecy and Voter Anonymity

It will typically be a requirement that the way any individual voter voted remain secret. Besides the natural desire for privacy, ballot secrecy serves to prevent coercion or vote buying. The key point is that there should be no way a third party can determine which way a voter voted, even with voter cooperation.

Note that absolute assurances of total secrecy may not be realistic here. In certain exceptional circumstances secrecy will be violated: for example, if all the votes went one way.

Instead of ballot secrecy we might require voter anonymity. At first glance one might suppose that these are equivalent. We define anonymity to be that the observable behaviour of the system remains the same under arbitrary permutations of the input ballots. This approach is formalised in [6] using the process algebra CSP. Using this definition, the scenario of everyone voting for the same candidate would still be deemed to satisfy anonymity but would fail the ballot secrecy requirement. Given that such a scenario is perfectly admissible and that the violation of ballot secrecy seems inevitable, this would seem to suggest that voter anonymity is the more appropriate requirement.

We will continue to use the informal name "secrecy" for the set of attributes and requirements about constraints on what one can learn about another's vote.

How would one define dependability requirements with respect to the secrecy attribute? Vote secrecy is limited by many factors outside the election system proper, so that a practically zero probability of secrecy violations in the election system itself, even if possible, may not be required. Instead, we will have a notion of "secret enough": a bound on the number and pattern of compromises of secrecy that are rare and limited enough not to endanger the general integrity of the process. However, this bound will depend more strongly than the one for "accurate enough" on the situation around an election: e.g., in a society where reprisals against opposition voters are likely and severe, the knowledge that even a small sample of votes can be known to the government may be enough to allow widespread intimidation.

3.3 Voter interface issues

An e-voting system that is used by a large and diverse group of people must have a readily understandable user interface, and the tallying and auditing mechanisms must be throughly understood by the people responsible.

More subtly, the voting procedure should not introduce any bias into the choice that the voter makes. If the ballot includes questions that are not of interest to a voter, then he or she may tend to chose (for example) the first of a number of options. An assertion about the lack of bias of a voting procedure (procedural invariance) would need to be substantiated by a cognitive argument.

3.4 Verifiability

A voter should have grounds to trust that his or her vote has been properly counted. Further, if it has not been properly counted, the voter should have a means of recourse to demonstrate this. However, this means of recourse must not be able to be used by a third party to coerce the voter into revealing his or her vote.

Demonstrating to yourself that your vote was counted is important, but it is also important that any voter can be assured that all votes are counted. This is called universal

verifiability in [7]. These properties go a long way towards providing transparency of the mechanisms, and their presence in a voting system would make an important contribution to an argument for public trust.

4 The Prêt à Voter Scheme

We now present an overview of the Prêt à Voter scheme. For full details see [5]. Prêt à Voter is based on the Chaum scheme [3], but uses a radically different mechanism to represent the encrypted vote value in the ballot receipt. In place of the visual cryptographic techniques of the Chaum original, the voters are provided with a familiar-looking ballot form. The voter makes her selection in the usual way by placing a cross against the candidate of choice. Thus a ballot with a vote for the Sophist candidate is indicated thus:

Nihilist	
Buddhist	
Anarchist	
Sophist	X
Solipsist	
	$7rJ94K$

To cast the vote, the voter now separates the right hand (RH) and left hand (LH) strips. The LH strip should be discarded, by, for example, feeding it into a shredder. The RH strip is placed under an optical reader or similar device. This records the information on the RH strip: the random-looking value at the bottom of the strip and the position of the X, i.e., the numerical representation of the cell into which it has been entered. The RH strip is now returned to the voter to retain as her receipt:

X
$7rJ94K$

The ballot forms would be augmented with various anti-counterfeiting devices, and a digital signature applied to the receipt when the vote is cast.

Thus far, aside from the retention of a receipt, the process of casting a vote is entirely familiar, to a UK voter at least. Now, an objection at this point is that possession of a receipt would open up the possibility of coercion or vote-buying. The trick that sidesteps this is that the order of candidate lists on the ballot forms are randomised. Choose a ballot form at random, and the order in which the candidates are shown will be unpredictable. Clearly, with the LH strip removed, the RH strip alone does not indicate which way the vote was cast.

Now the problem is how the votes will be extracted and counted. This is where the random value printed on the bottom of the receipt comes into play. Buried cryptographically in this value is the information needed to reconstruct the candidate list shown on the LH strip and visible to the voter when they cast their vote. This information is encrypted under the secret keys of a number of *tellers*. These tellers are automated, but have a similar role to the human tellers that count the votes in a manual election. Thus, only the tellers acting in consort (in an anonymising mix [4]) are able to reconstruct the candidate order and so interpret the vote value encoded on the receipt.

Once the election has closed, all the receipts are transmitted to a central tabulation server which posts them to a secure web bulletin board (WBB). This is a write-only, publicly visible facility. Only the tabulation server can write to this and, once written, anything posted to it will remain unchanged. Voters can visit this WBB and confirm that their receipt appears correctly.

After a suitable period in which voters can verify that their receipts have been correctly posted, the set of tellers take over and perform a robust, anonymising, decryption mix on the batch of posted receipts. Intermediate stages are also posted to the WBB for partial random audits, and so is the final list of decrypted, anonymised ballots.

In summary, the *tellers* perform a set of publicly available algorithms, aimed at guaranteeing that all votes are properly decrypted and also that accidental or malicious vote tampering (altering or deleting ballot contents or adding spurious ballots) has only a minuscule probability of going undetected. We omit the details of this here, but they can be found in [5].

All this is fine as long as all the steps are performed faithfully. If we are prepared to trust the entities executing this process then we can be confident that the election will be accurate and the vote values kept secret. However, the aim of schemes like the ones devised by Chaum and Neff and Prêt à Voter is to achieve these goals without the need to place such trust in any of the components of the scheme. In section 6 we outline the mechanisms used to detect any malfunction or misbehaviour by the devices or processes that comprise the scheme.

5 Satisfying the dependability requirements

The function of the election system is to transmit the information from the voting booth to the counting stage. Prêt à Voter (like Chaum's scheme [3]) is remarkable for its *error detection* mechanisms, which, when coupled with appropriate error treatment, allows a highly dependable system to be built out of undependable components. Assurance in

the error detection mechanisms themselves, in turn, requires proof of properties of their algorithms, plus verification of their implementation.

Accuracy is assured by a series of checks on the mechanisms that perform stages of the transmission and counting. Some are performed directly by the voter in the booth (Section 6.1). The posting on websites of voters' (encrypted) retained receipts, together with people *actually checking* the posted data against those left with the voters detects errors during the transmission of the votes from the booth to the initial teller. The auditor detects accuracy errors in each decryption-based stage of the transmission.

So far, we have not considered denial-of-service attacks. In a viable design, detected errors in the core system (that part of the system defined thus far) must lead to attempts to recovery, repeated if necessary. Eventually, therefore, either the result is considered valid by the error detection mechanisms (and can thus be either a real success or a secrecy failure or an accuracy failure) or the election does not complete.

To evaluate the whole election system, it is necessary to consider the threat profile: probability of attacks of each kind. Importantly, this is affected by social and cultural factors and by the would-be *attackers'* perception of the effectiveness of the error detection and/or recovery mechanisms, and of the severity of consequences that they can trigger for the attackers. This assessment requires understanding of social and psychological factors of deterrence that are certainly outside the competence of reliability engineering. Yet, this dependability analysis suggest clearly which questions the decision makers need to ask their social scientists, historians and psychologists.

In the high-level system design stage, a designer uses the dependability requirements (Section 3) to apportion responsibility to the various subsystems, and judge whether the subsystems proposed are suitable for the system to satisfy the overall system requirements. For accuracy and secrecy failures, $Pr(undetected\ failure) \leq Pr(error\ in\ core\ system) \times (1 - coverage)$ where the $coverage$ of the error detection mechanisms is their probability of detecting an error, if an error did occur. Given the overall requirement (left-hand term), high-level design must determine the two right-hand terms. This will typically be done separately for different categories of failure modes and causes. Errors may be caused by misfortune (accidental faults) or malice (attacks, also called intentional faults). For the former, familiar techniques can be applied towards a conservative assessment of their likelihood and the coverage of the error detection mechanisms. The threat profile is an input for the designer of the detection systems that have to assure the required coverage. Technical analysis of Prêt à Voter gives a high coverage, conditional on assumptions on the type of attack (e.g., no collusion between multiple tellers and/or auditors) that must be satisfied via further technical and organisational mechanisms.

As for non-completion failures (ncf), if the reaction to a detected error were to abort an election we could write $Pr(ncf) = Pr(error)(coverage) + Pr(false\ alarm)$

We observe that shifting responsibility for avoiding undetected failures from the security of the mechanism collecting the votes to high detection coverage may shift the focus of some adversaries to causing non-completion failures, by increasing their attacks on either the core system or the error detection mechanisms. On the other hand, for a given strength of the security of the core system, adding detection mechanisms will have the desired result of reducing undetected failures and increasing the risk for attackers, but will still increase the probability of non-completion failures, via possible errors in detection mechanisms.

6 Recovery Mechanisms and Strategies

The core scheme provides a high level of assurance that any accidental or malicious corruption of votes will be detected. But these error detection mechanisms by themselves do not reduce the probability of failures. Detected errors must also be dealt with properly; aborted elections are still failures.

In this section we propose some possible strategies and explore their implications for the robustness of the scheme.

Leaving aside for the moment problems that might arise outside the core system, there are basically three failure modes with respect to the accuracy requirement:

- Ballot forms might be incorrectly constructed in that the information buried in the cryptographic value on the receipt might not in fact match the candidate order shown on the LH strip.
- The receipt might be incorrectly recorded or transmitted to the WBB.
- The tellers might fail to correctly decrypt the receipts.

For all of these there are random auditing mechanisms in place to (probabilistically) detect any malfunctions or corruption. Details can be found in [5].

Turning to the effects of attacks, our response strategy should be based on patterns of detected errors rather than isolated errors. It is possible for individual errors to look quite innocent and yet a group of errors taken together may constitute a collusion attack resulting in the deliberate corruption of a vote. In defining our response strategy therefore, we must take account of the nature and patterns of errors. This is reminiscent of the challenge faced by intrusion detection in general.

6.1 Ballot form errors

Voters are urged to confirm that their ballot sheets appear correctly in the input column of the web site. If the voter fails to find a copy of their receipt posted or finds that the posted receipt does not match their copy, then they should report this. The process of collecting and collating such reports needs to be handled carefully: the dependability of the scheme depends on appropriate response to such reports. It is important that voters are made aware of a recognised and dependable reporting authority, for example the returning officer.

It may be necessary to check voter claims of missing or corrupted receipts. This can be done by requesting voters to provide their receipts and confirming each report. If a particular booth is confirmed to have lost or corrupted a number of ballot receipts, it should be assumed to be either defective or malicious and taken out of service. It may be necessary to rerun votes cast at that booth.

6.2 Teller errors

The auditor performs checks on a random sample of links on the WBB for each teller to establish whether or not the decryptions have been correctly performed. All resulting errors need to be collated and analysed.

Requirements can be set on the tellers to deploy reasonable quality computing systems and cheap fault-tolerance to achieve very low probability of accidental corruption of ballots. The purpose here is not to avoid undetected errors at the system level, but to make even minor disruption of elections by accidental faults very unlikely, and thus: protect society's trust in the election system; prevent non-completion failures; and ensure that only massive effort by attackers can disrupt an election, and that such attacks cannot be disguised as technical "glitches". This improves diagnosis (discriminating malicious from accidental faults) and thus also improves society's confidence.

6.3 Secrecy failures

The notion of a detector for secrecy errors raises interesting issues. The main question is: what would such a detector detect? There are clearly plenty of blatant errors that could be detected. If say a teller applied the null permutation then this would be picked up at audit. On the other hand there appear to be a large class of secrecy failures that would not manifest themselves in the target system, especially when we note that there are restrictions on where we place our monitors.

Consider the following: we have just two tellers and they are in collusion: the first applies a "random" permutation and the second applies the inverse. This is clearly a violation of secrecy, or at least puts the system in a hazardous state w.r.t. secrecy.

It could be argued that if nobody but the tellers knows that this has occurred then no failure of secrecy has occurred. There is always the danger that this will leak out at some time to some group later. Also, the fact that such a fault could occur with our having no way to detect it is itself worrying and has the potential to undermine confidence. The fact that we may not be able to detect such hazardous states also means that it is hard to demonstrate that such a state has not occurred. Absence of an alarm does not imply absence of error state.

We should note that we could in principle audit all links and detect this attack, but this would then immediately violate the secrecy property we are trying to preserve.

7 Trust and Reputation

One of the primary assets of an election system is its reputation. Indeed, the reputation of the election system is an important factor in the mandate of any person elected by it. An attack on the reputation of an election system could be as damaging as an attack on the system itself.

One way of thinking about the reputation of an election system is in terms of the confidence that the voters place in it. We can be more precise, and say that voters have requirements of election systems expressed as dependability claims ("its accuracy is no worse than ... , its secrecy is no worse than ..."). One can never be sure that such claims are true, but on the basis of the available evidence one will have a certain *confidence* in their truth. We can expect that if confidence were to fall below a certain threshold, there could be serious social implications.

These ideas are close to recent formal approaches to confidence in reliability and safety claims for systems [2]. There, an attempt is made to model confidence as a (possibly subjective) probability that a dependability claim is true, based upon the evidence and reasoning of a dependability "case". The novelty in a voting system is that *intentional faults*, as well as accidental ones, can contribute to a reduction of confidence in a dependability claim.

Just as worrying as the scenario of public opinion mistrusting a dependable system is the scenario of a populace trusting an undependable system, leading to elections being stolen through vote tampering or vote buying. Efforts to avoid the first danger might well increase the likelihood of second. An adversary could even pursue strategies like minor sabotage of a few elections, causing many minor error reports which governments will be compelled to play down, so reducing the public's sensitivity to reports of problems, and reducing the probability of appropriate reactions to the real, all-out attack when eventually launched.

A comprehensive threat profile should consider attacks against reputation. This highlights again the importance of integrating the considerations from the social and human sciences, that should inform the dependability requirements for the election system, with the technical considerations from reliability and security engineering (including human factors and organisational issues) that determine the feasibility of satisfying the requirements via the various possible designs of election systems.

As an example of a possible attack against the reputation of the encrypted ballot receipt, consider the case where an attacker wishes to coerce people into voting in a particular way. He could (falsely) claim that he is able to decrypt the partial receipts retained by the voters, and insist that they are surrendered to him. Voters would then have to choose between believing the experts that told them the receipt was undecipherable, or believing the coercer.

Defences against such an attack might include things like specific public information measures to build public trust. These defences would necessarily need to be psychologically and sociologically informed, and indeed building a comprehensive threat profile would also require interdisciplinary thinking.

The reputation of a voting system could also be damaged without a malicious "attack", e.g. by even minor accidental errors being detected and requiring the telling procedure to be rolled back: the transparency of a system may work against the system on a societal level.

Attacks on the reputation of a system are not new, nor is defending against them. What makes the reputation of an electronic voting system interesting is the criticality of conveying the supporting arguments to such a large number of people. Compelling dependability arguments are usually designed to convince experts; here the general public must be convinced.

8 Conclusion

We have discussed at a high level the dependability requirements of an election system, which includes computer and communication technology as well as the people and organisations controlling it.

With its large scale and tight interconnection of technical and human components, an election system exemplifies the kind of systems that will become increasingly common. It is comparable in scale to some telecommunication networks or corporate IT systems, but is more similar to an anti-missile defence system in being an on-demand system with very stringent requirements on the single demand. It differs from all these other examples in having limits on its error detection capabilities imposed by a secrecy requirement, rather than just by physical constraints or costs; and its usefulness depends more heavily than with many other systems on preserving its reputation with the public, which is vulnerable to many kinds of attack.

The solution we have used as reference in our discussion relies on a combination of provable properties of cryptographic algorithms, and extensive error detection methods which in turn depend on both technical and social mechanisms for their functioning.

Our discussion has emphasised three aspects: requirement specification, with the need to translate society's informal requirements and to consider how threat profiles change compared to those affecting non-electronic elections; design issues for the fault-tolerant system, including the need to complement the basic cryptography-based ideas with explicit methods for assurance of the detection mechanisms and explicit recovery mechanisms, subsystem-level dependability requirements and the concern for denial-of-service attacks; and the need to integrate technical considerations with psychological and social ones that determine the threat profile, the voters' reactions and the effectiveness of the socio-technical mechanisms for error detection and recovery.

Acknowledgments This work was partly funded by the UK Engineering and Physical Sciences Research Council (EPSRC) through DIRC, the Dependability Interdisciplinary Research Collaboration. We wish to thank our DIRC colleagues at Newcastle and City Universities for many stimulating discussions on this topic.

References

[1] B. R. A. Avizienis, J.-C. Laprie and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable And Secure Computing*, 1(1):11–33, 2004.

[2] R. E. Bloomfield and B. Littlewood. Multi-legged arguments: the impact of diversity upon confidence in dependability arguments. In *International Conference on Dependable Systems and Networks (DSN)*, 2003.

[3] D. Chaum. Secret-Ballot Receipts and Transparent Integrity: Better and less-costly electronic voting at polling places. http://www.vreceipt.com/article.pdf.

[4] D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, Feb 1981.

[5] D. Chaum, P. Ryan, and S. Schneider. A practical, voter-verifiable election scheme. Technical Report CS-TR-880, University of Newcastle upon Tyne, 2004. http://www.cs.ncl.ac.uk/research/pubs/trs/papers/880.pdf.

[6] S. Schneider and A. Sidiropoulos. CSP and Anonymity. In *ESORICS*, volume 1146 of *LNCS*, 1996.

[7] B. Schöenmakers. A simple publically verifable secret sharing scheme and its application to electronic voting. In *Advances in Cryptology - CRYPTO'99*, volume 1666 of *LNCS*, pages 148–164, 1999.