Comments from Scytl on the CORE report from the electronic voting solution used in 2010 Victorian Election

Version 1

Scytl Secure Electronic Voting

STRICTLY CONFIDENTIAL

Use only for evaluation purposes

The property of the cryptographic mechanisms and protocols described in this document is protected by their owners

Table of Contents

1. Introduction 4

2. Evaluating the security of Pnyx.DRE 5

3. Comments on the CORE report 7

3.1. Open source code or free access to the source code 7

3.1.1. Statement in the CORE Report 7

3.1.2. Scytl's response 9

3.2. VVPAT or a paper ballot proof 10

3.2.1. Statement in the CORE Report 10

3.2.2. Scytl's response 10

3.2.2.1. Verifiability in electronic voting 11

3.2.2.2. the Verifiability comments based on CORE's report statements 11

3.3. Random Number Generation 13

3.3.1. Statement in the CORE Report 13

3.3.2. Scytl's response 13

3.4. Trusted components 15

3.4.1. Statement in the CORE Report 15

3.4.2. Scytl's response 16

3.4.2.3. Trust on the voting kiosk 16

3.4.2.4. Trust on the back-office 18

3.4.2.5. Trust on the Voice Interface 18

3.5. Decryption of votes in the voting kiosk 18

3.5.1. Statement in the CORE Report 18

3.5.2. Scytl's response 19

4. Conclusion 19

5. References 21

1. Introduction

Scytl would like to thank the CORE (Computing Research & Education) review team for all the time and effort that they have put into the independent expert review of the electronic voting platform used on the 2010 Victorian Election [1]. This was one of the three different auditing/review efforts done on the voting software, all supervised the Victorian Electoral Commission (VEC).

The analysed voting platform allowed to cast electronic votes from two different channels: voting terminals and telephone voting. The major part of the voting platform was built around Scytl's Pnyx.DRE software. A previous version of this software was already used in 2006 Victorian Elections. Besides Pnyx.DRE, telephone voting (introduced by the VEC in this election) required to use an Interactive Voice Response (IVR) system provided by a third entity directly contracted by the VEC. VEC also directly provided all required hardware, including voting terminals, servers and telephone terminals.

As explained in this document, the review process of the voting system included significant interaction between the different involved actors and Scytl. We strongly think that auditing processes based on independent experts (especially on security applied on electronic voting) is very important and constitutes a must have in any electoral process (even if the process is based on traditional voting channels). We strongly believe that both transparency and vendor collaboration are critical in the review of any voting system.

This document contains Scytl's comments on the report prepared by the CORE review team's in regards to the voting system employed in 2010 Victorian Election [1]. This document will be referred as the reference number [1], CORE's report or simply "the report". In Section 2, we explain in general terms the considerations to take into account when auditing the security of Pnyx.DRE, which will be useful to understand the rest of the document. In Section 3, we address each of the specific findings in the report, providing Scytl's position in regards of each claim done in the report. Finally, Section 4 summarizes our overall conclusions of CORE's report and Section 5 includes all related references.

This document will also make references to the report generated by BMM Australia Pty Ltd, a well known testing lab contracted by the VEC to perform a source code and security audit on the voting system of the solution.

2. Evaluating the security of Pnyx.DRE

This section provides in first place an overall view of the aspects that must be considered when evaluating the security of voting systems and how do they apply to Pnyx.DRE based system.

Pnyx.DRE, as the rest of Scytl's voting technology, is a software voting solution that implements cryptography at application level, i.e. it implements specialized technological measures to ensure a trusted voting process. These specialized measures go beyond generic security measures such as firewalls, Virtual Private Networks and alike. These generic security measures are used by the voting platform to protect the different communication channels, but Pnyx.DRE provides a new and specialized security layer on top of them.

But as with every voting (and other mission-critical) system, technology alone is key but not enough to guarantee the security of a voting process. Technology must be complemented with other elements to ensure a trustable voting process:

- Procedures that clearly define who can do what on which component and under which circumstances. Procedures are usually applied by certain individuals, which should not be considered trustable per se.
- An auditing process that ensures the voting system (the technology, usually complemented by the appropriate procedures) behaves correctly and provides the required security levels for the election before it happens. A successful audit process usually provides a certification on the system being audited.
- An auditing process that allows to verify that the voting system behaves correctly during an election while it is being executed or after it has finished.

All voting systems rely on trusted personnel to ensure a transparent voting process. For instance, in paper-based elections voters must trust that election officials will correctly tally the ballots cast or that they do not impersonate voters by placing ballots on their behalf in the ballot box.

Pnyx.DRE implements a cryptographic protocol that reduces the number of trusted components typically associated to a voting system. In fact, Pnyx.DRE reduces the trusted components to just three (Pnyx.DRE is composed of many different components) located in two different places:

- The Voting Terminal, used by the voters to cast their ballots. This software displays the different voting options and encrypts the ballots. It is composed by two modules named Voting Module and Verification Module. This software is stored in a bootable CD-ROM (Live CD) whose fingerprint is digitally signed by the Electoral Board. Therefore, any modification of it cannot be hidden (read-only media) and will be detected by the Voting Terminal or an auditor (digital signature).

- The Mixing Service, responsible for decrypting the ballots.
- The Key Generation service, responsible of generating all the cryptographic keys and digital certificates used by Pnyx.DRE.

Both the Mixing Service and the Key Generation Service, operate in an air-gapped, physically protected computer supervised by the VEC. These services, as the rest of Pnyx.DRE components, have been reviewed and compiled by BMM Australia Pty Ltd (BMM), installed and reviewed by the VEC, and have been checked against the reviewed (certification pending) version of the voting system before being used in the actual election, to ensure that the deployed components are the same than the audited ones. BMM also chose dates not known in advance to VEC or Scytl to verify that the software run by these services during the election was the same they reviewed, in order to ensure that the software was not manipulated before or during the election.

Therefore, provided the correct procedures are employed, these three trusted components are easy to protect in front of manipulations.

In the same sense, Pnyx.DRE also requires only to trust a reduced number of individuals to ensure a fair voting process:

- Poll workers, who must behave properly when authenticating voters and employing the correct Voting Terminals at each Polling Station.
- Election Officials, who configure the voting system.
- Electoral Board members, who are responsible of digitally signing the Voting Terminal software at VEC's Office before the election, and of decrypting the ballots and generating the results once the polls are closed. The previous actions must always be performed in a collaborative way (a single member of the Electoral Board cannot act alone). They all operated from VEC's Office.
- Auditors of the election, may they be members of VEC, CORE or a trusted third party such as BMM.

Notice that the previous individuals are the same that must be trusted in any kind of election, may it use paper ballots or electronic voting systems. In fact, Pnyx.DRE does not require to trust the technicians that may be required to install the system.

3. Comments on the CORE report

After a careful review of the report, Scytl has found several statements and findings that we agree with, and others that we disagree with. For this reason, we will use this report to provide our opinion on these disagreements.

Regarding the agreements, we can mention the overall conclusions of the report: that the system uses reasonable cryptography for protecting the system against external attacks, or that the reviewers considered the audited system to be in general more robust from outside attacks than alternative remote channels such as postal voting.

Regarding the findings and statements we disagree with, they can mainly be classified into five different categories based on CORE's report [1]:

- Open Source Code
- Voter Verified Paper Audit Trail (VVPAT)
- PRNG (Pseudo Random Number Generator) seeding
- Trusted components
- Decryption of votes in the voting kiosk

There are also other findings and statements that are related to other components of the voting platform out of the scope of the Pnyx.DRE, such as the IVR system. We will only consider in this response IVR findings if they are related to the Pnyx.DRE interface with this system.

Below, we analyse each of the categories we have used to classify CORE's findings. Each category includes a snip of the original text of the CORE report, followed by Scytl's comments on the finding subject.

3.1. Open source code or free access to the source code

3.1.1. Statement in the CORE Report

The report mentions several times the need for making public available the whole source code of the voting system. More concretely:

<u>*Extracted from Section 4:*</u>

(…)

It is challenging to balance the legitimate wish of software vendors to protect their intellectual property with the demands of transparency and scrutiny necessary for a voting system. Everyone agrees that a very high degree of transparency should be required for electronic voting systems. The controversy is over exactly how much, and in particular over whether the source code for the system should always be available for public scrutiny. Many computer scientists believe this should be a requirement, though some would say the requirement may be relaxed if the system provides genuine verifiability such as a human-readable printout.

Recommendation : Requirements for openness should be part of the initial tender and contract. The more openness, the more assurance that the system behaves as expected.

For example, from my protocol-level information I could identify some issues concerning vote privacy, but there may be other issues that were not detectable without source code.

The Council of Europe's recently published "Guidelines on the transparency of e-elections" include the statement that "Access to documentation including minutes, certification, testing and audit reports as well as detailed system's documentation explaining in details the operation of the system, is essential for domestic and international observers." These recommendations are (perhaps deliberately) vague on whether the "details [of] the operation of the system" include source code or not, but clearly they imply a very high degree of openness, probably exceeding what was provided by the VEC and Scytl in this trial.

Leaving aside the controversy over whether it is essential to have public source code, it is certainly feasible to do so. The ACT has run electronic voting on an open-source system for many years (the EVACS system from Software Improvements [3]), and the Norwegian government recently instituted a project for Internet voting, for which open-source software was a requirement of the tender (and Scytl was the winning bidder [4]). Also, if the VEC decided to produce its own system rather than purchasing one from a private vendor, it would then be free to publicise it openly.

(…)

<u>*Extracted from section 5:*</u>

(…)

3. Without the source code, I was not able to establish whether someone with access to the kiosk after voting could discover which votes were cast in order…
4. Without the source code, I was not able to investigate the generation of randomness for cryptographic operations
(…)

3.1.2. *Scytl's response*

As stated in the report, the reviewers were reluctant to sign a Non-Disclosure Agreement (NDA) with Scytl in order to get access to the source code for its revision. As with previous customers, Scytl is not reluctant to third party review of its software source code provided it is regulated by a mechanism that protects its Intellectual Property. If reviewers did not have access to the source code was due their personal position of not signing this NDA with Scytl.

In any case, we respect the position of reviewers, that considered of relevant importance the fact of having a public review of the source code, and plead for more openness requirements in future tenders and contracts. In fact, it is feasible to have public source code for voting systems, and Scytl has done that in the past based on customer requirements and payment of the associated costs: a public review does not allow Scytl to control who has access to the Intellectual Property developed by the company and therefore, it allows competitors to have unrestricted access to our technology. For this reason, Scytl requests a different license (more expensive) in case the customer wants to publish the software.

In any case, Scytl does not believe that having a public source code review implies that the code is more secure. A public review does not guarantee that people with enough expertise will be reviewing the code, or that if someone finds a bug, this will be properly notified to the software provider to fix it. Therefore, we think that it is more important to have a formal code review, made by security experts (which can work under an NDA) that allows the certification of this source code, and ensures that any bugs which may be found in the review are properly reported and fixed before the certification.

In contrast, as in 2006 election, BMM signed in 2010 an NDA with Scytl and had full access to the source code of all Pnyx.DRE components. Once reviewed, BMM compiled the source code and generated the executable program files used in all the voting platform components. During such source code revision, CORE's auditors pointed out to BMM different places where to look for potential vulnerabilities, and therefore, although indirectly, CORE was able to check how was built any Pnyx.DRE component. We do not understand why the report includes references to a lack of capability to review certain parts of the system. In the same sense, CORE pointed out to the auditors in charge of a penetration testing, different places where to try to attack the system to ensure no security gaps. BMM considered CORE suggestions when auditing the source code.

3.2. VVPAT or a paper ballot proof

3.2.1. Statement in the CORE Report

Reviewers also mention in the report the requirement of having a paper proof as part of the voting process. More concretely:

Extracted from Section 3:

(…)

The 2010 electronic voting trial did extend to voters other than the visually impaired, but it did not include a human-readable printout. […] Subsequent models should provide a human-readable printout, so that voters who are able to read can verify that their vote was recorded as they intended. This applies to all computerised voting systems, whether the interface is by telephone or computer. Printing out a real ballot that could then be placed in a ballot box would be a simple way to achieve this.

(…)

Extract from Section 4:

(…)

Openness about the system is a vital part of transparency, but it provides only partial assurance because it is infeasible to guarantee that the system so carefully analysed is the one that actually runs on the computers on voting day. That is why a human-readable paper record remains crucial for providing evidence of the correct outcome.

(…)

3.2.2. Scytl's response

In order to better understand our comments about the usefulness of the VVPAT system as a voter verification method, we will provide a brief introduction about verification in electronic voting before exposing our arguments.

3.2.2.1. Verifiability in electronic voting

One of the major concerns of electronic voting is the difficulty for the voter to verify the correct register of her vote, since it is usually done in a digital way that cannot be directly observed. However, the introduction of cryptographic techniques or information technologies in electoral processes can be used to provide means to the voters to verify the correct processing of their votes, also known as individual verifiability. This individual verification is equivalent in a traditional election to the verification that the vote cast by the voter is properly stored inside the ballot box. Therefore, cryptographic measures can improve the level of voter verification and election auditing compared to conventional electoral processes.

Individual verification measures comprise several sub-protocols inside the voting process, such as:

- The use of tracking codes (voting receipts) with random identifiers tied to the votes, in order to ensure that the vote reaches the counting phase without being modified;
- The use of Pollsterless methods or Return Codes in order to ensure that the encoded vote represents the voter intent (special codes are calculated from the encoded vote and compared with values pre-assigned to the voting options in a Voting Card held by the voter);
- VVPAT (Voter Verifiable Paper Audit Trail) methods, where a print-out of the digital vote is generated in order to let the voter verify her choices before the vote is encoded and registered.

Therefore, VVPAT is not the only way to provide individual verifiability of a voting system.

Other measures can also be implemented in order to get auditors and independent observers to verify the integrity of the election outcome, such as the verification of the integrity of the Ballot Box where votes are registered (i.e. with the inspection of immutable logs), and the integrity of the counting process, where the votes are collected from the Ballot Box, decoded and counted. Combining the individual verifiability and facilitating auditors to verify the integrity of the ballot box, it is possible to achieve a similar audit level as a traditional election.

3.2.2.2. the Verifiability comments based on CORE's report statements

In the CORE report, the authors strongly recommend the use of a human-readable printout generated by the Voting Terminal in order to let the voter to verify that her vote has been properly cast (VVPAT).

As introduced above, paper print-out is one of the possible ways for providing individual voter verification, but not for voters with visual impairments (they are unable to verify their vote on their own). In addition, there are also some technical aspects that have to be deeply analysed before adopting this technology, such as the reliability of the required hardware (printers), the potential issues with lost printouts, or the extra costs just to mention a few.

In fact, the decision of printing a paper trail is not only a technical matter, and requires also to analyse the introduction of this methodology from the procedural, legal, accessibility and cost point of view. For instance:

- Shall the print out have the same layout as a standard ballot? If so, large paper-format support printers (e.g., A3 plotter-like printers with auto-cutter) should be connected to each voting terminal (it is not possible to use the small format factor thermal printers usually used) in the case of VEC.

- Can the voters manipulate the paper trail or shall it be protected by a transparent surface? The former requires to define a valid process on how to invalidate an electronic vote in case the voter finds a problem in the print-out.

Therefore, any introduction of paper trails should be preceded by a detailed analysis of all its implications, technical and non-technical.

In any case, despite the use or not of VVPAT in the system, Scytl thinks that voter verification and other auditing methods are crucial for proving the integrity of the election outcome. Human-readable printouts are a way to perform these verifications. However, it is recommended not to use them as the unique verification of the election integrity, since they rely on the fact that the voters are going to review the printout and notify any anomaly to the election officers: a malicious application trying to change a small quantity of votes will not be noticed using this verification method, as it is mentioned in [3]. Therefore, additional verification / auditing measures must be implemented in the voting platform.

Pnyx.DRE provides voter verification and additional auditing methods: voters can verify that their votes have been correctly cast and have been taken into account in the final count, by (1) using a confirmation screen in the voting kiosk (handled by an independent cryptographic module); and (2) by using a printed voting receipt, which provides them traceability with their vote till the counting process. Also, immutable logs are generated in order to verify the integrity of the Ballot Box, parallel registers are used to verify the consistency of the votes registered in the voting kiosks, and the counting process is implemented in an isolated, secured and audited environment in order to ensure the fairness of the result.

3.3. Random Number Generation

3.3.1. Statement in the CORE Report

The report also mentions the importance of using reliable random sources for protecting the privacy of the election:

Extracted from Section 5 – Privacy:

(…)

Without the source code, I was not able to investigate the generation of randomness for cryptographic operations. I was surprised by the auditor's assessment that although the key generation was weak, this was not a practical vulnerability. In general predictable keys can make it much easier for outside attackers to decrypt messages.

(…)

3.3.2. Scytl's response

First of all, as mentioned before, the fact that CORE report reviewers did not have access to the source code was based on a personal position of not signing an NDA. BMM did it and had access to it.

The audit performed by BMM found a potential vulnerability on Pnyx.DRE related to a possible weakness in a Random Number Generator (RNG) used to generate cryptographic keys (in the Key Generation Service, not in the Voting Terminals). CORE included this finding in its report but did not mention BMM's evaluation on the risk of this finding. In fact, despite BMM considers this finding a potential issue, they also state that the risk of having a real problem was negligible, as detailed by BMM in its report [2] (excerpt included below).

(…)

The risk here is that an attacker capable of smart card creation could guess the value of the millisecond timer within a range of a few million values and re-run the key generation many times until it produces a smart card that works correctly in the live system. Since the secret *keys are not independent, the attacker could theoretically then re-generate every key for the whole system.*

Although testing a few million keys is quick for a computer, the attack here requires that a smart card be produced for each attempt and tried on a computer with the live election loaded. Access to the back office computer or the data centre computers is not possible due to physical security arrangements. Access to other computers such as issuing point laptops is also supervised and physically controlled. The most practical attack approach would be to obtain access to an issuing point laptop or its install CD for the live election and use that software to evaluate smart cards. The attack would be limited by the rate at which smart cards could be written and tested in the software. Trying several million cards would take a long time, and this is just to find one key value. A faster attack could involve hacking into the issuing point software and running many keys through its validation function.

Although the RNG does not gather additional entropy between keys the RNG is used for various purposes between generating each key, for example to randomly fill space at the end of a buffer. It is not the case that the software generates each key on successive calls to the RNG. The attacker would also need some inside knowledge of the Scytl software and knowledge of the order of back office key creation to enable generation of many keys from the cracking of a single key.

Assuming a successful attack, the attacker now holds smart cards that can access issuing points and other computers. However access to the data centre and back office is prevented by physical security and access to voting kiosks is impossible during working hours. After working hours the attacker could start a kiosk and cast votes. This activity would be logged and detected immediately by the monitoring system and the votes would be invalidated.

As a result of considerations in the foregoing discussion, BMM believes that the risk to election integrity based on an RNG attack is negligible.
(…)

This weakness is related to the initialization of the RNG, which uses the millisecond timer in its default implementation. Therefore, if somebody can discern the moment when the keys are generated, the value of the keys could potentially be predicted. However, the RNG that had this issue is used only by the Key Generation Service (a machine completely isolated from any network), and it is used for additional tasks between the generation of each cryptographic key (in a typical election thousands of keys are generated). Therefore, the problem was isolated to this specific RNG, and due to that additional tasks, it is very hard to know the exact time of each key generation and then guess its value from this timing information.

Although the practical risk is negligible, we have committed to change this in the future new version of the voting software.

3.4. Trusted components

3.4.1. Statement in the CORE Report

The report also highlights the fact that some components must be trusted to guarantee the integrity of the Election. More concretely:

Extracted from Section 5 – Integrity and Correctness

(…)

In secure systems of any kind, it is better to avoid placing trust in any single part of the system. For example, our system for paper-based voting is carefully designed so that no single person is completely trusted. The presence of scrutineers observing the count, the public location of the ballot box, etc., all make it very difficult for a single person to violate another's privacy or to compromise the integrity of the vote. An important exception is when one person relies on another to write their vote for them because they are unable to do so alone - then the writer must be trusted to keep it secret, and sometimes trusted to write it correctly because the voter is unable to verify this. This situation is undesirable, and indeed removing it is one of the prime motivators for electronic voting. Nevertheless, the electronic voting system itself has three trusted components:

1. *The kiosk is trusted. It can misrecord a vote undetectably. This problem would be solved by making it produce a human-readable paper printout.*

2. *The back-office is trusted. It could break voter privacy as described above. The extent to which it can change the votes depends upon which additional checks are performed. In its basic version, it performs both the decryption and the test of whether the decryption was performed correctly. It would be better if it exported the proof so that an independent machine could check its correctness.*

3. *The Secure Voice Interface is trusted. It can undetectably modify a telephone vote that it relays. This problem would be solved by directing the telephone interface to a human-readable paper printout.*

The system should be redesigned so that no single component can compromise the integrity or privacy of the votes.
(…)

3.4.2. Scytl's response

The existence of trusted components in an election is not only specific in electronic voting. In any traditional election it is also required to trust some components to guarantee the election integrity (e.g., electoral roll). As in traditional elections, it is important to implement correct procedures, such as audit and separation of duties, to guarantee that trusted components are behaving properly.

As described above, Pnyx.DRE implements a cryptographic protocol that reduces the number of trusted components typically associated to a voting system:

- The Voting Terminal, used by the voters to cast their ballots: this software is stored in a bootable CD-ROM (Live CD) whose fingerprint is digitally signed by the Electoral Board. Therefore, any modification cannot be hidden (read-only media) and will be detected by the Voting Terminal or an auditor (digital signature).
- The Mixing Service.
- The Key Generation Service.

Both the Mixing Service and the Key Generation Service operate in an air-gapped, physically protected and supervised computer. These services, as the rest of Pnyx.DRE components, have been compiled by BMM after a source code audit, and installed and reviewed by the VEC. Therefore, provided the correct procedures are employed, these three trusted components are easy to protect from attacks.

3.4.2.3. Trust on the voting kiosk

Regarding the fact that the kiosk is trusted, Pnyx.DRE has some features that prevent the kiosk (specifically the Voting Terminal application) from cheating without being noticed. As it has been explained before, the Voting Terminal application is composed of two modules: the Voting Module and the Verification Module. The Voting Module is in charge of registering the voter preferences. Once the voter has finished her selections, the Verification Module shows in a separate screen the voting options. Once confirmed by the voter, the voting options are encrypted and stored by both the Voting Module and the Verification Module. This Verification Module can be:

- Designed to be in an independent piece of hardware, so that it is infeasible for an attacker controlling the Voting Terminal application to take control of the Verification Module. Therefore, if the Voting Module has manipulated the voter options, it can be detected through this independent Verification Module.
- An independent and compact piece of code that runs inside the Voting Terminals that can be easily audited to ensure that it behaves properly. Therefore, the votes stored on it can be considered a reliable representation of the voters' intentions.

In the specific configuration for this project, since both the Voting Module and the Verification Module run together in the same machine, it could be feasible for the Voting Module to 'simulate' the Verification Module when showing to the voter the confirmation screen, and make the voter believe that she has voted successfully. This is prevented by auditing the code –the live CD that is installed in the voting kiosk- and verifying during the election that it corresponds to the audited version, as it was done in VEC 2010 project.

The LiveCDs in the voting kiosks, jointly with other applications of the election platform (other components of Pnyx.DRE, like the Key Generation Service, or the Mixing Service) were reviewed by a team of auditors in order to verify their integrity. This is reflected in the BMM report [2]:

(…)
Scytl provided a build environment so that BMM could build the software and take checksums of the installable outputs. The final release was built at BMM with Scytl staff present.

BMM chose times not known in advance to VEC or the providers to verify the software loaded on various EAV computers.
This included:

a) *The two servers at the data centre;*

b) *Issuing point laptops and kiosks at the Doncaster and Northcote early voting centres; and*

c) *The failover server and the back office desktop computer at the VEC head office.*

It was found that the checksums of software loaded on these machines matched the checksums obtained from the final build.
In general, it is difficult to prove that software actually running on a computer is exactly and only the software seen on the disk. Each computer runs a multitude of tasks and drivers. The level of assurance here rests on the fact that all computers were configured in a controlled

environment from tested and verified media and each computer has been kept in a either a secure location or a supervised environment since then.
(…)

For future implementations, we would like to suggest having an independent hardware piece with a screen where Verification Module operations –including voter verification- are performed. Therefore, it would be harder for the Voting Module to try to supplant the Verification Module.

3.4.2.4. Trust on the back-office

Regarding the comment about the trusted back-office (which contains the Key Generation Service and the Mixing Service), we would like to comment that, in order to ensure that this single entity behaved correctly, its code was audited, it was executed in an isolated and controlled environment in order to prevent any external attack, and any auditor or observer could be present to verify how it was operated and worked during the electoral process.

3.4.2.5. Trust on the Voice Interface

About the Voice Interface, this is a mechanism that relies of an component, the IVR, which was provided by a third party directly contracted by the VEC, and therefore it is not part of Pnyx.DRE. In any case, this interface was mainly introduced for facilitating the voting process to visually impaired voters. Therefore, a paper printout will be difficult, if not impossible, to verify by these voters. It is important to note that telephone voting was also used in the 2007 Australian General Election by the Australia Electoral Commission.

3.5. Decryption of votes in the voting kiosk

3.5.1. Statement in the CORE Report

Finally, reviewers also mention some concerns regarding the encryption of the votes in the voting terminal:

Extracted from Section 5 – Privacy:
(…)
Without the source code, I was not able to establish whether someone with access to the kiosk after voting could discover which votes were cast in order. Because the votes were encrypted using symmetric-key cryptography it would have been difficult to erase these values, hence possible for someone to decrypt them after polling. This would not be a

problem if they were stored in a randomised order, but could compromise privacy if the vote order was not properly randomized.

(…)

3.5.2. Scytl's response

We believe that this statement is related to a misunderstanding on how the encryption in voting kiosks works.

In the voting kiosks, the votes are encrypted using digital envelopes, where a random session symmetric key is generated to encrypt the vote, and the public key of the election is used to encrypt this symmetric key. Therefore, at the end of the election the Electoral Board can reconstruct the election private key and use it to decrypt each symmetric digital envelope key to decrypt the votes. This solution is known as the best trade-off between public key cryptography and secret key cryptography (fast encryption with a simple key management system).

The symmetric keys used to encrypt the votes are ephemeral random numbers (generated on-the-fly, unique per vote) which are encrypted with the election public key to prevent their disclosure. These symmetric keys are never stored in the kiosk hard drive when generated. After they have been used to encrypt the vote, they are securely erased from memory. Therefore, nobody at the kiosk can in any case decrypt the votes nor violate the privacy of a voter.

Regarding the claim that CORE members did not have access to the source code, again, this was due the personal position of the reviewers that were reluctant to sign an NDA.

4. Conclusion

The use of electronic voting in 2010 Victorian elections was preceded by a thorough revision of the voting software being field by the VEC. Three different audit/revision processes were made before the election started: a cryptographic protocol revision by CORE, a penetration testing by an independent entity, and a functional and source code audit by BMM. The two last processes included some tests/revisions suggested by CORE. Also, during the election and once it was ended, BMM performed different verification tasks. This document is related to the statements made by CORE in its own report [1].

As explained above, we do not fully agree with some of the statements in the CORE report. They are related to public source code reviews, the advantages of a human-readable printout for voter verification, the potential weaknesses of a 'by default' random number generator seeding, and the trust in a single component of the voting platform.

Regarding public access to the source code, Scytl wants to clarify that it never banned any access to the code. It only requested, based on the software license contracted by the VEC and in accordance to its Intellectual Property protection practices, the signature of a Non Disclosure Agreement (NDA) by CORE. In fact, BMM signed an NDA with Scytl, had access to the source code, and provided accurate conclusions about any potential issue they found (i.e., the NDA did not prevent the disclosure of findings), as reported in [2].

There are also several other findings and statements that are related to small errors/misunderstandings on how the Pnyx.DRE is used, like the possibility of decrypting the votes in the voting kiosks without having the election public key, or are related to other components of the voting platform out of the scope of the Pnyx.DRE, like the Secure Voice Interface.

Despite our disagreement with some of the findings, we are generally pleased with the overall conclusions that can be derived from this review report. In particular, we are pleased with the overall conclusion of the report, that the system uses reasonable cryptography for protecting the system against external attacks.

We consider this review as a very challenging but useful exercise that can help close the existing gap between the vendor and academic communities. The review proves that the good-will collaboration of vendors and academics can significantly help states evaluate new voting systems.

Furthermore, this kind of review allows us to improve our software. For instance, we will implement a different seeding method of the RNG in the Key Generation module to eliminate this potential risk.

Finally, we consider that task of the CORE team should not finish after this election. We encourage them to collaborate with the VEC and other Australian Electoral Commissions to define a set of guidelines that could be used to compare electronic voting solutions in future elections.

5. References

[1] Report on the VEC-Scytl electronic voting system for the 2010 Victorian Election (Draft , 3rd Dec 2010), Dr. Vanessa Teague on behalf of CORE.

[2] Electronically assisted voting audit 2010. Draft November 26th 2010, BMM Australia Pty Ltd.

[3] Testimony on voter verification: presentation to senate committee on rules and administration. Ted Selker, MIT. June 2005. Caltech/MIT Voting Technology Project.

[4] E-vote 2011: Election system with solution for electronic voting, Norway 2011. Final tender.