

A robust new e-voting security model using anonymous public key certificates

Pre-print version of a paper submitted to IEEE Security & Privacy Magazine October 2007

Stephen Wilson
Lockstep Technologies Pty Ltd
11 Minnesota Ave Five Dock NSW 2046 Australia
swilson@lockstep.com.au

Abstract

Most electronic voting solutions have so far been complex and correspondingly difficult for regulators to validate. The full scale of the problem was revealed by independent testing in California and the subsequent high profile de-certification of several voting machines in that state in August 2007.

We propose a robust new security model based on public key technology and smartcards. Highly tamper resistant digital signatures and public key certificates protect both the ballots and individual voters' electoral enrolment. The solution can be deployed on a variety of modern smartcards with built in cryptographic processors, the likes of which are widespread in Asia, Europe and the US.

Each ballot cast would be unique and anonymous, unable to be replayed, nor modified. Each voter could only vote once. The security model, based on mature public key infrastructure standards, is simple. It is therefore inexpensive to implement yet straightforward to independently validate and certify.

Introduction and background

World wide experience of electronic voting to date has almost universally raised concerns about the quality and security of the underlying technologies and information systems. At best, e-voting systems have been criticised as lacking transparency. For instance, the Open Rights Group in its critical review of recent

trials of e-voting in the United Kingdom¹ commented that "E-voting is a 'black box system', where the mechanisms for recording and tabulating the vote are hidden from the voter. This makes public scrutiny impossible, and leaves statutory elections open to error and fraud" —[1]. At its worst, e-voting has been described as a "fiasco" for what some argue is a demonstrably disappointing standard of software engineering [2].

Following Britain's local government internet voting pilots of May 2007, the United Kingdom Electoral Commission concluded that "the level of risk placed on the availability and integrity of the electoral process was unacceptable. There are clearly wider issues associated with the underlying security and transparency of these e-voting solutions ... which need to be addressed" [3]. The commission went on to strongly recommend a central process for testing and approving e-voting solutions.

One of the first places to attempt such a process was the state of California. After mixed experiences with commercial off-the-shelf e-voting solutions, the Secretary of State there initiated a "top-to-bottom" review of no fewer than four products, culminating in the high profile and unprecedented de-certification of all of them, in August 2007 [4].

Despite the challenges and the recent apparent setbacks, we should still strive for secure electronic voting and, ultimately, trustworthy Internet based e-voting.

¹ For more background on the United Kingdom trials, refer to the UK Electoral Commission website www.electoralcommission.org.uk.

The important potential benefits include:

- improved voter turnout
- better availability and efficiency for absentee voting²
- reduced cost by avoiding the need for special voting equipment, which needs to be archived and/or maintained between elections.

While the focus of this paper is political elections, we should also remember that the medium of the Internet is ideal for a range of other polling and survey activities, such as opinion polling, deliberative polling, citizen initiated referenda, and company board elections. The demand for Internet based election and polling solutions is set to grow strongly.

Emerging standards for e-voting

While many critiques have been recently published (see e.g. [1], [2], [3] and [5]), perhaps the most elaborate and comprehensive attempt to standardise the requirements for e-voting has been that of the US National Institute of Standards and Technology's *Voluntary Voting System Guidelines* (VVSG) [6]. With respect to the security model of e-voting, there are two particularly significant requirements set down by NIST.

The first such requirement is for *Software Independence*, meaning that "an undetected error or fault in the voting system's software is not capable of causing an undetectable change in election results". NIST requires that all voting systems must be "software independent" in order to conform to the VVSG. A crucial feature of our anonymous certificate based e-voting solution is that it is decoupled from the voter's PC platform software, and is purely reliant on the compact secure firmware of a smartcard, which is far more amenable to independent verification.

The reality of intrinsically insecure e-voting software – or to put it more optimistically,

² Providing defence force personnel with the capacity to vote when on duty overseas has been a special policy goal of Australia and the United States, amongst other nations, in recent years.

the reality that software can probably not be proven to be secure; see *The nature of the software engineering challenge* below – led NIST and other analysts to further mandate the feature of *Voter Verifiability*. The NIST guidelines require that all voting systems include "a vote-capture device that uses independent voter-verifiable records (IVVR). IVVR can be audited independently of the voting system software *but do not necessarily have to be paper-based*" (emphasis added) [6]. The design we propose here provides for a redundant, tamper-resistant *soft copy* of one's ballot to be retained privately on a smartcard, from where it can be retrieved and checked at any time.

The nature of the software engineering challenge

So why has developing robust e-voting systems been such a struggle? From first principles, we really should expect difficulty when marrying mission and security critical applications to commercial operating system platforms. Complex fat client software is always hard to test, and fundamentally may be impossible to fully verify. Software quality professionals are familiar with the tenet that "Finding all errors in a large system is generally held to be impossible ... or else highly demanding and extremely expensive" [5].

It becomes especially prohibitive to manually inspect application code when its design makes it dependent on operating system code for its security functions; this is the case with almost all commercial software. Not only are many thousands, even millions of lines of code involved; it is not unusual for operating system vendors to keep details of their own software secret, in the interests of intellectual property protection. Such restrictions can be tolerated in most business applications, but with e-voting the social stakes are enormously greater.

Cryptographer and security expert Ron Rivest has written specifically on the e-voting challenge:

There is a fundamental problem we must face when trying to design remote electronic voting systems: the 'secure platform problem.' Cryptography is not the problem. ... The problem is interfacing the voter to the cryptography. Almost all proposed cryptographic voting protocols assume that a voter ... has a secure computing platform that will faithfully execute her portion of the protocol [7].

We see the same fundamental 'interface' problem in all types of secure transaction system. Recent US Government standards for strong authentication over the Internet of high risk transactions recognise the inherent difficulty of protecting against attack when cryptographic keys are not protected in some form of hardware [8]. The National Institute of Standards and Technology (NIST) has determined that to withstand "man-in-the-middle" attack (in which a spoof website might be put up in order to corrupt an online election) the only practical solutions entail hardware security devices and public key cryptography [9].

Sidebar: What is public key cryptography?

Public key (aka "asymmetric") cryptography is the technology that underpins most Internet security today. Discovered by pure mathematicians in the 1970s (and by British spy agency researchers some years before that³) public key cryptography allows all-important security keys to be distributed across large groups of real world users with relative ease.

³ One of the first and still most important public key cryptographic algorithms, "RSA", was patented (US 4405829) in the late 1970s by MIT engineers Ron Rivest, Adi Shamir and Leonard Adelman, whose initials give the algorithm its name. However, it was revealed much later that in 1973 British intelligence agency mathematicians had discovered essentially the same principles, which they called "non secret encryption", but kept them secret for national security reasons.

Conventional "symmetric" cryptography uses the same key to encrypt and decrypt messages. While encryption is usually thought of as the means to keep a message confidential from prying eyes, it is also used to check (that is, "authenticate") that a message has truly come from the expected sender. If users Alice and Bob are using a pre-arranged key to encrypt and decrypt, and if they trust that no one else has an illicit copy of their shared key, then any act of successful decryption proves that a message had to come from the other holder of the shared key.

The great problem with symmetric cryptography is key distribution: if Alice and Bob are far removed, how can they arrange to reliably share their key? In public key cryptography, the problem is solved by using pairs of keys operating in a special asymmetric algorithm: if one key encrypts a message then *only the other matching key can decrypt it*. Furthermore, knowledge of one key in such a pair tells us nothing about the other key.

In a public key cryptosystem, each user has at least one key pair. One key in each pair is made *public* so it is available to all other users, while the matching so-called *private* key is retained by its owner and kept secret. To send a confidential message to Bob, Alice encrypts it using a copy of his public key. The only way to decrypt that message is through Bob's matching private key.

Authenticating messages uses an inverse process, and a different public-private key pair. If Alice wishes to prove herself to be the originator of a message – that is, *digitally sign* the message – she will encrypt it with her *private* key. Anyone in the system who can sensibly decrypt that message with a copy of Alice's *public* key can thus be sure it came from her.⁴

To make public key cryptosystems useful in practice, we also require a trusted means for distributing faithful copies of public

⁴ In order to save processing time and bandwidth, practical asymmetric digital signatures don't involve the encryption of an entire message but rather only a compressed digest of it.

keys and associating them with their rightful users. A public key infrastructure or "PKI" is a managed system of business processes and rules for managing these associations. Each holder of a public key (whether it is used for confidentiality or authentication) is issued a *public key certificate* by a trusted "certification authority" (CA) which in effect notarises a copy of the key and links it to the holder.

The certificate can contain other information about the holder of the public key, vouched for and notarised by the certificate issuer. In the process of validating a digital signature against the public key held in the originator's certificate, any other information contained in the certificate is automatically and irrevocably bound to the transaction as well. Such information can include static personal data, membership numbers, affiliations and so on.

Conventional PKI names each user in their public key certificate(s). However, Lockstep's original research into electronic health record privacy has shown that public key certificates issued when the private key is held safe in a smartcard or similar secure device can have the name of the user removed and replaced by a pseudonym, to completely de-identify sensitive transactions. We discuss this research in the next section.

De-identification by anonymous certificate

In 2005, we published a detailed account of how anonymous public key certificates can be used to bind individual smartcard holders to de-identified electronic health records [10]. This technique can best be understood as creating a logical "triangle" that binds together (1) an individual, (2) a smartcard that has been issued to them, and (3) a public key certificate issued to the smartcard that conveys some important attribute (in the case of an electronic health record, that individual's unique health identifier). Importantly, the issuer of the smartcard and the issuer of the certificate need not be the same entity.

The first objective of the de-identification scheme was to ensure that when an

electronic health record is created, all users of that record can be assured that it pertains to one particular individual, without revealing who that person was. It is critical to healthcare information management that electronic health identities are resistant to tampering and forgery, and that they cannot be stolen or "replayed". We now consider that these same attributes are equally important in e-voting.

Let us now review how the de-identification solution works in its original context. Referring to Figure One, consider an individual patient named Smith to whom a Health Department has issued a Unique Health Identifier (UHI). If the UHI were to be carried around in an ordinary memory device and copied into transactions like regular data, then it would have no "pedigree"; that is, once the identifier is presented by the cardholder in a transaction, it looks like any other number and is just as vulnerable to attack.

To better safeguard a UHI using a smartcard, the identifier can be sealed into a digital certificate, as follows:

1. generate a fresh private-public key pair inside patient Smith's smartcard
2. export a copy of the public key
3. create a certificate around the public key, including as an attribute the UHI
4. have the certificate signed by (or on behalf of) the Health Department.

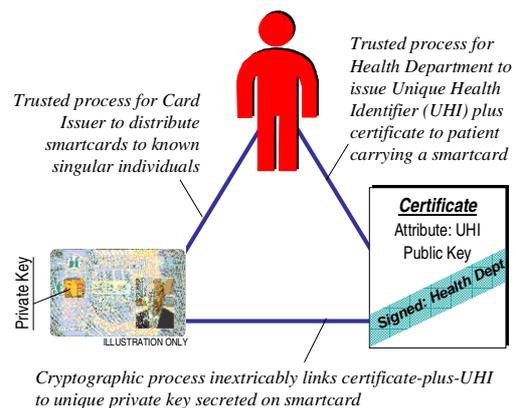


Figure 1: Logically "triangulating" a smartcard, a patient and a unique health identifier

The result is a logical triangle that inextricably binds cardholder Smith to her UHI and to a specific smartcard. The

certificate signed by the Health Department attests to Smith's ownership of both the UHI and a particular key pair unique to her smartcard. Private keys generated inside a smartcard are retained internally, and never divulged to outsiders. It is impossible to copy the private key to another card, so the logical triangular relationship cannot be cloned, reproduced or counterfeited.

Using anonymous certificates in e-voting

We now propose extending the concept of anonymous certificates to secure e-voting,

by arranging for a voter's de-identified electoral credentials to be notarised in a certificate associated with a smartcard, and using that smartcard to sign a ballot. As a result, individual ballots and the poll as a whole would be blinded with regard to the voter, but across the system we can be assured that each ballot is genuine and that no one has voted more than once.

Figure Two shows the major elements and processes of the e-voting solution, which are explained below.

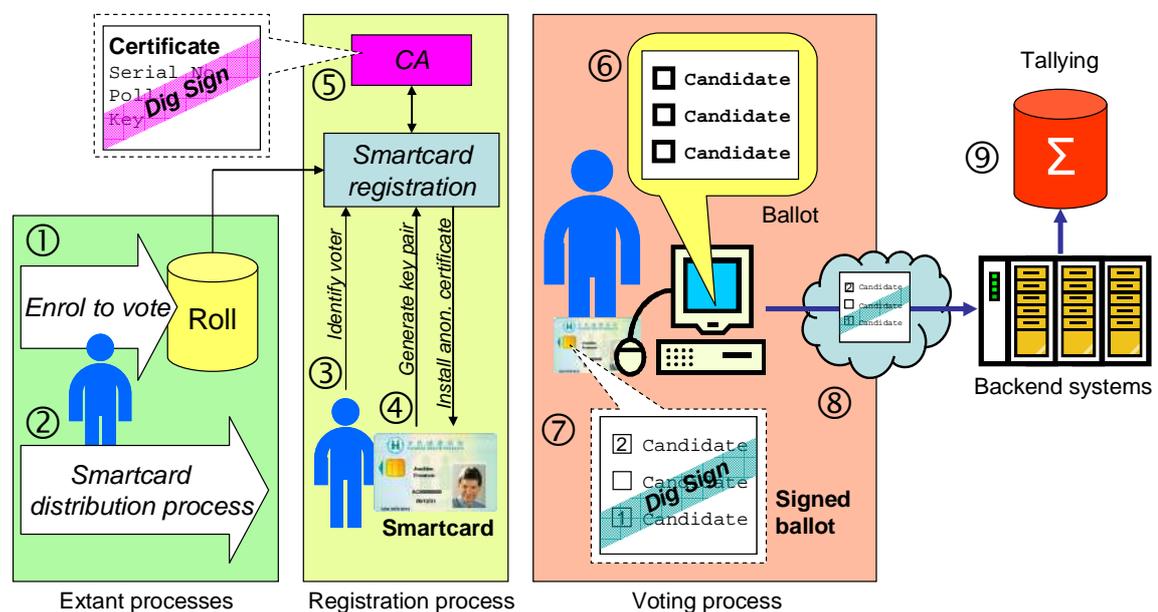


Figure 2: Components and workflows of the proposed e-voting solution

There are three major elements of the solution. Firstly, a **smartcard** is used by each voter to secure their cast ballot. The smartcard⁵ carries a secure voter registration code, encapsulated in an anonymous certificate in like fashion to the de-identified electronic health record discussed above, and an associated private

key which is used to create a unique digital signature on the ballot.

Secondly, an anonymous public key **certificate** is issued to each registered voter, through a separate process that associates a smartcard with the voter's electoral enrolment. The certificate contains:

- a public key matched to a private key secreted in the smartcard, where that private key is used to create and verify a signature on the ballot,
- an indication of which election the certificate was issued for, and
- a serial number or other unique code that proves the uniqueness of the certificate.

⁵ Note that the solution can be implemented using several alternatives to smartcards. As long as they have the requisite public key capability, the solution can use wireless PKI enabled cell phones, USB "crypto keys", hardware security modules and so on.

The serial number is not ordinarily linked anywhere in the system to the identity of the voter. The certificate is signed by the relevant electoral authority, acting through a Certification Authority (CA), and contains nothing that identifies the voter. We will explain a little later that a 'master file' may be retained if needed under local voting rules that records the certificate serial numbers against voter identities.

Thirdly, a **digitally signed ballot** is created using the smartcard and the public key certificate, interacting with standard PKI-enabled electronic forms software, using a pre-arranged formal structure such as XML for instance, with which to compose ballots to enhance standardisation and data processing.

There are five important work flows or business processes in this proposal. Three of them are a conventional part of any election system, but two are novel and underpin the new security model. The process of casting a vote need not differ much from any other Internet polling system today, except for the inclusion of a smartcard in the work flow. Referring to the numbered steps in the diagram:

1. Enrol to vote: Step ① shows the voter following whatever existing procedures and government policies are used to create and maintain a trusted electoral roll.

2. Distribute smartcards: Step ② indicates an existing procedure for issuing suitable smartcards to individuals, including the business processes that deal with lost and stolen cards, card renewals and so on. Our proposal can be deployed into general purpose smartcards already distributed for banking, government services, identification and so on (see below), or alternatively, the e-voting solution could make use of dedicated cards.

3. Register a smartcard to carry voter entitlement: Once individuals have smartcards, they will need to register their particular card to carry entitlements to vote in a given poll. This registration step can be carried out remotely in many cases – if there are remote access control arrangements in place that can be used to verify the

card holder – or it can be carried out face-to-face at an electoral office or similar designated outlet. In detail, the technical steps involved are as follows:

- At step ③ present the smartcard in a reader, either remotely or in person, and answer an identity challenge to prove ownership of the card as well as the good standing of their electoral enrolment (as established already in step ①)
- At step ④ generate a public-private key pair in the smartcard chip (a seamless embedded process that goes un-noticed by the cardholder)
- At step ⑤ generate an anonymous voting certificate, and have it signed by or on behalf of the electoral authority (also seamless). Each certificate will contain a serial number or some other unique code allowing repeat votes to be detected when the signed ballots are later collated for counting.

When implementing the proposed security model in a real voting system, it is possible between steps ③ and ⑤ to retain a 'master file' that links certificate serial numbers to the identities of voters. In many electoral systems, an absolute guarantee of voter anonymity is desirable, and the natural implementation of our proposed solution would have the master file securely destroyed at this point. Yet in some jurisdictions – most notably the United Kingdom – it is a legal requirement that every voter's ballot be available and identifiable after polling closes, in order to resolve disputes. Our solution allows for this possibility if the master file is archived rather than destroyed.

4. Cast votes: Polls will open according to the particular election rules and electronic ballots will be posted on a designated voting website for access by the public. Voters will log on to the site through a conventional security protocol such as the "secure sockets layer" (SSL) built into all web browsers and familiar to most users of Internet banking. The voter views the ballot at step ⑥ and indicates their choices

through a suitable user interface.⁶ When satisfied, they indicate their wish to submit the ballot, at which point the data with their selections is formatted and sent to the smartcard for signing. The ballot is signed using the unique private key matched to the anonymous voting certificate, at step ⑦. The signed ballot is then sent to the voting system backend at step ⑧. For convenience when validating signed ballots, the client side software will also send a copy of the voter's anonymous certificate.

Note that any number of additional copies of the signed ballot may be saved at this point, for disaster recovery in the event of a loss of data later during counting, audit, and/or independent voter verification. For voter verification, a logical place to cache a copy of a signed ballot is the smartcard itself. The integrity feature of the digital signature means that all cached copies of a signed ballot are equivalent; none can be tampered with to create repeat votes or to alter a vote, because the signature will only validate against the unique anonymous certificate. If two *different* signed ballots are found with signatures that validate against the *one* certificate then we can be sure that the smartcard was used twice.

5. Count votes: At step ⑨ after the poll closes, all cast votes are first checked for eligibility. In particular, all ballots' digital signatures must correspond one-to-one to voting certificates issued for this particular election. Repeat usage of a single smartcard can be readily detected by looking for multiple instances of the same certificate. All validated signed ballots are then counted, and a poll result determined.

⁶A host of accessibility and human factors engineering considerations lie outside the scope of this paper, where we have focused on the security model alone. Our new solution is compatible with any number of interface options as supported by today's standard PC and Internet platforms. Importantly, the ballot remains open to review and modification by the voter until they confirm their choices and instruct the smartcard to sign their ballot. That is, the proposal meets the new requirement of *Voter Editable Ballot Device* specified by the NIST VVSG [6].

Smartcard compatibility

Our proposed e-voting solution is compatible with most PKI-enabled smartcards, as in widespread use for health services, government services, personal security, and national identity. Examples include the US military Common Access Card (CAC), the new US government Personal Identity Verification (PIV) card [11], the identity cards of Belgium, Estonia, Hong Kong, Malaysia, Sweden and Thailand, the government PKI services card of Taiwan, and the new generation health cards of Australia (*Access Card*), France (*Sesam Vitale*) and Germany (*Gesundheitskarte*).

Technically, any multi-programmable smartcard with a public key cryptography co-processor and on-chip key generation will be suitable for the proposed solution. A key length of 1024 bits or higher is nowadays standard, and an exposed application programming interfaces (API) is required for the voting server to update the smartcard and to trigger the signing of the ballot. Increasingly this sort of basket of security functions is being normalized by standards such as "FIPS 201".⁷

Special properties of the solution

Before enumerating the specific security benefits of our proposal, let's review two special properties of the design.

Firstly it delivers a high fidelity form of voter verifiability of their ballots. At a minimum, an Independent Voter Verifiable Record (IVVR) may be produced as a tamper resistant soft copy of the digitally signed ballot. The copy may be cached in the voter's smartcard, where it would be available for review at any time. The smartcard's inherent PIN protection protects the privacy of the IVVR. Further, depending on implementation, there is the option of the voting system producing a printed copy of the ballot as cast, including a numeric rendition of the digital signature.

⁷ The Federal Information Processing Standards (FIPS) series is published by the National Institute of Standards and Technology; see <http://csrc.nist.gov/publications/fips>.

Secondly, in the context of the NIST requirements [6], our proposed solution is entirely independent of the software in the voter's host computer (typically a PC). The core security model is essentially implemented by firmware in the smartcard, wherein the voter's ballot is digitally signed using their dedicated and anonymously certified key pair. The only necessary cryptographic primitives are 'encrypt with internal private key' together with certificate lifecycle management commands such as 'generate key pair', 'fetch public key' and 'install public key certificate'. All of these are standard features of modern PKI-enabled smartcard firmware, and all are part of the target of evaluation for independent smartcard security accreditations.

Security benefits of the e-voting proposal

With respect to our central claim of a simple and readily validated security model, the e-voting solution has the following benefits:

- The design makes use of a restricted set of mature cryptographic primitives that come built-in with most standard PKI-enabled smartcards today.
- The use of anonymous but otherwise entirely standard public key certificates and primitives requires no additional smartcard applets and involves zero perturbation to the design of most PKI-enabled cards; thus a smartcard's prior security accreditation is unaffected by the e-voting solution.
- The e-voting solution security is software independent with respect to the voter's host PC, to simplify end-to-end accreditation of the system.
- There is a strict, mathematically robust one-to-one mapping of signed ballots to anonymous digital certificates, making repeat voting essentially impossible.
- Each ballot cast is cryptographically unique, preventing replay attack; a vote cannot be cast without an authorised smartcard properly loaded with the anonymous registration certificate, and ballots once cast cannot be intercepted during transmission or storage and used to synthesise fake votes.

- Voter registration cannot be counterfeited, thanks to the digital signature of the electoral authority on the anonymous certificates.
- Copies of ballots can be cached as Independent Voter Verifiable Records, to provide redundancy and protection against system outages, denial of service attack and so on, without compromising the integrity of the ballot; all cached ballot copies can be reconciled later without possibility of double counting.
- The poll is readily auditable as each voter's public key certificate, while anonymous, is unique.

General benefits

In addition to the security model, our proposal brings several other general benefits for the conduct of trustworthy elections and other types of polling:

- Ballots are strongly blinded; voters remain anonymous and cannot be linked to their ballots (unless the registration master file is archived as may be required in certain jurisdictions).
- The smartcard represents a physical "two factor" authentication method (that is, it represents something the user knows plus something the user possesses) and so provides added protection against voter identity theft.
- Many smartcards can be configured with additional "mutual authentication" of the remote server, protecting the voter web site against spoofing.
- The solution is independent of detailed smartcard and card reader design (across a relatively wide class of PKI-capable cards) and can therefore be implemented as a value-add on numerous current platforms, such as driver licences, government services cards, identity cards and banking cards.
- The use of a thin client host platform means the solution is location independent and highly accessible, as is necessary for effective absentee Internet voting solutions.
- The design is simple and therefore inexpensive to develop and deploy; no special host platform is required.

References

- [1]. *May 2007 Election Report Findings of the Open Rights Group Election Observation Mission in Scotland and England* Open Rights Group, June 2007;
www.openrightsgroup.org/wp-content/uploads/org_election_report.pdf.
- [2]. *The role of software engineering in electronic elections* Steven J. Murdoch, University of Cambridge, 13 July 2007
www.lightbluetouchpaper.org/2007/07/13/the-role-of-software-engineering-in-electronic-elections.
- [3]. *Electronic voting May 2007 electoral pilot schemes* UK Electoral Commission August 2007
www.electoralcommission.org.uk/files/dms/e-votingresearchsummary_6782-6321__E__N__S__W__.pdf.
- [4]. *Top-to-bottom review of certified voting machines* California Secretary of State August 2007
www.sos.ca.gov/elections/elections_vsr.htm.
- [5]. *On the notion of "software independence" in voting systems* Rivest R. R. & Wack J. P. July 2006
<http://vote.nist.gov/Sl-in-voting.pdf>.
- [6]. *Voluntary Voting System Guidelines: Recommendations to the Election Assistance Commission* National Institute of Standards and Technology, August 2007
<http://vote.nist.gov/VVSG-0807/Final-TGDC-VVSG-08312007.pdf>.
- [7]. *Electronic Voting* Rivest R. L. 2001;
www.vote.caltech.edu/Rivest-ElectronicVoting.pdf.
- [8]. *Electronic Authentication Guideline* National Institute of Standards and Technology Special Pub SP800-63 2006
http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf.
- [9]. *Electronic Authentication in the U.S. Federal Government* Burr W., National Institute of Standards & Technology, at the Asia PKI Forum, Tokyo 2005
http://asia-pkiforum.org/feb_tokyo/NIST_Burr.pdf.
- [10]. *A novel application of PKI smartcards to 11anonymise Health Identifiers*, Stephen Wilson, Lockstep Consulting, AusCERT 2005 Security Conference Refereed Stream, Gold Coast, Australia, 2005
www.isi.qut.edu.au/events/conferences/auscert2005/proceedings/wilson05novel.pdf.
- [11]. *Personal Identity Verification (PIV) of Federal Employees and Contractors* FIPS PUB 201, National Institute of Standards and Technology, 2005; available at <http://csrc.nist.gov/npivp>.