1. Introduction

The voting process is a widespread democratic procedure in use throughout the world. Today the well known paper voting system is still the most commonly used, although several countries are in the advanced stages of conversion to an electronic voting system. The benefits of the electronic voting system are:

- Capability of diverse control of the electoral process
- Fast and accurate electronic vote tallying without human intervention
- Casting and tallying audibility

The expected difficulties during the process of the incorporation of the electronic voting system are:

- Explaining the security and the reliability of the system to the public
- Assisting the public to become familiar with voting in a different way
- The high cost of implementation and assimilation of a new voting system

The market offers a variety of electronic models, with various levels of security and of reliability.

The purpose of this paper is to analyze the Wombat Model, which was designed and developed in Israel with the cooperation of the Interdisciplinary Center in Herzliya and Tel Aviv University. I chose to analyze this specific model because its designers proclaimed a high standard of security, transparency and reliability during the casting process and the tallying process.

This model offers a dual ballot structure, electronic voting combined with the old fashioned way of paper ballot voting. The integration of the two processes may contribute to the understanding of the voting process and strengthen the system’s credibility in the eye of the voter. On the other hand, it can create a problem of synchronization of the two parallel processes. It is important to mention that the analysis of this model is not a purely academic endeavor, but rather a practical analysis in terms of the real world.

The purpose of this analysis is to offer an objective point of view to anyone considering implementing this system. Section two of this paper focuses on understanding the model, the voting flow chart and its cryptographic/security components, while examining the risks and threats inherent in any voting system. Security components attempt to handle known specific threats. After understanding this model and its components, section three uncovers the new threats that arise from the integration of these components. In section three, all the parties that participate in the election process are examined, under the assumption that no party is trustworthy.
2. System Overview

This section reviews the structure of the Wombat System and the designers' vision for achieving security and privacy. The system allows the voters to cast their votes and, at the end of the procedure, counts them, while keeping the results secure from as many threats as possible. Maintaining the voter's privacy means that no information regarding any vote is revealed to anyone, or even to the system itself. The voter has the option to audit the voting machine with a simple audit ballot which will be explained in detail below. Any vote can be verified if it has indeed been counted, without compromising the voter's privacy. The implementation should be by open source, which means that the source code of the system is available to the public. The open source concept is part of the system's transparency, ensuring that even the developers themselves cannot cheat the election results.

2.1. The Voting Process

The flow chart below shows each step in the voting process. The steps are numbered and explanations of each one are listed beneath the flow chart.
1. **Entering the Booth**: The identification system is independent and not included in the Wombat Model. After identification, the voter enters the booth where there is a machine that includes a touch screen interface, a printer and a computer (without any storage device), which runs a live Linux operating system from a CD ROM.

2. **Cipher Random Part is Digitally Signed**: The voter makes his choice in the booth on the touch screen, and receives a printed ballot with his vote on it. The vote is encrypted by ElGamal encryption. For cryptographic reasons, the encryption uses randomness, which is different from one encryption to another. In the Wombat System, the randomness for the encryptions is taken from two Smart Cards, which are installed in the booth computer. Each smart card produces part of the randomness, which is digitally signed by its producer smart card. Each Smart Card is from a different manufacturer.

3. **Ballot is Digitally Signed**: The encrypted vote is digitally signed by either one of the Smart Cards which was used to produce the randomness.

4. **Printing the Ballot**: The data and the encryption are in the barcode. The encrypted vote is printed and then the voter has a choice: he can use this ballot as a voting ballot or as an audit ballot. If the voter wants to use it as a voting ballot, he receives the ballot as is. If the voter chooses to use it as an audit ballot, the printer prints the randomness on the ballot. The encrypted vote can be decrypted by the randomness, in the audit station, and will confirm that the vote is indeed correct. If the voter chooses to use this ballot as an audit ballot, it can only be used for that purpose and not for voting.

5. **Ballot Structure**: One part of the ballot is printed with the plaintext and can be folded and sealed. The other part has the following data as a barcode:
SN_1 – Serial Number from Smart Card 1.
SN_2 – Serial Number from Smart Card 2.
$g^{r_1}, \text{sign}_{SC1}(g^{r_1}\mid SN1)$ – Cipher random part and its digital signature from Smart Card 1.
$g^{r_2}, \text{sign}_{SC2}(g^{r_2}\mid SN2)$ – Cipher random part and its digital signature from Smart Card 2.
E($v, r$) – The Encrypted vote.
Sign_SC(E($v, r$)\mid SN1\mid SN2) – The digital signature of the encrypted vote by one of the smart cards.
Hash which represents this ballot in the tallying participation verification. This Hash can be read by the voter.
r_1+r_2 – The randomness of the vote encryption (only if this is an audit ballot)

6. **Casting the Vote**: The plaintext part of the ballot is folded, sealed and torn off. Then it is placed in the ballot box. The barcode is scanned and sent to the poll. Then, it is stamped at the poll and becomes the voter's receipt for his choice.

7. **Scanning The Barcode**: The ballot's barcode is scanned to the poll computer. The poll computer executes the following verifications:
- The digital signature of the Cipher random part from Smart Card 1.
- The digital signature of the Cipher random part from Smart Card 2.
- The digital signature of the encrypted vote from one of the Smart Cards.
- The entries of the serial numbers which should be in chronological order according to the order of the voters.

8. **Storing and Publishing the Votes**: If the vote passes the verification, it is stored in the database. All the votes in the database are published on the web bulletin board. The voters can access the bulletin board in order to verify that their votes have been counted.

9. **Mixing The Votes**: The encrypted votes are mixed in every candidate's server and every supervisory authority's server, in random order, by a Mixnet algorithm. After the Mixnet, every encrypted vote is re-encrypted and the vote's permutation is randomized.
10. **Tallying The Votes**: All the candidates and the supervisory authorities have a part of the private key prepared by threshold encryption algorithm. Only 70% of the candidates are needed for decrypting the encryptions and tallying them.

2.2. **Generic Attacks on Voting Systems**

There are potential attacks on every electronic voting system. These fall into three categories:

- **Integrity**
- **Privacy**
- **Denial of service**

First and most important is **Integrity**. Attacks in this category can change the result of the election and make the voting process meaningless. The attacks in this category:

- **Registration Frauds**: A voter is identified but casts his vote in another person's name. In this case, one person can cast more than one vote using the names of other voters.
- **Repeating**: A voter casts his vote more than once.
- **Ballot Stuffing**: "Stuffing" votes without voters, or entering more votes into the system, without the casting process. In systems in which people can vote at one poll only, at the end of Election Day the poll workers can ascertain who did not come to vote. They can then "stuff" the votes of all those who did not show.
- **Altering Ballots**: The voter knows the candidate he voted for, but the ballot has been altered and does not reflect his vote
- **Altering Returns**: A ballot selection is entered into the voting system, but it is counted differently.
- **False Count and False Returns**: Falsifying the tallying while counting the votes, or tallying honestly and returning a false result.
- **Chain Voting**: The attacker wants a group of voters to cast according to his choice and all the voters in the group agree. The attacker gives every voter a sealed ballot that is "ready for launch". Each voter votes with the attacker's ballot. This attack is called Chain voting because in "the old style paper voting" every voter gets only one envelope. The attacker gives one voter a sealed and ready envelope. After he casts his vote, a new empty envelope is brought back from the voter. Then, the attacker, again, puts a new ballot inside and prepares it for the next voter. It operates like a chain, from one voter to the next.

The second category is **Privacy**. An attack in this category can violate the voter's privacy by revealing his plaintext vote. In addition to the voter's distress, there is imminent danger to the voter who will be exposed to extortion by political parties that aim to influence his vote. The attacks in this category:
• **Voter Exposure**: This allows the attacker to know the choice of a specific voter, or to obtain some clues about it.

• **Encryption Compromise**: Knowing or getting clues about the plaintext of an encryption regardless of who the voter was.

• **Subliminal Channel**: The encryption machine hides a subliminal message in the ciphertext. This can be done by adding extra bits to the encryption, or by hiding an extra message in the randomness, which is of course not so random.

The last category is **Denial of Service**. This category includes every attack whose purpose it is to prevent the system or one of its components from functioning properly. Cutting off the power supply is an example of a Denial of Service attack.

### 2.3. System Cryptographic Components and Security Tools

The Wombat has several cryptographic and security tools designed to protect the system against some of the potential attacks mentioned in section 2.2. In order to better understand the security functions of this model there are explanations below of the cryptographic and security components and their purposes:

**Identification Process**: The Wombat does not include an identification system. Every ballot is anonymous and does not include any voter name or any other clue about his identity. It is important to have an effective identification procedure, in order to prevent attacks such as "Registration Frauds" by impostors, "Repeating" by voters or "Ballot Stuffing" by the poll workers. A weak identification procedure can compromise the entire election process.

**Paper Ballots**: The combination of electronic voting and the old style paper voting, is not really a combination. Actually these are two separate voting systems parallel to one other. It is possible to eliminate one of them, and still use the Wombat voting system. But when both of them are in use, they can crosscheck one another. The paper ballot tallying result can be compared and crosschecked with the electronic tallying result, by sampling a few polls, or by checking the results of the entire election.

**Live CD OS**: The OS in the booth computer is Live CD OS. There is no hard disk, so no data can be saved in this computer and there are no "leftovers" from the voting process. No information can be retrieved through violation of the computer because no data is saved and the software itself cannot be changed since it is run by a CD.

**Zero Knowledge 1 of L**: A zero knowledge algorithm in the booth computer verifies that the voter has indeed selected a valid candidate,
1 of the L valid candidates, without revealing the candidate himself. The computer system, which encrypts the vote, approves the correctness of the plaintext to the Smart Card, which has CPU capability. If the booth computer's system attempts a "Subliminal Channel" attack, by adding extra data to the encryption, the Smart Card will not approve it.

**Encryption:** The electronic vote is encrypted asymmetrically by ElGamal encryption. The booth computer encrypts the vote with the public key. Each cipher vote has two parts, the first is the cipher random part and the second is the cipher candidate part. In the tallying process, the votes are decrypted with the private key. Any attack of privacy violation on the encrypted vote cannot break the encryption without the private key. The ElGamal encryption can also be decrypted by the randomness which was used to encrypt it. Every encryption is executed with a unique randomness. Therefore, one piece of randomness can decrypt only one encrypted vote. The encryption strength does not lie in the encryption algorithm alone, but also in its randomness strength which is detailed in "randomness generator". Another component that reinforces the encryption strength is the zero knowledge which is detailed in "Zero knowledge 1 of L".

**Threshold Encryption:** The private key is divided by using a threshold encryption algorithm to N entities, where every entity is one of the candidates in the election or a supervisory authority active in the election process. In order to decrypt the encrypted votes, K entities of the N entities are needed for getting the private key (K<=N). Any attacks like "Encryption Compromise", by vote decryption with the private key, requires the cooperation of at least K entities. Of course, all the N entities would not collaborate, because every entity has a different interest, and K should be large enough to ensure that the K candidates would not collaborate. Actually, the candidates take part in the generation of the keys for the encryption through the threshold encryption algorithm. Each candidate publishes his part of the public key before Election Day, and anyone can calculate the encryption's public key. At the end of Election Day, K entities calculate the private key for decrypting and tallying the votes.

**Randomness Generation:** Two separate Smart Cards produce randomness. The two random portions together complete one piece of randomness used for one vote encryption. Every Smart Card is from a different manufacturer, so in order to forge the randomness, the cooperation of both manufacturers is needed. A "Subliminal Channel" attack, forging the randomness or only part of it, has to install software in the operating system for adding the desired data into the cipher random part and digitally signing it. Both manufacturers of the Smart Cards and the operating system manufacturer must cooperate in order to execute the attack.
Digital Signatures: The randomness of each Smart Card is digitally signed by its smart card producer. The completed randomness is signed by one of the smart cards (either one). Any attack like "Altering Ballots" or "Subliminal Channel", carried out by replacing the randomness, changes the digital signatures of the Smart Cards, something which cannot be done without using the Smart Cards' manufacturers' private keys.

Audit Ballot: The voting machine can be audited. The printer prints the cipher text of one vote and then, the voter can choose to audit the system by printing an audit ballot, or continue with a regular voting ballot. An audit ballot has an addition to the randomness of its encryption, which is used for decrypting the encryption while auditing, and to check if it is indeed, the correct vote. The audit ballot is used only for auditing in the auditing station and cannot be used for voting. The purpose of the auditing process is to verify that the booth system is trustworthy and outputs the cipher text of the correct plaintext which is input in the casting process. The auditing process can be performed by any voter, but it is recommended that it be carried out at random intervals by the poll workers. If the booth computer software is attacked by an "Altering Ballots" attack, it will be revealed in the auditing process.

Serial Number: Every ballot has different serial numbers, provided by the Smart Cards that supply the randomness for the same ballot's encryption. Every ballot has several serial numbers, as many as the number of the Smart Cards in the machine. The serial numbers increase with every ballot. An attack like "Chain Voting" can be prevented by scanning only ballots in the increasing order of the serial numbers. It is impossible to print a number of ballots and use them later after someone has printed and used the next ballot.

Hash for Tallying Verification: Every ballot has a unique string, which is a hash of the cipher text of its vote. On a bulletin board accessible to the public a voter can check whether his vote is on the list by entering his ballot's hash. Assuming that the bulletin board results are consistent, any attack like "False Count" and "False Returns" by deleting votes from the bulletin board will be revealed when the voter does not succeed in confirming his participation.

Voter’s Receipt: In the scanning process the ballot is stamped at the poll and remains in the possession of the voter. In the event that the voter does not find his vote on the bulletin board, he has a receipt to confirm his vote. The reason for the physical signature, the stamp, is to prevent attacks, where there are false complaints of non-participation in a ballot which has not been scanned into the system.
Mixnet: The algorithm mixes the votes' permutation and re-encrypts them before decrypting and tallying. After the mixing, the entity which decrypts and tallies the votes cannot connect an encrypted vote to its former encryption. There is not only one entity that runs the Mixnet, every candidate or supervisory authority runs the Mixnet separately on his own server. That is to say, an entity cannot recognize the encryptions after a different entity has mixed the votes. In order to perform an attack of privacy violation, all the entities must collaborate. If all the candidates collaborate they behave as one individual. The chance that all the candidates will cooperate is negligible, because each candidate has different interests. There is a clear separation between the voter and his encrypted vote and the re-encrypted vote and its decrypted vote (the plaintext). Thus an attack of privacy violation in the tallying process can be prevented.

Mixnet Zero Knowledge: Every entity that runs the Mixnet, proves its mix by a zero knowledge algorithm. The meaning of this is that an entity that was mixing the votes can prove that the new encrypted votes are the same votes as the old encrypted ones without revealing the new permutation of the votes. In the case of an attack of "Altering Returns" in the Mixnet process, the new votes that appear after the mixing are not the same ones from the old encryptions. The candidate that performed the attack will be easily revealed, because he cannot prove the new encryptions by zero knowledge.

3. Security Threats Implementations

Every system is exposed to a variety of threats. These threats are a consequence of the system's security weaknesses, combined with the attackers' motivation. An electronic voting system is a system with many potential motivated attackers, because the consequence of a successful attack may have great significance to people of influence. The purpose of this section is to present the threats to this model. It is important to mention that these threats refer only to the model and not to the implementation. Different implementations of this model can diminish some of those threats, or result in more threats.

3.1. The Participating Parties

The table below lists potential attackers. Every attacker is a single party in the security surroundings.

<p>|   | Polling Technicians and Staff | The polling staff is found inside the poll area and has access to the equipment (by deception or by job definition) in the poll area. The assumption is that all of them have technical knowledge. |</p>
<table>
<thead>
<tr>
<th></th>
<th>Party Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Supervising Authorities and Candidates System Administrators</td>
<td>Any authority that takes active part in the election, by taking part in the Mixnet and the threshold encryption processes. The authorities are all the candidates and various active supervising authorities. The administrators have control of the authorities' servers. They are responsible for generating the encryption keys by threshold encryption algorithm and executing their part in the Mixnet process.</td>
</tr>
<tr>
<td>3</td>
<td>Bulletin Board Administrators</td>
<td>Responsible for the information that is published to the public.</td>
</tr>
<tr>
<td>4</td>
<td>Voters</td>
<td>Every voter who is eligible to vote.</td>
</tr>
<tr>
<td>5</td>
<td>Impostors</td>
<td>Every voter who enters the poll area with a fake identity.</td>
</tr>
<tr>
<td>6</td>
<td>Vote Buyers/ Coercers</td>
<td>The vote buyers attempt to influence the voters through extortion or by offering benefits.</td>
</tr>
<tr>
<td>7</td>
<td>Smart Card Manufacturers</td>
<td>Supply Smart Card, SC, which contains the digital signatures and produces the randomness for the vote's encryption.</td>
</tr>
<tr>
<td>8</td>
<td>Printer Manufacturer</td>
<td>Supplies the printer that prints the ballot.</td>
</tr>
<tr>
<td>9</td>
<td>Scanner Manufacturer</td>
<td>Supplies the barcode scanner, which scans the barcode of the encrypted vote from the voter's ballot.</td>
</tr>
<tr>
<td>10</td>
<td>Linux OS CD provider</td>
<td>Supplies the CD of the live Linux Operating System, OS, which runs on the booth computer.</td>
</tr>
<tr>
<td>11</td>
<td>External Attackers</td>
<td>Every attacker who can attack from outside the poll area.</td>
</tr>
</tbody>
</table>

### 3.2. Potential Threats from a Single Party

Below is a list of the worst case attacks of a single attacker acting alone. For each party its own threats and its own capability for potential
damage. These potential threats are explained according to the parties and the chronological numbers of the table from section 3.1.

1. Poll Technicians and staff:
   - **Ballot Stuffing:** They can print many votes for a specific candidate and "stuff" them at the end of Election Day in place of those who did not vote, or those whose non-participation was known of in advance, exactly as in a classic "Ballot Box Stuffing" attack. An effective identification process can make this attack more difficult, for example, by using one identification system for all the polls, allowing people to vote wherever they wish. Thus the poll worker has no list of those registered to vote in a specific polling station.
   - **Voter Exposure:** They can also switch the Linux OS CD, of the booth machine to an alternative CD which stores the plaintext and the ballot's serial number in a device installed in advance. That is to say, they can access all the votes printed in this specific booth and thus determine to which ballot any plaintext vote belongs according to the ballot's serial number. If all the booths in one poll are compromised in such an attack, this poll's votes are all exposed and voter privacy is violated.
   - **Authentication Disabling:** They can sabotage the scanner's computer and disable all the scanning authentications. Disabling digital signatures authentication means that there is nothing to stop the forging of Smart Cards that produce fake randomness. Fake randomness, like in the attack above, means that the votes can be easily decrypted. Another authentication, which can be disabling, is the "Chronological Order" authentication, which makes executing "Chain Voting" attacks much easier, since the adversary can print as many ballots as he wishes and use them whenever he wants, without considering the chain protocol of a "Chain Voting" attack. It is possible to prevent these attacks by double checking all the authentications on the server side before storing the votes in the database.
   - **Smart Card Forgery:** Assuming it is possible to extract the digital signatures from the Smart Card, a new Smart Card can be forged that has a valid extracted digital signature, but contains a predicted randomness. Knowing all the randomness of all the votes in the poll lets the adversary have a finite number of optional randomness for a dictionary attack for decrypting the encrypted votes(using the randomness). That is to say, the votes can be easily decrypted and the voter's privacy is compromised.
   - **Booth Denial of Service:** They can switch the paper in the printers to paper with invalid values for a "Denial of Service" attack. This attack can prevent synchronization between the result of the paper tallying ballots and the electronic tallying.
   - **Wired Network:** They can have the configurations of the
computer and the network. Therefore, they can connect to this network from another place and "sniff the data". This attack cannot be ignored although it allegedly cannot do any direct damage. However "sniffing" all the cipher votes of one poll, can be used for other attacks mentioned in this paper.

- **Booth's OS CD Manufacturers Capabilities Identity:** They can forge an alternative Linux OS CD and change it with the original CD, therefore, all the threats that apply to the Linux OS CD provider, will apply to the poll's personnel as well.

2. **Candidates System Administrators:**

   **Actions That Pose No Threat When Carried Out by a Single Party:** They have a part in the Mixnet process and in the threshold encryption process. Their part in the Mixnet process is proved by Zero Knowledge Proof. Therefore they cannot cheat as long as the Zero Knowledge protocol is not compromised. Their part in the threshold encryption process alone is not very significant, because in order to perform the threshold encryption process, only K of N entities are needed while K<N. Therefore, the candidates' system administrators alone will have difficulty attacking the system.

3. **Bulletin Board Administrators:**

   **False Results Publishing:** The bulletin board can exhibit incorrect data. It can publish false positive results, meaning, publish votes that were not cast as confirmed votes. Or they can publish false negative results, meaning, not publish votes that were cast. Therefore, only the bulletin board administrators can execute a "Denial of Service" attack, by not publishing the results for those voters who seek confirmation.

4. **Voters:**

   - **Booth Denial of Service:** Voters can sabotage the equipment and add more physical ballots to their own vote, for a primitive "Denial of Service" attack.
   - **False Accusation:** A voter can print an extra ballot and hide it on his person and then forge the stamp on its receipt. After voting with his real ballot, he can complain that his vote, the one with the false ballot's receipt, did not participate in the election. In order to prevent this kind of attack, the physical signature on the receipt must be a "hard to fake" signature, for example, a signature which includes poll's digital signature.

5. **Impostors:**

   **Registration Frauds:** They can vote in another person's name, but this attack depends on the effectiveness of the identification
process. An effective identification process like biometry identification would pose a greater challenge for impostors.

6. Vote Buyers:
   - **Vote Buyers Motivation**: They can have an interest in getting the plaintext vote of the voter, whose vote was bought or extorted, in order to verify whether he indeed voted as agreed. But as a single party they can only obtain the cipher vote from the voter's receipt.
   - **Voter Exposure**: A party can install devices, like a camera in the booth, in order to get the plaintext of the voter's ballot. But this attack can be used in any kind of voting system.

7. Smart Card Manufacturer:
   **Actions That Pose No Threat When Carried Out by a Single Party**: There are two Smart Cards in the system, so each manufacturer has only half of the randomness, meaning, he can forge only half of the randomness and sign on its half cipher random part. Therefore, he cannot forge all the randomness alone and one corrupted Smart Card manufacturer cannot compromise the system.

8. The Printer Manufacturer:
   - **Voter Exposure**: They can install a chip for saving the cipher votes and the plaintexts which it printed inside the printer, or the chip can send the data to a remote device by a wireless link. Thus, they have the plaintext vote and the cipher vote of each voter who has printed his vote in the printer. They can connect the plaintext vote to the cipher vote from the voter's receipt. In order to prevent these kinds of attacks, the printer should be checked carefully for suspicious components.
   - **Printing Subliminal Channel**: The printer can be programmed to identify the plaintext vote and mark the selected candidate, in addition to the normal data on the ballot: for example, a specific mark on the rest of the barcode or on the frame of the barcode or on any other place on the ballot. Therefore, it is possible to see the vote by looking at the ballot without decrypting the encryption. A careful check of the printed ballot can help to reveal this kind of attack.
   - **Booth Denial of Service**: The printer can print a wrong text, at a specific time on Election Day for a "Denial of Service" attack without being exposed before the specific attack's time.

9. Scanner manufacturer:
   **Partial Voter Exposure**: They can install a chip inside the scanner for saving the cipher votes, or send the data to a remote device by
a wireless link. The list of the scanned encrypted votes can be attached later to the list of voters. This list of voter names and cipher votes can be used only in combination with a list of cipher votes and plaintext from another attack. The scanner should be checked carefully for suspicious components in order to prevent such attacks. This attack cannot cause damage on its own, but it is important to mention its potential danger when combined with other attacks.

10. The Linux OS CD Provider:

- **Booth OS Built-in Backdoor**: They can supply APIs for saving the randomness produced by the two Smart Cards. Anyone who wants to change the code of the CD and to make a new one can use the API to easily do it. A checksum on the CD after code review can prevent this altered OS CD attack. Before Election Day, it is possible to validate every CD's authenticity by its checksum.

- **Altering Ballots**: Assuming it is possible to know when a ballot audit will be executed, an altered OS CD can print a normal ballot at the time of the audit, but vote for a specific candidate the rest of the time. The altered encrypted vote will not match the plaintext vote, because if it does match, the voter will easily notice the attack. Therefore, after this attack, the tallying of the encrypted votes will not match the tallying of the plaintext votes. A strict audit protocol at various intervals during Election Day can prevent this attack.

- **Subliminal Channel**: The selected candidate can be represented by some bits of the cipher random part as "Subliminal Channel" messages. In order to plant this message, the CD Provider can program the OS to program a suitable randomness from the Smart Cards. Or to be more specific, to get randomness from the Smart Cards over and over until a suitable randomness appears. Thus, a clue about the plaintext vote can be hidden in the cipher random part attached to the encrypted vote. Anyone who knows how to find this clue can know the plaintext vote without decrypting the encryption. For example, if there are eight candidates, three bits are required to represent them. Assuming that candidate four was chosen, the three bits should be "011". The probability of arriving at the suitable randomness, in this case, is 1/8, which is the rational probability for executing this attack. In order to detect this attack, the authentication component should note a significant increase in the serial numbers from the Smart Cards, because each request for randomness increases the serial numbers.

- **Printing Subliminal Channel**: An altered OS CD can mark the candidate voted for, in addition to the normal data on the ballot,
for example, by a specific mark on the rest of the barcode or on the frame of the barcode or on any other place on the ballot. Therefore, it is possible to identify the vote by looking at the ballot without decrypting the encryption. A checksum on the CD after the code review can prevent this altered OS CD attack. In addition, careful checking of the printed ballot can help to reveal this kind of attack.

11. External Attackers:
   - **Collecting External Information:** They can count voters who entered the poll, by times and photos. It can compromise the privacy of the voter's identity. This attack is trivial and relevant for any voting process, but should be noted.
   - **Denial of Service**: They can sabotage the power and communication for a "Denial of Service" attack.

3.3. Potential Threats from Collaboration of Multiple Parties

No system is fully immune if all the parties collaborate to attack it. For this reason it is important to try to be as realistic as possible while trying to predict the number of parties able to collaborate to attack the system. If all the parties collaborate, the election process is meaningless because the results are predictable. After identifying each party and its potential threats, this section contains the most significant threats resulting from the collaboration of multiple parties.

1. Polling Staff and Impostors:
   - **Registration Frauds:** The polling staff can allow impostors to vote under another name by letting them bypass the identification station without arousing suspicion. An effective identification process, biometric for example, can prevent this kind of attack by having the voter identify himself by electronic verification and not only by human confirmation.

2. Polling Staff and Voters:
   - **Repeating:** The polling staff can allow voters to repeat their vote. That is, one voter casts a number of times without arousing suspicion, namely in order to perform a "Repeating" attack. It is possible to count the number of voters and to count the number of scanned votes, while comparing them online and signaling when there are more scanned votes than voters. This action can prevent this kind of "Repeating" attack.

3. Polling Technicians and Staff and Linux OS Live CD Provider:
Booth's OS CD Manufacturers Capabilities Identity: Any of these parties, which are equally effective, can make a forged CD (it is open source), and the polling worker can replace it. The poll technicians or even the most junior official in the poll, who has access to the booth computer CD, can control the OS which runs on the booth computer.

4. Polling Staff and Vote Buyers:
   - Partial Exposure Voter: The polling staff can leak the list of all voters and their cipher votes to the vote buyers. Thus the vote buyers can confirm whether a specific voter indeed cast the agreed vote. But it is still a list of only the encrypted votes. This list can be useful for vote buyers when performing a "Chain Voting" attack, for example.

   - Voter Exposure by the Ability to Decrypt the Votes: As detailed in polling staff capabilities, they can access the randomness from the booth machine by replacing the Linux OS CD with an altered one and save the randomness on a chip that installed in advance, or alternatively, sent by a wireless link. The randomness will be sent to the vote buyers who can use this information. The vote buyers have conformation that a specific voter indeed cast a specific choice by decrypting his encrypted vote by the randomness from the encryption machine. In order to prevent such an attack, the booth machine should be checked carefully for suspicious components.

5. Polling Staff and Printer Manufacturer:
   Voter Exposure by the Ability to Decrypt the Votes: Again, as in the previous attack, this is another way to implement the same principle by using hardware to access the randomness from the booth machine. The polling staff can install a Linux OS CD which saves the randomness on the printer chip which was installed in advance with the help of the printer manufacturer. It is can also be done, by sending the data to a wireless link. This attack means that the polling staff can decrypt all of the encrypted votes at this poll. To prevent this kind of attack, the printer’s electronics must be examined by a third party supervisor.

6. Polling Staff and Bulletin Board Administrators:
   Ballot Stuffing: If the implementation of the system includes publication of the names of the voters who did vote, a "Ballot Stuffing" attack by the polling staff, can be performed if the bulletin board administrators will not publish the names of the voters who unwillingly participated in the election with the "Ballot Stuffing" attack.
7. Printer Manufacturer and Vote Buyers:
   **Voter Exposure:** The printer manufacturer can install a wireless link inside the printer which sends all of the printer's output, the cipher votes and the plaintext of the vote, to the vote buyers. Thus the vote buyers can identify the specific voter by the encrypted vote on his receipt, and the candidate selected, according to the plaintext vote. This attack can be prevented by examining the printer's electronics and checking for suspicious components.

8. Vote Buyers and Voters:
   **Chain Voting:** These parties can perform "Chain Voting", but using the general protocol of chain voting cannot be carried out in this model because while a single vote is scanned into the system, printing order authentication is carried out. This authentication means that another ballot printed after the first ballot in the same booth machine, but scanned before it to the system, is invalid. Therefore, the vote buyers cannot print several ballots and distribute them to his voters, while the same booth keeps on working normally and prints ballots for immediate scanning. A primitive "Denial of Service" attack for disabling the booth machine, right after printing several ballots, can make those ballots valid for a certain time window, until the specific booth machine becomes active again. To prevent this kind of "Chain Voting" attack, another authentication can do the trick. Allowing a few minutes (a specific time window) for each ballot, could be more than enough time for printing and voting, but not enough time for printing, ballot distribution and voting.

9. Vote Buyers and Polling Staff:
   **Chain Voting:** Another way to execute "Chain Voting" and overcome the authentication obstacle that was discussed in the previous "Chain Voting" attack section, is to use the capability of the polling staff (detailed in the polling staff capabilities) to disable the authentication in the scanner computer. This action would allow for the option of printing as many ballots as the vote buyers want and scanning them at any time. To prevent this kind of "Chain Voting" attack, random authentications auditing by an external team should be executed at each poll.

10. Two Smart Card Manufacturers:
    **Voter Exposure by the Ability to Decrypt the Votes:** Each one provides half of the randomness and together they provide all the randomness. Therefore, together they can fake a predictable stock
of randomness. If this scenario is realized, it means that there is a finite list of optional randomness which is not random at all. This list is a dictionary for decrypting any encrypted vote. To make this attack harder to execute, it is possible to add an additional Smart Card to the system from a third manufacturer. Three corrupted manufacturers are less likely than two.

11. The Two Smart Card Manufacturers and the Linux OS CD Provider:
   **Subliminal Channel**: By collaborating they can also disable the Zero Knowledge 1 of L component which is used to authenticate the “purity” of the encrypted votes and plant a "Subliminal Channel" message in the encrypted vote itself by adding a few more “wicked” bits to the encryptions. That means that the CD provider can program the Linux OS to plant a non-encrypted message in the encrypted vote, for example, in the plaintext vote itself. In addition, it is important to remember that the poll staff can also forge a Linux OS. Therefore, they can collaborate with the Smart Card manufacturers instead of with the CD provider.

12. K Candidates System Administrators:
   **Voter Exposure by the Ability To Decrypt the Votes**: If the number of candidates \( \geq K \), then they hold \( k \) parts of the secret key needed for using a threshold encryption algorithm in order to decrypt all the encrypted votes. Although each candidate has different interests, the \( k \) candidates can decide together that decrypting and identifying the votes is in their common interest regardless of the fact that the “rival” candidates will see them as well. This attack means that the encrypted votes are decryptable and a privacy compromise of the votes is possible. If this happens, it means that the implementation of the threshold encryption algorithm using \( k \) of \( N \) is too low, and a higher \( k \) is needed.

3.4 Conclusion of Security Threats

This section lists all the parties who can compromise the system, their capabilities, and the capabilities inherent in their collaboration. Some threats can do significant damage to the voting process and some less but it is important to mention them all. Many of these attacks are not trivial and require the investment of substantial resources, time, planning, coordination and preparation, and sometimes substantial financial investment. Therefore, the execution of complex attacks is less likely, as they are more complicated. It is easy to see that the most significant threat comes from the booth machine, which contains the booth computer which runs a Linux live CD, the Smart Cards and the printer. Therefore the parties who are the most significant threats are the ones closely associated with the booth machine. The poll
personnel are the ones with the greatest access to the booth machine and are therefore the most significant threat to the system. We believe that good implementation and strict physical security protocols can give a satisfactory solution to the threats described in this section.

4. Conclusions and Recommendations

The analysis of the Wombat Model is important for understanding the risks involved but even if it is perfectly secure, malicious attacks on the election process are possible if implementation of the system is not carried out with due diligence. The purpose of this section is to highlight several recommendations based on the material detailed in the previous sections. Implementation of these recommendations can prevent most of the attacks and can provide a satisfactory solution to the threats to the system. Assuming that the system's program code is perfectly secure and the security configurations are well set up, our purpose is to maximize the security of the model. Each threat should be evaluated according to the probability that it will occur and the magnitude of the threat to the system, both relative to the costs of the proposed solution.

Identification Process: This model does not include an identification system, but, as was stated in the previous sections, it is important to have a reliable one. There are a several recommendations:

- People can vote anywhere and not only at specific poll in order to prevent "Ballot Stuffing" by poll workers at the end of Election Day.
- Biometric identification can provide a good solution for the "Registration Frauds" threat.
- For the "Repeating" threat, it is possible, when implementing the Wombat System, to limit each scanned vote to a single identification by software.

Live OS CD: The CD should be digitally signed by the Live OS CD provider, in order to prevent other parties, (besides the Live OS CD provider) from altering the OS program. It is also important, to verify the digital signature in every booth machine before the start of Election Day. By implementing this recommendation, we can automatically be assured of the reliability of the poll staff, when a CD is altered. Therefore, regarding the Live OS CD, only the CD provider's honesty is at issue.

Booth Machine: The booth machine includes custom electronic components: a computer, a printer and a touch screen. All are vulnerable to the installation of malicious electronic components as detailed in section three. The booth machine should be checked for suspicious components, extra chips and wireless devices, before the start of Election Day.

Physical Stamp: Each ballot that was scanned is stamped and becomes the voter's receipt. In order to prevent a malicious voter from faking a stamp on a ballot which has not been used for voting, the stamp should be "hard to fake". It is recommended that each stamp contain a unique poll's digital signature. This action will make it easy to check if the stamp is a forgery.

Bulletin Board: Each voter can verify if his vote was tallied or not by checking if his cipher vote indeed exists on the bulletin board. The bulletin
board can give a false answer to the voter who asks if his vote was tallied or not. In order to prevent the bulletin board server from choosing who gets a correct answer and who gets a false answer, it is recommended that at the implementation stage of the bulletin board server, that the server send a digitally signed list of all the cipher votes to the client browser, or to any other client application even before such a request is made. In that way, the voter knows that all the cipher votes appearing on the bulletin board are correct.

**Booth Machine Auditing:** The Wombat System provides the option of auditing the booth machine, meaning that it is possible to verify whether the cipher vote printed from the booth machine is indeed the encryption of the same vote performed on the touch screen. Since one audit does not provide sufficient proof that the booth machine is secure, we must determine how many are needed. Referring to the "Altering Ballots" attack in section 3.2.10, the booth machine casts to a specific candidate only at some interval in time and the rest of the time, casts normally, according to the voter's choice. The question is how many times we need to audit the machine until we can be assured that there are no attack runs on the machine. Assuming that every voting takes one minute and Election Day is 12 hours. Casting can be a voter's casting, an attack's casting or an audit. Below there is a chart which represents the probability of catching one attack of intervals from a three hour attack to a fifteen minute attack. Each group of columns in the chart represents the probabilities of disclosing these attacks, when running audits at a different number of times from two audits in a day to audits an hour.

We can see in the chart above that the probability of identifying a fifteen minute attack is very low, even if an audit is run every hour. But the truth is that one fifteen minute attack in a 12 hour day will not make a big difference on Election Day. In order to effect a real change, at least one quarter of the votes in one booth must be altered, meaning that an attack would have to last at least three hours out of 12. Below is a chart representing the probability of uncovering attacks which run a total time of three hours. The probability of each attack is the sum of the geometric distribution of each attack, \( \sum_{i=k}^{1} (1-p)^{i-1}p\) when \(p\) is the probability of one attack from the chart above and \(k\) is the number of times that this attack
repeats itself for a total of three hours. For example, for attack of one hour, \( k = 3 \) and an audit every two hours during this attack, \( p = 0.5 \) according to the above chart.

![Graph showing audit frequency and attack duration]

We can see that when running an audit every two hours in one booth machine, the probability of discovering any attack is more than 80%. Of course, the same attack on more than one booth machine, using the same audit protocol, will give a higher probability. We think that an audit every two hours in each booth (at a random time in every two hour window), will provide a satisfactory solution.

**Ballot Scanning Authentications Auditing:** Each ballot that was scanned goes through a series of authentications in the verification process. Referring to the "Authentications Disabling" attack in section 3.2.1, these authentications can be disabled. In order to prevent this attack, advance preparation of a special "authentication audit ballot" is recommended. This ballot will be similar to a regular "voting ballot", only with incorrect digital signatures and serial numbers. This ballot should be used to audit the verification process. The verification process should be audited every time the booth machine is audited.

**Ballot Printing Auditing:** Referring to the "Printing Subliminal Channel" attack in section 3.2.8, the printer can print a mark on the ballot that identifies the candidate. In order to prevent this attack, the printed ballot can be audited by printing two regular "voting ballots", where each ballot is a vote for a different candidate. In order to check if there is any difference between these two ballots, except for the encrypted data, they have to be scanned in an OCR program designed for this purpose. The printed ballots should be audited every time the booth machine is audited.

For the moment, there is only one implementation of the Wombat Model, which is the creation of the model designers. It is still an academic project, whose purpose is to examine its implementation and how it performs in real time. There was an election pilot in 2011 at the Herzliya Student Union Interdisciplinary Center election and an actual election in 2012, at the of the Meretz Party primaries in Israel. They both were small scale elections of a few
thousand voters, but involved a variety of people, from students and soldiers, to older voters. They were both successful election processes from the aspects of user interface and system efficiency. However, the implementation has not yet been security audited.

References


