



PhD-FSTM-2020-55
The Faculty of Sciences, Technology and Medicine

DISSERTATION

Defence held on 25/09/2020 in Esch-sur-Alzette
to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG
EN INFORMATIQUE

by

Marie-Laure ZOLLINGER

Born on 30 December 1986 in Saint-Avold (France)

FROM SECURE TO USABLE AND VERIFIABLE
VOTING SCHEMES

Dissertation defence committee

Dr Peter Y.A. Ryan, dissertation supervisor

Professor, Université du Luxembourg

Dr Martina Angela Sasse

Fakultät für Elektrotechnik und Informationstechnik, Lehrstuhl für Human-Centred Security

Professor, Ruhr-Universität Bochum

Dr Yves le Traon, Chairman

Professor, Université du Luxembourg

Dr Olivier Pereira

Professor, Université Catholique de Louvain

Dr Peter B. Rønne, Vice Chairman

Research Associate, Université du Luxembourg

UNIVERSITY OF LUXEMBOURG

DOCTORAL THESIS

From Secure to Usable and Verifiable Voting Schemes

Marie-Laure Zollinger

Supervisor:

Prof. Peter Y.A. Ryan

Daily advisor:

Dr. Peter B. Rønne

The author was supported by the **FNR-INTER-SeVoTe** project.



Fonds National de la
Recherche Luxembourg



Abstract

Elections are the foundations of democracy. To uphold democratic principles, researchers have proposed systems that ensure the integrity of elections. It is a highly interdisciplinary field, as it can be studied from a technical, legal or societal points of view.

While lawyers give a legal framework to the voting procedures, security researchers translate these rules into technical properties that operational voting systems must satisfy, notably privacy and verifiability. If Privacy aims to protect vote-secrecy and provide coercion-resistance to the protocol, Verifiability allows voters to check that their vote has been taken into account in the general outcome, contributing to the assurance of the integrity of the elections. To satisfy both properties in a voting system, we rely on cryptographic primitives such as encryption, signatures, commitments schemes, or zero-knowledge proofs, etc. Many protocols, paper-based or electronic-based, have been designed to satisfy these properties.

Although the security of some protocols, and their limits, have been analysed from a technical perspective, the usability has often been shown to have very low rates of effectiveness. The necessary cryptographic interactions have already shown to be one contributor to this problem, but the design of the interface could also contribute by misleading voters. As elections typically rarely happen, voters must be able to understand the system they use quickly and mostly without training, which brings the user experience at the forefront of the designed protocols.

In this thesis, the first contribution is to redefine privacy and verifiability in the context of tracker-based verifiable schemes. These schemes, using a so-called tracking number for individual verification, need additional user steps that must be considered in the security evaluation. These security definitions are applied to the boardroom voting protocol F2FV used by the CNRS, and the e-voting protocol Selene, both use a tracker-based procedure for individual verifiability. We provide proofs of security in the symbolic model using the Tamarin prover.

The second contribution is an implementation of the Selene protocol as a mobile and a web application, tested in several user studies. The goal is to evaluate the usability and the overall user experience of the verifiability features, as well as their

understanding of the system through the evaluation of mental models.

The third contribution concerns the evaluation of the voters' understanding of the coercion mitigation mechanism provided by Selene, through a unique study design using game theory for the evaluation of voters.

Finally, the fourth contribution is about the design of a new voting scheme, Electryo, that is based on the Selene verification mechanisms but provides a user experience close to the standard paper-based voting protocols.

Acknowledgements

First, I would like to thank my supervisor Prof. Peter Y.A. Ryan, for having given me this great opportunity to join his group APSIA. I am grateful for his support and for the freedom offered to explore several aspects of my topic without objections.

Then, I would like to thank my daily advisor Dr. Peter B. Rønne, who supported me professionally as well as personally in all aspects of my PhD. I achieved so many things thanks to his encouragement, guidance and trust in my research, and I am very thankful for that.

Next, I express my sincere gratitude to Prof. Olivier Pereira, Prof. Martina Angela Sasse and Prof. Yves le Traon for being part of my defense as jury members without hesitation. In particular, Prof. Olivier Pereira was a member of my PhD supervision committee and followed my research from the beginning. His regular output helped me to organize my research plan.

I would like to thank Prof. Steve Schneider for the interesting discussions and projects about voting. He gave me valuable advice for my research, and I have always felt very welcomed by his team.

I also give my warmest regards to all Apsians, I was lucky to be part of a supportive group, where working days are very enjoyable. In particular, I thank Najmeh for her warming cups of tea, and Itzel for the mind-clearing climbing sessions. They were also part of my progression.

I would have never reached the end of this PhD without the support of my family and friends, who never doubted that I could achieve this.

In particular, I thank Marjorie, Carole, Baptiste and Kenzo for their support and for accepting being my guinea pigs in pilot studies. I am also grateful to my former colleagues, Alexandre, Jérôme, Renaud and Guillaume, for always cheering me up and encouraging me when I was feeling nostalgic.

Of course, I have a thought for my late father, who gave me a sense of fairness and a taste for justice, that led me where I am today. I will be always grateful to him.

Finally, my husband Aurélien gave me all his love and support through the years; my

mother and my brothers never questioned my wish of going back to the University. They all strongly believed in my capacity of success even when I myself doubted it. Your confidence was the biggest support I could ever have.

Contents

Abstract	i
Acknowledgements	iii
Contents	v
Publications	ix
1 Introduction	1
1.1 Definitions	1
1.1.1 Main properties	1
1.1.2 Other properties	3
1.1.3 Examples of approaches	3
1.2 Selene as a Use Case	4
1.2.1 The protocol	4
1.2.2 The voter’s experience	6
1.2.3 The coercion mitigation mechanism	7
1.2.4 Limitations	8
1.3 Contributions	8
1.4 Outline	9
2 Formal definitions of security properties	11
2.1 Introduction	11
2.2 The Tamarin Prover	13
2.2.1 Semantics	13
2.2.2 Adversary	14
2.2.3 Security Properties	15
2.3 Security properties for voting	15
2.3.1 Verifiability properties	16
2.3.2 Privacy	18
2.4 Voting Protocols	21
2.4.1 The CNRS protocol	21
2.4.2 The Selene protocol	22

2.5	Models	22
2.5.1	General setting	23
2.5.2	The CNRS protocol	23
2.5.3	Selene	26
2.6	Results and discussion	31
2.6.1	The CNRS protocol	32
2.6.2	Selene	33
2.7	Related Work	35
2.8	Conclusion and Future Work	36
3	Design and implementation of a usable system	37
3.1	Introduction	37
3.2	Related Work	39
3.3	Following a User-Centred Approach	39
3.3.1	Reminder of the protocol's steps	39
3.3.2	A user-oriented approach	40
3.3.3	Cryptography	41
3.3.4	Trust assumptions	42
3.4	Use Case: A Usable Interface for Selene	42
3.4.1	From paper to mobile: a participatory design	43
3.4.2	Android application	43
3.4.3	Administration page	45
3.4.4	Bulletin Board	45
3.4.5	Version 2	46
3.4.6	Web application	47
3.5	Metrics	49
3.5.1	Usability	49
3.5.2	User Experience	50
3.5.3	Psychological needs	51
3.6	Results from three user studies	51
3.6.1	User Experience and User Needs	51
3.6.2	Usability	54
3.6.3	Usability and User Experience of the coercion mitigation mechanism	56
3.7	Conclusion	62
4	Evaluating the voters' understanding through their mental models	63
4.1	Introduction	63
4.2	Related Work	65
4.3	Design of the studies	66
4.3.1	The interviews study	66
4.3.2	The drawings study	68

4.4	Mental Models	71
4.4.1	The interviews study	71
4.4.2	The drawings study	75
4.5	Discussions	80
4.5.1	The interviews study	81
4.5.2	The drawings study	83
4.5.3	Recommendations	85
4.6	Conclusion	86
5	Trust and understanding of voters: evaluation of a coercion mitigation mechanism	87
5.1	Introduction	87
5.2	Related Work	89
5.3	Trust	90
5.3.1	Definitions	90
5.3.2	Our metric	92
5.4	Game design to evaluate voters	92
5.4.1	Methodology	92
5.4.2	Voting question	94
5.4.3	A tutorial to show the coercion mitigation feature	94
5.4.4	Vote-buying Game	95
5.4.5	Ethical approval	96
5.5	Results: Evaluation of Voters' Understanding of the coercion mitigation mechanism	96
5.5.1	Quantitative results	96
5.5.2	Qualitative results	97
5.5.3	Relations between variables	100
5.5.4	Analysis	102
5.6	Conclusion and Future Work	105
6	Electryo: to a paper-based e-voting protocol	107
6.1	Introduction	107
6.1.1	The Essence of Selene	108
6.1.2	The Essence of Electryo	109
6.2	Related Work	109
6.3	A paper-based version of Selene	110
6.3.1	Participants and Primitives	110
6.3.2	The Voting Experience	112
6.3.3	Protocol	114
6.4	Security proofs	117
6.4.1	Tamarin	118
6.4.2	Results	120

6.5	A trusted protocol?	122
7	Conclusion	124
7.1	Summary	124
7.2	Future work	126
	Bibliography	127
A	Questionnaires	139
A.1	System Usability Scale Questionnaire	139
A.2	User Experience Questionnaire	140
A.3	Psychological Needs Questionnaire	142
B	Mental Models	143
B.1	Drawing material	143
B.2	Mental Models	145
C	Trust and Understanding	147
C.1	Study Description	147
C.2	Post-hoc Tukey between feelings and UEQ scores	148

Publications

The research leading to this thesis has been previously published or are under submission in conferences. Several of them are made from collaborations with two other PhD students: Verena Distler (University of Luxembourg) and Karola Marky (Technische Universität Darmstadt).

1. Peter B. Roenne, Peter Y.A. Ryan, Marie-Laure Zollinger. **Electryo, In-person Voting with Transparent Voter Verifiability and Eligibility Verifiability.** *In E-Vote-ID 2018 (TUT Proceedings).*
2. Verena Distler, Marie-Laure Zollinger, Carine Lallemand, Peter B. Rønne, Peter Y.A. Ryan, Vincent Koenig. **Security – Visible, Yet Unseen? How Displaying Security Mechanisms Impacts User Experience and Perceived Security.** *In CHI 2019.*
3. Marie-Laure Zollinger, Verena Distler, Peter B. Rønne, Peter Y.A. Ryan, Carine Lallemand, Vincent Koenig. **User Experience Design for E-Voting: How mental models align with security mechanisms.** *In E-Vote-ID 2019 (Tal-Tech Proceedings).*
4. Marie-Laure Zollinger, Peter B. Rønne, Peter Y.A. Ryan. **Mechanized Proofs of Verifiability and Privacy in a paper-based e-voting Scheme.** *In Financial Crypto 2020 (Workshop in Secure Electronic Voting).*
5. Marie-Laure Zollinger, Peter B. Rønne, Peter Y.A. Ryan. **Verifiability and Privacy of Tracker-Based Voting Protocols.** *To be submitted.*
6. Marie-Laure Zollinger, Ehsan Estaji, Peter Y.A. Ryan, Karola Marky. **“Just for the sake of transparency”: Exploring Voters’ Mental Models of Verifiability in Internet Voting.** *Submitted to CHI conference.*
7. Karola Marky, Marie-Laure Zollinger, Peter B. Roenne, Peter Y.A. Ryan, Tim Grube, Kai Kunze. **Investigating Human Factors in Individual Verifiable Internet Voting.** *Submitted to CHI conference.*

The following paper is not in the scope of the thesis:

Karola Marky, Marie-Laure Zollinger, Markus Funk, Peter Y. A. Ryan, Max Mühlhäuser. **How to Assess the Usability Metrics in E-Voting Schemes?** *In Financial Crypto 2019 (Workshop in Secure Electronic Voting)*.

Chapter 1

Introduction

Voting is the foundation of democracy. Voting research has developed many new schemes in the last decades, including one new property: Verifiability. Verifiable schemes aim to provide proof of integrity and accuracy of the result of an election. They can combine several techniques: cryptography, using verifiable computations; allowing the voters' participation by asking them to verify that their vote is included in the result; and the public display of the ballots/votes, to allow anyone to recount and verify the result. It differs from the standard and largely deployed election schemes where voters must trust the system and its administrators. While the general idea of involving the citizen in the process sounds appealing, there are many obstacles that we will discuss in this thesis, namely security, usability, and understanding issues. In this chapter, we will introduce the concepts and properties behind voting protocols. We will define the main properties that are Privacy and Verifiability, and additional desirable properties. We will focus in detail on the Selene e-voting protocol that is the main use case in this thesis.

1.1 Definitions

We will start by giving the main properties associated to voting, applicable to *paper voting* and *electronic voting* (or e-voting). Then we will give a few other properties of interest and finally some examples of approaches.

1.1.1 Main properties

The first property that we will mention is Privacy¹. The privacy of the vote is a fundamental human right and is written in the Universal Declaration of Human Rights [5]. The reason behind this property is to let anyone vote as intended without any external influence. To prevent this threat, the *secrecy* of the ballot has been

¹As a convention, concepts will be written with a capital letter through all the thesis.

designed to allow anyone to cast a vote without any pressure.

To allow voters to follow their vote and verify the process, some early schemes (e.g. [34, 20, 31]) have developed a receipt to help voters to track their vote on a public bulletin board, ensuring the correct recording. However, those schemes allowed voters to prove their vote to a third party, meaning an opportunity for vote-buying and other coercion threats. Benaloh and Tuinstra [23] defined a new concept of *receipt-freeness*, where voting systems must not provide any receipt to the voters and focus on transparency.

Finally, in [66], authors described additional coercion attacks that do not require an explicit receipt: forced abstention, usurpation of credentials, and randomization attacks (force to vote for a random candidate).

Hence, **Privacy**, in the context of voting, has been defined with those three notions:

- *ballot-secrecy*: the system must not reveal the voter's vote.
- *receipt-freeness*: the system should not give any evidence to a voter proving to a third part how he voted.
- *coercion-resistance*: the voter cannot cooperate with a coercer to prove to him how he voted.

The other main property that a voting system nowadays must ensure is Verifiability. Verifiability aims at providing tools to check the integrity of the elections, decreasing the need of trust in machines and polling clerks.

Sako and Killian [111] defined **Verifiability** informally as follows:

- *individual verifiability*: a voter can verify that his vote is included in the set of all votes.
- *universal verifiability*: any observer can verify that the tally has been correctly computed from the sets of votes.

End-to-end Verifiability (E2E-V) means that three-steps must be verified, i.e. that votes are *cast-as-intended*, *recorded-as-cast* and *tallied-as-recorded*. Individual verifiability is an opportunity for individual voters to verify that their vote are recorded-as-intended; Universal verifiability allows any voter or observer to check that votes are tallied-as-recorded, by verifying the result. The first E2E-verifiable protocol has been designed by Chaum in [30], in which voters could use pseudonyms to allow individual and universal verification of the tally.

In the concept of Verifiability, some definitions also consider the *eligibility verifiability*: any observer can verify that all votes have been cast by eligible voters.

One might notice that Privacy and Verifiability are antagonistic properties: one aims to protect and hide the data while the other aims to give information to voters

to help them verify. How can these two properties coexist? To make it work, one solution is to use cryptographic protocols and to carefully choose trust assumptions to achieve privacy while providing verifiable procedures. On the legal aspect, the Council of Europe produces a set of guidelines and good practices translating the security properties into legal requirements [46, 4].

1.1.2 Other properties

From the security point of view, the development of verifiable schemes is a good solution and leads to protocols that have a good level of Privacy and ensure several verifiability properties. So far, the price for a good security protocol was a lower usability. **Usability** is the capacity of a system to let its users perform the tasks safely, effectively, and efficiently while providing them satisfaction. In other words, usability measures the capability of a user to succeed in a given task. The first studies conducted on voting protocols using cryptography have shown that usability was low (e.g. [7]). In particular, a part of the tested voters did not manage to cast their vote or to verify the elections. But usability is not the only aspect to take into account. Recently, a few studies have shown that usability was not enough to bring satisfaction and trust in the voting system [50, 88].

Elections must also satisfy side aspects necessary for a system to work and gain the voters' trust. **Accountability** aims to detect any fraud or error in a system, being able to identify the cause. This concept is to be related with **Dispute Resolution**, which aims to figure out when a voter's claim is legit or not.

Software independence was introduced by Ronald Rivest [60, 105]: "A voting system is software independent of an (undetected) change or error in its software cannot cause an undetectable change or error in an election outcome". In other words, such a system should detect any attempt of outcome change by a malicious software, and be able to correct it. A fully verifiable system should be software independent, as we should not depend on the correctness of the software but on the verifiable procedures.

1.1.3 Examples of approaches

To obtain a verifiable voting protocols, we can distinguish the protocols using cryptography and the protocols that do not. Among those using cryptography, we have polling-based protocols using paper ballots, e.g. Prêt-à-Voter [108], polling-based protocols using electronic ballots, e.g. STAR-Vote [22], and remote protocols with electronic ballots, e.g. Helios [10]. Those three examples of protocols provide a certain level of Privacy and allow voters to verify their individual vote, giving them a receipt to track their encrypted ballot.

As we will discuss in this thesis, voters tend to prefer paper-based protocols as they have a better understanding of all steps, and they are more used to the paper process. Hence, the idea of developing verifiable protocols without involving cryptography could help to convince the skeptical voters regarding the procedure, without involving too much expertise in the process. As example of non-cryptographic protocol, ThreeBallot [106] and Origami Voting [26] are using such techniques.

In this thesis, we will consider a protocol which is in the category of remote voting protocol with an electronic ballot, namely Selene. This protocol has been designed to be more usable and intuitive than its precursors, limiting the vote casting part to a simple candidate selection, and allowing the voters to verify their plaintext vote instead of some cryptographic digits, which might seem obscure to the standard users.

Moreover, this protocol has not been explored in the existing literature, except in [29, 63, 112], which let us the opportunity to contribute further to the ongoing research. We will give details regarding Selene in the next section.

1.2 Selene as a Use Case

In this thesis, the e-voting protocol Selene [109] was taken as a use case to study security and usability of voting protocols. Selene is an e-voting scheme developed with the goal of being more usable than the previous cryptographic systems. The idea is to let voters verify their plaintext vote with a tracking number. This tracking number, or tracker, is delivered to them after the election is over, and is displayed next to their plaintext vote. Of course, to not break privacy, several mechanisms are used to keep the tracker private and to make it accessible only to the relevant voter. While vote casting and verification are made usable, the challenge is to make the security understood by the voters. Indeed, seeing their plaintext vote can make them insecure and lower their trust.

In the following, we detail the protocol by giving the cryptographic setting and the voter's experience as it was described initially in [109]. We will highlight the security mechanisms and the limitations.

A reminder of the voter's experience and how we implemented the protocol will be given in the chapters mentioning the usability of the protocol.

1.2.1 The protocol

1.2.1.1 Cryptographic primitives

We consider a group which is satisfying the DDH assumption. We consider the following parties in the protocol: n Voters, t Tellers and an Election Authority (EA). A public bulletin board (BB) is used for all verifiable communication. We use an

ElGamal (k, t) -threshold encryption scheme, we note pk_T the election public key (PK) and the secret key (SK) is shared among the t Tellers. Each voter i has a pair of public and private keys that we note resp. pk_i and x_i , such that: $pk_i = g^{x_i}$. We note $\{m\}_{PK}$ an encryption of a message m with the public key PK . An ElGamal encryption can be written as a pair $(\alpha, \beta) = (g^r, pk^r \cdot m)$, where r is the randomness used for the encryption, pk is the public key and m is the message. The protocol also use signatures and non-interactive zero-knowledge proofs of knowledge (NIZKPoK) that we will not detail here.

1.2.1.2 The protocol's steps

The Selene mechanism assigns to each voter a unique, private tracking number, used to directly check his vote in plaintext in the final tally board. This number is revealed to the voters only after all votes and trackers have been published.

Selene uses ElGamal encryption, which is homomorphic and can be used as a commitment scheme. Every voter V_i has a public key $pk_i = g^{x_i}$ where x_i is the secret trapdoor key. The election key pk_T is used to encrypt votes and trackers. A subset of t Tellers T_j perform distributed threshold decryption. A Mixnet (e.g. [127]) performs distributed re-encryption and shuffling of votes and trackers.

Selene's workflow is as follows:

(Setup) Unique tracking numbers t_i are generated. We compute g^{t_i} and its encryption with the election key then we mix, with permutation π , before associating the result to a voter's ID. Before the election starts, the obtained tracker $\{g^{t_{\pi(i)}}\}_{pk_T}$ is "transcripted" to an ElGamal encryption under the voter's key $\{g^{t_{\pi(i)}}\}_{pk_i}$, that we can write as a pair (α_i, β_i) .

The transcription works as follows: for a voter i , each Teller j produces a pair

$$(\{h_i^{r_{i,j}}\}_{pk_T}, \{g^{r_{i,j}}\}_{pk_T})$$

where $h_i = pk_i$ is the public key of the voter i , $r_{i,j} \in \mathbb{Z}_p$ is a random number. We form the (pair-wise) product across the first elements of the pairs to produce:

$$\{h_i^{r_i}\}_{pk_T} = \prod_{j=1}^t \{h_i^{r_{i,j}}\}_{pk_T}$$

We multiply this element together with the encrypted tracking number to form:

$$\{h_i^{r_i}\}_{pk_T} \cdot \{g^{t_{\pi(i)}}\}_{pk_T} = \{h_i^{r_i} \cdot g^{t_{\pi(i)}}\}_{pk_T}$$

Tellers perform a threshold decryption and we obtain the commitment to the tracker $\beta_i = h_i^{r_i} \cdot g^{t_{\pi(i)}}$.

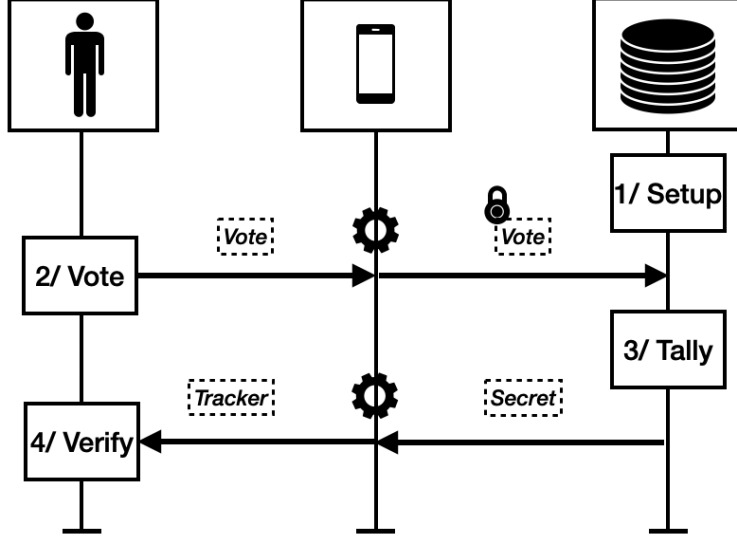


Figure 1.1: The voter’s experience.

The β -term is displayed on the bulletin board, as a trapdoor commitment. The encrypted α -terms $\alpha_{i,j} = g^{r_{i,j}}$ are kept secret. The Bulletin Board now contains for every voter the following tuple:

$$(pk_i, \{g^{t_{\pi(i)}}\}_{pk_T}, \beta_i)$$

(Vote) Each voter V_i sends a signed and encrypted vote $Sign_i(\{vote_i\}_{pk_T})$ together with a zero-knowledge proof of knowledge π_i (see [49, 25]) of the plaintext to the bulletin board, which checks the signature and the proof before appending the encrypted vote.

$$(pk_i, \{g^{t_{\pi(i)}}\}_{pk_T}, \beta_i, \{vote_i\}_{pk_T}, \pi_i)$$

(Tally) At the end of the election, votes and associated trackers are extracted, mixed in parallel and threshold decrypted in a public verifiable manner to be displayed on the bulletin board.

$$(g^{t_{\pi(i)}}, vote_i)$$

(Verify) The encrypted $\alpha_{i,j}$ are combined, threshold decrypted and sent to the voter, who can build his encrypted tracker by putting it together with the public commitment β -term. He decrypts it with his trapdoor key to retrieve the tracker, and verify his vote.

1.2.2 The voter’s experience

In [109], the voter’s experience has not been given in detail, and authors assumed that voters already possess their keys. In the following, we won’t go further, and we

assume that voters can cast their vote with a device that we let unspecified here. We will specify these aspects in chapter 3 where we will detail our interface’s implementation.

First, the voters receive an invitation to vote. With their device containing their keys, they input their choice and the device encrypts and signs the vote. The resulting ballot is sent to the Election Server.

When the voting phase is over, the tally is computed and pairs of (tracker, vote) are displayed on the BB. After some delay, the voters receive an invitation to verify their vote. They are notified of the secret α -terms, that they can input in their device to compute the tracking number from the β -term and trapdoor key. Once the tracker has been retrieved, voters can verify that their vote appear on the BB.

An overview of the different steps are given in figure 1.1.

As in other verifiable protocols, the verification step is optional.

1.2.3 The coercion mitigation mechanism

One important feature brought by the protocol is a coercion mitigation mechanism that voters can use to fake their tracker, if they need to. Selene has been designed carefully to preserve vote privacy even if a plaintext vote is available for individual verification on the bulletin board. The mechanism works as follows: when the bulletin board displays the pairs of (tracker, vote) on the bulletin board, a coerced voter can have a look and notes down a tracking number which will please the coercer. With the help of her trapdoor key, she is able to input in her device the wished tracker and the device will display this tracker instead of her real one.

Nevertheless, the proof that the coercer will see has to be convincing, he might request more than the tracker itself. Let’s remind how the tracker retrieval works: the voter i receives α -terms from the Tellers, and can combine them with the commitment C_i , which is the β -term needed to obtain the complete encryption of the tracker t_i over the voter’s key. Then, with the trapdoor key x_i , the voter computes:

$$\frac{C_i}{\alpha^{x_i}} = \frac{g^{x_i \cdot r_i} \cdot g^{t_i}}{g^{x_i \cdot r_i}} = g^{t_i}$$

When choosing another tracker $g^{t'_i}$ on the bulletin board, the voter can compute a fake α -term with his trapdoor key:

$$\alpha' = \left(\frac{C_i}{g^{t'_i}} \right)^{x_i^{-1}}$$

It is intractable to compute an alternative α -term that will open to a valid tracker without the trapdoor key because of the Diffie-Hellman Inversion Problem. The only

requirement to allow a voter to fake his tracker is that the delivery of the α -terms must be done through an untappable channel to make them deniable.

1.2.4 Limitations

This protocol has several limitations that we will discuss now. Partial solutions have been brought already, and we give them below.

First, as for all remote voting schemes, we assume that the voter can cast his vote alone, without the constant pressure of a coercer. This can be mitigated by establishing a revoting policy, again assuming that the voter will be able to cast a vote alone at some point during the voting phase.

Another limitation mentioned in [109] is that for the coercion mitigation mechanism, an unlucky voter could choose the tracker of the coercer himself, or the coercer could claim that the given tracker is his own tracker in all cases. This situation is solved in the original paper by introducing a variant to Selene, where $c + 1$ trackers are assigned to each voter, c being the number of candidates. The voter can safely choose another tracker as all of them are his.

As mentioned above, the delivery of alpha-terms must be done through an untappable channel, meaning that the coercer must not monitor the alpha-terms delivery to leave the opportunity to the voter to fake them.

Finally, dispute resolution is pretty tough with Selene: as we give them a mean to deny their vote, some dishonest voters could try to claim an error in the elections, claiming their tracker is not showing their vote. This is solvable only by breaking privacy in camera and looking at the encrypted vote that has been sent.

1.3 Contributions

The first contribution concerns the definitions of Privacy and Verifiability in the symbolic model. We propose new definitions in the context of tracker-based verifiable schemes, such as Selene. Those schemes use a tracking number for individual verification, that lead to additional steps for the setup and the verification phases. We considered these additional steps in the security evaluation of Privacy, and in the definition of individual and universal verifiability. We provide security proofs in the symbolic model using the Tamarin prover.

The second contribution concerns the development of a usable interface for Selene, following a user-centred approach. The methodology that we follow builds on users needs rather than the protocol itself, even though the security elements must be taken into consideration. We did evaluate our interface in three user studies, iterating on our participants' feedback to improve our results. We evaluated our participants through quantitative metrics, using standard questionnaires on one side, and through qualitative metrics, using semi-structured interviews and a drawing session on the

other side. This will give us the first insights regarding the voters' understanding and expectations of our verifiability procedures.

The third contribution is the evaluation of the understanding of the coercion mitigation mechanism in Selene, using a unique study design where participants play a theoretic game. By proposing incentives depending on their choices in the voting application, the hope is to observe that the users follow the dominant strategy, which correspond to a correct use of the coercion mitigation mechanism. We also evaluate the participants' feedback with questionnaires, in particular we relate this result to their trust through a new questionnaire.

The last contribution concerns the design of a new voting scheme, Electryo, that is based on the Selene verification mechanism, but provides a user experience close to a standard protocol using paper ballots. We provide a first evaluation of the security in the symbolic model.

1.4 Outline

This thesis is structured in 7 chapters. In the following, we outline the contents of each remaining chapter:

Chapter 2: Formal definitions of security properties In this chapter, we redefine Privacy and Verifiability in the context of tracker-based verifiable schemes. We give security proofs using the Tamarin prover.

Chapter 3: Design and implementation of a usable system In this chapter, we introduce a methodology for a user-centred development of a usable interface, for the e-voting scheme Selene. We give the usability and user experience results from three user studies.

Chapter 4: Evaluating the voters' understanding through their mental models In this chapter, we analyse interviews and drawings from two user studies, we explore the mental models of our participants to evaluate their understanding of our voting application. We provide guidelines for future development of voting schemes.

Chapter 5: Trust and understanding of voters: a new game theoretic study In this chapter, we run a new user study with the idea of evaluating the understanding of the coercion mitigation mechanism in Selene. We used game theory to design an online game where participants have to take a decision in a specific

scenario. We also evaluate the trust of participants in our application and relate it to their understanding.

Chapter 6: Electryo: to a paper-based e-voting protocol In this chapter, we describe a new e-voting protocol, namely Electryo, that implement the Selene mechanism for individual verification in a standard paper ballot voting scheme. We describe the protocol and give proofs in the symbolic model for Vote-Privacy and Individual Verifiability.

Chapter 7: Conclusion and future works In this chapter, we conclude our work and suggest future plans for the research.

Chapter 2

Formal definitions of security properties

Security protocols are designed to ensure specific properties. Voting protocols, in particular, aim to provide two main features: Privacy and Accuracy of the announced outcome. Privacy concerns Vote-Privacy, Receipt-Freeness and Coercion-Resistance. Accuracy is guaranteed through verification procedures, leading to the development of verifiable schemes. In this chapter, we look at tracker-based verifiable schemes, and we present new definitions of Privacy and Verifiability, reflecting the additional features introduced by tracker verification. We modelled two protocols with the TAMARIN prover: the CNRS boardroom protocol and the Selene e-voting protocol, both use trackers for individual verification. While the first is very simple and does not use cryptography, the second relies on advanced cryptographic primitives. We prove certain privacy properties and verifiability of these schemes under certain trust assumptions.

2.1 Introduction

Research into secure voting systems over the past few decades has led to the development of many new protocols and properties. In particular, *verifiability* is a novel requirement that enables voters and observers to confirm that votes are accurately included in the tally, with minimal trust assumptions. Privacy can be refined to Ballot-Secrecy (Vote-Privacy), Receipt-Freeness and Coercion-Resistance. Verifiability includes individual verification, where voters can check that their vote is cast-as-intended and recorded-as-cast; universal verification, where any observer can verify the outcome of the election; and, in some cases, *eligibility verification*, where anyone can verify that votes are cast by eligible voters only.

It is of course essential that such protocols are rigorously analysed against these requirements and with respect to precise threat models. One approach consists in formal verification in the symbolic model. Many tools exist to perform automated

verification such as [27, 33, 53], that aim to create proofs for security properties such as *Vote-Privacy* and *Verifiability*. However, each tool has limitations for describing the protocols and executing the proposed model, e.g., false attacks due to approximations or limitations in terms of the cryptographic primitives that can be handled.

We categorized verifiable voting schemes fall into two categories: “conventional”, in which voters check the presence of an encryption of their vote on a bulletin board, and “tracker-based”, in which the voter can identify their plaintext vote in the published election result using a private tracker. Tracker-based protocols include [109, 75, 63, 107], but also schemes where plaintext votes are cast via anonymous channels together with a pseudonym/tracker. These include many simple implementations involving cryptocurrencies, but also more advanced constructions [58].

In this chapter, we present novel, formal definitions for Verifiability and Privacy, for tracker-based protocols. Our Verifiability definitions build on the definitions from Cortier et al. [36], considering the additional step of tracker retrieval. Our Privacy definitions are based on a similar setting as the definitions from Delaune et al. [39], that is showing an equivalence between processes considering two honest voters swapping votes. We keep the same definition for Vote-Privacy, but we define Receipt-Freeness as indistinguishability between a voter telling the truth and a voter lying about his tracker. We elaborate on this new approach to redefine Coercion-Resistance, with a new but limited version of this property as well. We test our definitions on the CNRS boardroom voting protocol described in [13] and on the e-voting protocol Selene [109]. Models are implemented with the tool TAMARIN [16]. TAMARIN allows user-defined equations and implements some cryptographic primitives.

The CNRS protocol is a simple boardroom voting protocol, which seeks to provide simplicity, privacy and full verifiability [13]. The protocol does not use cryptography, and trust assumptions must be made on some entities. The idea is to provide each voter with a tracker before vote casting, voters must cast their vote with a voting device. The ballot is a concatenation of the vote and the tracker. The votes are displayed on a screen and each voter can verify that their tracker shows the correct vote.

On the other hand, the Selene e-voting protocol [109] employs a number of advanced cryptographic primitives, such as threshold encryption and commitment schemes. Voters can verify their vote with a tracking number, delivered after the final tally is published, and Selene also gives the possibility to fake it in case of coercion. A commitment to the tracker is published at an early stage on the bulletin board associated with the voter’s ID. The information needed to open the commitment, the so-called *alpha-term*, is shared among a set of Tracker Tellers. After a standard voting phase, the tally is computed and the authorities publish a set of anonymised (*tracker, vote*) pairs. To counter coercion threats, Selene adopts the strategy of only notifying voters of their tracker after the publication on the bulletin board. This gives coerced voters

the opportunity to choose a tracker that points to the coercer’s required candidate, that they then claim as their tracker. Thus, voters are notified with alpha-terms, only after the anonymised (*tracker, vote*) pairs have been computed and publicly revealed. From the published commitment, the shared alpha-terms and their private key, the voters can open their tracking number and verify their vote on the bulletin board.

Contributions

In this chapter, we define Verifiability and Privacy for tracker-based verifiable voting protocols. In particular, we formalize two Verifiability properties (i.e. Individual and Universal Verifiability), and two Privacy properties (i.e. Receipt-Freeness and Coercion-Resistance) in the symbolic model. Then we propose models and their security proofs, including Vote-Privacy, for two tracker-based verifiable schemes: the CNRS boardroom voting scheme [13] and Selene [109] using the TAMARIN prover [16].

The chapter is organized as follows: In section 2.2 we introduce the TAMARIN prover, its semantics and how security properties are modelled. In section 2.3 we formalize our properties with predicates and give our new definitions. In section 2.4 we give more details regarding the voting protocols and the known attacks. In section 2.5 we detail our models, specifically the trust assumptions, the list of entities and roles, and the formal properties we aim to prove. In section 2.6 we show our proofs and attacks, besides a discussion about the model and findings. We discuss related work in 2.7 and conclude in section 2.8.

2.2 The Tamarin Prover

The TAMARIN prover is a security protocol verification tool, which has been released in its first version in 2012 [16].

2.2.1 Semantics

In TAMARIN, messages are represented as terms. A term is a variable t or a function $f(t_1, \dots, t_n)$ of arity n , where t_1, \dots, t_n are terms. We denote the set of terms \mathcal{T} . The set of operators with their arities are defined as the signature \mathbf{sign}_{Op} . TAMARIN uses equational specifications for cryptographic operators. An equation is an unordered pair of terms s and t , written $s = t$. Let E denote a set of equations. An equational theory is the smallest congruence closure containing all instances of E . The equivalence modulo E between terms s and t is defined as the smallest, symmetric and transitive closure of the equational theory E , and we write it $s =_E t$.

$$\begin{aligned}
& \{ \text{Out}(x) \rightarrow \mathsf{K}(x), \mathsf{K}(x) \xrightarrow{\mathsf{K}(x)} \text{In}(x), \text{Fr}(\sim x) \rightarrow \mathsf{K}(\sim x), \\
& \quad \square \rightarrow \mathsf{K}(\$x) \} \cup \\
& \{ \mathsf{K}(x_1), \dots, \mathsf{K}(x_n) \rightarrow \mathsf{K}(f(x_1, \dots, x_n)) \mid f \in \text{sign}_{Op} \text{ of arity } n \}
\end{aligned}$$

Figure 2.1: Message deduction rules of the adversary.

Protocols are modelled through multiset rewriting rules. These rules use sets of Facts that represent the current state of the system. A set of Facts is defined as $\mathcal{F} = \{F(t_1, \dots, t_n) \mid t_i \in \mathcal{T}, F \text{ is a fact of arity } n\}$. We define fresh values with the annotation \sim and public values with $\$$. Facts are user-defined except: **Fr** for fresh nonces, **In** and **Out** for inputs and outputs of a rule, and **K** for the attacker knowledge¹. An exclamation mark **!** before a Fact will define it as persistent and can be consumed many times.

We define a rule as a tuple (id, l, a, r) with id a unique identifier, and l, a, r are multisets of Facts. A rule is written $id : l \xrightarrow{a} r$, we say that l is the premise, a is the label and r is the conclusion. A set of multiset rewriting rules is a labeled transition system. The initial state of a labeled transition system is the empty multiset of facts \emptyset .

We define an execution of a protocol as the sequence of states and rule instances: $S_0, (l_1 \xrightarrow{a_1} r_1), S_1, \dots, S_{n-1}, (l_n \xrightarrow{a_n} r_n), S_n$ with $S_0 = \emptyset$ and we demand that $(S_{i-1}, (l_i \xrightarrow{a_i} r_i), S_i)$ are valid according to the step rule for all $i \in \{1, \dots, n\}$.

2.2.2 Adversary

In TAMARIN, the adversary is a standard Dolev-Yao style attacker [42], that is controlling the network and can apply all operators. The message deduction rules are described in figure 2.1.

The adversary learns all messages sent by participants when they are output with the **Out** fact. He can send messages to the participants with the **In** fact, that is we assume that every input could be given by the adversary. The adversary can also generate fresh values and knows all public values. Finally, he can apply all functions available in the signature sign_{Op} .

¹In TAMARIN, a distinction is made for the attacker's knowledge on terms on which he can apply construction rules and deconstruction rules. Here we won't detail these aspects, see [43] for more details.

2.2.3 Security Properties

2.2.3.1 Observational equivalence

Observational equivalence [17] is used to prove indistinguishability between two executions of a system. To prove observational equivalence, TAMARIN is using what is called a *bi-system*, that is a multiset rewrite system where terms can be built using a special operator $\text{diff}(\cdot, \cdot)$. This operator lets us instantiate a term with two possible elements. When using it, TAMARIN creates two systems (that we will refer to as a left-hand side (LHS) and a right-hand side (RHS)) with identical rules where the only difference is the value of the term instantiated with the diff operator.

When using the diff operator in TAMARIN, the tool automatically creates an Observational Equivalence lemma. The algorithm behind relies on *dependency graphs*. A *dependency graph* is a data structure that represents the system execution. Each node is a rule defined in the model, and there is a direct relation from a rule r_1 to r_2 iff r_1 outputs a fact to the input of r_2 . Dependency graphs induce an equivalence relation that is stronger than, i.e. implies, observational equivalence. The equivalence between dependency graphs depends on the notion of *mirrors*. Mirroring is defined as an isomorphism between two graphs, that is: given a dependency graph on the LHS, its mirrors contain all graphs on the RHS of the bi-system defined with the diff operator, where nodes are instances of the same rules and edges are the same.

In the voting context, we can express privacy-related properties in terms of indistinguishability between two systems. We give our definitions for privacy in the next section, and we detail the proofs in section 6.4.2.

2.2.3.2 Traces

Another way to prove a security property in TAMARIN is to use the traces of the protocol. These trace properties are expressed as first order logic formulas. These formulas also use the temporality of the protocol with new variables, for reasoning about the order of events. The detail of the semantics for trace properties is given in [113].

In the voting context, these trace properties can be used to define the verifiability properties. We give our verifiability definitions in the next section, and we detail the verifiability proofs obtained for Selene in section 6.4.2.

2.3 Security properties for voting

Voting properties refer to Privacy and Verifiability. We detail below our new definitions of Verifiability for tracker verifiable schemes based on Cortier et al. [36]. We also give definitions for Vote-Privacy and Receipt-Freeness based on Delaune et al. [39] and we give a new definition of Receipt-Freeness based on indistinguishability between voters' behaviour. From this last definition, we propose a new concept of

Coercion-Mitigation, where a coercer cannot distinguish between a voter following his instructions and another voter, with limitations that we will describe below.

2.3.1 Verifiability properties

Verifiability has been defined by Sako and Killian [111] as:

- **Individual Verifiability:** a voter can verify that her vote is included in the set of all cast votes on the bulletin board.
- **Universal Verifiability:** any observer can verify that the election outcome corresponds to the ballots on the bulletin board.

One other aspect that can be included in universal verifiability is *Eligibility Verifiability*: anyone can check that each counted vote has been cast by a registered voter and there is one vote per voter. In this chapter, we will not address eligibility verifiability.

In the existing literature, Kremer et al. [71] and Cortier et al. [36] define verifiability in symbolic models. Here we propose a definition adapted from [36] for tracker verifiable protocols [109, 75, 63, 107, 13]. In [36], authors build on type-based systems to test their definition, while we will use a verification tool. Comparing to their definition, the additional information that we need to provide is the tracker that is used to verify the vote².

Tracker verifiable protocols consist in verifying with a tracking number, delivered privately to the voter, that his vote is taken into account and appears on the bulletin board. We need to introduce the following predicates to define our notion of verifiability:

- **Voting($i, vote$):** the voter i wants to cast his vote $vote$.
- **SetBallot($i, vote, b$):** the voter i computes a ballot b containing his vote $vote$.
- **GetTracker($i, t, rvote$):** the voter i computes a tracker t that shows a recorded vote $rvote$. The way the tracker is computed differs from one scheme to another.
- **VHappy($i, vote, BB$):** the voter i looks for his cast vote $vote$ on the bulletin board BB .
- **JHappy(BB, r):** a third-party (judge) is happy when the result r corresponds to the content of the bulletin board BB .

We are now ready to define our Verifiability properties

²In their definition, Cortier et al. define a “ballot” and verify that it appears on the bulletin board, e.g. in Helios.

2.3.1.1 Individual Verifiability

Individual Verifiability is the ability of a voter to check that his vote is recorded-as-intended on the election bulletin board. Regarding the predicates given above, we define Individual Verifiability as follows.

Definition 1 (Individual Verifiability). For every execution, for every voter i , vote $vote$, tracker t and bulletin board BB , we have: $\forall \text{Happy}(i, vote, BB) \Rightarrow \text{Voting}(i, vote) \wedge \exists t \in BB. \text{GetTracker}(i, t, vote)$

The examples of the use of `VHappy` and `GetTracker` in our models, see figures 2.3 and 6.2.

2.3.1.2 Universal Verifiability

Universal Verifiability can be defined as the ability of any observer to verify the result of an election [37]. It concerns the correct execution of the protocol, that can be verified by validating Zero-Knowledge Proofs if used in the protocol, e.g. [18]. It also means that any observer can verify that the final tally is correct.

In the symbolic definition in [36], the authors give a “sanitized” version of the Bulletin Board with invalid votes removed. In our protocol models, we assume that voters are honest and are allowed to cast their vote only once, but we will introduce a similar concept in the definitions for generality. In [36] they also provide a predicate to verify the correct counting of votes. For tracker verifiable schemes, we slightly modified those elements by adding a predicate that links the submitted vote by the voter and the published tracker on the final bulletin board. Also, to ensure the correct counting of votes, we ensure the correctness of the Bulletin Board (see below).

Clean the Bulletin Board We define the predicate `CleanBB`(BB, BB') to check if a sanitized bulletin board BB' is correctly constructed. The predicate `CleanBB`(BB, BB') holds true iff for all $b \in BB$ such that `SetBallot`($i, vote, b$) holds true for voter i and a vote $vote$, then $b \in BB'$.

Correctness of Ballots Before introducing the correct counting of votes, we need to define the well-formedness of a ballot sent by a voter to the authorities. To achieve this, we define a predicate `VoteInBallot` to verify that a given vote is inside a ballot. A ballot b corresponding to the vote $vote$ for voter i satisfies:

- (i) `VoteInBallot`($vote, b$) $\Rightarrow \exists i. \text{SetBallot}(i, vote, b)$,
- (ii) `VoteInBallot`($vote1, b$) $\wedge \text{VoteInBallot}(vote2, b) \Rightarrow vote1 = vote2$. The first condition ensures that the ballot b contains a vote $vote$ set by a voter. This is true if and only if the voter knows the plaintext vote, as the `SetBallot` predicate corresponds to the computation by voter i of the ballot. The second condition ensures that a ballot cannot contain two different votes.

Correctness of the Bulletin Board Finally, we need to provide a new predicate for verifying the correct outcome of the election. As stated in [24], universal verifiability guarantees that:

$$\text{Correct}(BB) \Rightarrow \text{Correctly tallied}$$

We consider the predicate $\text{Correct}(BB)$ to describe that the bulletin board BB is valid. We have that $\text{Correct}(BB)$ holds true if for all pairs of (tracker, vote), noted (t, vote) , such that $BB[j] = (t, \text{vote})$ for some index j , there exists a ballot b such that $\text{VoteInBallot}(\text{vote}, b)$.

Definition 2 (Universal Verifiability). For every execution, for every bulletin board BB and result r , we have: $\text{JHappy}(BB, r) \Rightarrow \exists BB'. \text{CleanBB}(BB, BB') \wedge \text{Correct}(BB')$ where BB' is the cleaned bulletin board.

2.3.2 Privacy

Privacy in voting research refers to Ballot-Secrecy (or Vote-Privacy), Receipt-Freeness and Coercion-Resistance. In the symbolic model, privacy properties are defined as an equivalence between processes [110]. We will give below a definition for Vote-Privacy and Receipt-Freeness following the definitions by Delaune et al. [39]. We also re-define Receipt-Freeness, with a new approach using two honest voters both outputting their data to the adversary. Our definition of Receipt-Freeness will shift to Coercion-Mitigation, as long as we have at least one honest vote for the requested candidate.

2.3.2.1 Vote-Privacy

In the symbolic model, we usually define Ballot-Secrecy, or Vote-Privacy, as the protocol must not reveal the intention of each voter [11]. Delaune et al. [39] define it as an equivalence between two system where the votes of two voters have been swapped. The outcome of the election will remain the same, the adversary tries to observe a difference in the voters' intention of vote. We rely on this approach for our definition of Ballot-Secrecy.

Using the notations provided by Basin et al. in [18], we denote $x \approx y$ the observational equivalence of a left system x with a right system y . We use the notation $P_{v_1 \leftarrow v'_1, v_2 \leftarrow v'_2}$ to define the protocol P where all occurrences of v_1 are replaced by v'_1 and all occurrences of v_2 are replaced by v'_2 .

Definition 3 (Vote-Privacy). Let P be a protocol as defined above. We denote v_A and v_B the terms for voter A and voter B's votes, respectively. Let v_1 and v_2 be message terms. The protocol P guarantees Vote-Privacy iff we have: $P_{v_A \leftarrow v_1, v_B \leftarrow v_2} \approx P_{v_A \leftarrow v_2, v_B \leftarrow v_1}$.

2.3.2.2 Receipt-Freeness

Informally, Receipt-Freeness has been defined as follows by Benaloh and Tuinstra [23]: no voter should be able to convince any other participant of the value of her vote.

In the symbolic model, Receipt-Freeness is defined as an equivalence between two executions where the coerced voter reveals all their private information and the other does not [39]. The relation with Benaloh and Tuinstra’s definition is that the private information revealed should not give information regarding how the voter really votes.

We define the protocol P' that is similar to P in Vote-Privacy, except that voter A gives all his private data to the adversary, and always claim voting for candidate v'_1 .

Definition 4 (Receipt-Freeness (Delaune [39])). Let P' be the protocol obtained from P as described above. We denote v_A and v_B the terms for voter A and voter B’s votes, respectively. Let v_1 and v_2 be message terms. The protocol P guarantees Standard Receipt-Freeness iff we have: $P'_{v_A \leftarrow v_1, v_B \leftarrow v_2} \approx P'_{v_A \leftarrow v_2, v_B \leftarrow v_1}$.

For our new definition, we denote P'' a protocol similar to protocol P defined above, except that voter A and voter B give their private data to the adversary, and voter A fakes his data in the left system and voter B fakes his data in the right system. This definition considers the ability of a voter to lie about his vote and sometimes to fake data in tracker-based verifiable scheme. The idea is to show that an attacker cannot distinguish between a cooperative voter and a non-cooperative one, while the standard definition focuses on one voter. With this construction, both voters provide data to the adversary and always claim voting for the same candidate. However, the adversary cannot say which voter is lying.

Definition 5 (Receipt-Freeness (New)). Let P'' be the protocol obtained from P as described above. We denote v_A and v_B the terms for voter A and voter B’s votes, respectively. Let v_1 and v_2 be message terms. The protocol P guarantees New Receipt-Freeness iff we have: $P''_{v_A \leftarrow v_1, v_B \leftarrow v_2} \approx P''_{v_A \leftarrow v_2, v_B \leftarrow v_1}$.

The private data that the voters give away to the adversary depends on the protocol, we will discuss this aspect in the section 2.5.

The timing of the tracker delivery is important for receipt-freeness, especially the voters need to be able to lie by knowing other trackers. For some protocols, e.g. the CNRS boardroom protocol, the tracker is known by the voter before vote casting. However, voters are still able to deny their tracker after the tally is published. Hence, we will call “Tracker Deniability” the ability of a voter to lie about his tracker after the tally has been published.

2.3.2.3 Coercion-Resistance and Coercion Mitigation

Coercion-Resistance is defined as follows in [39]: a voter cannot cooperate with a coercer to prove to him that she voted in a certain way. In particular, the coercer can prepare messages that he wants the voter to send. Delaune et al. adapted the definition of Coercion-Resistance by defining a simulation relation, called “adaptive” simulation, which let the second voter (not coerced) adapt his vote to correspond to the coercer’s choice. Hence, the coerced voter can always claim a vote corresponding to the coercer’s instruction, but keep his vote intention.

If we consider our new definition of Receipt-Freeness above, and the idea of an adaptive simulation, it follows that we could cover some aspects of Coercion-Resistance if there exists at least one honest vote on the bulletin board for the requested candidate (that is not the vote of the coercer himself) which all voters can point to in case of coercion. Indeed, if the coercer cannot distinguish between a cooperative voter and a non-cooperative one by looking at their receipt, he won’t be able to distinguish between an execution where the voter follows his instruction and another where he does not, while there is an honest vote for the requested candidate on the bulletin board.

We denote by P''' a protocol defined as protocol P'' above, but, in addition, the coercer gives to voter A an instruction I to vote.

Definition 6 (Coercion-Resistance (New)). Let P''' be the protocol obtained from P'' as described above. We denote v_A and v_B the terms for voter A and voter B’s votes, respectively. Let v_1 and v_2 be message terms, f_A and f_B are functions taking 1 parameter I that is the instruction of the adversary. The functions adapt the instruction to fit the voting term. The protocol P guarantees New Coercion-Resistance iff we have: $P'''_{v_A \leftarrow v_1, v_B \leftarrow f_B(I)} \approx P'''_{v_A \leftarrow f_A(I), v_B \leftarrow v_2}$.

Note that for this definition we assume that a voter can cast a vote in private without the coercer at his side.

In the case of a tracker-based protocol, this definition states that voter B votes for the requested candidate if voter A does not follow the instruction, letting voter A to “reveal” a tracker that will satisfy the coercer³.

The form of the coercer’s instructions to the voter can have an impact on the voter’s capabilities. If the coercer is limited to a vote instruction (i.e. mention a candidate’s name), it gives more power to the voter to protect his vote. We define “Coercion-Mitigation” the ability of a voter to cast his intended vote when the coercer is limited to a candidate’s name for the vote instruction.

Definition 7 (Coercion-Mitigation). Let P''' be the protocol obtained from P'' as described above. The protocol P guarantees Coercion-Mitigation iff P''' satisfies Coercion-Resistance where the adversary is limited to give a candidate’s name as an instruction to the voter.

³Note that a stronger definition would avoid a collusion between voter A and voter B.

2.4 Voting Protocols

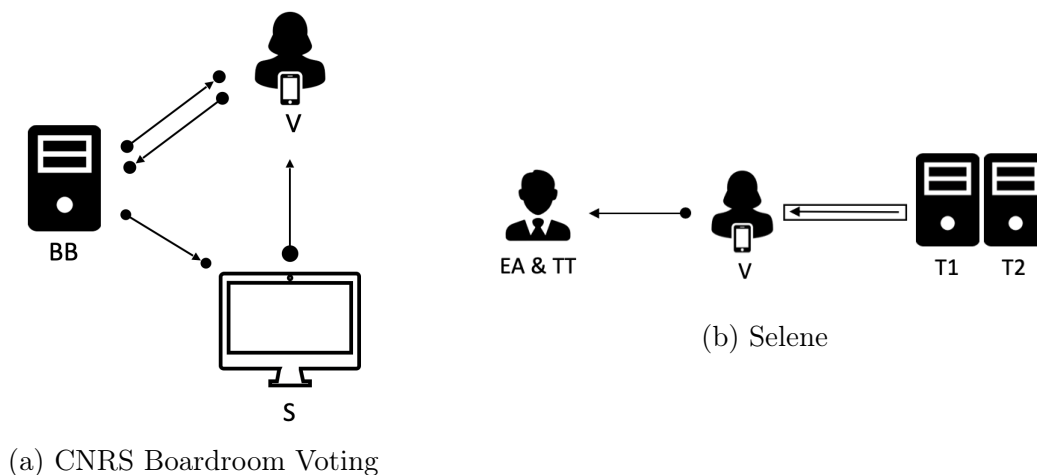


Figure 2.2: The channel rules between participants.

2.4.1 The CNRS protocol

First, we consider a simple tracker-based scheme that has been described in [13]. This protocol has been developed to be used in boardroom voting, and provides privacy and verifiability (with some limitations). Three variants of this protocol have been proposed, in this section we consider only one version; in the next section we will mention the second variant which is a simplified version.

2.4.1.1 The protocol

We consider a Ballot Box BB that collects the ballots and tallies the votes, a list of voters V_i , a Screen S and an auditor A . The protocol workflow is as follows:

(Setup) The Ballot Box generates one tracker per voter and sends them to the voting devices.

(Vote) Each voter casts his vote with his voting device. The vote is sent back to the Ballot Box along with the tracker.

(Tally) Then the Ballot Box tallies the votes and sends the (tracker, vote) pairs to the Screen.

(Verify) Each voter can verify that their vote has been correctly recorded. They let the auditor know if their check was successful, and the auditor can validate the result.

2.4.1.2 Known attacks

In [13], the authors describe a few possible attacks: if the ballot box is dishonest, it can generate the same tracker for 2 different voters who vote for the same candidate, and add a new ballot of its choice – the so-called clash attack. Furthermore, with a dishonest ballot box privacy cannot be guaranteed as the ballot box can leak how each voter has voted. Or, the ballot box can retain one ballot and the voter not seeing his ballot on the screen will complain, revealing his identity to the attacker. Finally, an attack on privacy could be to encode the identity of the voters in the random tracking numbers: e.g. a voter i would receive a tracker $r_i \equiv i \pmod{p}$, an attacker would deduce the vote by reducing the value in the ballot $\langle r_i, v_i \rangle$. For Vote-Privacy, we will hence assume an honest ballot box.

The protocol is not receipt-free if the coercer can ask for the tracker before voting as, at this point, the voter does not know other tracker/votes and cannot equivocate.

It has been shown that this protocol does not satisfy universal verifiability with a corrupted ballot box, because of the clash attack: all voters will confirm the correct recording of their vote to the auditor while the auditor will count a result that does not reflect the voters' intention. In our model, we also capture this attack on correctness.

2.4.2 The Selene protocol

Selene is an e-voting protocol described in [109] and chapter 1. This protocol is much more complex than the CNRS protocol as it involves many cryptographic primitives to ensure stronger security properties than the CNRS protocol. As we already introduced Selene in the chapter 1, we won't provide all details here.

Known attacks In [29], the authors proved Vote-Privacy of a simplified version of the protocol. They found an attack, already discussed in [109], on Receipt-Freeness (with respect to definition 4) if one of the two voters is dishonest and colludes with the attacker. In their model, two trackers were created and one of them was always known by the attacker. The honest voter faked his tracker for the coercer, leading to an attack.

2.5 Models

For Verifiability and Vote-Privacy properties, the proofs are computable on a standard laptop.

For Receipt-Freeness and Coercion-Resistance, however, we had to use additional resources on a high performance server, using up to 500GB of RAM and 28 Intel Xeon cores. Furthermore, we simplified slightly our models to obtain termination. The description of the modifications are given with the formal properties.

2.5.1 General setting

2.5.1.1 Channel rules

The rules between participants in our models are detailed on figure 2.2. As in [91], we define by $\bullet \rightarrow$ an *authentic* channel, that means the adversary cannot modify the messages or their sender, but he can access it. A *confidential* channel, written $\rightarrow \bullet$ means that only the intended receiver can read the message but everyone, including the adversary, can send a message on this channel. The confidential channel allows us to define a *secure* channel $\bullet \rightarrow \bullet$, which is both confidential and authentic. Hence, an adversary can neither modify nor learn messages that are sent over a secure channel. He cannot copy the content of a message, as the message will be “hidden” behind a persistent Fact. More details about TAMARIN channel rules can be found on the manual web page [16].

We will use authentic, secure, and we also define *untappable* channels that we denote with \boxRightarrow . We model these channels with a linear Fact, the content is not accessible to the attacker. Those Facts can be consumed only once, preventing any replay from the attacker.

2.5.1.2 Attacker

We consider a Dolev-Yao adversary [42] who controls the network by listening to all messages sent over the network (on the standard input and output), building new messages and sending messages. The adversary has access to all channels from compromised agents.

2.5.2 The CNRS protocol

2.5.2.1 Trust assumptions

The boardroom protocol is simple. We used the same channel rules that were described in [13].

Some attacks are already known and a dishonest ballot box will undermine Vote-Privacy. For privacy properties, we assume that both the ballot box and the screen are honest, and we developed a model with 2 honest voters. For verifiability properties, we modelled a malicious ballot box, giving the same tracker to two voters.

2.5.2.2 Model

We will use the built-in *multiset* available in TAMARIN to concatenate the list of ballots on the screen.

The protocol does not use any cryptography, we use a simple signature consisting of the functions $\text{pair}(\cdot, \cdot)$, $\text{fst}(\cdot)$ and $\text{snd}(\cdot)$ together with the equations:

$$\text{fst}(\text{pair}(x, y)) = x, \text{snd}(\text{pair}(x, y)) = y$$

We consider a Ballot Box BB , a screen S and voters V_i . The voters and their devices are considered as the same. The values of all identifies are public as well as the candidates.

An overview of the model is given in figure 2.3. Note that the symbol \sum is used as the multiset union.

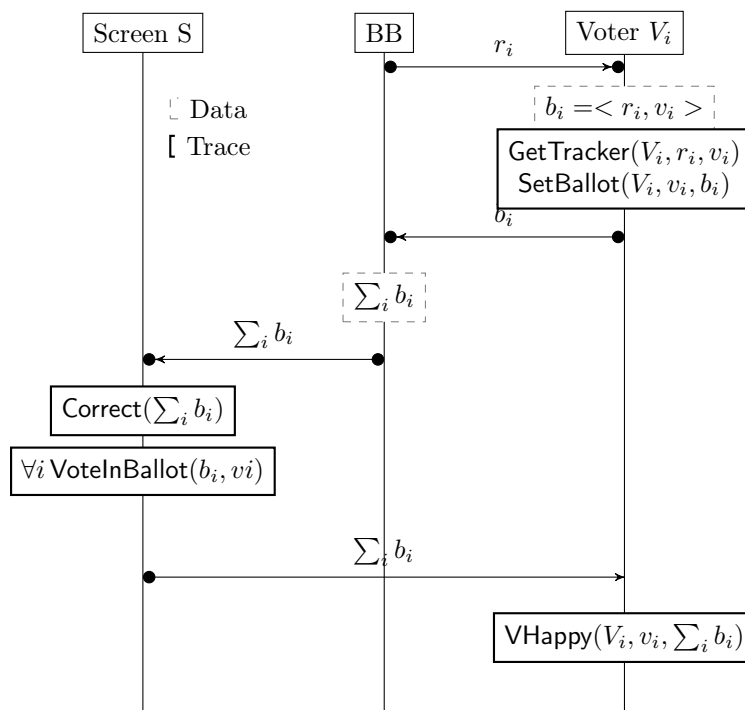


Figure 2.3: The boardroom voting model.

2.5.2.3 Formal properties

Privacy The privacy properties are verified with Observational Equivalence using the TAMARIN diff operator. For Vote-Privacy, we simply swap votes between two voters when defining their vote intention as follows:

```
Voter($V1, diff($vote1, $vote2))
Voter($V2, diff($vote2, $vote1))
```

For Receipt-Freeness, even on such a simple protocol, the amount of resources needed for the proofs increased and TAMARIN could not terminate on a server with 28 Intel Xeon cores and 500GB of RAM within 24 hours. Hence, we used a variant of the protocol also described in [13]: the trackers are generated by the voters and we assume that there is no collision between trackers. Also, as the ballot box is not malicious, we merged it with the screen, meaning that the tally is given by the ballot box to the voters directly. Voters swapped their vote, and they output their vote and tracker to the adversary in the end. As mentioned in the previous section, the protocol is not receipt-free because the tracker is known before voting. To model our definition, we need voters to know other trackers. We test our definition by revealing the trackers and votes at the end of the protocol, when the tally is known. Our definition states that one voter tells the truth and the other lies, and vice versa. The adversary should not be able to tell the difference.

The protocol is of course not coercion-resistant as the tracker is generated by the voter at voting time. A coercer could directly request a voter to cast a vote with a tracking number that he can verify on the screen.

Verifiability The verifiability properties are defined as TAMARIN lemmas. We labeled our model's rules with events corresponding to the different predicates that we defined in section 2.3. An overview of these events is given in figure 2.3 (traces with the thick border).

The lemma for individual verifiability is defined as follows:

```
lemma ind_verif: all-traces
  " All V vote ni1 vote1 ni2 vote2 #i .
    VHappy(V, vote, <ni1, vote1> + <ni2, vote2>)@i
  =>
    Ex t #j . Vote(V, vote) @j & GetTracker(V, t, vote) @i
      & ( ( t = ni1 ) | ( t = ni2 ) ) "
```

This formula expresses that a voter V is happy with the ballot box containing $\langle ni1, vote1 \rangle + \langle ni2, vote2 \rangle$, then there exists a tracker t that the voter has received and that is on the bulletin board and linked to his vote intention $vote$.

For correctness and universal verifiability of the ballot box, we defined the following two lemmas following our definitions given earlier:

```
lemma Correctness: all-traces
  " ( All vote1 vote2 b #i . VoteInBallot(b, vote1) @i & VoteInBallot(b,
    vote2) @i
  => vote1 = vote2 )
  &
  ( All vote b #i . VoteInBallot(b, vote) @i
  => Ex V rand #j . SetBallot(V, vote, <rand, vote>) @j ) & #j < #i "
```

```
lemma uni_verif: all-traces
  " All r1 vote1 r2 vote2 #i .
```

$$\begin{aligned} & \text{Correct}(\langle r1, \text{vote1} \rangle + \langle r2, \text{vote2} \rangle) @i \\ \implies & \text{Ex } \#j. \text{VoteInBallot}(\langle r1, \text{vote1} \rangle, \text{vote1}) @j \ \& \ \text{VoteInBallot}(\langle r2, \\ & \text{vote2} \rangle, \text{vote2}) @j'' \end{aligned}$$

The first formula states the correctness of the ballots, by verifying that a given ballot (of the form $\langle r, \text{vote} \rangle$) cannot contain two different votes; and that a given ballot must have been set by a voter earlier in the process.

The second formula (implemented here for two voters) verifies that a bulletin board is correct if we can find the corresponding votes in ballots.

As we consider honest voters in our model, we do not model the predicate `CleanBB`.

2.5.3 Selene

2.5.3.1 Trust assumptions and limitations

The TAMARIN prover has some limitations that we will detail below. It constrained us to make the following simplifications over our models.

First, the voters and their devices are considered as the same. We assume that the two voters are honest (necessary to test our privacy definitions with observational equivalence) and can cast their intended vote without being monitored by a coercer. For Coercion-Resistance, the coercer gives instructions to one voter. The Tally Teller and Election Authority are honest. One Tracker Teller gives his alpha-terms away to the adversary. The randomness used to encrypt the votes remains private and unknown to the adversary, otherwise he could learn the votes of voters during the vote casting phase (as we use an authentic channel for vote casting). To avoid this, we could use a re-encryption trusted party through a secure channel before publishing the encrypted votes, but we decided to go for a simplification of the model and keep the randomness secret.

The advantage of Selene is that the voter can lie to a coercer by giving a fake tracker. Here, we assume an untappable channel between the voters and the tracker tellers to ensure Receipt-Freeness. However, a standard channel is enough for Vote-Privacy as the adversary cannot use the alpha-terms without knowing the private trapdoor key of the voters. We assume that the adversary is not a voter, but an external observer, meaning that he has no information regarding any tracking number.

One of our Tracker Teller is corrupted, however, the adversary could ask for the other alpha-term part from the honest Teller to the voters. In the real protocol it would be possible to give a fake partial alpha-term to cheat the adversary, but in our model and with the multiset operator as a combination operator for alpha-terms, it is not possible to simulate this. We always assume that the adversary requests the combined alpha-terms.

Finally, for Receipt-Freeness and Coercion-Resistance, the model has been simplified due to the memory required by the tool. Using the assumptions above, the execution could not terminate and had a memory error using a server with 28 cores

and 500GB of RAM. We will detail the modifications below in the paragraph 2.5.3.4.

2.5.3.2 Equational Theories

This protocol has more complex cryptography and we use the following signature:

$$\sum_{\text{Selene}} = \{\text{pair}, \text{fst}, \text{snd}, \text{open}, \text{commit}, \text{fake}, \text{cp}, \text{dcp}\}$$

where the functions and equations are detailed in the following paragraphs.

Trapdoor commitments equations To be able to fake the data, we need a theory that includes a trapdoor commitment scheme for the trackers' notification. In [43], authors develop an equational theory for TAMARIN to support additional convergent theories, i.e. that are confluent and terminating. This allows to define a new convergent theory for the trapdoor commitment scheme, that is defined by the functions `open`, `commit`, `fake` and by the equation given in figure 2.4.

The randomness r used in the equations will be split between tracker Tellers using the multiset operator $+$.

$$\begin{aligned} \text{open}(\text{commit}(m, r, td), r) &= m \\ \text{commit}(m_2, \text{fake}(m_1, r, td, m_2), td) &= \text{commit}(m_1, r, td) \\ \text{open}(\text{commit}(m_1), r, rd), \text{fake}(m_1, r, td, m_2) &= m_2 \\ \text{fake}(m_1, \text{fake}(m, r, td, m_1), td, m_2) &= \text{fake}(m, r, td, m_2) \end{aligned}$$

Figure 2.4: Trapdoor commitment equations.

Encryption equations We define a probabilistic asymmetric encryption scheme:

$$\text{dcp}(\text{cp}(m, r, \text{pk}(sk)), sk) = m$$

where r is the randomness used for the encryption, sk is the secret key and $\text{pk}(sk)$ is the corresponding public key. This will let us send encrypted votes over the network without the adversary being able to know which candidate has been chosen⁴. For this, we assume the randomness remains secret.

⁴The candidates are public knowledge and the adversary can use construction rules such as encryption to build messages. Without randomness, the adversary would be able to build encrypted votes for both candidates and find an equality with the sent vote.

Distributed Trust To implement Selene in the symbolic model, we split the randomness used for the trapdoor commitments between two authorities as described in section 2.4. To share the randomness, we considered several operators available in TAMARIN: the XOR operator \oplus [44], the multiset operator $+$, or create a function of two parameters as a new user equational theory.

The XOR operation has associativity and commutativity properties that are already defined in TAMARIN. The cancellation property has been added. In [44], the authors showed that the new equational theory for XOR is compatible with previous equational theories implemented, such as Diffie-Hellmann exponentiation [113], bilinear pairing and multisets, and with any user-defined equational theory. However, the proofs are slower for verifiability with this operator compared to the other options above, and we could not obtain termination with observational equivalence.

Alternatively, we could have created a function of two parameters with no specification associated. In this case, TAMARIN creates a term from the 2 inputs and no specific property is applied to it. The commutativity is not specified hence we must indicate that the first parameter is the one delivered by the Teller 1, and the second by the Teller 2.

Finally, we chose to use the multiset operator $+$, as it already has associativity and commutativity properties. It removes the specification on the order of parameters we had with a function of 2 parameters. The voter will need both terms to open the commitment and retrieve the tracking number. One Tracker Teller will output his alpha-term to the adversary.

Mixnets and zero-knowledge proofs In Selene, mix-nets (e.g. [127]) are used several times to randomise the data. A mixnet permutes and re-encrypts a set of input ciphertexts to ensure privacy.

To lower the complexity of the model and to simulate a mix between two terms, we will use again the multiset operator $+$ to model the reordering of the data. We write $\text{Out}(t1 + t2)$ as an output of a rule R1 and $\text{In}(t3 + t4)$ as an input of a rule R2, where $t1, t2, t3, t4$ are terms. We can find a positive match between those rules with both substitutions $\{t1 \rightarrow t3, t2 \rightarrow t4\}$ or $\{t1 \rightarrow t4, t2 \rightarrow t3\}$.

Also, with the aim of simplifying the model, we don't model zero-knowledge proofs to reduce complexity.

2.5.3.3 Model

An overview of the model is given in figure 2.5. The data given to the adversary is indicated in the figure with the **Out** keyword. We also indicate the predicates used for verifiability definitions. This figure shows an overview of the model used in Vote-Privacy and Verifiability. The paragraph 2.5.3.4 gives the simplified assumptions for Receipt-Freeness and Coercion-Resistance.

The key generation is suppressed in figure 6.2. Fresh keys are created and their

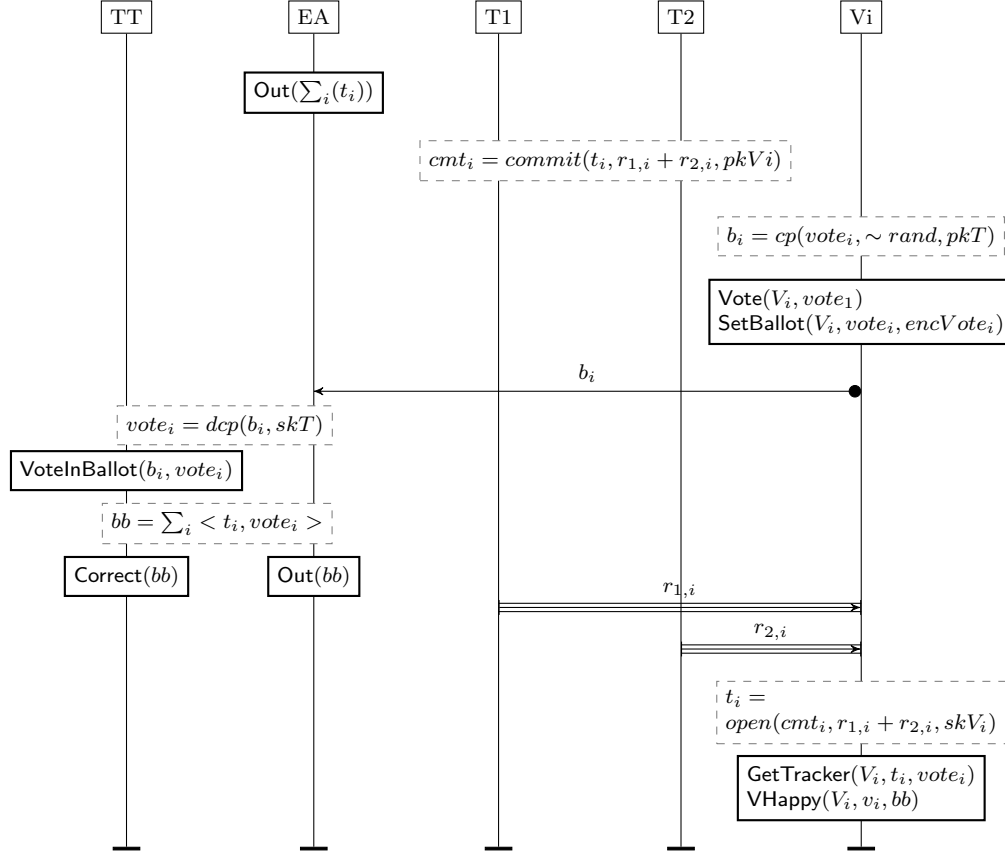


Figure 2.5: The Selene model.

associated public keys are output to the adversary. A setup rule initialises the keys and the states of the entities.

The state of each agent is initialised with a Fact that contains their knowledge: the Tally Teller (TT) is initialised with the election’s public and private keys; the voters (Vi) are initialised with their vote intention, private keys and the election’s public key; the Tracker Tellers (T1 and T2) are initialised with the voters’ public keys; and the Election Authority (EA) generates the tracking numbers and outputs them over the network.

After the setup, the first step is the generation of the tracking number commitments. This is performed by both Tracker Tellers and the EA. The Trackers Tellers generate the alpha-terms as two nonces combined with the multiset operator; T1 is colluding with the adversary and output his nonces. The EA gives the two tracking numbers, and we use the equations given in figure 2.4 to compute the commitments. The Tracker Tellers store their alpha-terms and the EA stores the commitments.

Once the commitments have been computed, the voters can cast their vote. The voting rule is defined as follows: each voter creates an encryption of the candidate choice and outputs the encryption over the network on an authentic channel. Further,

the predicate `SetBallot` is instantiated and it is used in the Correctness lemma above.

The next step is to wait for all votes to reach the EA. A synchronisation rule achieves this, and it also means that all voters must vote. Abstention is not allowed hence we don't test for forced abstention attacks (i.e. a coercer preventing a voter to vote).

After this synchronisation phase, the Tally Teller and the Election Authority collaborate to compute the tally and to output results on the bulletin board. The Tally Teller decrypts the encrypted votes and the Election Authority publishes the pairs (tracker,vote) on the Bulletin Board (removing the link to the voter). The predicate `VoteInBallot` is instantiated at that stage, indicating that a decrypted vote *vote* was contained in a received ballot *b*.

The predicate `Correct` is also instantiated as the bulletin board is output over the network hence available for verification.

When the tally is published, the Tracker Tellers can send the alpha-terms to the voters through an untappable channel, necessary for Receipt-Freeness.

From the alpha-terms, the voters can compute their tracking number with the commitment β and their private key. Then they can verify their vote. The predicate `VHappy` is instantiated here with the voter's intended vote and the bulletin board.

2.5.3.4 Formal properties

Privacy The privacy properties are verified with observational equivalence with the `diff` operator.

For Vote-Privacy, as we did for the CNRS scheme, we simply swap the votes between two voters.

For Receipt-Freeness and Coercion-Resistance, as for the CNRS protocol, the complexity of the model leads to a space explosion. Hence, we simplified the protocol as follows: we removed the operator `+` for the multiset, hence we modelled one tracker teller. Furthermore, without the multiset operator, we removed the output of the final tally. We also merged the tally teller and the election authority to reduce the model's size. Similar simplifications were done in [29] in the elaborated version of Selene, where the multiset operator and the tally output have been removed for similar reasons.

In our definitions, both voters must give their private data to the adversary, but only one voter at a time is faking it. We use the `diff` operator as follows:

```
// Voter 1
Out(diff(<vote2,~ltkV,cmt,fake(ni,r1,~ltkV,ni2),ni2>, <vote,~ltkV,cmt
    ,fake(ni,r1,~ltkV,ni),ni>))

// Voter 2
Out(diff(<vote2,~ltkV,cmt2,fake(ni2,r1,~ltkV,ni2),ni2>, <vote,~ltkV,
    cmt2,fake(ni2,r1,~ltkV,ni),ni>))
```

where ni is V1’s true tracker, $ni2$ is V2’s true tracker, $\sim ltkV1$ and $\sim ltkV2$ are the private keys of resp. V1 and V2, cmt and $cmt2$ are the commitments of resp. V1 and V2, $vote$ is the true vote of V1 and $vote2$ is the true vote of V2, and we use the `fake` function to compute the related alpha-term. Note that in both executions, both voters will output the same vote to the adversary.

For Coercion-Resistance, we give an additional instruction to Voter 1. Again with the `diff` operator, we make the voter follow or not the instruction when computing his ballot. In the first case, the instruction is just a candidate name and the voter computes his ballot as follows:

```
encVote = cp(diff(vote, I), ~rand, pkT)
```

where $vote$ is the voter’s true vote and I is the attacker’s instruction.

In the second case, in which Selene will not be coercion-resistant, the instruction is an encryption of a vote, the voter does not know its content. The `diff` operator is used in the output, when the voter sends his ballot to the election authority:

```
Send_to_EA('V1', diff(encVote, coercerVote))
```

where $encVote$ is the voter’s ballot and $coercerVote$ is the coercer’s instruction.

Verifiability The verifiability lemmas are the same that we described for the boardroom voting scheme. The only difference is on the ballot’s definition: a ballot b is defined by an encryption of the $vote$. In the CNRS scheme, it was defined as a pair $\langle rand, vote \rangle$.

2.6 Results and discussion

We start by giving a few remarks regarding our models. The models consider two voters only. In [12], the authors gave conditions under which three voters are enough to find all attacks on Vote-Privacy and Receipt-Freeness, specifically two honest voters and one dishonest voter. Here, we were limited by the tool, and we could not model a third voter and have termination⁵. Hence, we needed to simplify our assumptions. Our two voters are honest and the attacker is not a voter but an external observer of the protocol. In [45], the authors consider two voters as well but one can be dishonest. To test our privacy definitions, we need two honest voters (who lie to the attacker).

Some attacks are not covered by our model, in particular “fault” or “randomization” attacks [39]: some instructions provided by a coercer to a voter would lead to an observable difference in the execution of the protocol because it would interrupt

⁵We contacted the development team regarding this issue and an bug is opened here at submission time: <https://github.com/tamarin-prover/tamarin-prover/issues/368>

Results	Privacy					Verifiability		
	VP	TD	RF	CM	CR	Correct.	IND-V	UNI-V
CNRS (std)	✓	✓	✗	✗	✗	✓	✓	✓
CNRS (mal. BB)	✗ ([13])	NA	NA	NA	NA	✗	✓	✓
Selene (std)	✓	✓	✓	✓	✗	✓	✓	✓
Selene (mal. TT)	✓	NA	NA	NA	NA	✓	✗	✓

Figure 2.6: Synthesis of the results. A ✓ indicates a proof while ✗ indicates an attack. The acronyms are: Vote-Privacy (VP), Tracker Deniability (TD), Receipt-Freeness (RF), Coercion Mitigation (CM), Coercion-Resistance (CR). Non-applicable (NA) is indicated when a test could not be run. The attack on VP for CNRS with malicious BB was also known from [13]

its correct execution. For Selene, we modelled two behaviours from the coercer: on one side, his instruction is a candidate’s name and on the other side, the instruction is an encrypted vote to cast. If a coercer tries to elaborate an attack with a ballot that is not well-formed, it could cause an error in the execution.

Also we did not model a copy attack: the attacker copies an encrypted vote and re-encrypt it as his own, and will learn a voter’s vote when verifying his vote. In the original protocol, copy attacks are prevented by a proof of plaintext knowledge of the vote sent at vote casting by the voters.

A synthesis of all proofs and attacks can be seen in figure 2.6.

2.6.1 The CNRS protocol

2.6.1.1 Privacy

We followed the standard definition of Delaune et al. [39] to prove Vote-Privacy. With an honest ballot box, the CNRS protocol satisfies the definition 3 of Vote-Privacy. In the paper [13], authors described a few attacks on vote privacy with a malicious ballot box that we did not model here, as a malicious ballot box could anyway leak the relation voter-tracker.

For Receipt-Freeness, we run our proof on the simplified version of the protocol. The protocol is not receipt-free as the trackers are known before vote submission, voters can reveal their tracker before casting their vote. Yet, they need to know other trackers in order to lie about their own. However, we proved in TAMARIN that the protocol has Tracker Deniability, i.e. voters can fake their tracker when the tally is published on the screen.

The protocol is not coercion-resistant. If we limit the coercer to just give a voting instruction (i.e. a candidate’s name), the voter can claim another tracker on the bulletin board. In this simple protocol, especially with the variant where the tracker is created by the voter, a coercer would, however, also give a tracker to be

submitted by the voter.

2.6.1.2 Verifiability

The lemmas given in the previous section, modelling our verifiability definitions, have been proven under our trust assumptions. As discussed in [13], one attack on verifiability and in particular correctness of the tally has been found in case the ballot box is malicious. We modelled this attack as well where the ballot box provides the same tracker to all voters (assuming they vote for the same candidate) and adds another vote to the tally. With this assumption, our **Correctness** lemma is not verified.

2.6.2 Selene

2.6.2.1 Privacy

Vote-Privacy In Tamarin, to prove Vote-Privacy, we followed the standard definition (i.e. Delaune et al. [39]), given in section 2.3 definition 3 and we show an Observational Equivalence between two executions of the protocol.

With one corrupted tracker teller, Selene satisfies the definition 3 of Vote-Privacy.

The intuition behind the privacy of Selene, according to [109], is as follows: the encryption of the vote ensures the secrecy of the vote choice. The two mixnet shuffles involving encrypted tracking numbers implies that an adversary would need to control the mixnet to learn the association between a tracker and a voter. Furthermore, the commitments β_i are perfectly hiding. Even if the alpha-terms are revealed, they won't give enough information to the adversary as it is just a part of an ElGamal encryption of the tracker, and an adversary would still need the private trapdoor key of the voter to break privacy.

Receipt-Freeness The information given to the voter to verify the vote is the tracking number. As the voter can fake it, we assume that the notification of the alpha-term must remain private to achieve Receipt-Freeness. In addition, we assume that the randomness used in the encryption of the votes is never revealed and kept secret by the voter. If not, the adversary would be able to look for the encrypted vote⁶.

In [29], the authors have detected an attack on the standard definition of Receipt-Freeness given in section 2.3 definition 4, if the coercer knows one of the tracking numbers, i.e. when the other voter is the attacker. Receipt-Freeness is possible only in the extended version of Selene where no collision is possible between tracking numbers (as mentioned in the original paper).

⁶Without randomised encryption, the adversary can build an encryption of the candidate with the election key and modify or compare the ballots with his own terms.

Our new definition of Receipt-Freeness aims to show that an adversary cannot be convinced by a receipt as he cannot tell which voter is lying. Both voters reveal their data and both claim the same vote on the bulletin board. As mentioned in the previous section, we simplified our model by removing the shuffling through multiset, which contributes to memory issues during the execution. Hence, our model was executed without the tally output.

Under this constraint, Selene satisfies the definition 5 of Receipt-Freeness.

Coercion-Resistance To test our definition of Coercion-Resistance, we split the voting phase into two different rules: voter V1 receives instruction from the adversary and voter V2 can vote as intended. First, we assume that the coercer does not have control over the voter during the vote casting phase. Like Receipt-Freeness, both voters output (fake) data to the adversary: on the left-hand side, V1 outputs fake data and shows V2’s tracker for candidate B; on the right-hand side, V1 follows the instruction and show the real data to the coercer, while V2 claims the same tracking number.

In the first case considered, voter V1 receives an instruction which is an encrypted vote. Under this assumption and with the simple encryption modelled, Selene does not satisfy Coercion-Resistance.

If we limit the coercer to simply give a candidate’s name, then Selene satisfies Coercion-Mitigation.

2.6.2.2 Verifiability

The lemmas modelling our verifiability definitions have been proven under our trust assumptions. We detail the results below.

Individual Verifiability To prove individual verifiability, we need to prove that voters have cast their intended vote and that the delivered tracking number shows a corresponding recorded vote. The lemma `ind_verif` has been proven with TAMARIN. Selene satisfies the definition 1 of Individual Verifiability.

However, we found an obvious attack in the special case where the coerced Tracker Teller sends fake alpha-terms to the voters. In the model, this is done by giving instruction to one of the Tellers when notifying the voters with the alpha-terms. The computed tracker is not on the bulletin board, leading to an attack on the lemma `ind_verif`. This is not a stealthy attack as it would raise an alert to the authorities.

Universal Verifiability Our definition of Universal Verifiability is based on the correctness of the bulletin board, and on the correct counting of votes.

In our model, the correctness of the bulletin board relies on the correctness of ballots that can be sent only by honest voters. Unlike the CNRS protocol, the ballot

does not contain the tracker, and the same tracker cannot be assigned twice. The link between tracker and voter is unknown, and the uniqueness of the trackers is verifiable in the protocol.

The formal properties `Correctness` and `uni_verif` have been proven with TAMARIN. From these lemmas, it follows that the votes are correctly tallied.

2.7 Related Work

Tamarin models of voting protocols

Bruni et al. proposed a symbolic model of Selene with TAMARIN in [29]. They proved Ballot-Secrecy and Receipt-Freeness of the protocol. They also used the trapdoor commitments equations initially developed in [43] to model the tracking number commitments in Selene. However, they proposed a model for a very simplified version of the scheme, with only one Teller and assume trust in the Teller and in the Election Authority. Basically, all data was output through an untappable channel (linear facts), the adversary had no chance to tamper with the data, hence attack the model. Furthermore, in our model, we have introduced another Tracker Teller for distributed trust on the commitments generation and a Tally Teller for the decryption and tally of votes. We also considered a judge for verifiability properties.

Basin et al. [18] proposed a new protocol for random sample voting, as well as the security proofs in TAMARIN. They proved Receipt-Freeness and End-to-End Verifiability. The distinction between the voter and the device in use has been made in the model, in our implementation we consider the device and voter as the same entity.

In other papers (e.g. [43]), voting protocols such as FOO [51] or Okamoto [97] protocols are modelled in TAMARIN as proofs of concept for TAMARIN extensions.

Security properties

Cortier et al. [37] propose a survey of existing verifiability definitions in the symbolic and the computational model. They propose a general definition of Verifiability. We focused on the definition given in [36] which provide definitions for individual and universal verifiability, as well as end-to-end verifiability considering an additional condition (*no-clash* property). Their definitions have been tested on Helios. The individual verifiability definition checks that a ballot that has been computed by a voter appears on the Bulletin Board. The universal verifiability definition checks the correctness of the bulletin board, and the correct counting of ballots. We slightly modified those definitions to take into account the trackers that are used for individual verifiability in tracker-based schemes. It also had an impact on the content of the bulletin board and we define correctness of the tally.

Kremer et al. [71] define Verifiability in the symbolic model, and consider three aspects of verifiability: individual, universal and eligibility properties. We have not explored eligibility verifiability (even though the election authority accepts votes sent by the expected voters in our model), the definition of individual verifiability does not take into account tracking numbers but ballot checking (as in Cortier et al. [36]).

In [75], authors provided a formal definition in the computational model for Privacy and Verifiability. Verifiability definitions are based on a goal the protocol must achieve in order to evaluate the mentioned properties. The concepts are related to Accountability (also in [77, 76]) where a judge assesses the soundness and completeness of a protocol, and the authors have proved that individual and universal verifiability are not enough to satisfy end-to-end verifiability in some cases. The examples given used a dishonest bulletin board.

2.8 Conclusion and Future Work

In this chapter, we looked at tracker-based verifiable voting protocols. We proposed new definitions in the symbolic model for Verifiability properties (individual and universal) and for Receipt-Freeness, Coercion-Mitigation and Coercion-Resistance. We have developed models for the CNRS boardroom protocol and for the e-voting protocol Selene, and we tested our definitions and discussed their limitations. As future direction, we could work on new formal definitions for other concepts in voting, including eligibility verifiability and accountability. Also, we could work on pushing the boundaries of Tamarin and other tools to allow verification of more precise models including an equational theory for zero-knowledge proofs and signatures for eligibility.

Chapter 3

Design and implementation of a usable system

This chapter presents a user-centred approach that we used to design our voting applications. In particular, we designed a mobile application and a web application for the Selene e-voting protocol. The resulting interfaces were tested in three user studies. The first one was conducted with 38 participants, and studied the impact of the verification phase and of displaying security mechanisms. The evaluation was performed with the User Experience Questionnaire and the User Psychological Needs Questionnaire. The second was conducted with 24 participants and an evaluation of the usability was done with the System Usability Scale. The third study was conducted with 300 participants and used a web application in an online study. The evaluation of the web app was performed with the User Experience Questionnaire and the System Usability Scale. Additional elements were evaluated during those three user studies. In the chapters 4 and 5, we will go further by analysing: 1) the mental models revealed during interviews, 2) the mental models revealed during drawing sessions and 3) the understanding of participants through a game. This chapter aims to give a first taste of the performance of Selene among voters.

3.1 Introduction

Voting protocols are carefully designed to satisfy certain security properties, most importantly Privacy and End-to-End (E2E) Verifiability. Some notable privacy properties are ballot-secrecy, receipt-freeness and coercion-resistance. E2E-verifiability is usually separated into the votes being cast-as-intended, recorded-as-cast, and tallied-as-recorded.

E2E-verifiable schemes often require voters to handle encrypted ballots [10, 22, 66]. As a reminder, the Selene e-voting protocol [109] has been designed in order to hide the cryptographic operations from the voter. As described in the introduction (see chapter 1), each voter is assigned a private tracking number, which lets them

verify that their vote has been correctly included in the tally. In the setup phase, a unique tracker number is secretly associated with each voter and cryptographically committed to the bulletin board. At the end of the election, the votes are posted on a public bulletin board along with the associated tracking numbers. To avoid coercion, the voters are notified of their tracking number only after the vote/tracker pairs have been published. This gives coerced voters the opportunity to identify a tracker that points to the coercer’s candidate that they can then claim is theirs. The hypothesis is that this procedure is more intuitive, transparent and easy-to-use than the usual E2E verifiable schemes, where voters should check the correct encryption of their vote (e.g. Benaloh challenge [21]) and then presence of this encryption on the bulletin board.

User tests on voting protocols have shown that schemes that provide security often have usability issues [7, 88, 74]. According to [121], *usability* measures the effectiveness, efficiency, and satisfaction of a software in a specified context of use. Effectiveness is the accuracy and completeness with which the users achieve their goals. Efficiency represents the resources expended for effectiveness. Satisfaction is defined by the comfort and acceptability of use. In the papers [7, 88, 74], the effectiveness of vote casting, that is the ability to cast successfully a vote, has been at most 81.25% [88]. In addition, the meaning of the verification phase is not always well understood, which can lead to voters not performing the verification task or unintentionally aborting the task. Ensuring system usability is further complicated by the fact that elections rarely occur and voters are expected to understand and use a system they are not familiar with.

User Experience is defined as “a person’s perceptions and responses that result from the use or anticipated use of a product, system or service” [122]. It considers emotions, psychological needs and temporal aspects of the interaction between the system and the user, and can measure a person’s perceptions of system qualities such as attractiveness, ease of use and novelty, in addition to usability aspects. To improve the user experience, user-centred methodologies have been developed in order to include the final users in the development of a product [96, 95, 35, 78]. We will describe the process in detail below.

In this chapter, we present the first development of a prototype interface for smartphones for the Selene e-voting protocol, following a user-centered design process [96, 95, 35] called User Experience Design (UXD) Process [78]. We will discuss the impact of our implementation choices on the protocol’s security. In section 3.3, we will give the methodology used to create the interface. We will provide the details of our interface in section 3.4. Following the same process, we designed a web interface for Selene, including the coercion mitigation mechanism that we did not implement in the mobile application.

Finally, we have done two user studies with two versions of the mobile application and one user study with the web application: in section 3.6, we will give the first usability and user experience results.

In the next chapters, we will focus on the voters’ understanding of the protocol using the same applications, exploring their mental models and their trust in the system.

3.2 Related Work

In [41], Dodier-Lazaro et al. highlighted the importance of taking the user into consideration in a security implementation. They argue that a “value-sensitive design”, meaning considering what users value, can help to provide more suitable interfaces to the users. Users usually value security but the available tools don’t let them reach their primary goals easily.

Similarly, Bartsch and Sasse [15] have shown that employees care about security but the policies in use were not compliant with their needs. As a consequence, employees deployed their own security measures. In [69], this is what authors called “shadow security”: when employees cannot comply with the security policy of their organisation, they create new mechanisms that fits their needs. However, the authors point out that these policies might be less secure than the official one.

In the context of voting, voters value privacy and have security concerns (see chapter 4). However, the security features proposed in the existing e-voting schemes are often confusing. Indeed, the usability of e-voting protocol has been investigated in many previous works confirming its importance [7, 50, 48, 73, 86, 126, 88]. A bad usability at vote casting can easily lead to errors, those will have an impact on the final result hence can impact the integrity, see [7] for an example with the Helios protocol [10]. A bad usability at vote verification can lead to undetected mistakes, false alerts, or even the impossibility to verify correctly their vote, see [50] for an example with the Norwegian e-voting scheme [54].

About the study of a coercion-mitigation feature, to our knowledge only Neto et al. in [94] have run a user study on the JCJ protocol [66] and have shown the difficulties of the required tasks.

3.3 Following a User-Centred Approach

3.3.1 Reminder of the protocol’s steps

3.3.1.1 Protocol overview

Most verifiable voting schemes involve voters seeing and handling cryptographic data which can lead to errors or misuses [7]. As a reminder, Selene [109] is an e-voting protocol that has been designed to provide an easier and more intuitive verification procedure for voters. It lets the voters verify that their votes have been included in the tally using a unique tracking number. To protect against coercion threats, i.e. achieve receipt-freeness and coercion mitigation, voters first learn their private

tracking number after the votes have been posted. Selene uses ElGamal encryption, that is homomorphic and can act as a commitment scheme. An ElGamal encryption is a pair (α, β) . For a given voter, the tracking number is encrypted using ElGamal and the β -term is published at the beginning of the election. The α -term is kept secret and shared between several entities called Tellers. After the tally has been published, the α -terms are sent to the voter, who can decrypt the tracking number with her key.

3.3.1.2 Voter experience

As in Selene [109] we assume that the voter already has the cryptographic key material needed for the protocol, i.e. we skip the key setup phase. We assume that the keys are accessible through the application. The voting ceremony without coercion is as follows:

- (1) The voter receives an invitation to vote.
- (2) The voter makes her vote choice in the provided application, encrypts and signs the vote, and sends it to the Election Server.
- (3) (*Optional*) The voter later receives an invitation to visit the bulletin board when votes and tracking numbers are published.
- (4) The voter receives the α -term, and can retrieve the tracking number to verify her vote.

The third step is optional as it is only necessary if the voter is being coerced. For our first implementation (mobile app), we will assume that no coercion is happening and thus the third step is not available. In our second implementation (web app), the coercion mitigation mechanism will be available. We also simplified the fourth step by automating the α retrieval and tracker computation. The detailed methodologies deployed during the user tests are described in section 3.6.

3.3.2 A user-oriented approach

We followed a user-centred design methodology, which has originally been described by Norman [96] and then detailed as a design process [35, 95]. In particular, we followed the User eXperience Design (UXD) process by Lallemand et al. [78]. It consists of five steps which are *planning*, *exploration*, followed by an iterative process (shown in figure 3.1) with *ideation*, *generation*, *evaluation*. In the following we will detail the process for our first implementation of the Selene mobile application, a similar process was followed for the web application.

The exploration phase includes a collection of user needs, and can be done using various methods, such as a literature review of previous studies, interviews, focus groups or observations. In our case, we discussed the voting issues mentioned in several papers [7, 8, 57, 61, 68, 88] during meetings with Human-Computer Interaction (HCI) experts who helped us develop and test prototypes of the e-voting application in a user-centred process in close collaboration.

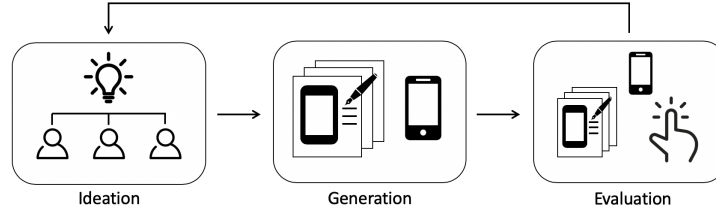


Figure 3.1: Iterative process.

Then we focused on the iterative process: we worked together with the HCI experts for the generation of ideas for the design during group sessions with up to ten group members. We then came up with the concept for a mobile application, that will have both features of voting and verification. We developed a first version that is a low-fidelity paper prototype. Then, we evaluated this version with HCI experts. The received feedback on design and understanding of security allowed us to iterate and develop a second paper prototype, that was tested with both HCI and security experts. More details about the low-fidelity prototypes will be given in the next section. The final iteration was a high-fidelity software version that will be described in detail in section 3.6.

A particular challenge for the user experience of Selene is that a certain level of understanding might be necessary to achieve fulfillment of privacy needs: as secure and easy-to-use as the application might be, displaying the plaintext cast vote to the voter after election could seem insecure. Further, the verification phase is not commonly used in standard elections and is largely unknown to users. Following the UXD process, we tried to anticipate users’ expectations on voting and their questions on such a protocol, hence we designed an interface that, hopefully, is more understandable.

3.3.3 Cryptography

Selene uses several security mechanisms, however, the cryptographic details can be hidden from the user during the voting and verification phases. As mentioned above, we assume here that the voter doesn’t have to configure her device with her secret keys or explicitly handle other keys such as the election public key. As a reminder, in the original paper [109], the details concerning the device’s configuration were not provided.

The tracking number retrieval (fourth step in the voting experience) is simplified here and the voter simply has to click on a single button to use the α -term, to download the β -term and to compute the tracking number. It will highlight the result on the bulletin board, displayed in-app, automatically. The voter’s trapdoor key won’t be explicitly manipulated by the voter and it is embedded in the phone, unlocked by the voter’s credentials.

The other primitives used in Selene do not require direct interaction with voters (e.g. zero-knowledge proofs). Hence, these are not mentioned in the conducted user experience test. In a real election implementation all of this can be public and verifiable by observers and interested voters.

We emphasise that our implementation is a first step that provides a user interface, in order to answer our research questions on user experience. This application is not ready to be used in a real election as both software security and the full cryptographic features have not been integrated yet. As described in the protocol, the public key, the encrypted tracker, the commitment and the encrypted vote should be displayed on the bulletin board after the voting phase. In our studies we simplified these aspects and only update the bulletin board in the end with the pairs (tracker,vote).

3.3.4 Trust assumptions

Even if not all the cryptographic primitives have been integrated in this version, we can already discuss the consequences of the design choices on the security properties. Firstly, we assume that the voting device is trusted for privacy. Further, in this test we have used a single device for voting and verification. In real scenarios, we would recommend that different devices, or at least apps, are used for vote-casting and vote-verifying for improved security.

The reason for using only one device was to simplify the experience for the participants focusing on a basic voting and verifying experience and to test this. The tracking number retrieval is also automated: the voter does not have to manually combine the α and β terms and decrypt the tracking number. Since the α term is not shown to the voter, no visible α term needs to be faked. The coercion-mitigation mechanism have been implemented in the web app only where the users can fake the tracker directly by choosing one on the bulletin board. Further, the level of receipt-freeness in Selene will also depend on the chosen vote-casting method, e.g. a Helios type of electronic ballot will only achieve software-dependent receipt-freeness. The feedback from the participants will help us to take the correct direction in the future developments and iterations.

Finally, the verification phase was mandatory as a part of the test procedure. But in a real election it is to be expected that not all the voters will verify their vote. We have not investigated the voters' motivation yet.

3.4 Use Case: A Usable Interface for Selene

The first application has been developed with the Android native language (Java) and the back-end server is developed in php and deployed on an Apache server. A second web interface has been developed in php. No security analysis has been performed as

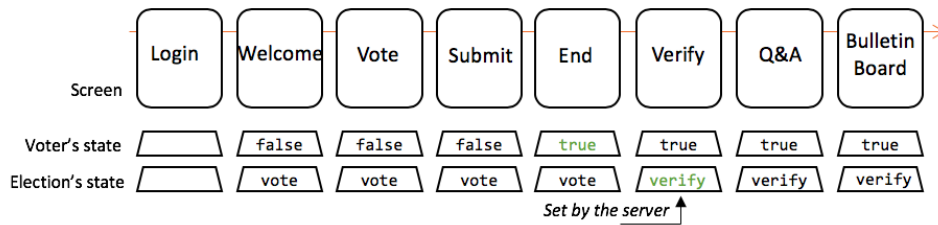


Figure 3.2: Application workflow with states.

the goal of this interface is to run user tests. The security of the application remains basic. We describe below our two interfaces, mobile and web.

3.4.1 From paper to mobile: a participatory design

To start the design and the development of our application, we wanted to have feedback quickly. If our first ideas were wrong, we needed to limit the impact of any modification. The goal of this methodology is to start the implementation of the application with a good representation of what the final user would be expected.

As mentioned in the previous section, following a user-centred approach, the first prototype of our voting application was made in paper. We printed the number of screens needed, and we drew directly what we wanted to show to the final users. For our application, which is linear, this low-fidelity prototype was easy to build, and we received valuable feedback from the HCI and security experts.

3.4.2 Android application

The first application implements two modules: one for voting and one for verification. The application retrieves flags related to the voter after authentication, that indicates: the voter's state (has voted (`true/false`)), and the election's state (`vote/verify`).

Figure 3.2 shows the organization of the application with the corresponding flags. The application has been developed with a linear workflow, the voter only has a minimal choice for navigation, namely going backwards or forwards.

The application starts with a login screen. The voting phase screens are given in figure 3.3. The workflow is straightforward, there is one validation screen after the selection and then the vote is directly sent.

The verification phase screens are given in figure 3.4. We provide explanations regarding the tracking number. Our idea here was to be positive about verification, and mention that privacy is preserved. The voter can retrieve his tracking number and the latter is highlighted in the application.

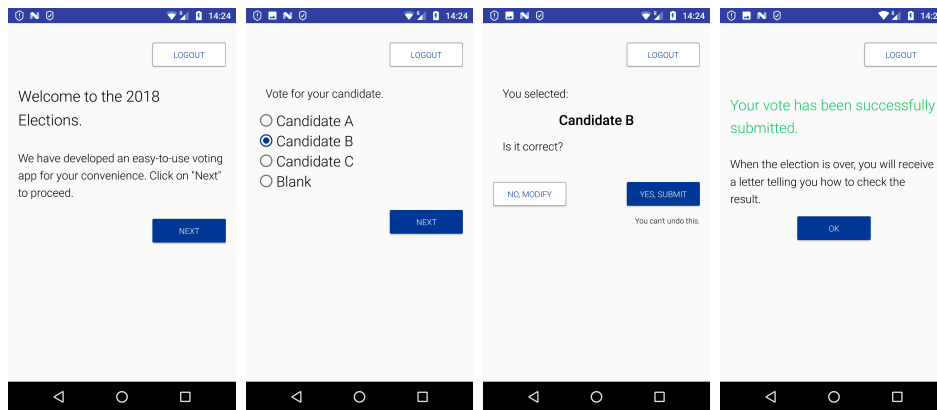


Figure 3.3: The voting screens.

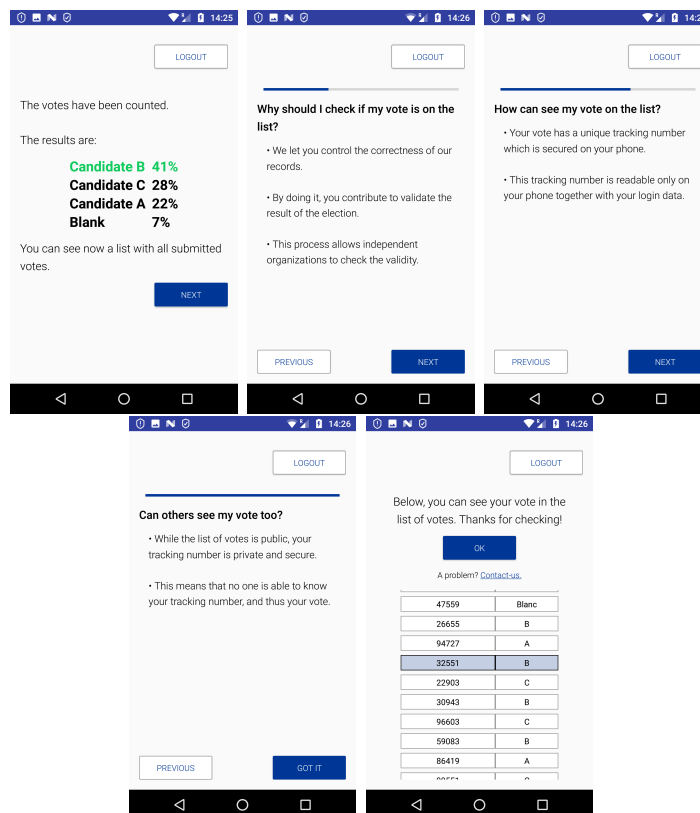


Figure 3.4: Verification phase screens.

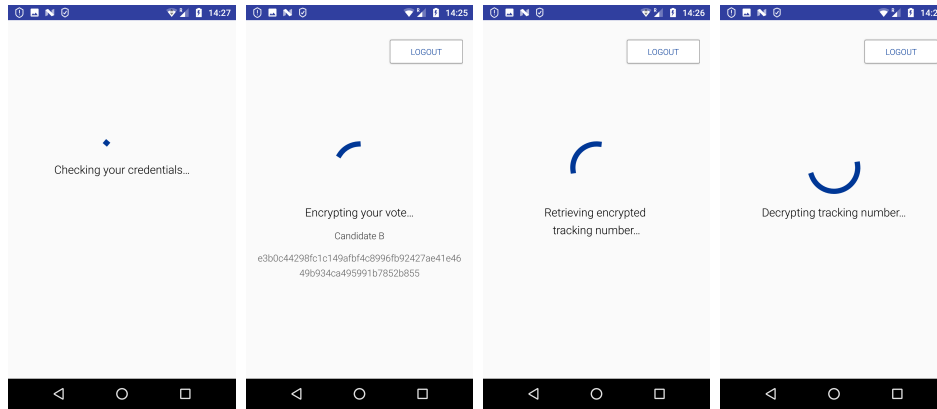


Figure 3.5: Security screens in the first version.

Security screens In addition of the previous interface, we designed specific screens in order to communicate the security of the protocol to the participants (see figure 3.5). Our first study, that we will describe in the next section, have shown that this screens remain unseen by all participants, even if the security perception is slightly higher [40].

Remark. One should notice that in the context of a real election, a voting application might contain more screens with additional information. Here, we designed an application for mobile phones, reducing the amount of information that can be shown. Furthermore, this being designed for a user study, we needed to simplify the amount of information to fit the study context. In a real election, a broader context must be created to inform and teach future users about the protocol and its security.

3.4.3 Administration page

The back-end server is an API that is used to verify voter eligibility during authentication, to receive votes sent by the app, tally the results, and send the tracking number information during the verification phase.

The administrator can access it through a user interface from a browser, to manually tally the results, and change the state of the election from “vote” to “verify”. When tallying the results, all pairs (tracker, vote) are counted and published on the Bulletin Board.

3.4.4 Bulletin Board

The bulletin board is retrieved during the verification phase by the application. An additional button lets the voter highlight her vote. The bulletin board is accessible inside the mobile app but can be accessed directly from any browser, however it

only contains minimal data needed for the user test. Indeed, the original protocol stipulates more elements should be published to be verified:

- Publication of trackers during the setup phase: verification of the uniqueness of trackers.
- Mix of encrypted trackers: the encrypted trackers go through a sequence of verifiable re-encryption mixes.
- Well-formedness of the α -terms: the tracker tellers provide proofs of knowledge that the α creation are well-formed.
- Correctness of sent votes: the votes must be sent with a proof of knowledge of the plaintext.
- Eligibility of the voter: each voter must sign their ballot.
- Mix and decryption of tracker-vote pairs: before decryption, the extracted (tracker,vote) pairs are put through a verifiable re-encryption mixnet, and Tellers perform a verifiable decryption.

All these additional steps, requiring more expertise than the individual verification phase, are taken out of the application, and would be anyway hidden from the standard users.

3.4.5 Version 2

We can highlight here some improvements we have made between versions and which lessons we have learned.

As mentioned above, the security screens were never noticed by the participants in our first study. One can argue that the communication has been done in the wrong place, as those screens are also loading screens. In the updated version of the application, we tried to slightly improve this by adding more security-related words inside the other screens.

When evaluating the voters' understanding of Selene, we noticed that the purpose of the tracking number was sometimes misunderstood. Some participants said that it was the number of people who had voted the same way as they did. In the updated version, we created unique tracking numbers including letters to digits, removing this possible interpretation.

Finally, we separated the application into two different applications: one for voting and one for verifying. The reason is that one might believe that the revealed vote during the verification phase was saved from the vote casting phase, and not retrieved. By using two different apps, we wanted to emphasize the separation between the two phases and their independent execution. Furthermore, security-wise, using two applications, or even two devices, makes it harder for an attacker to steal

data. The voter signing and verification keys could be easily separated, ensuring a better privacy protection.

From this three updates, we will see in the next chapter what impact these changes have had on the participants' understanding.

3.4.6 Web application

In addition to the mobile application, we designed a web interface following the same approach for another study that we will detail in section 3.6 and in chapter 5. As we already had a satisfactory interface for smartphones, we did not prepare a low-fidelity prototype and developed a first prototype directly in the browser. We re-used the same API already available on the server, and we improved it to be compatible with our voting client.

3.4.6.1 Design

The main difference compared to the mobile application is the navigation. For the mobile application, we developed a linear workflow that brings the users to the next page without them needing to think about where to go. In a browser, the workflow must provide more freedom to its users, hence we chose a different design and gave the opportunity to navigate in the different pages, looking for information themselves.

Following the idea of a participatory design, the application went through several iterations with security and HCI experts, and with pilot participants. Similarly, the user can connect using credentials that we provide. Once connected, the user can access the following pages:

- Home: A page summarising the purpose of the app.
- Voting: While the user has not voted yet, this page display a choice of candidate. After choosing, the user is explicitly asked to encrypt the vote (by clicking a button) and send it to the election server.
- Verifying: From this page, information about why one should verify his vote is provided. From there, the user has 2 options: go for verification, or go for tracker deniability (see next paragraph). This page is shown in figure 3.6.
- About: This page explain the entire protocol to the users (in a user-friendly way).
- Contact: This page provides a contact to any user who wishes to ask more questions about the app.

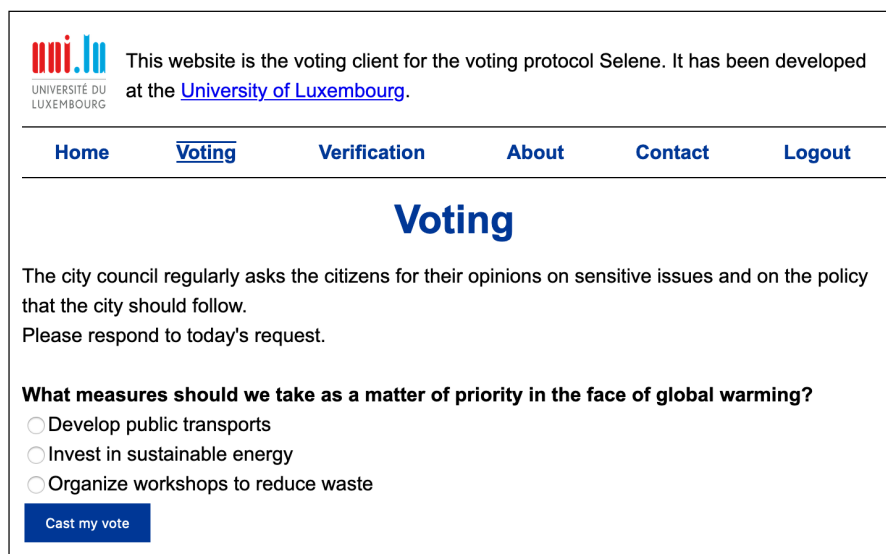


Figure 3.6: Verification page of the Selene web application

3.4.6.2 New feature: the coercion mitigation mechanism


With this new application, we want to test the coercion mitigation mechanism provided by Selene, or the ability for a voter to fake his tracking number in case of coercion. As explained in chapter 1, a voter can choose another tracker of his choice on the bulletin board, and show this tracker to the coercer. Here, the calculation of the α -term is hidden to the voter, we did not provide such detailed explanation. Instead, we let the voter select any tracker available on the bulletin board and validate that this tracker will be the one he will be notified with (see figure 3.7).

3.4.6.3 Iterative tests

As mentioned above, the web application and the new feature have been tested with HCI and security experts, as well as end-users on the platform Prolific during a pilot study. The participants were selected for having a high approval rate ($> 95\%$) in Prolific¹. While the first tests of the application were promising, the first pilot study on Prolific revealed a behaviour that we did not expect: the selected participants tried to finish the study as fast as they can (5 to 14 minutes despite the 25 minutes provided). The absence of linearity was problematic in this context, hence, to be tested on Prolific with end-users, our test version of the application must be guiding and force the participants to follow a specific workflow.

Rather than doing an application opened to self-exploration, we transformed the application into a tutorial that we will detail in the study description in section 3.6.

¹The amount of participants and the setup have been done following the recommendations of Prolific.


 This website is the voting client for the voting protocol Selene. It has been developed at the [University of Luxembourg](#).

[Home](#) [Voting](#) [Verification](#) [About](#) [Contact](#) [Logout](#)

Hide my vote

To receive a tracker showing a vote that is not yours, please proceed as follow.
 We can provide another tracking number of your choice.
 Please visit the [Bulletin Board](#) and type below a tracking number that suits you.
 Alternatively, you can type one of the possible responses below and a list of trackers will be displayed.
 To learn more about the process, you can visit the page [About](#).
After clicking on "Proceed", keep in mind that you will give up on your real tracker.

Tracking number:

Figure 3.7: Coercion mitigation mechanism

3.5 Metrics

In this section we will talk about the standard questionnaires in use when evaluating the usability of an application. In chapter 5, we will introduce a new metric to evaluate Trust.

3.5.1 Usability

Usability of an application concerns the effectiveness, efficiency and satisfaction of a protocol. In particular, the criteria are defined as:

Effectiveness The accuracy and completeness with which specified users can achieve specified goals in particular environments.

Efficiency The resources expended in relation to the accuracy and completeness of goals achieved.

Satisfaction The comfort and acceptability of the work system to its users and other people affected by its use.

3.5.1.1 Effectiveness

Based on the above definition of effectiveness, we need to determine the binary success, meaning if a voter manage or not to cast/verify a vote, and the level of success, meaning the process the voters went through [124]. In one of our study

below, the effectiveness of verification was measured by introducing a manipulated vote in one of the round. The detection of such a manipulation let us know if the verification was effective.

3.5.1.2 Efficiency

According to the above definition, we can measure efficiency by assessing the completion time of different tasks, e.g. cast or verify a vote.

3.5.1.3 Satisfaction

Satisfaction is commonly measured with questionnaires, the most common being the System Usability Scale - SUS (see ISO 9241-11 [121]), we provide the complete questionnaire in the appendix A.1. It has been used in many e-voting studies, meaning that it can be used to compare new results with previous work. However, this scale remains pretty restrictive and as stated in [90], it is recommended to go for the User Experience Questionnaire (see next subsection) which measures additional aspects of a user experience, hence gives a bigger picture.

3.5.2 User Experience

User experience is defined as “a person’s perceptions and responses that result from the use or anticipated use of a product, system of service” [122]. To measure the user experience, we can rely on the User Experience Questionnaire (UEQ) which covers the overall attractiveness, pragmatic/usability aspects (perspicuity, efficiency, dependability) and hedonic aspects (stimulation, novelty) of a user experience. In particular, the criteria are defined as:

Attractiveness General appreciation of the system.

Perspicuity Easiness to learn and to use.

Efficiency Resources expended by the user, promptness of the system.

Dependability Feeling of control and security.

Stimulation Motivation to use.

Novelty Creativity of and interest in the system.

In addition of considering the usability of the product, this questionnaire helps to consider users’ expectations and feelings. Some correlations can be drawn between the different criteria. The UEQ contains 26 items that are contrasted pairs of words separated by a 7-points scale, e.g.:

secure ○ ○ ○ ○ ○ ○ ○ not secure

The complete questionnaire is given in appendix A.2.

3.5.3 Psychological needs

Sheldon et al. [119] proposed a questionnaire to evaluate the psychological needs of users in ten questions. The goal of such a questionnaire is to evaluate which needs are fundamental to users, more generally humans. As suggested by authors, the fulfillment of human psychological needs could help to lead to a positive user experience. Lallemand and Koenig [79] proposed an update of this scale, now containing thirty items divided in seven subscales, that are: *Competence* (5 items), *Autonomy* (4 items), *Security* (5 items), *Pleasure* (4 items), *Relatedness* (4 items), *Influence* (4 items), *Self-actualizing* (4 items).

The items are rated using a 5-points Likert scale, from 1 - *Not at all* to 5 - *Extremely*. The questionnaire is given in appendix A.3.

3.6 Results from three user studies

3.6.1 User Experience and User Needs

This study is a common work with Verena Distler from University of Luxembourg. More details can be found in the paper [40].

This is the first study that we have run with the Selene mobile application. In the next chapter, we will also discuss the mental models explored during this study. Here, we look only at the quantitative results on UX.

3.6.1.1 Methodology

Participants We recruited 38 French participants (19 male and 19 female) through social networks, trying to ensure a fair distribution of our sample in terms of gender, age and education level. The average age was 35,4 years old (Min=19, Max=73, SD=12,45). The education level broadly varied as well: no diploma (13%), A-Levels (29%), some college degree (21%), Bachelor (18%), Master (16%) and PhD (3%). The study has been run in French and the data presented has thus been translated into English.

To make their answers consistent and accurate, we selected participants that had participated at least in one political national election in France.

Procedure We provided each participant with a paper sheet explaining the context of the user test, that is a national election in France, together with the candidates' programs. Two personalized letters were distributed to each participant to provide them their individual credentials to access the application. Then the sessions were split up into 4 phases:

- (1) the voting phase,

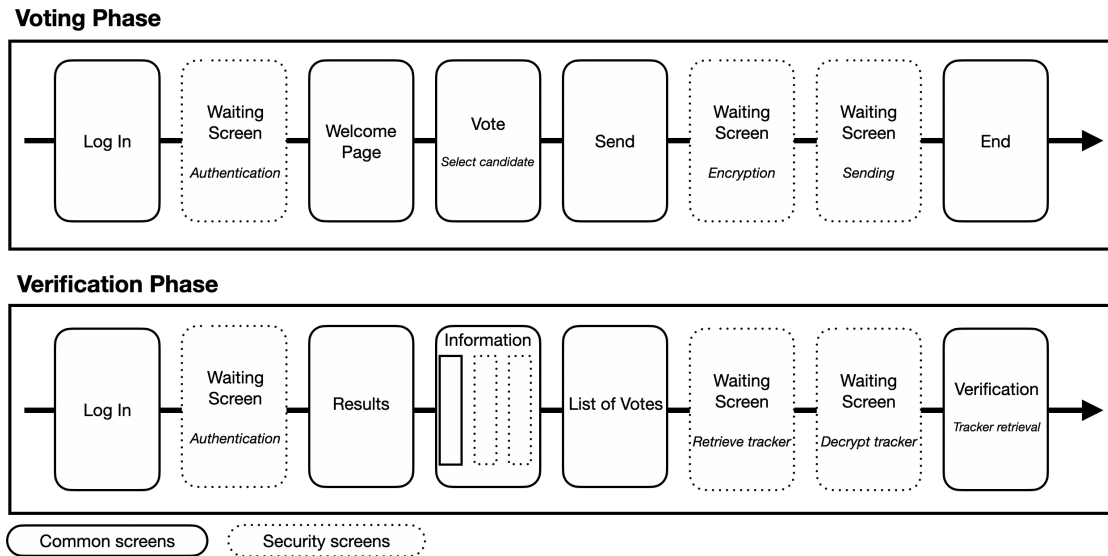


Figure 3.8: Overview of the application workflow

- (2) two questionnaires: the UEQ [115] and UX needs [120, 79]) followed by a semi-structured interview,
- (3) the verification phase,
- (4) the same procedure as in step (2).

The questions asked during the semi-structured interviews will be given in the next chapter.

Before the verification phase, we gave them a second letter which was an invitation to verify their vote using the application.

We did not prime participants with security aspects, to avoid usable security bias [52]. We told participants that our goal was to evaluate the UX of the app, and no reference to security was made during interviews unless participants brought the subject themselves.

After the study, we had a debriefing phase where we let participants know our goals regarding security.

Material For this user study, we split the participants into two groups: one of them used an application that displays the security screens and the other group used an application without the security screens (security screens are given in figure 3.5). Also, explanations was partially removed for the second group, in particular screens mentioning the privacy of the tracking number. In the following, we will refer to D and ND for resp. the app with security display and the app without security display. An overview of the workflow in the app is shown in figure 3.8.

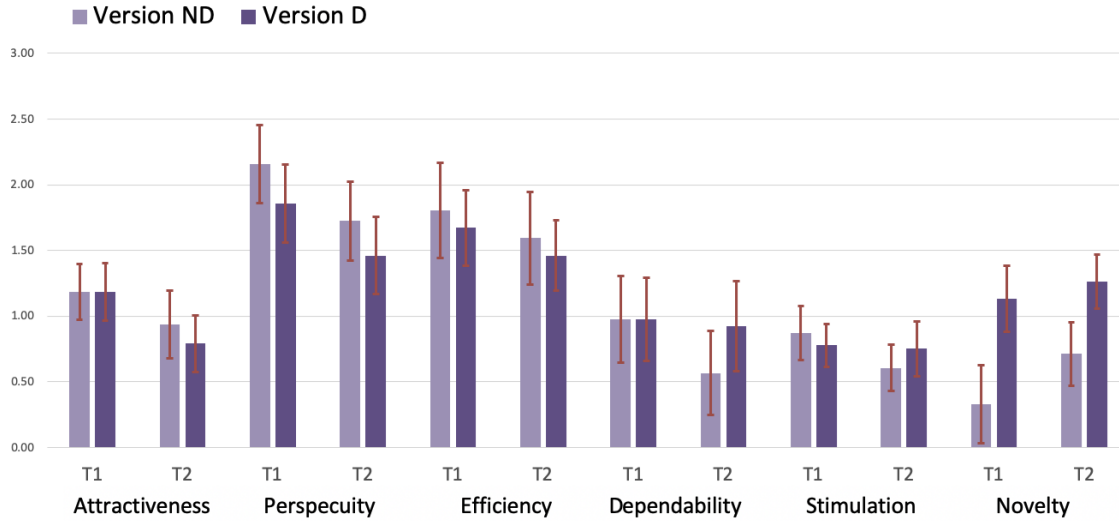


Figure 3.9: Means and Standard Errors of the Mean for the UEQ.

3.6.1.2 Results

User Experience Questionnaire The summary of the UEQ results are displayed in figure 3.9.

We can see that the version D at T1 scored higher on Novelty than version ND. Version ND at T1 scored slightly higher for usability aspects ($M = 1.64$, $SD = 1.41$) than version D ($M = 1.5$, $SD = 1.41$). At the subscale level, results hence indicate that Perspicuity (e.g., understandable/not understandable, difficult to learn/easy to learn) was experienced higher in Version ND ($M = 2.16$, $SD = 1.29$) than in Version D ($M = 1.90$, $SD = 1.30$) at T1. We can also observe that the scores are lower after phase T2 except for the Novelty scale.

UX Needs questionnaire The measurements from the UX needs questionnaires (Autonomy, Competence and Security subscales) are displayed in figure 3.10.

Participants had the same assessment of their need for Security in both versions D ($M=3.8$, $SD=0.71$) and ND ($M=3.51$, $SD=1.00$). The need for Competence was higher in version D after T1 than after T2. We can't observe any difference in the need of Autonomy. In [40], we mentioned a strong correlation between Security and Competence for version ND and no correlation in version D, especially after the verification phase (T2).

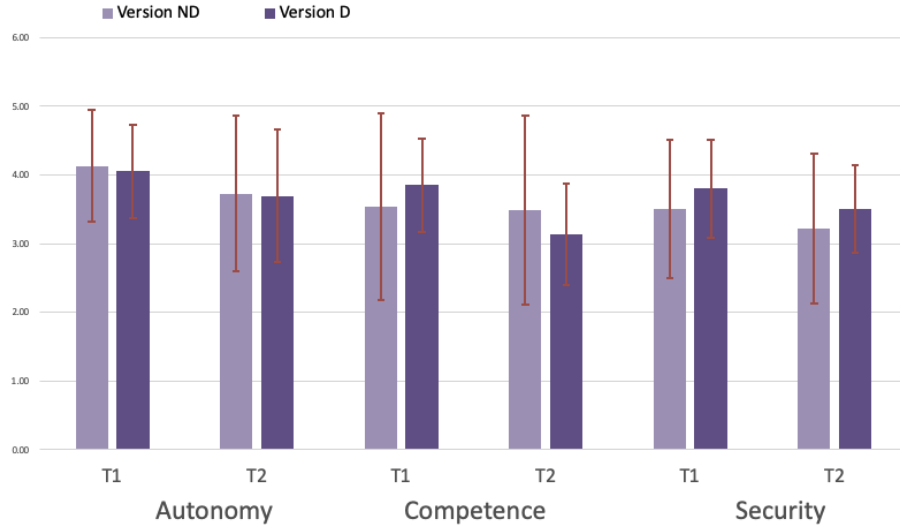


Figure 3.10: Means and Standard Deviations for the UX Needs questionnaire, for the Autonomy, Competence and Security scales.

3.6.1.3 Discussion

First, we observed 100% of effectiveness for vote casting: all participants were able to cast their vote successfully.² The application was designed in order to be easy-to-use and responding to users’ expectations, and we can argue that it is the linearity of the vote casting in the Selene implementation that leads to this excellent result. The quality of this straightforward behaviour was mentioned several times by participants, e.g. “we follow the workflow, but we can’t really make a mistake” (P3).

Almost all participants who had used the version D of the application did not notice clearly the security information that was displayed to them, and we can observe from the questionnaires that there is no significant differences in the UX needs questionnaire.

From [115], we can say that our application scored pretty well on the usability aspects and the application received an overall positive feedback, especially regarding the perspicuity: the app was seen as understandable and easy to use. However, as we will discuss further in the next chapter, this does not guarantee an acceptance of the system and many participants remain skeptical regarding the overall security and regarding the utility of the verification phase.

3.6.2 Usability

This work results from a collaboration with Karola Marky from Technische Universität Darmstadt. The usability results have been compared with 3 other voting

²See [7] where only 60% of ballots were effectively cast with Helios [10].

schemes and the corresponding paper is under submission. For this thesis, we will look at the usability results of Selene only.

3.6.2.1 Methodology

Participants We conducted a user study with 24 participants. They were recruited by mailing lists, social networks, flyers and poster advertisements. Fifteen participants identified as male, eight as female and one as other. Most of the participants were in the range of 19-30 years old, five were more than 31. The average age was 24.8 years (Min=19, Max=40, SD=5.37). Within the 24 recruited participants eight of them have a background in computer science, either as a student or as a professional. 14 participants were students in different areas. We recruited participants that were technology-savvy since prior research indicated that this is a good starting point before investigating the broader voter population [88].

Procedure We used the mobile application described in previous sections, with additional improvements mentioned in the subsection 3.4.5. We separated the system into two apps, one for voting and one for verifying. We also added a button that clearly mentions that the votes are getting encrypted before submission. Finally, we changed the tracking number syntax by adding letters to help the participants to understand its purpose.

The apps simulated an election for a parliament in Germany, to give a realistic scenario as recommended by related work [90]. Therefore, we used the ballots and results from the last election in the constituency where the study took place. The election had two contests, the first one had six candidates and the second one 20.

During the study, we used the System Usability Scale - SUS (see ISO 9241-11 [121]). To measure the effectiveness effectively, we should measure whether a participant performed a mental task or not. To capture this, we randomly manipulated one of the two contests for all participants, as recommended in the literature [90, 116]. This means that the voting option next to the tracker did not correspond to the voter's choice. In the end, each participant cast two votes since we wanted them to experience the voting scheme with and without a manipulated vote.

The study protocol was as follows:

- Consent form and demographics
- Study material: we provided voting instructions to the participants to preserve their privacy, and the paper slip was always in front of them for them to remember the voting instruction.
- Interaction and Questionnaire: the participants use the applications to cast and verify their vote twice, first without the manipulated vote and second with the manipulation. Then they were asked to fill the SUS questionnaire³.

³A drawing session was also performed and the analysis are given in the next chapter.

- Final questionnaire and Debriefing: a final questionnaire with open-ended questions were given, and finally the debriefing let the participant ask any questions. The manipulation was revealed.

3.6.2.2 Results

The overall results are given in table 3.1.

To measure the effectiveness, we look at the number of participants who managed to detect the vote manipulation. Surprisingly, even though the tracking numbers show a plaintext vote, only 84% clearly mentioned that their vote was wrong after the manipulation.

The efficiency measured the execution time in seconds for both voting and verification phases. Here, the participants took between 201 and 663 seconds in total.

Finally, the satisfaction measured with the SUS questionnaire gave a pretty high score with a mean of 82.10.

Effectiveness	Efficiency					Satisfaction				
	Mean	Median	SD	Min	Max	Mean	Median	SD	Min	Max
84	326.68	298.00	102.124	201.00	663.00	82.10	90.00	15.08	40.00	100.00

Table 3.1: Usability results for Selene.

3.6.2.3 Discussion

Our mobile application for Selene scored pretty well on the Usability assessments. The surprising result was that not all participants have detected a manipulation, despite the high usability and the verifiability phase showing a plaintext vote. To evaluate the reasons behind those mistakes, we can explore the voters' understanding of the system as we will do in the next chapter. One idea is that a participant who is unsure about his understanding of the application will not mention a possible mistake as he might believe that he is wrong.

3.6.3 Usability and User Experience of the coercion mitigation mechanism

In this section, we describe our methodology to test the coercion mitigation mechanism of Selene. We obtained ethical approval to test this mechanism in a specific scenario involving a vote buyer. In this chapter, we will give a first glance on the usability and user experience results, in chapter 5 we will focus on the evaluation of the voters' understanding and their trust in the application.

3.6.3.1 Methodology

Participants We recruited 300 participants on the crowd-sourcing website Prolific [103]. We used the pre-screening feature to select participants: to ensure that they have a similar experience in voting, we chose UK citizens living in UK. The average age was 33 years (Min=18, Max=73, SD=11). They come from various backgrounds, the education level differed: No diploma (0,67%), A-Levels (13,33%), College Level (19,33%), Bachelor (42,33%), Master Degree (20%), PhD (1,33%) and other (3%).

Procedure We provided a study description on the crowd-sourcing website Prolific, explaining the context and the workflow they will have to follow in the app. The study protocol was as follows:

- Consent form: we provide information about the goal of the study and the participants' rights regarding their data.
- Demographics: we asked basic information but also their opinion about how to contain the COVID-19 crisis. Their answer is used to configure the game's step (see below) but its value is not of interest.
- Video: we prepared a video explaining the basic operations of Selene. In particular, we mention the trackers' creation, the tally process, and a description on how to vote and verify a vote in the app [131].
- Tutorial: to test the application and help the user to understand how it works, we let them access the application but we provided a guide, giving instructions next to the app. More details are given below.
- Questionnaire: the participants were asked to fill a questionnaire to evaluate the usability and the user experience⁴, as well as three questions to evaluate their understanding of the protocol⁵.
- Game: we asked the participant to use the app again and this time, a vote buyer is trying to buy their vote. We used the answer provided in the demographics to ask them to vote for the other available candidate. Again, instructions regarding the vote buyer's request are given in the app, but the participants are free to decide how to use the application. Specific payoffs for selling the vote or keeping it are given. The details and results regarding this game are explored in chapter 5.

⁴We also evaluated their trust that we will detail in chapter 5.

⁵This was done to ensure a better focus during the study as we will explain below. Even though the questions were easy and the answers were clearly available in the text or the video, we know that it is a lot of information to receive at once and we won't evaluate the participants' performance on this.

- Final questions and debriefing: after completing the game, users were asked which choice they think they have made and why. We also asked about their feeling regarding the application. Finally, we debrief the participants by giving more insights on our tests and redirect them back to prolific.

Participants were paid 2.5 £ for the study (20 minutes), and we added an extra 1 £ as a bonus payment for having played the game.

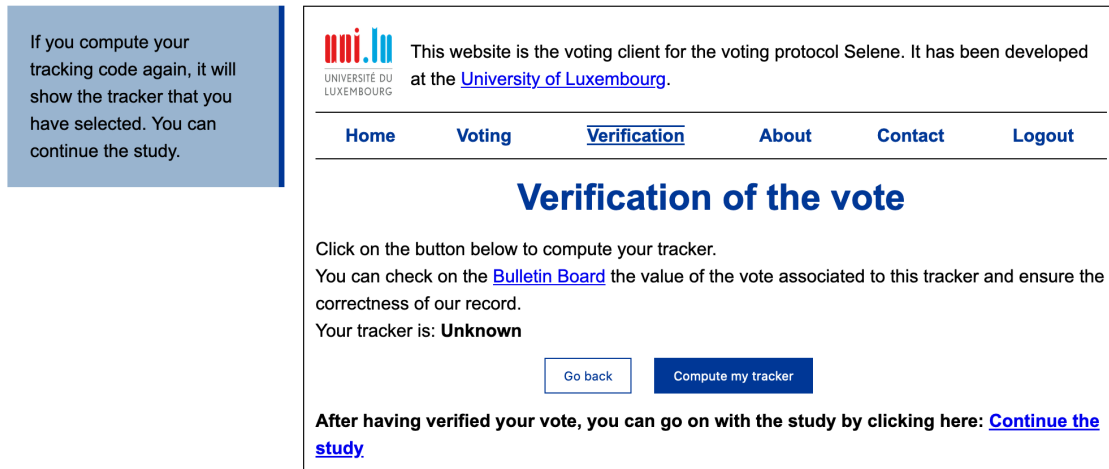


Figure 3.11: An example of a page in the tutorial.

Context and ethical approval To test the coercion mitigation mechanism of Selene, we have elaborated a scenario that involves a vote buyer, to which the participant can decide to provide (or not) a tracking code corresponding to his will. We obtained ethical approval from the University of Luxembourg to run the study online.

We introduced the scenario in the study description (see appendix C.1). In the video, we explain the possibility of concealing the vote, for any reason.

We will evaluate the voters' understanding regarding this scenario in the next chapter.

Voting questions To make the system more realistic, we gave a context of use to the participants. We introduced the voting page saying that the city council asks the citizens about decisions to be made in the city, i.e. our app was for a use in a local election.

Our voting question for the tutorial concerns the global warming issues. The game setup is given in chapter 5 and concerns the covid-19 crisis.

What measures should we take as a matter of priority in the face of global warming?

- Develop public transports
- Invest in sustainable energy
- Organize workshops to reduce waste

Application test and tutorial After having adapted our voting application to the Prolific users (see section 3.4), we ran a second pilot study on Prolific, lasting 25 minutes and with 5 participants, before launching the study with 300 participants.

To ensure that they won't miss any available feature in their evaluation, we forced them to follow a specific workflow that will bring them through all features. We design the application test by adding instructions next to the application as shown in figure 3.11. It forced the participants to use every screen, the navigation workflow is given in figure 3.12.

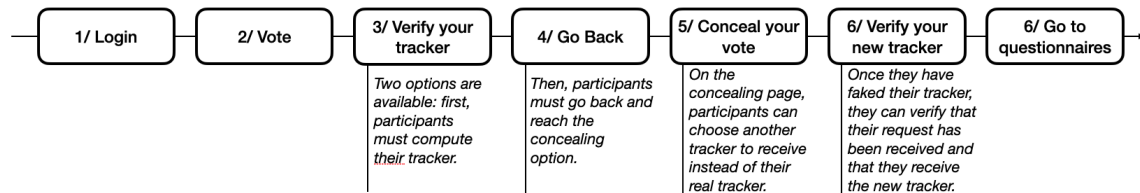


Figure 3.12: The navigation workflow in the tutorial phase.

In addition to this tutorial, in the study description, we emphasized the importance of watching the video or read the available information. We indicated that a few easy questions will be asked in relation to this. By doing this, we were hoping for a better focus and more investment in the app.

We ran our second pilot study after these modifications. Participants have spent the right amount of time and several of them mentioned the concealing feature in their feedback.

3.6.3.2 Results

In this section we will explore the results obtained for the user experience and the usability questionnaires.

User Experience Questionnaire The summary of the UEQ results are displayed in figure 3.13.

Compared to the mobile application, we can see that the web application performed poorly. The attractiveness has been rated as -0.09 (SD=0.07), the usability aspects received the score of 0.16 (SD=0.08) and the hedonic aspects received the score of 0.33 (SD=0.06). At a subscale level, dependability received the higher score with 0.6 (SD=0.06).

Where perspicuity (difficult to learn/easy to learn) was the highest score in the mobile application, here it was the lower score received in the UEQ with -0.4 (SD=0.09).

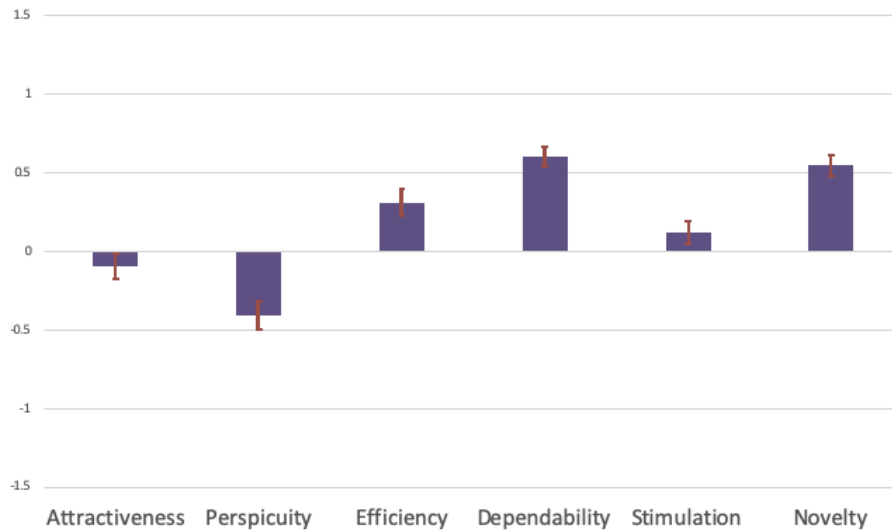


Figure 3.13: Means and Standard Errors of the Mean for the UEQ.

System Usability Scale The summary of the SUS results are given in table 3.2. We measured effectiveness by asking the participants to give a self assessment of their individual verification step. We asked them to tell us if they found their tracking code on the bulletin board. Only 86% of the participants answered that they found their vote, even though we know that all participants have computed their tracking number.

We measured the efficiency by measuring the time taken by the participants to vote and to compute their tracking code. The average time is 57 seconds.

Compared to the mobile application, again, the web application performed poorly on usability. We can also note that participants were on average six times faster to vote and verify, while the minimum amount to cast a vote is almost twelve times faster with the web app, questioning the participants' commitment to the test.

Effectiveness	Efficiency					Satisfaction				
	Mean	Median	SD	Min	Max	Mean	Median	SD	Min	Max
86	57	45.5	39.65	17	324	48.67	45	22.81	0	100

Table 3.2: Usability results for the web application.

3.6.3.3 Discussion

Compared to the mobile application results, the participants assessed a pretty low score to our voting application, even if it is the same protocol and the same information for most screens.

We have several ideas to explain those assessments. First, we noticed that Prolific’s users are used to fill surveys and not participate to complex studies as this one. Several participants mentioned the complexity of the tasks, reflected in the perspicuity score of the UEQ. We run the study in pilot with 10 participants before launching it, and it appears that participants were rushing to the end, probably to maximize their payoff. The measured efficiency supports this assumption. Furthermore, we have been contacted by several participants because they did not know how to connect to our app, while this information was written in the study’s description. One other assumption is that the participants evaluated the study’s complexity rather than the voting protocol itself.

Regarding the protocol itself, the user workflow has changed compared to the mobile version to introduce the coercion-mitigation mechanism. First, this new feature adds complexity to the standard linear workflow: the voters can choose between two options when verifying their vote. In the case of using the coercion-mitigation mechanism, the voter must check the bulletin board on another page and choose another tracking code. Therefore, the bulletin board has to be accessible aside the main application, and not displayed directly to the voter when computing the tracking code. This also adds two steps for the voter if he chooses to verify their tracking code: open a new page to display the bulletin board, and search by hand his tracking code. From a security point of view, these design choices are better as the verification relies on the voter, and will not be saved in the browser’s history. However, the extra-steps, added to the standard workflow, could be the cause for a lower usability and user experience.

Regarding the design, compared to our mobile application, this web application was not linear, even if for the test, we helped the participants by guiding them through instructions. They were still allowed to navigate. We let the participants explore the application, that could make sense for a real election app but might have added too much complexity for a user study. More of these aspects will be explored in chapter 5.

Finally, Prolific provides a high number of participants in a very small amount of time. The payoff that we propose was rated as good by Prolific (£7.50 per hour, plus the extra payment of £1). However, this remains pretty low compared to a lab study, and participants might get less involved by an online study than a lab study where their opinion is explored more deeply.

3.7 Conclusion

In this chapter, we have used a user-centred methodology to design an interface for Selene. We have described the several steps of the methodology and the resulting applications: first a mobile application and second a web application. We also gave the results regarding usability and user experience. Those were first insights regarding the possibilities offered by Selene. We have seen that we can obtain very good results for usability, but also a very bad one, when adding complexity to the tasks. In the next chapters, we will explore more in details the understanding of voters.

Chapter 4

Evaluating the voters' understanding through their mental models

In this chapter, we introduce the concept of mental models for evaluating the voters' experience. Mental models represent the people's understanding of how things work (Norman in [95]). Exploring mental models helps us to go further in the analysis of voting systems, by evaluating the voters' understanding, their expectations and the gap between their perceptions and the evaluated scheme. We present results from two user studies: the first one is an analysis of interviews, where the mental models are compared with the theoretical security notions in voting; the second is an analysis of drawings where participants were asked to represent their understanding. Based on our findings, we conclude with recommendations for future implementation of Selene as well as for the design of online voting systems.

4.1 Introduction

To improve access to elections, several countries introduced ways to conduct elections over the Internet (e.g., Estonia [47], Switzerland [117], and Australia [38]). To uphold democratic principles, voting researchers have proposed secure and robust systems to ensure the integrity of Internet elections. The goal is to satisfy two main security features among others: privacy and verifiability.

We quickly remind the goals of these two properties here. In the context of voting, Privacy, in particular vote-secrecy, is a well known-concept as it is also mandated by the law in many countries. Verifiability comprises *individual verification* meaning that each voter checks that their vote has been correctly recorded, and *universal verification* meaning that the outcome of the election can be confirmed by any observer [9]. Verification mechanisms, in general, intend to guarantee the correct outcome and execution of an election.

While Privacy aims to keep secret and protect voters' votes, verifiability must

provide a convincing proof to any voter that their votes are correctly recorded-as-cast and correctly counted-as-recorded [9]. To achieve this, Internet voting schemes rely on cryptography, often at the expense of usability (cf. [7, 88, 89]). Research on voting has shown that voters are concerned by the risks related to security [125, 118]. This affects their trust in the system, especially as they see verifiability mechanisms as privacy breaches [114, 93], or their necessity was questioned by the voters [7, 88]. This can be due to the novelty of verification, which is only implemented and utilized in few real elections with high stakes (see [47, 117, 38]). However, it can also come from the complexity of the verification, requiring the voters to perform extra steps during the voting phase, to understand complex mechanisms or to compare cryptographic data to verify their votes [10, 22, 19].

To counter this, the Internet voting scheme Selene has been developed to minimize the voters' interaction with cryptography while providing individual and universal verifiability [109]. We have demonstrated Selene's usability in other studies [40] mentioned in the previous chapter. However, usability studies with other Internet voting protocols have shown that mere usability is not sufficient in convincing voters about the correct processing of the votes [7, 50, 88]. This might be because the voters' mental models do not align with the verification procedure [88].

Contributions. In this chapter, we evaluate voters' mental models of the Selene Internet voting protocol through two user studies. First, we studied the gaps between voting research and users expectations for a voting system, by exploring the mental models of voters for Privacy and Verifiability expressed in the semi-structured interviews. Those results are also available in [130].

In our second study, we analyzed the mental models of 24 tech-savvy participants. After letting them interact with Selene, we asked the participants to draw their understanding of Selene. The drawing session was guided by the examiner who asked them questions to help them express their understanding and perceptions. We performed an analysis of the drawings and of the answers and extracted four categories of mental models: 1) technology understanding, 2) meaning of the verification phase, 3) security concerns, and 4) unnecessary steps. We also classified the understanding of participants into levels of sophistication of their mental models. Those results are under submission.

Finally, we discuss our findings and propose a list of recommendations applicable to Selene and to other internet voting systems, focused on 1) the education of voters on risks, 2) the need of correctness and transparency, 3) the integration of simple interactions with security features and 4) the design of several levels of verification.

The remainder of the chapter is organized as follows: Section 4.2 gives the related work, Section 4.3 gives the methodologies followed in our two user studies. Section 4.4 gives the observed mental models, and Section 4.5 proposes a discussion around our findings. We conclude in Section 4.6.

4.2 Related Work

The study of mental models is useful to align the system design with the users' expectation of a system, reducing the possible interaction errors that could lead to additional security (or safety) issues. The subject has received little attention in voting, we relate our work to the few publications here and discuss them in detail in section 4.5.

Mental models are internal representations that humans derive from the real world to interact with a technology [65]. The level of sophistication of a mental model can differ among humans [65, 28, 67] and the mental models must be sound enough that users can effectively interact with a technology [72]. Generally, two types of mental models can be observed: functional models, and structural models [6]. Functional models mean that users know how to use a technology, but they do not how it works in detail. Structural models offer a more detailed understanding of how a technology works. Once a mental model has been established, it is difficult to shift [123]. Misconceptions within mental models might lead users to behaviours that do not represent their true intentions. There are different ways to capture mental models of users, among them are interviews, sketching, or think-aloud techniques. Related work in the domain of privacy has demonstrated that the combination of sketching and think-aloud is effective to capture the mental models of users [129].

Mental models of verifiability in postal voting and paper voting have been explored by Olembo et al. through a survey conducted in Germany [99]. They suggested breaches in the procedures that could lead to integrity issues and asked participants about different aspects of verifiability.

Another paper from Acemyan et al. [7] analyzed mental models for three voting schemes which are Helios [10], Prêt à Voter [108] and Scantegrity II [32]. The experiment aimed to study the participants' mental models through drawings and interviews after using each of the voting systems. The analysis of participants' feedback showed that many participants did not see the E2E-verifiable schemes as being more secure than a standard paper-based voting method. The authors also highlighted that participants tend to focus more on the voting phase.

Human factors in security were highlighted by Kulyk and Volkamer in [74]. They extract five concepts including concern and self-efficacy, as we did here: we noticed a lack of concern for verifiability and a lack of self-efficacy (in the sense of knowledge and understanding).

Trust was pointed out by Schneider et al. in [114] as an important factor for participants, as people are aware of potential security issues.

4.3 Design of the studies

4.3.1 The interviews study

This first study concerns data that was collected during the same study described in section 3.6.1. We quickly recall the study protocol and give additional data regarding the analysis methodology. While the previous chapter mentioned quantitative results, here we focus exclusively on the qualitative analysis and the mental models.

4.3.1.1 Participants

We recruited 38 French participants (19 male and 19 female) through social networks, trying to ensure a fair distribution of our sample in terms of gender, age and education level. The average age was 35,4 years old (Min=19, Max=73, SD=12,45). The education level broadly varied as well: no diploma (13%), A-Levels (29%), some college degree (21%), Bachelor (18%), Master (16%) and PhD (3%). The study has been run in French and the data presented has thus been translated into English.

To make their answers consistent and accurate, we selected participants that had participated at least in one political national election in France.

4.3.1.2 Procedure

We provided each participant with a paper sheet explaining the context of the user test, that is a national election in France, together with the candidates' programs. Two personalized letters were distributed to each participant to provide them their individual credentials to access the application. Then the sessions were split up into 4 phases: (1) the voting phase, (2) a semi-structured interview, (3) the verification phase and (4) a semi-structured interview. Before the verification phase, we gave them a second letter which was an invitation to verify their vote using the application.

4.3.1.3 Methodology

The goal of the present analysis is to identify which mental models participants have of privacy and verifiability in e-voting. The semi-structured interviews entailed the following topics: general opinion about the application, trust, control, understanding of the verification phase and of the bulletin board. The three first topics were addressed after both the voting and verification phases. The two last topics were addressed after the verification phase only. We avoided security priming by not addressing security-related topics (such as privacy) until the very end of the study in order to avoid influencing participants' answers. In most cases, they mentioned by themselves the different security issues they could face with regard to e-voting. We describe in the next chapter which mental models we identified. Information about the verification procedure was provided through paper letters and inside the application. The Q&A screens are mandatory in the workflow and the participants have

to go through them before verification. We told the participants that the tracking number let them verify that their vote has been counted in the final tally, that it helps to validate the election results, that this tracking number is unique and that the count can be verified by anyone. As we did not want to prime participants with possible security issues, we have not mentioned the risks of using one device and the associated trust assumptions.

4.3.1.4 Data analysis

The user test was devised as a between-subjects study, and two versions of the e-voting application have been tested with our participants: half of the participants tested a baseline version and the other half an extended version where security aspects are additionally displayed to the participant. It contains additional information about the ongoing process in the application yet with no extra interaction. The impact of displaying security-related mechanisms was the topic another published paper [40], mentioned in chapter 3, alongside with additional factors impacting UX (attractiveness, novelty, etc.) and psychological needs (autonomy, competence, etc.). Interestingly, we noted that this additional layer of communication remained largely unseen by the participants, with the perceived security being rated as only slightly higher in the elaborated version. The analysis of this paper focuses on interviews only and explore the feedback of participants regarding the security properties of voting, to check their understanding as it will be described in the next section.

To analyze the data retrieved from the interviews, we followed the methodology described in [80]. We coded the data through a theoretical thematic analysis, to look for patterns relating to the voting security properties. We organised the participants' answers into a list of concepts. We classified these concepts in categories given in the next chapter to understand voters' mental models of security. The categories were organised to match with the known theoretical models of security in e-voting: privacy properties including ballot-secrecy and coercion-resistance, and verifiability.

The qualitative analysis of answers in the semi-structured interviews leads to similar concepts for both versions, and we will thus analyze the participants' feedback in a similar way without considering the tested version.

One goal of user-centred design is to achieve a better alignment between participants' mental models and researchers' security vision, by "ensuring that products do fit real needs, that they are usable and understandable" [95], we will discuss this below.

4.3.1.5 Ethics

The study follows the guidelines provided by the ethics commission at the University of Luxembourg and conformed to GDPR.

4.3.2 The drawings study

This second study concerns data that was collected during the same study described in the section 3.6.2. We quickly recall the study protocol and also provide the extra-steps that are important for the analysis of mental models.

4.3.2.1 Methodology

We implemented two mobile apps for the Selene protocol. The first one was for voting and the second one for verification. The apps simulated an election for a parliament in Germany to give a realistic scenario as recommended by related work [116, 90]. Therefore, we used the ballots and results from the last election in the constituency where the study took place.

Before interviewing the participants to investigate their understanding, they interacted with the voting scheme. Hereby, we wanted to know whether the participants were able to successfully verify their votes. To capture this, we randomly manipulated one of the two contests. This means that the voting option next to the tracking number does not correspond to the voter’s choice. This models a threat to vote recording, and we will study the impact of such a manipulation on the mental models. The procedure of our study was as follows.

1) Welcome and Consent Form. We welcomed the participants by explaining that they are going to vote in an internet election followed by an interview. Then, we let them read the consent form and the study’s data protection policy. Each participant could then ask questions about the procedure and finally signed the consent form.

2) Demographics. Each participant provided demographics consisting of age, gender, education, and occupation. We also asked about previous voting experiences and how often they use smartphones, and other electronic devices to access the internet.

3) Introduction. The participants were introduced to the voting material and devices. Each of them received a letter with sealed voting credentials as if they would in a real election. The credentials consisted of a voter ID and a password. Each participant received randomly chosen voting instructions meaning a voting option for each ballot. This was to preserve the participants’ vote privacy in the study since we took screen-recordings [90].

4) Voting and Verifying. Each participant cast two votes since we wanted them to experience the voting scheme with and without a manipulated vote. In each round, the participants were asked to cast a vote matching the instructions. When

they reported completion, the examiner explained that two weeks have passed and the election results are now available. The participants were asked to use the verification app to have a look at the result and check it¹.

5) Drawing Session. After the interaction with Selene was completed, we proceeded with the interview part. The participants were told that we would like them to draw their understanding of the following questions and that there are no wrong answers. We told the participants that we start the recording and proceeded with the following questions²:

- You have just cast and verified a vote. Could you sketch how that worked according to your understanding?
- What, to your understanding, is the purpose of the tracking number that you have received?
- Why, to your understanding, is it necessary to see the list of all votes and not only your own one or is it not necessary at all?
- How, to your understanding, does the vote verification work?
- Why do you think voters are asked to verify?
- Consider an election, would you want information on how the vote verification works? Where or when would you like to receive this information?

In each question, the participant could integrate cards with pictures that we provided into their drawings. The cards had pictures of the following components: an icon representing the voter, a ballot, a ballot box, a smartphone, the icon of the app, an icon representing internet, an icon for encryption and a server. Then, the examiner stopped the recording and notified the participant about it.

6) End. Each participant was given the opportunity to ask questions, or to provide further feedback. Finally, the participant could fill in the consent form to participate in a raffle for an Amazon voucher.

¹The emphasis was placed on the individual check of the tracking number. We did not explicitly ask the participants to recount the votes for universal verifiability, but the interview is designed to explore this element of understanding.

²We did not ask explicitly in this study if the vote manipulation was noticed.

4.3.2.2 Data Analysis

We transcribed the interviews and used a deductive coding methodology to categorize the data. The categories were discussed before starting the coding, and emerged from the questionnaire given to participants and the existing literature on the analysis of voters perception. Then, two researchers coded independently the interviews and compared their findings. They discussed the categorization and solved the disagreements. We tracked the disagreements and the Cohen's Kappa was calculated at 0.822, that is we had almost perfect agreement.

The drawings were categorized by two researchers as well, ordering them per accuracy. The drawings were then related to the participants feedback about their experience with the apps.

4.3.2.3 Ethical Considerations

The study follows the guidelines provided by the ethics commissions from the University of Luxembourg and Technische Universität Darmstadt, and is conformed to strict national law and conformed to GDPR. In particular, our studies must limit the collection of personal data in order to preserve the privacy of the participants.

To anonymise the data, we proceeded as follows: each participant received a randomly assigned identifier. As mentioned above, before the study starts, each participant signed a consent form. The consent form was recorded separately from all other data such that data cannot be linked to participants' identity. The following information were provided to the participants: goal and procedure of the study, risks associated with the participation, and how data storage and analysis is handled. Finally, a paragraph regarding data protection policy was also given. We can emphasize that the institution where the study was run does not require to follow a formal independent ethics committee for the kind of user study we conducted. But, we took the necessary precautions to preserve participants' privacy.

4.3.2.4 Limitations

First, we consider that the biggest limitation to our study is the sample size. As with similar user experiences based with interviews and qualitative analysis, we can only work on small samples with the idea of providing improved designs for bigger experiments.

Although we took precautions and also recruited participants beyond the university campus, our sample is mainly composed by students and well-educated professionals. This population usually provides a better feedback [59] even if the areas of occupation were diverse.

The participants come from Germany, providing feedback regarding their local experience on voting. Depending on the local habits regarding the voting technologies, the feedback and the understanding of certain tasks might differ.

Another limitation is that the study was run in a lab hence a controlled environment [81, 79], potentially leading to biased answers from the participants. However, for the voting area, it is hard to conduct experiments over real elections to preserve voters’ privacy [90].

4.4 Mental Models

4.4.1 The interviews study

In [6], Norman defined mental models as being “people’s views of the world, of themselves, of their own capabilities, and of the tasks that they are asked to perform, or topics they are asked to learn”. The interactions they have with the environment make them form internal models of the system they are interacting with. We here propose a categorization of participants’ feedback. An overview is given in figure 4.1. From the identified concepts, we derive a categorization of the mental models expressed by participants. We will discuss how these results should impact the future development of the application in section 4.5.

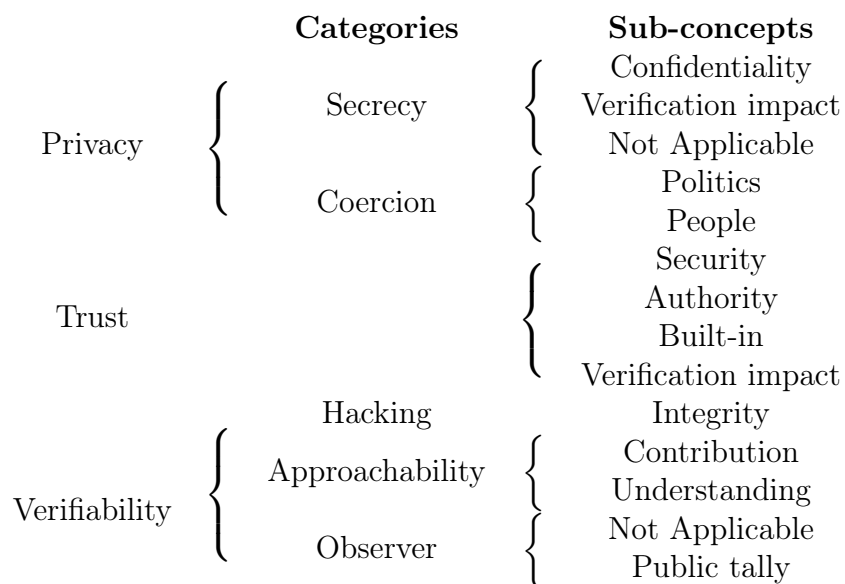


Figure 4.1: Mental model categorisation

We now explain this structure in detail.

4.4.1.1 Privacy

Secrecy Mental Model Participants mentioned that their vote must be kept *confidential and anonymous*. They questioned the data management, wondering if someone knows the relation between identities and votes. Some participants stressed

the importance of the booth, another argued that the booth is not private either, e.g. P10 said:

“Some people are looking.”

The *verification* could have a negative impact on secrecy of votes as well, for instance:

“It is like someone else could see it too.” (P22)

Finally, another concept was *not applicable*, as some people did not feel concerned by secrecy, for instance:

“Others know who I am voting for.” (P13)

“You have to take responsibility for your political decisions.” (P38).

Coercion Mental Model Coercion from *people* (in the sense of a physical attempt to coerce) was mentioned several times. In particular, participants mentioned the advantage of being able to vote at home, as other people won't influence them:

“Here I don't have interactions with other people.” (P5)

“I am sure to make my own choice [...] I feel less pressure than in polling station, with people behind.” (P2).

Vote buying was mentioned once by P23:

“We can be manipulated, one could buy our vote, but we need to evolve.”

Finally, the *political aspect* of coercion was also mentioned a few times, as some parties could try to cheat and to steal credentials from voters:

“We must pay attention to parties, ensure there is no violation, that the elderly or other vulnerable persons do not get their vote stolen.” (P10)

4.4.1.2 Trust

The concept that appears the most for Trust is *security*. Participants mentioned that their trust in the application is highly dependent on the security provided:

“I don't trust it, how could we know if it is really secure?” (P29)

“I trust it, there are breaches everywhere but I think we can secure this.” (P10)

In particular, it was reflected on their other mental models related to privacy and efficiency. Hence, we can derive this security concept with the following sub-concepts: coercion, secrecy, and understanding that increased or decreased the security perceived.

Another concept was *authority*, mentioned as trust-transference in [11]. Participants refer to some trusted third party to emphasize their own trust:

“If it is done by an authority, I will trust it.” (P2)

“I trust the government, they will do what is necessary to ensure vote security.” (P12)

They rejected the verification process arguing with their trust in the authority, e.g. P12 said:

“If I trust the application I don’t see why we should verify that the vote has been taken into account.”

Some participants expressed a *built-in* trust:

“I always trust technology.” (P14)

“I trust it as I would trust any mobile applications.” (P38)

A *verification impact* was raised, mostly decreasing trust, for P33 for instance:

“I don’t trust the application after verification, even if the tracking number is private.”

Even though an opposite positive effect on trust was also mentioned by some users:

“The second phase makes me feel secure.” (P4)

4.4.1.3 Verifiability

Hacking Mental Model Participants were concerned by the security of internet technologies and had many preconceptions. Even if participants didn’t master the complexity of internet security, they were aware that it could be an issue. For example, they mentioned problems they heard about other voting systems with electronic ballots:

“In United States there was this elections hacking. Paper is more reliable.” (P15)

Others feared internet technologies in general:

“I think internet is vulnerable, even if the app is secure.” (P24)

Ballot stuffing was also mentioned as a big problem:

“There are people who can buy hackers’ services to have thousand of votes added, we will never know.” (P28)

Integrity is a concept that often appeared during the interviews. Participants questioned the good behavior of the application as they did not receive any proof of it. The reliability of the system is questioned:

“It does not guarantee that it is really my vote.” (P33)

Some participants also expressed the need for a procedure in case of encountering an issue:

“Who should I call in case of problem? And if my vote is not in the list?” (P19)

“If I voted A and it shows B, what should I do?” (P32)

Approachability Mental Model Some participants were convinced of the good behavior of the system due to the verification phase. It was mentioned as a proof of their personal *contribution* to the elections:

“Seeing that my vote is taken into account, seeing others’ votes, it lets me believe that I contribute to something.” (P18) “It is important to see that my vote has been counted.” (P27)

Most participants understood that they were seeing a confirmation of their tallied-as-intended vote. But they expressed their [lack of] *understanding* in the process of verification like:

“I feel in control maybe because I can see what I did, I can see my vote again.” (P11)

“I wonder why this is here, seeing results with percentages is enough for me.” (P3)

We tried to rate participants’ understanding of Selene’s mechanism, through the two last questions stated in section 4.3. To help participants to answer, we provided some light explanation of the verification phase meaning. However, many participants did not manage to provide a complete description of the verification phase after using the app. Furthermore, the tracking number has not always been understood as such, but rather as a counting of votes:

“We can see our candidate and the number of people who voted for him.” (P6)

Observer Mental Model Some participants stressed the importance of observation. In France, voters are allowed to go to the polling-station to observe the public count of votes. However, most of our participants did not notice the link between this real-life procedure and the availability offered by the bulletin board:

“The list is not really informative.” (P35)

“I can’t see if there is any interest to see this list with all details.” (P17).

This can be explained by their lack of understanding of the procedure, as compared to a physical count of votes, in which they can see and understand each step:

“In polling stations you can verify by yourself, on internet it’s questionable.” (P24)

Finally, the *individual* aspect seems to be enough to participants, e.g.:

“Seeing percentages with general results, and my individual vote is enough” (P17).

4.4.2 The drawings study

In the previous study of Selene, we have shown that the protocol provides a good user experience to the study’s participants, but some misconceptions remain. We have already looked at the mental models of voters using Selene, reflected during interviews, by categorizing them using the security properties of voting. Here, we want to go further by asking explicitly the participants how they imagine the system and represent it in drawings, to reveal their understanding of the verification mechanisms and their beliefs regarding internet technologies.

A detail description of the participants’ feedback and their mental models is given in appendix B.2.

4.4.2.1 General observations

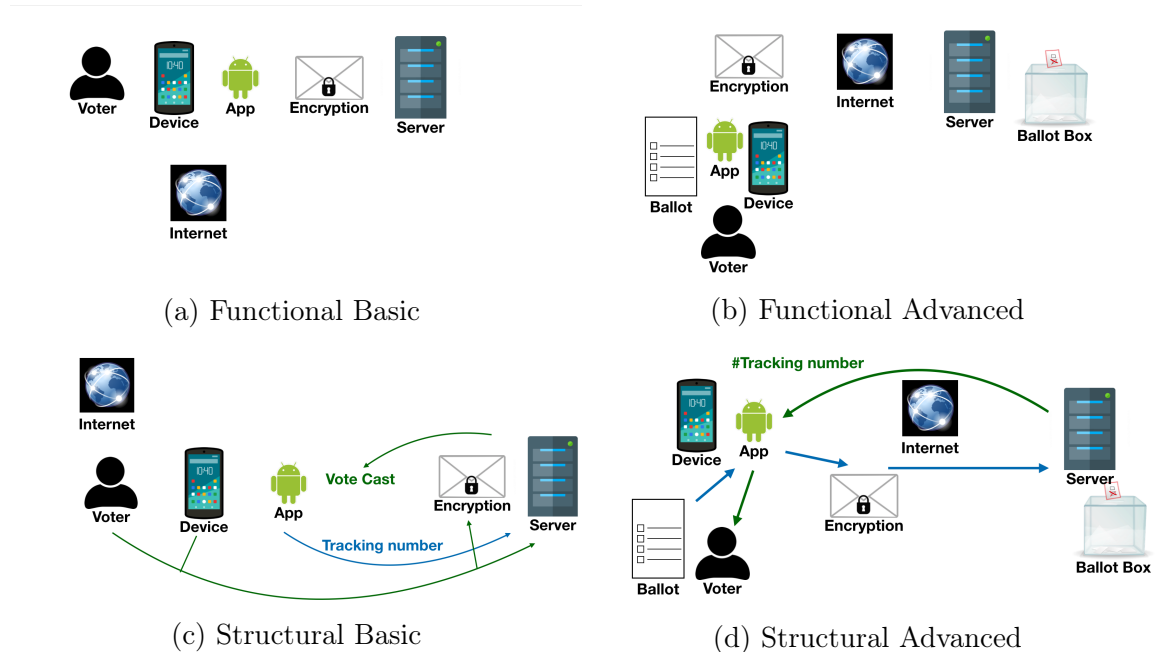


Figure 4.2: Four levels of understanding.

Many participants had a good overview and managed to provide a detailed explanation on how the system works according to their understanding. We classified

the drawings in two types of mental models as described in [6]: one functional model and one structural model.

The functional model is defined with components that were used as concepts, to help the participants express their understanding of the features in the app, without being able to link those components together. In the structural model, we classified the drawings depending on the use of components and their relations to each other. Inside these two categories, we can deduce two levels of understanding: a basic and an advanced one. Figure 4.2 shows an example of drawing (reproduced) for every category mentioned below. We describe the four levels as follows:

- functional basic (one participant): some components are used but are not detailing the entire experience. In figure 4.2a, P22 used some components but the ballot and ballot box are missing.
- functional advanced (seven participants): the components are used in a specific order to express the functional tasks that were performed. In figure 4.2b, P21 used all components and grouped them while explaining his experience.
- structural basic (nine participants): some components are used and related to each other. In figure 4.2c, P14 used some components and tried to relate them but misplaced and did not use all of them.
- structural advanced (seven participants): the components and their relations were correctly set. In figure 4.2d, P01 used all components, and explained the correct structure and relations between all of them.

Regarding the vote manipulation, we counted 20 participants who reported clearly that they have seen a problem to the examiner.

4.4.2.2 Technology Understanding

All participants gave their vision on how the voting system is designed and how they understood the technology behind the apps.

As described in Section 4.3, we provided several components to participants. Some of them were related to a standard paper-based system (ballot, ballot box) and others were related to online technologies (internet, app, device, encryption, server).

Over 24 participants, seven did not use the paper-based components in their drawings. The other components were all used but sometimes misplaced. In particular, six participants believed that the encryption, mentioned in the app as being the encryption of the vote after the candidate selection (see figure 3.2), occurs on the server side, instead of the application. Also, three participants misplaced the ballot box as being the smartphone itself, for instance:

“The smartphone is the ballot box and the app is a tool.” (P16).

Nineteen participants provided a good description of their experience and the technology in use. For example:

“All the data is hopefully encrypted locally, transmitted over the internet and sent to a server where the ballot box is kept up to date and I can check it at some point.” (P03)

“The smartphone uses the app to apparently retrieve the data from the server of the electoral authorities and show the voter which vote he has cast based on this tracking number.” (P06)

“The encrypted vote is then forwarded via the Internet and placed in the digital ballot box and added. And this ballot box is stored on a server, where all election results are then uploaded. And where they can then be retrieved again by the voter after the election, for example by the verification app.” (P12)

Regarding the verification phase, the procedure itself for individual verification was understood but the overall process remained unclear. As described in chapter 3, we mention the existence of a tracking number in the verification phase, but we do not provide details about how the connection between the tracker and the vote is made, just that it is unique and private to each voter. Nevertheless, seven participants tried to describe how their vote was linked to their tracking number. For example, participant P19 said:

“It’s probably generated from some data from my smartphone, because it has to store it somehow, because I didn’t have to enter it anywhere”.

Only one participant (P17) misinterpreted the content of the bulletin board and thought that he was given a link to voters:

“We receive the list of, as far as I understood, all voters.”

4.4.2.3 Meaning of the Verification Phase

In the interview, we asked the participants to explain the purpose of the tracking number, but also why seeing the entire bulletin board is needed. Finally, we ask them why this verification phase exists. Hence, we describe their answers into the three paragraphs below.

Individual Verifiability. 23 out of 24 participants gave an explanation of the individual check mechanism with the tracking number. 15 of them mentioned explicitly the *correctness* of records, others explained a comparison between the record and their vote intention. For example, we have the following comments:

“Here is a list of numbers and votes, and these will be sent to my smartphone and I can compare them with my tracking number to see if what I

voted for finally reached the server.” (P04)

“The user can check if his choice was recorded correctly.” (P06)

“I as a voter can check if I have voted correctly.” (P12)

Only one participant did not manage to explain the tracker’s purpose, and mention that:

“With this tracking number you can ask the server what kind of election was registered for you” (P07).

Universal Verifiability. To explain why they were seeing all votes and not only their own, participants had more trouble finding a good reason. Eleven participants talked explicitly about recounting even if the app did not offer an easy way to do it, for example the participant P02 said:

“It wasn’t possible with the app, but theoretically I could use all the votes to check if everything is correct and of course I need all the votes for that.”.

Two participants also mentioned that the entire list was necessary to find their own vote, as the tracking number was known locally on the phone. For instance participant P09:

“I need them all at some point, because I have to find my own.”

General Purpose. Three participants compare this new feature to the actual voting system that does not allow them to perform such a check. For instance, participant P01 said:

“It offers a way to see if the vote is present at all, because in old systems that’s not there at all”.

Four participants mentioned transparency as a general purpose, for example:

“[...] that we can offer the citizens a certain transparency.” (P16)

“I do think it is necessary, just for the sake of transparency.” (P23)

Five participants also mentioned it as a confidence or trust feature, like participant P11 mentioned that it is *“to give the system a little more confidence.”*

4.4.2.4 Security Concerns

All participants mentioned security concerns during the interviews. With respect to the previous section about the meaning of the verification phase, the correctness of the result and the integrity of the elections was mentioned by 15 participants as a security concern, e.g. participant P07 said:

“There is a bit of certainty that it was done correctly”.

All participants noticed the encryption of the votes in the application. Three participants questioned the encryption of other parts in the system, for instance encryption of the channel between the app and the server and encryption of the data on the server itself, e.g. P05:

“I didn’t pay attention to it, but I hope there was an encrypted connection to the infrastructure of the election office, via the Internet”.

Sixteen participants mentioned that they wanted to have information regarding the verification phase in advance, during the registration process. Three different reasons were given: four said that it will help them to decide whether they choose this application to cast their vote; 11 mentioned that they would evaluate the reliability of the application and six said that it will provide more time for voters to understand how it works.

Nine participants questioned the implementation of the system that had a direct impact on their trust. For example, some participants question the origin of the tracking number and what is the proof that this tracking number is really showing their cast vote. For instance, participant P15 said:

“It was cryptic in the sense that I just received it from the server, I couldn’t understand if this is really the vote I cast.”.

Also, three participants questioned the system by describing it in a skeptical way, e.g. P05:

“Hopefully the votes cast are stored there in encrypted form”.

Attacks or bugs in the system were mentioned by 9 participants, for example ballot rigging, or possible manipulations were brought up, for instance:

“I would know theoretically whether they were manipulated or not” (P20)
“The votes that were cast could also, as it was the case with me once, simply have been wrong somehow.” (P10)

Four participants mentioned anonymity as one of their concerns, linked to the tracking number made up for verification in a private way, e.g. P01:

“They are anonymous, because nobody has any idea which tracking number the other person has.”

Concerns about dispute resolution was also mention by three participants, two of them noticed that they cannot prove how they voted afterwards so questioned how to prove a mistake, e.g. participant P19:

“Somehow nobody can prove that you have actually chosen something else.”

Only two participants mentioned the decryption of the tracking number in the app, and one of them questioned the origin of the keys in use in the app.

Finally, two participants believe that from the list of tracking numbers, one might figure out whom a specific voter voted, hence breaking privacy.

4.4.2.5 Unnecessary Steps

Thirteen participants claim that some steps in the verification phase was unnecessary. In particular, the list containing all trackers and votes were useless to them. For example, participant P09 mentioned:

“If I already know what my tracker is, I honestly don’t see the point of seeing all of them”.

Also, even if some of them cited the recounting of the votes as an option, those participants were not interested in doing it therefore receiving the complete list. For example, participant P06 said:

“You could say that it’s about comparing this list of votes with the overall election results, of course. But then again, I do not see how the normal voter in Germany should actually do that with several million eligible voters or several million votes cast”.

Even if most of the people understood the purpose of individual verification, two were not always convinced by the provided information and questioned the need of making the system verifiable. For instance, participant P03 mentioned that

“It also doesn’t help me to check if this is really part of the final result or not”.

4.5 Discussions

From this two studies, we categorized the mental models with two different goals. First, from the interviews, we extracted the participants’ expectations on e-voting, and we compared them with the technical properties designed by the security experts. The idea was to evaluate the gap between the voters’ idea of voting, and what the research propose as solution.

From the drawings, the approach was focused on the understanding of the voters, by asking them to explain to us their vision of their experience.

In the following, we will discuss our results obtained from the two studies.

4.5.1 The interviews study

Norman in [6], and Cooper et al. in [35], show that three models must be considered in the design of a user interface: the system or implementation model that reflects how the system actually works, the system image or represented model that reflects what is shown to the user and the mental model that is the projection made by the user. Here we focus on the discrepancies between those three distinct categories.

4.5.1.1 Comparison between mental models and security properties

The properties on which we base the implementation model of a voting scheme are Privacy properties and Verifiability. Selene provides ballot-secrecy, receipt-freeness and has a coercion mitigation mechanism. It also provides individual and universal verifiability. However, as mentioned in the previous subsection, the coercion mitigation mechanism has not been implemented in this study.

Despite this, voters were concerned about Coercion and about Privacy during elections in general. Mental models for Privacy were consistent with the properties of the system, and the reason might be that Privacy is a mandatory element required by law during elections in France, and it is taught to people at an early age at school.

On the other side, the novelty of the verification phase seemed to prevent participants from properly explaining their experience. However, indirect properties and potential issues were mentioned, such as hacking and integrity, and public tallying. It appears that participants were able to point out the potential issues of online voting without seeing that the verification mechanism was part of the solution.

As for privacy, we can argue that an early education on verifiability could lead to a better understanding and acceptance of the concept.

Trust is not a security property of voting protocol. However, it plays an important role for voters and impacts the use of a system. This aspect is important for people to accept the system they use.

Even if the convenience of online voting was mentioned many times, voters stressed their lack of knowledge about internet technologies as a big drawback. Paper-ballot voting contains steps that are understandable and accessible to people, and this is not in general the case for online voting. Even if this aspect is not required by law as in Germany, it seems reasonable that voters are more willing to trust a process they fully understand.

4.5.1.2 Impact on the voting experience

We have seen here in our analysis that the perceived security of a voting application is an important factor for trust and it was emphasized for the secrecy and verifiability concepts. Moreover, the information related to security in the expanded version of the app was shown during loading screens. It might be that the progress bar prevented

the participants from reading the information displayed below³. In a small study [73], the authors found that the voters chose more secure systems as their preferred scheme even if they scored lower on the SUS scale. It is thus interesting whether allowing more cryptographic interactions could increase the acceptance, even if it reduces the usability, we will discuss this in the next section where, in our next study, we made the encryption of the vote more noticeable in the app.

Many participants did not understand fully, or were not able to describe the verification phase despite the explanations provided in the Q&A screens. As Acemyan et al. observed in their drawings study [8], participants focused on the voting steps while avoiding the verification steps. The verification phases of each tested system was considered useless in many cases, like in our observations. On the other hand, participants who understood the interest of seeing their vote in the app did not understand why they were seeing others' votes, as their own vote and only this vote was highlighted in the application.

Olembo et al. [98] showed that specific messages could motivate voters to verify their vote, as they understand better the objective of such a procedure. In particular, they focused on risks, norms and analogies. In our application, the focus has been done on norms only, i.e. we explained what is the purpose of verification and what it brings to society. We emphasized democracy protection and integrity of votes records. Now, according to voters' models for Coercion and their concerns on hacking, a stronger emphasis on the incurred risks and solutions provided by verification might help the voter to understand.

Some people understood that the tracking number was instead the number of people voting like they did. A simple improvement is to add letters to the tracking number.

In this version of the application, the bulletin board was not accessible before the individual verification phase. One improvement could be to make the bulletin board available before the individual verification. The possibility to request a fake tracker must also be implemented at this stage. In addition, once the Selene check is available, we could show the individual vote first (with fake or real tracking number) and let the voter optionally consult the bulletin board.

4.5.1.3 Limitations

The results of our study are bounded by some limitations.

First, the user tests were done in a laboratory that had a reassuring impact on participants. Some of them admitted that they were not really feeling any threats for their vote as they were part of an experiment. The influence in a lab context on user studies is discussed in [79].

We mentioned to our participants that the elections were related to the national elections in France, however we did not use real candidate names nor run an election

³We will see in the next analysis with drawings that it was indeed the case.

that already happened, as suggested in [90].

Also, the participants had a very limited amount of time to understand the verification procedure, and the novelty of such a protocol might require more time to be understood and accepted. A broader context would be provided in real elections, giving users time to understand the process of verifiability of the application. We also assumed in our study that the configuration of the devices was already done. The ease of use might be questioned if the registration to online voting and key configuration must be performed by voters. However, this configuration could be done only once for several elections.

Finally, we asked the participants to verify their vote right after the vote casting phase. In other protocols and user studies, the verification is done during the vote casting (e.g. Benaloh challenges, or return codes). In this protocol, the verification is performed after the results have been published and due to experimental constraints, the participants had to do it right after vote casting, which could have disconcerted them.

4.5.2 The drawings study

In this section, we report the insights learned from the second user study with drawings.

Impact of Vote Manipulation. In our study, the participants used the voting application twice. In the second round, the cast vote has been modified since vote manipulations are recommended to measure the execution of a given mental task⁴ [116, 90]. In our case, this manipulation has shown a possible source of errors in an online system to the participants, and 20 participants clearly reported it. In the previous studies on the Selene protocol [40, 130], such threat was not shown to the participants and the participants had more trouble explaining why verification is useful. This, combined to the explanations provided in the application, had a positive impact on the understanding of the participants. Indeed, almost all of them had a good idea of the meaning of such a procedure in particular the correctness. Of course, we cannot trigger such an attack and make voters experience errors in a real election but it shows that being aware of the risks helps to understand the meaning of a given task.

Olembo et al. studied what could motivate voters to perform additional security checks, in particular verification steps [98]. Risk communication makes voters aware of the possible threats therefore understand better the verification mechanisms. Also, Huang et al. studied the impact of giving users a good perception of information security, with the intention of motivate them performing additional security steps [62].

⁴Manipulations are also recommended as a security check, to ensure that voters would be able to spot a mistake in the result.

Different Needs for the Users. As mentioned above, several participants had a background in computer science. Even though the sample size is small, we could not detect any relation between the background of those participants and the feedback provided. However, we can note that several participants have expressed a need of learning more details about the setup and the origin of tracking numbers, or wanted to have additional proofs. The correct understanding of the available features was not enough to convince them. This had a negative impact on participants as it raised many questions and affected their trust in the system. Many participants said that they would prefer to have information regarding the system before the elections, to ensure their correct understanding and the reliability of the system. Selene can provide additional proofs and it is specified in the original protocol that more data is available to the public and verifiable.

Besides, more than half of the participants did not consider it necessary to see the complete list of votes, even if some of them explained clearly the possibility of recounting the votes and the transparency that it provides. Transparency was also suggested in the application, as one purpose of having access to the list was the “verification by external organisations”, but was given less prominence. As mentioned in Chapter 1, one important security feature in the protocol design, not tested here, is the accessibility of the bulletin board in order to let a possibly coerced voter choose another tracking number. It has been highlighted in previous studies that this missing feature might help the voters to understand better the opportunity of accessing the complete list of votes [130].

In [14], Bada et al. acknowledged that risk awareness and understanding are prerequisites to change security behaviours. However, they also highlighted that additional factors must be taken into considerations. In particular, the adaptability to the audience and to its needs is encouraged.

Impact of the Security Communication. In the application, the security was communicated in several screens. First, several loading screens between the direct interactions with the users show the following information: authentication, encryption of votes, and decryption of the tracking numbers. Furthermore, before the vote encryption, the users are explicitly pushing a button indicating “Encrypt” to encrypt their vote. Finally, the anonymity of the trackers is explained inside the application before the verification. In the two prior studies of Selene [130, 40], the authors highlighted that the security, even visible, remained unseen by the participants of their study. A possible reason for that could have been the lack of interactivity with the security features. In our study, participants had to push an “encryption button” and we observed that all participants mentioned this feature. However, the drawings revealed that the location of the encryption’s computation was sometimes unclear. This might be due to a lack of knowledge in software design but it did not have a negative impact on the participants. On the contrary, the interaction with the encryption had a positive impact on the security concerns of the participants, as it

makes them aware that a security feature is implemented. One other possible reason for seeing encryption better is the demographics of our participants. Surely, our participants were tech-savvy users that might be naturally prompt to notice security features. However, similarly to the previous studies, participants did not notice the decryption of tracking number, since it was mentioned only in the loading screen.

4.5.3 Recommendations

We conclude with four recommendations to inform the design of future verifiable voting schemes.

Transparency and Correctness. The security concern regarding correctness was often mentioned during interviews when explaining the meaning of the verification phase. As discussed above, one reason might be the impact of the vote manipulation, but we can also mention the verification app that gives several insights on verifiability to voters, among which the correctness of records was cited. On the other side, a few participants only mentioned transparency, but this was not enough to convince them for seeing all votes. For future implementations, our first recommendation concerns the clear designation of each entity that a user might deal with and their purpose, to ensure a complete understanding of the expected tasks.

Educate voters on risks. As mentioned above, the vote manipulation made participants aware of possible risks related to online voting and let them understand better the meaning of the verification phase. We have highlighted that communication of the risks, control over verifiability procedures and easy security interactions will lead to a better understanding of the tasks one must perform. To be verified, an online voting system needs to convince enough voters to perform those additional individual checks. Early access to the voting system is recommended to train and to educate voters on the possible risks related to online voting, and how to counter them. The Swiss Post voting protocol provides such access to voters [117].

Provide simple interactions with a security emphasis. The simple interaction with the encryption button has shown to raise the awareness of participants regarding the security implementation. Also, the other screens in the app, where security was shown without interaction, were mentioned by participants only twice. This confirms the previous studies with this voting protocol [40, 130]. Therefore, we recommend communicating security through simple interactions whenever possible. Following the example of the encryption, naming the security tasks on a simple interaction like pushing a button is enough to raise the awareness of users. In a study on the Swiss Post system, Marky et al. also give this recommendation [89].

Provide different levels of verification. Many participants understood but were not always convinced by the provided proofs, while other participants have found some information unnecessary. One of our recommendations concerns the verifiable data and while the steps should remain easy to perform, the information could be split and displayed only on demand. We can distinguish three levels of verifiable information: 1) a minimal display with only the individual vote, 2) a full user-friendly display with the entire list of votes, and 3) a detailed specification on how to perform additional checks. This last level will let all voters (and external observers) who have an expertise in computer science to verify more steps of the protocol.

4.6 Conclusion

In this chapter, we have seen the mental models from two different points of view: first we studied the gap between the properties developed in voting research and the voters' expectations, then we looked at the understanding of voters. We have seen that the properties aimed by the voting protocol are consistent with the voters' expectations; however, they don't always understand verifiability as a solution for ensuring the integrity. Then, we have seen that simple security interactions make the protocol's steps more understandable to the voters; still all of them don't have the same need of control on the voting system. Showing risks associated with a system also helped to understand the reasons behind the verifiability steps. Indeed, the manipulation of the vote in the second study has shown the advantage of vote verification in case of a problem. Overall, we have learned how to improve voting interfaces to be closer to what voters' expect from a security point of view, but also to help them understand the purpose of extra features such as individual verifiability.

Chapter 5

Trust and understanding of voters: evaluation of a coercion mitigation mechanism

In this chapter, we want to evaluate the voters' understanding of the system at a larger scale. In the previous chapter, we did a qualitative analysis of interviews and drawings, which has the advantage of going deep in the evaluation and our understanding of the voters' expectations. We found that despite a user-centred design and a good usability, participants had trouble to understand some aspects of Verifiability. Some of them questioned the purpose of the procedure, or had trouble explaining how it works. In this chapter, we evaluate again the voters' understanding with respect to an additional feature of Selene, namely the coercion mitigation mechanism. We take a different approach, we evaluate the voters' understanding of the Selene protocol through a game: we give instructions to participants with a specific scenario and incentives associated to their choice. Our goal is to observe the strategy adopted, we want to assess their understanding with our game design. This allows us to evaluate voters at a larger scale. Furthermore, we have seen in the previous chapter that trust was an important aspect to take into account when evaluating the understanding of a protocol. In this chapter, we also propose a definition of trust in the context of voting and propose a new questionnaire evaluating the voters' trust in terms of acceptance and feeling of security regarding the protocol. We relate this information to the understanding results, and see if there exists a correlation between them.

5.1 Introduction

In the previous chapter, we have seen that the study of mental models helps to evaluate the gap between a system's implementation and the voters' perceptions. We have learned that this gap might come from a lack of understanding of the procedure, as

our participants could not properly explain their experience nor understand the reason behind the verifiability procedures. One hypothesis is that the Selene experience was not complete in our previous work, as not all features were implemented. Also, a lack of information in the app could cause a misunderstanding, and we have already seen that communicating clearly on encryption helps the participants to acknowledge this security feature.

In this chapter, we evaluate the voters' understanding using a web application, in which we have implemented additional features (see description in chapter 3). To evaluate the voters' understanding, we designed a user protocol with a specific scenario inspired by game theory, as experimented in [83]. Game theory is the mathematical study of interaction between agents, each of them having their own interest. It has been used widely in security, here we produce a scenario involving a vote buyer. When the game starts, a vote buyer asks a voter to cast a specific vote in exchange for a reward. The voter has the choice to follow or not his request. The voter receives a reward if he follows the buyer's instruction, and another if he keeps his vote intention. Selene offers the possibility to do both thanks to the coercion mitigation mechanism. By doing this experiment, we want to see if the participants understood this aspect of Selene: a participant who did understand will manage to receive both incentives.

One hypothesis is that a participant will have a rational behaviour in the sense of game theory: when the system is well understood, the rational behaviour is to maximize the pay-off, and to go for the dominant strategy. In addition to the understanding's measurement, we asked the participants to give feedback on their decision to have more insights on their understanding. From this feedback, we categorized the participants' answers into several categories that will let us know more about their experience.

Besides, we want to evaluate the voters' trust in the application. Trust has been a recurring factor in several user studies on voting, e.g. [98, 114, 88, 130]. However, there is no questionnaire available today to assess this metric in voting. In this chapter, we define trust in the context of voting and propose a new questionnaire assessing the voters' trust in the tested protocol. This questionnaire has two subscales: acceptance of the system and security perception. Trust in security and in particular in voting has been discussed in [102]. The author builds a link between the voters' trust and the explanation provided of a security feature, showing that a good understanding of a system could increase the voters' trust. From the measurement of trust and understanding, we have evaluated the correlation between those two concepts, and we discuss our findings in the next sections.

Our contributions are:

- A new definition and questionnaire to evaluate trust in the context of voting,

- A unique game design to assess the voters' understanding of a system at a large scale,
- An evaluation of the relations between understanding and trust,
- A list of recommendations for future voting systems.

The chapter is organised as follows: Section 5.2 gives the related work, in Section 5.3 we define trust and give our questionnaire. Section 5.4 provides our user protocol and the game design. Section 5.5 gives our study results. We conclude in Section 5.6.

5.2 Related Work

Llewellyn et al. [83] used a similar approach for the evaluation of Prêt-à-Voter (PaV) [108] (without any coercion threat). They designed a simple game that they run into groups of 12 participants. Each subject must cast a vote with the PaV system. In PaV, voters cast a vote with a paper ballot on which the set of candidates is displayed according to a permutation. Each ballot is identified with a so-called *onion*, encrypting the permutation of candidates. After putting a cross next to their favorite candidate, voters must detach the ballot into two parts, one containing the candidates' list, and the other their mark choice with the onion. The candidates' list is destroyed, while the other part is scanned and taken as a receipt. After the tally, the scanned receipts are published on the bulletin board, voters can verify if their ballot appears. In the game proposed by Llewellyn et al., participants had the choice between publishing their receipt or not. The game aims to evaluate whether a participant understands that making public their receipt will not reveal their vote choice.

The game works as follows: each participant marks their choice on the PaV ballot, and must indicate if he wishes to post the receipt or not on the bulletin board. In case the participant agrees to post the receipt, he receives £1. Otherwise, he receives nothing. Then, the receipts are posted according to the participants choices. Each participant must guess other participants' votes, whether there is an available receipt or not. For each correct guess, a participant receives £0.5. For every correct guess from another participant, a participant loses £0.5. Of course, publishing the receipt does not reveal any information regarding a vote due to the encryption, so the idea was that a participant who understood the system will publish the receipt as it brings him £1.

The authors have run up to 6 rounds in one experiment, letting them observe any evolution in the understanding while using the system several times. However, the low number of participants makes it hard to give any conclusion.

The understanding of voters is usually evaluated by looking at their mental models. We have learned some elements in the previous chapter, and other papers also explored those aspects. In [8], authors let voters draw their mental models with three voting schemes. This study reveals that voters focused much more on the voting phase in all three protocols, as the verification features remain unclear for them. So far, most of the studies focused on the usability and appreciation of voters for a given system, but a true evaluation of their understanding is rarely performed.

5.3 Trust

Trust is an element that comes back in many studies about voting [114, 73, 88, 130]. It is rather complex to evaluate as trust can be impacted by many aspects related to elections or to the media used: trust in politics, trust in internet technologies (in the case of internet voting), understanding of the app, etc.

Today, there is no standard questionnaire available to evaluate trust of users, in particular for voting systems. We studied the literature and we designed our own questionnaire. In the following, we will introduce the notion of Trust and our questionnaire.

5.3.1 Definitions

General definition From a general perspective, Riegelsberger et al. [104] defined trust with *contextual properties* and *intrinsic properties*. The contextual properties concern the elements that will provide a motivation to trust: temporal (repeated interaction), social (general reputation) and institutional (trusted third parties) embeddedness. The intrinsic properties concern the ability, the motivation based on norms and the benevolence of a trustee. The authors point out that trust is necessary in situations where a detailed knowledge about the object of interaction cannot be obtained. The contextual properties will be important in the first interactions while intrinsic properties are important in continuous exchanges. In [87], authors studied trust and privacy implications in a learning process. They found out that users are more likely to trust their colleagues when there are trust signals. Signals are split in two categories: the symbols, with an arbitrary meaning (e.g. security logo, uniforms, etc.) and the symptoms, showing a specific quality of a system (e.g. a good usability).

Trust by Design In [70], the authors highlighted the need to consider trust in security design¹. In particular, they discussed the usual behaviour from organiza-

¹This paper applies in the context of an organization with the relationship employer-employee, here we look at the use of a security procedure outside a company, e.g. Verifiability required during an election.

tions in information security which consists in contextually-incentivize trustworthy behavior of users with enforcement. This approach assumes users are not motivated to exhibit trustworthy behavior and one of the resulting problem is distrust. Authors proposed a list of improvements to gain trust which are: improving usability, improve awareness and education, improve participation and improve accountability. We will see in the next sections that our studies lead us to similar guidelines in the context of voting. In the next paragraph we will define trust in the context of voting.

Trust in the voting context First there is a difference between ‘confidence’ and ‘trust’. In [84], Luhmann defines confidence as “self-assurance of the safety or security of a system without knowing the risks or considering alternatives”; and trust as “self-assurance by assessment of risks and alternatives”. Confidence can be obtained without any additional explanation, in particular security does not need to be perceived to be acknowledged. In contrast, trust requires an evaluation from the users’ of their security perception, that involves a minimal level of understanding, to be granted.

In [102], Pieters highlighted the relationship between trust and explanation. In the previous chapter, we have studied mental models that are the users’ perception of a system, and our goal in the design of our voting app was to reduce the gap between the system itself, its image given through an interface and what the final user understands from it. One insight to reduce this gap is the explanations given to the users [102]. In our previous work on mental models, we have seen that trust was related to security, in particular to the concepts of coercion, secrecy and understanding of security concepts. Explanations are there to fill the gap between ‘actual security’ and ‘perceived security’, and help users in their understanding, but also to increase their trust. As we already mentioned in chapter 4, awareness and education are key improvements to a good security design.

Pieters [102] highlights that a voting system can obtain the voters’ confidence if it works properly. A system that guarantees a correct result will not worry the voters. But, when a new system implementing new procedures, such as verifiability features, comes in with a comparison to the old system which has the confidence of voters, trust and especially distrust takes the place of confidence. Today, most of the voting systems used in national elections don’t implement verifiability, with the exception of Australia [38], Switzerland [117], and Estonia [47]. Meaning that most of the systems in use today are based on the voters’ confidence, and we need the voters to understand verifiability to implement a new system and convince them to use it. Previous works have already mentioned the relationship between trust and the explanations. Glass et al. [55] conclude that trust depends on granularity of explanations and on transparency of the system.

In this study, we aim to provide a reasonable amount of information regarding our protocol, in order to increase our chances of having a good trust rating. However, the participants will have a limited amount of time to evaluate our app, so we must

not provide too much information that could overwhelm them.

5.3.2 Our metric

Here we propose a new voting-oriented questionnaire containing eight assessments that a user study’s participant must answer. These aspects are separated into two subscales: the acceptance of the tested application, and the feeling of security. It will measure the effective trust in our system.

As mentioned above, Trust needs an evaluation of the security perception to be granted. Hence, we evaluate the feeling of security on one side, and the acceptance, or willingness to use the application again on the other side.

Usability and understanding are also key aspects when evaluating trust. Hence, those elements must be evaluated as well for trust.

Our questionnaire is given in table 5.1 and evaluates the participants with a Likert scale.

To obtain a score from this metric, we use a similar logic as for the System Usability Scale. We proceed by giving 10 points per question: for the one expressed with a positive tone (1 to 7), we remove one point per question ensuring that a “strongly disagree” will give 0 point. For the one expressed with a negative tone (8), we change the score by doing $6 - score$ to ensure that a “strongly agree” will give 0 point. We sum up the obtained score per question, and multiply the total by 2 to obtain a score up to 10.

The total score can be up to 80 points.

5.4 Game design to evaluate voters

5.4.1 Methodology

To evaluate the understanding and the usability of Selene, we designed a user game inspired by game theoretic experiment in [83]. We recruited 300 participants on the crowd-sourcing website Prolific. The details about participants and the user protocol are already given in chapter 3, we quickly recall the steps here:

1. Consent form and demographics
2. Video explaining the protocol
3. Application tutorial: test of the application with additional instructions to guide the users
4. Questionnaires: we gave the User Experience Questionnaire and the System Usability Scale, results are available in chapter 3. Participants also filled our

	Strongly disagree					Strongly agree
[Acceptance] I trust the system and I would use it in a real election	1	2	3	4	5	6
[Security] I believe that the personal information (vote included) is kept private	1	2	3	4	5	6
[Security] I think that the system ensures the integrity of the elections.	1	2	3	4	5	6
[Security] I think that the system is transparent and lets me know everything about its behaviour	1	2	3	4	5	6
[Acceptance] I think that the verification phase is important	1	2	3	4	5	6
[Security] I was convinced by the verification phase that my vote was correctly recorded	1	2	3	4	5	6
[Acceptance] I would use such a verification system if it was available	1	2	3	4	5	6
[Security] I think that the result of the election can be changed by an attacker	1	2	3	4	5	6

Table 5.1: Our Trust Questionnaire

Trust Questionnaire, and answered three questions regarding their understanding of the protocol.

5. Game: the participants are asked to use the application a second time, taking into account the context of vote buying. A scenario is described, where a vote buyer asks the participants to cast a vote in exchange of a monetary reward (the details are given below).
6. End: to finish the study, participants are asked to tell which choice they have made and why, as well as how they felt during the test. A debriefing page is shown before redirecting them on Prolific.

The complete study description given to the participants is available in the appendix C.1.

5.4.2 Voting question

In the demographics' questionnaire, we asked the participants to give their opinion about the COVID-19 crisis to help us configure the game. The question was displayed as follows:

Regarding the recent events related to the COVID-19 pandemic, according to you, what would be the best policy to adopt at the beginning of the epidemic?

- A strict confinement for all
- No confinement but detection tests available for everyone

We had no interest in the actual answer, but this let us configure the game by changing the vote buyer's instructions according to their opinion. If they chose "A strict opinion for all", the vote buyer asks for "No confinement but detection tests available for everyone" and vice versa.

5.4.3 A tutorial to show the coercion mitigation feature

As described in chapter 3, we let the participants use the application through a tutorial before they play the game. This was done as a training round, to ensure that they see and test all available features in the application, and know all the tools they need in order to play our vote buying game. We evaluated their feedback on trust through our questionnaire after this tutorial phase. The reason was that we did not want to influence their trust rating through a coercive scenario, but grab their general impression of the app.

Also, we put a few attention checks at the beginning of our questionnaire. This information was announced in the study description, to increase the attention given to our explanations in the app. We asked three questions to which answers were highlighted in the app or clearly mentioned in the video:

1/ How many phases are described?

2; 3; 4

2/ 'When are the tracking codes created and associated to the voters?'

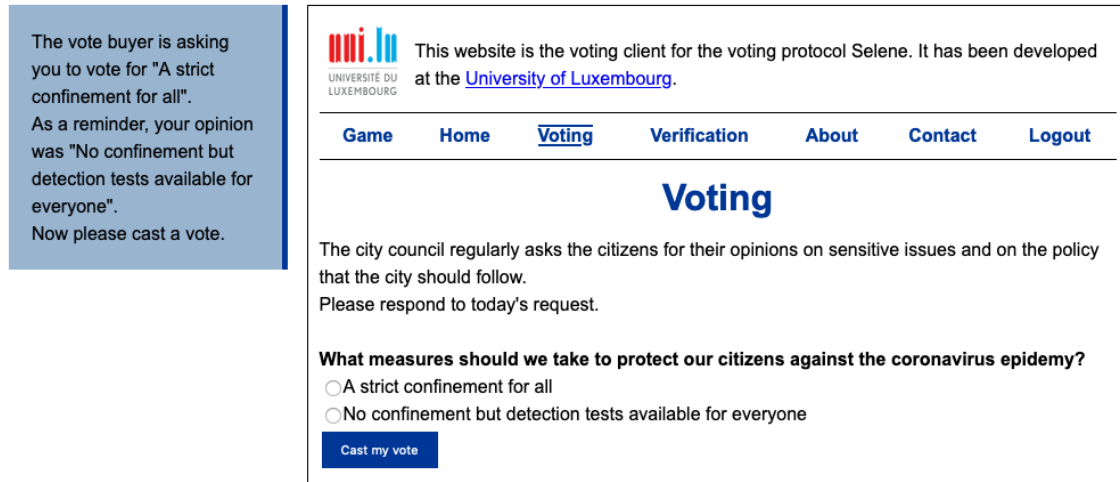
Before the elections starts; After vote casting; Just before being sent to the voters

3/ How can you conceal how you voted?

By contacting the authorities; By choosing another tracking code on the bulletin board; I can't

Our goal was not to punish participants who give a wrong answer but to help them focus on the information inside the app.

5.4.4 Vote-buying Game



The screenshot shows a web interface for a voting client. On the left, a blue box contains the text: "The vote buyer is asking you to vote for 'A strict confinement for all'. As a reminder, your opinion was 'No confinement but detection tests available for everyone'. Now please cast a vote." The main content area has a header with the University of Luxembourg logo and text: "This website is the voting client for the voting protocol Selene. It has been developed at the University of Luxembourg." Below the header is a navigation menu with links: Game, Home, Voting (underlined), Verification, About, Contact, Logout. The main heading is "Voting". The text below reads: "The city council regularly asks the citizens for their opinions on sensitive issues and on the policy that the city should follow. Please respond to today's request." The poll question is "What measures should we take to protect our citizens against the coronavirus epidemy?". There are two radio button options: "A strict confinement for all" and "No confinement but detection tests available for everyone". A blue button labeled "Cast my vote" is at the bottom.

Figure 5.1: The vote buyer's instructions

We introduced the game by indicating to the participant that he will receive instructions from a vote buyer. The rules are given as follows:

A vote buyer wants to buy your vote by giving you a vote instruction. He may ask you how you voted and to reveal your tracking code, in which case you can give an alternate code.

If you send a tracking code for the requested candidate, you will receive 70 pence from the vote buyer.

If you want to keep your vote intention, you will receive 30 pence.

These incentives will be provided as bonus payment after the study.

When a participant starts, he was asked to vote as he did in the tutorial but additional instructions given by the vote buyer are written on the left side (see figure 5.1). The participants could decide to follow the vote buyer or not. Our idea was to determine whether the participants understood that they can keep their vote while convincing the vote buyer that they follow his choice. Indeed, the dominant strategy for a player, regarding the possibilities offered by the application, is to keep his vote intention while selling a tracker to the vote buyer. After computing his tracking number, the participant could choose to send it to the vote buyer or not by clicking on a button behind the instruction.

To finish the study, the participants were asked to tell us which choice they have made - keep their vote intention or follow the vote buyer's instructions - and why. Our last question was about how they felt during the game.

5.4.5 Ethical approval

We obtained ethical approval from the Ethics Review Panel of the University of Luxembourg. Our work is compliant with GDPR and research terms of Prolific.

5.5 Results: Evaluation of Voters' Understanding of the coercion mitigation mechanism

5.5.1 Quantitative results

5.5.1.1 Trust

As mentioned above, the questionnaires have been filled after the tutorial phase and before the game. The reason was to let the participants give an evaluation of the app and of its features before we collect the data regarding their understanding. We did not want a specific scenario to influence their opinion on the protocol itself.

The results concerning the UEQ and SUS questionnaires are already given in chapter 3². Here we add the results received on the trust questionnaire described above. Overall, trust received an evaluation of 46.81 ($SD = 16.132$, $Min = 4$, $Max = 78$). On the subscale level, the acceptance (over 30) was rated 18.59 ($SD = 7.264$, $Min = 0$, $Max = 30$) and the feeling of security (over 50) was rated 28 ($SD = 20.093$, $Min = 0$, $Max = 48$).

We have computed the Pearson correlation coefficient $r = 0.561$ ($p = 0.01$) between our trust and satisfaction measures, meaning that there is a moderate positive correlation between trust and usability. Similarly, the coefficients computed between Trust and the UEQ's scale are given in table 5.2.

	Attractiveness	Perspiciuity	Efficiency	Dependability	Stimulation	Novelty
r	0.14*	0.135*	0.149**	0.151**	0.173**	0.063

Table 5.2: Correlation coefficients between Trust and UEQ scales, *: $p = 0.05$, **: $p = 0.01$.

The values for r are below 0.2 indicating a weak positive relation.

5.5.1.2 Understanding

We have counted the number of participants who kept their voting intention and managed to fake their tracker. In total, 54 participants have taken the dominant

²As a comparison, we have also checked the SUS and UEQ results from the first pilot study. Satisfaction was rated at 80,5 ($SD = 8,75$, $Min = 65$, $Max = 90$), and UEQ items were rated higher too.

strategy to receive the reward for keeping their vote, while receiving the reward from the vote buyer.

We have also counted the correct answers given to our attention checks. To the first question, ‘How many phases are described?’, 106 participants gave the correct answer of four. To the second question, ‘When are the tracking codes created and associated to the voters?’, 81 participants answered correctly ‘Before the election starts’. Finally, to the question ‘How can you conceal how you voted?’, 296 participants gave the right answer that was ‘By choosing another tracking code on the bulletin board’.

5.5.2 Qualitative results

We analysed the answers from the game and also the feedback provided through our two last questions. We categorized the answers into the following categories:

- To the question “Why have you made this **choice** in the game?”: money, integrity, understanding, experimenting and miscellaneous.
- To the question “How did you **feel** during the study”: overwhelmed, stressed, offended, good, interested, confident, confused, observed.

Then, two researchers coded independently the interviews and compared their findings. They discussed the categorization and solved the disagreements. We tracked the disagreements and the Cohen’s Kappa was calculated at:

- 0.841 for the self-explanation question, that is we had almost perfect agreement,
- 0.714 for the feeling question, which means that we had a strong agreement.

In the following, we detail our findings.

5.5.2.1 Self-explanation

Regarding the understanding, we measured if a participant managed to fake his tracker while keeping his own vote, by looking at his decisions recorded on the server. As mentioned above, we counted that 54 of the 300 participants succeeded in doing so. To our first question regarding their choice, 19 participants simply explained that their correct **understanding** of the feature:

“I could vote as I wanted and still be paid thanks to the concealed voting.”

(P21)

“The game made it perfectly possible for me to select what I personally wanted to vote for, yet fool the vote buyer into thinking I had voted for what they wanted. Win/win situation.” (P178)

“I voted as I wanted but then concealed my vote by changing to what the vote buyer wanted. So I cast the vote I wanted but made it appear to him that I voted for what he wanted.” (P291)

We found that 155 participants mentioned **integrity** as being their motivation behind their decision. Inside this category, participants believe that their intended vote matters, that vote-selling is illegal, or just that they care about their own integrity and could not disregard their opinion for money:

“It’s important not to buy votes, even in a game.” (P10)

“My vote should not be changed for monetary gain.” (P22)

“There is no point in voting if it is not fair.” (P48)

“I think it is important to vote for yourself and disregard any external influence.” (P109)

“I feel strongly against rigging any form of democratic elections.” (P175)

“I am not followed by greed and I did as I wished.” (P257)

“Voting is important and you shouldn’t let anything sway your vote.” (P287)

We can also cite P106 who understood the feature but felt bad about using it: *“I don’t like the idea of selling a vote, or of using a feature of the system intended to enhance privacy in order to lie and gain some reward.”*

Then, we found that 60 participants confessed having followed the vote buyer’s instructions because of the **monetary** incentive, with or without justification:

“I knew the situation was not real and I would prefer to receive a higher reward, I would probably not do this in a real situation.” (P32)

“I got more money for doing so.” (P67)

“There was an extra incentive to change my vote.” (P225)

We also found that 22 participants tried to use the application for **experimenting**, to prove a point, or to see what will happen if they follow the vote buyer, or tried to test a feature. In particular, P5 followed the vote buyer and explained: *“I was intrigued to see what would happen if I selected that option”*.

Other participants wanted to prove the lack of security of the voting protocol:

“I wanted to show how corruptable this method of voting is.” (P160)

“It shows that there are indeed vulnerabilities in the system and that it could be easily influenced.” (P239)

Some other participants put their trust in the system and wanted to see if the application can deal with corruption, e.g. P273: *“To test the system and how corruption can be detected and avoided”*.

Finally, 44 participants gave a feedback that could not be classified in the above categories, for various reasons: it does not answer our question, e.g. P28 *“That’s how I felt”*, or their feedback was unclear, e.g. they might be explaining their voting choice *“It is very important to contain the pandemic”* (P161).

5.5.2.2 Feeling

In this section, we analyse the various emotions reported by the participants.

First, 89 participants mentioned their **confusion** and difficulties to understand the application. Especially, the verification and the concealing feature were hard to understand for participants:

“A little confused when it came to the conceal vote part.” (P12)

“The verification part is confusing.” (P128)

“I felt a bit confused by the complexity, why is all the verification necessary? Normally when I vote, I vote. And that’s it.” (P171)

“I felt a bit confused about whether I could tell the voter I’d voted for what he wanted even though I hadn’t.” (P235)

Then, 66 participants said that they felt **good**, enjoyed the study, were calm or relaxed:

“I felt good. I enjoyed this task.” (P82)

“Pleased that I was able to conceal my true vote.” (P137)

“During the test I felt calm and in control” P(267)

In this category, some participants said being a bit confused but after the tutorial they felt comfortable, e.g. P70: *“I felt fine, it was slightly confusing first time round.”*

We counted that 34 participants felt **confident** or focused during the study, e.g. P91: *“I felt confident in picking my own vote and not changing. Felt safe that my personal information wouldn’t be compromised.”*, or P183: *“I felt very positive and very focused”*.

Then, 28 participants were **interested**, curious or motivated by the study:

“Interested to see the potential future of voting.” (P273)

“During the test I found it interesting and creative however it was a bit hard to use.” (P80)

We counted that 30 participants were annoyed, **offended** or frustrated by the study’s steps, especially the vote-buying scenario, for example:

“I felt annoyed by your frustrating voting system.” (P77)

“I felt frustrated that someone could buy peoples vote to get them to vote the way that they wanted.” (P125)

“I felt unsure at points. Offended at the thought of a lack of true democracy.” (P68)

Then, 19 participants highlighted that the study was **overwhelming** and tiring, complaining about their lack of understanding or complexity of the tasks again. For example:

“During the test I felt exhausted and tired.” (P14)

“I felt overwhelmed by the system” (P103)

Finally, 6 participants emphasised that they were feeling **observed** or manipulated, e.g. P113 who said that *“during the test, I felt manipulated”*.

5.5.3 Relations between variables

While the questionnaires were filled after the first phase (tutorial) of the user study, the understanding of participants and the qualitative data were collected after the second phase (game). In particular, the vote-buying scenario might have impacted some participants’ feedback especially their feeling regarding the study. The following correlations need to consider this limitation.

5.5.3.1 Trust and Understanding

When defining trust, we made the assumption that the explanations provided were important to give transparency and to increase the voters’ understanding in the application. During the study, we gave explanations through video and text, participants followed a tutorial before playing a game designed to evaluate their understanding of the features. Naturally, we check the correlation between the Trust results given above and the understanding of voters, measured through their decision in the game.

The understanding has been measured by looking at the capacity of a participant to keep his vote while faking the tracker for the vote buyer. We obtained one group of 54 participants who understood, and another group of 246 participants.

To measure a correlation between trust and understanding, we performed an independent t-test. The study found that the participant who understood the concealing feature gave a statistically higher evaluation of trust ($Mean = 51.22$, $SD = 15.372$) compared to participants who did not understand it ($Mean = 45.84$, $SD = 16.163$), $t(298) = 2.236$, $p = 0.026$. Further, Cohen’s effect size value ($d = 0.34$) suggested a small to moderate practical significance.

5.5.3.2 Understanding and Time Spent in the Study

When a participant logged in our platform, we recorded the session length. We counted 47 participants who finished the user study in less than 20 minutes (which was the amount of time we planned). The mean is around 2155 seconds (35 minutes and 55 seconds) for all participants. Participants took more time than what we planned, probably because of our attention checks, added after the pilot studies. We run a one-way ANOVA test, which has shown no significant difference between the group of participants who understood the game and the other participants.

5.5.3.3 Self-explanation/Feeling and Understanding

Out of the 54 participants who faked their tracking code to send it to the vote buyer, 26 mentioned integrity, 3 money, 17 gave an explanation about their understanding. On the other side, 2 participants explained correctly how the system works, but did not fake their tracking code for the vote buyer.

Regarding their feeling, 22 participants over 54 mentioned that they were confused, 25 that they were feeling good, confident or interested in the system, the remaining 7 were feeling observed, stressed, overwhelmed or frustrated.

We run a Welch ANOVA test between the decision categorization and the understanding, which has shown no significant differences between the five groups ($p > 0.05$). Hence, the understanding of participants is not related to the reason for following or not the vote buyer.

Similarly, we found no significant differences between the 8 groups of feelings ($p > 0.05$). Hence, the understanding of participants is not related by the feeling of participants.

5.5.3.4 Self-explanation/Feeling and Trust

The relation between the decision's categories and the trust assessments is analyzed with a 1-way ANOVA. The ANOVA test shows a significant difference between the five categories ($F(4, 295) = 2.872, p = 0.023$). A post-hoc Tukey is run to locate differences between categories, and found that participants who mentioned integrity rated trust better (8 points) than those interested in money ($p = 0.016$).

On the other side, there was no significant differences between the 8 groups of feelings ($p > 0.05$). Hence, the participants' trust (evaluated after the tutorial) was not influenced by their feelings (evaluated after the game).

5.5.3.5 Self-explanation/Feeling and Usability

We run a 1-way ANOVA test to investigate a relation between the SUS assessments and the self-explanation provided. The test shows a significant difference between the five categories ($F(4, 295) = 2,729, p = 0.029$). A post-hoc Tukey found that

participants who mentioned an experimentation gave a better evaluation than those doing the test for money ($p = 0.049$).

We run again a 1-way ANOVA test to find a relation between the feeling’s categories and the SUS assessments. The ANOVA test shows a significant difference between the 8 categories ($F(7, 292) = 3.446, p = 0.001$). A post-hoc Tukey found that participants who felt interested in the application rated better than those feeling overwhelmed or stressed ($p < 0.05$).

The details of the analysis are given in table 5.3 (we report those with a significant difference only).

	Difference between the means	P value
Experimenting over Money	15.48	0.049
Interested over Overwhelmed	21.34	0.039
Interested over Stressed	21.34	0.009

Table 5.3: Post-Hoc Tukey significant results between feelings and SUS scores.

Similarly, we run a 1-way ANOVA to find relations between the UEQ items and the categories for self-explanation and feelings. For self-explanation, no relation was found ($p > 0.05$). We found a relation between the feelings’ groups and the UEQ items with statistical significance ($p < 0.001$). The details are reported in appendix C.2. Overall, participants having a positive feeling regarding the app rated it better than the other participants with $p < 0.05$.

5.5.4 Analysis

5.5.4.1 Limitations

While having used Prolific brought many advantages, including the reachability of many participants in a small amount of time and good demographics samples, we found some limitations.

As mentioned earlier, some correlations were shown between the participants’ feelings and other measurements, done before the vote-buying game. The feelings of participants might have been impacted by the game and their relation with usability and trust can be mitigated.

As already mentioned in chapter 3, our first pilot study has shown that participants are rushing to probably increase their reward per hour. Without any guidance, we could not hope that participants will visit all pages in our app, forcing us to make them test the app through a tutorial rather than exploration. This bias in mentioned as “satisficing” and is acknowledged by Prolific [103]. To counter this, we

asked participants that questions regarding their understanding in the app will be asked in the questionnaire: these “attention checks”, are recommended by Prolific, helped us to lower this bias.

We also noticed that some participants do not read the study description, where information like connection credentials and guidance is provided. Many participants contacted us during the study because they did not know how to proceed, especially which credentials to use, while all information was given there. This might have given them frustration while doing the study and influenced their evaluation.

Another limitation concerns our scenario with vote-buying. As for studies in lab, participants might have a bias to give a good image of themselves, hence answering what would be ethically acceptable [79]. On the other hand, some participants justified themselves for having followed the vote buyer because “this is just a game”. Finally, as for other studies, we ask participants to understand new features in a very limited amount of time. More time would be necessary to understand the feature and especially why we implement them.

5.5.4.2 Discussion

The big appeal for *integrity* was not expected in our experiment, as our hypothesis was that participants will pick the dominant strategy if they understand the features and are rational. The feedback provided by the participants has shown that voting is an important matter for them and even if they can deceive a vote-buyer, their own integrity matters more. Unfortunately, we can’t say that our understanding measurement is exhaustive. Furthermore, no relation has been found between the understanding of participants and their self-explanation or feeling regarding the application. However, we have seen that trust and understanding are correlated, which is still satisfying our hypothesis.

In chapter 3, we have seen that the user experience and the usability were badly rated. Here we found a positive moderate correlation between satisfaction and trust, but only a small correlation between UEQ items³ and trust. In the SUS questionnaire, some items concerns the acceptance of the tested application, which is one aspect of our trust questionnaire, that might explain the higher score in the correlation.

However, we can still argue that a good user interface will benefit a voting application. In [87] and [70], authors mentioned the signals impacting trust, including usability. We had good results regarding the effectiveness and the efficiency, but we failed at convincing the participants that our application was easy to use and enjoyable.

To explain this, we can look at the feelings that were formulated by the par-

³There was no correlation for the item Novelty.

ticipants. The most expressed feeling was *confusion*: the participants were unsure about the steps to follow in the app. One reason could be the lack of linearity, even if our instructions forced participants to follow a certain workflow in the application. Another reason that was highlighted was the complexity of the study, while Prolific’s users are used to surveys, which are linear and require less commitment (in the sense of direct interactions influencing the behaviour of the app) from the user. The other feelings that were expressed by participant were *stress* and *frustration*: while we tried to provide more guidance to ensure that our app will be correctly tested, it has removed the freedom to navigate and has added complexity to the tasks. However, we also found that 128 participants had a positive feeling about the study (feeling *good*, *interested*, or *confident*), mentioning their curiosity for online voting or their satisfaction regarding the security of the app. Those participants also rated the usability and UX of the application better than the others, supporting our previous idea of the benefits of a good interface.

5.5.4.3 Recommendations

Here we provide a list of recommendations for future user study designs.

Focus on understandability In this study, we have found that participants who understood our security features have rated trust higher than the other participants in general. However, it was highlighted many times that our application was still confusing and tasks were too complex. It remains crucial to provide an interface that is transparent, with features that are understandable from voters, as it will increase their trust hence acceptance of the voting system.

Provide an easy-to-use interface While we must provide understandable and transparent information to participants, it also remains important to keep the interface as simple as possible. People who got stressed and overwhelmed by the application were less satisfied by their user experience. Indeed, we found that the participants who rated the application better had a positive feeling during the study. Moreover, we already highlighted in chapter 3 that the coercion-mitigation feature has added complexity to users who did not have a need for it. Hence, we recommend remaining simple and straightforward, keep the workflow as linear and guiding as possible, even though a minimal amount of information must be provided.

Raise awareness and improve education Many participants highlighted the illegality of our procedure in their country. To them, the fact that the law is already designed to counter some threats is sufficient to trust the system. However, if a voting system is not robust enough nor software independent, it opens a door to attackers who will not risk being caught. We recommend communicating on good practices in security, on possible risks that could exist in voting and could raise from

a misuse of the procedure. A good education, as already highlighted in our previous work on mental models and in [70], is one key to trustful applications.

Lower complexity and simplify user studies We have discussed above that many participants were confused during the test, even though instructions were available and guidance was provided during the entire user study. Besides, in our previous studies, we have already learned that the concept of Verifiability could be hard to understand in the small amount of time that we provide. Here, in addition of Verifiability, we have tested a coercion mitigation feature that has increased the complexity. Furthermore, we have learned that the Prolific’s participants needs guidance to follow a study correctly, as they won’t take time to explore an application. We recommend simplifying user studies and make them more linear in this context.

Adapt the scenario to the audience We have discussed above that the participants mentioned many times the importance of integrity, and many argued that the scenario was illegal. As we have seen in the previous paragraph, we can’t add complexity or information to the study as it is better to simplify our interface. Furthermore, this also highlight the absence of need for a coercion-mitigation feature for most our participants. For future user study designs, we suggest adapting a scenario that will be more realistic to the tested audience. Indeed, participants did not see the necessity of such a feature, as the associated scenario should not happen thanks to the law. Of course, it does not mean that such a scenario is unlikely, especially because we also had positive feedback from people believing it happens, but it will benefit both the participants, who will feel more comfortable regarding the tasks, and the research, because the measure of understanding would be more accurate.

5.6 Conclusion and Future Work

In this chapter, we have defined trust and have proposed a new questionnaire to assess trust in voting systems. We also described a user study where we have evaluated the Selene voting system, including the coercion mitigation mechanism. Our application has been tested by 300 participants, we have evaluated their experience by measuring their understanding through a unique game design, and their trust in the system. We found a relation between trust and understanding, and we gave a list of recommendations to follow in order to increase trust and usability of voting applications. Our recommendations are: 1) Focus on the understandability, 2) Provide an easy-to-use-interface, 3) Raise awareness and improve education, 4) Lower the complexity and 5) Adapt the scenario to the audience. We also have found some limitations in our study, that one should try to mitigate in future studies. This was one of the first user study that investigate a coercion-resistance feature (see [94]

for another example). For future research, it would be interesting to compare the feedback from another country, where our scenario is more common.

Chapter 6

Electryo: to a paper-based e-voting protocol

In this chapter, we will adapt Selene to a conventional paper-based system. The Selene mechanism can be applied to many e-voting schemes, but here we present an application to the polling station context, resulting in a voter-verifiable electronic tally with a paper audit trail. The system uses a smartcard-based public key system to provide the individual verification and universal eligibility verifiability. The paper record contains an encrypted link to the voter’s identity, requiring stronger assumptions on vote privacy than normal paper voting, but with the benefit of providing good auditability and dispute resolution as well as supporting (comparison) risk limiting audits. We will also present partial security proofs for Vote-Privacy and Individual Verifiability, that were computed with the TAMARIN prover.

6.1 Introduction

We propose combining the highly transparent counted-as-intended verification mechanism of the e-voting scheme Selene [109] with paper ballot, in-person voting. The aim is to keep the vote casting experience close to paper ballot voting with optical scanning, while enabling the intuitive voter-verification of the Selene scheme. The resulting scheme provides improved dispute resolution and supports risk limiting audits.

For most end-to-end verifiable schemes the voter verifies the presence of an encryption of her vote in the input to the tally on the bulletin board. In contrast, Selene lets a voter check that her vote appears correctly, in the clear, in the final tally via a tracking number system. This provides a highly transparent and intuitive verification, but, if naively implemented, could lead to vote-selling and coercion.

The main idea of Selene is to mitigate the coercion threats by notifying the voters of their tracking number only after the full list of tracking numbers and votes has been published. Coerced voters can then simply choose a tracker pointing to the required

vote and claim it as theirs. The notification provides the voter high assurance that it is the correct, i.e. unique, tracker while being deniable in the event of coercion.

In a paper ballot election the voters enjoy ballot secrecy thanks to the isolation of the voting booth at the polling station - giving good resistance against coercion and vote-buying attempts. Normally, but with UK as a prominent counter-example, the ballots are also anonymous and unmarked, extending the ballot secrecy to the tally phase. However, the integrity of the election relies on trust assumptions for the talliers, and many real attacks and errors are known, as shown in [56].

In Germany, for example, the tally process is public [3] and at least gives the voters the possibility to oversee the tally ceremony, however considerable trust is still required in the chain of custody of ballots.

To improve on this situation we propose here introducing the Selene mechanism to allow voters to verify that their vote is counted-as-intended. This requires the introduction of a carefully protected link between the ballot and the voter. The vote casting experience of the system is close to the optical scan paper ballot systems with the difference that the paper ballots will have a (QR-)code printed onto them which contains an encryption of the voter’s identity.¹ We assume that voters have smartcards to store and prove their ID. Before getting into the details, we recall the key elements of Selene.

6.1.1 The Essence of Selene

Selene revisits the old idea of enabling verification by posting the votes in the clear on the BB along with a private tracking numbers. The new twist is that voters are only notified of their tracker some time after the vote/tracker pairs have been publicly posted, giving a coerced voter the opportunity to choose an alternative tracker that will placate the coercer. Notification of the trackers is carefully designed to provide assurance that it is the correctly assigned tracker, i.e. unique to the voter, while being deniable. The key goals of Selene are:

- Ensure that each voter is assigned a unique tracker number.
- Notify the voter of her tracker after the vote/tracker pairs have been published in a manner that provides high assurance and yet is deniable in the event of coercion.

This is achieved, in essence, by publishing a list of trackers, n_i , verifiably encrypting and shuffling these and assigning them to the voters under trapdoor commitments according to the secret permutation π resulting from the shuffles. The commitment for the i th voter takes the form:

$$C_i := \text{pk}_i^{r_i} \cdot g^{n_{\pi(i)}}$$

¹This might be troublesome in some jurisdictions.

Where pk_i is the voter’s public trapdoor key. C_i is a Pedersen commitment to the tracker but can also be thought of as the second term (β) of an exponential ElGamal encryption of the tracker under the i th voter’s trapdoor public key pk_i . The corresponding first term ($\alpha = g^{r_i}$) is not published, but is communicated to the voter over a private channel at notification time. On receipt of the α -term, the voter can combine this with the β -term and decrypt using her trapdoor key.

If she is coerced, she can choose an alternative tracker that will satisfy the coercer and compute, using her trapdoor key, an alternative α . Without the trapdoor, it is intractable to compute an α that will decrypt to a given tracker. This observation simultaneously underpins the assurance that the tracker is correct, and removes the need to authenticate the α as communicated to the voter.

6.1.2 The Essence of Electryo

The key innovation of Electryo is to introduce a protected link between the paper ballot and the voter ID, in such a way as to guarantee the integrity and the secrecy of the link. This link is used to associate the encrypted vote, scanned from the paper ballot, with the voter ID on the BB, thus enabling the Selene mechanism to kick in. An additional feature is that at the time of scanning the ballot, a fresh, random *receipt code* is generated and printed for the voter to retain. This is required later to access the tracker number, providing an extra layer of privacy, as explained in detail later.

Now that voters are able to verify their vote in the clear, we can omit the usual checks required in cryptographic, end-to-end verifiable schemes: Benaloh challenges and correct posting to the BB of the encrypted vote. A corollary of this last observation is that the voter does not need to retain a copy of the encrypted vote, just the receipt code which is unrelated to the tracker, which helps ensure receipt-freeness.

The voting system provides individual verifiability via the Selene check, allows universal verifiability of the setup phase and of the tally as well as eligibility via the digital signatures. The paper record provides a basis for dispute resolution, while risk-limiting audits will strengthen the link between the paper and digital record – all of this while preserving a good measure of coercion-mitigation.

6.2 Related Work

Several in-person voting protocols mix paper ballots or a paper-audit trail with a public digital record of the votes:

Prêt-à-Voter [108] is a paper-based voting scheme with voter-verifiability, a version of which has been trialled in a state election [38]. Contrary to the present scheme, these schemes does not provide transparent verification or directly support RLAs, see however [85] for a version with a human-readable paper-audit trail.

Wombat [19] combines paper-ballot voting with cryptographic tabulation and end-to-end verifiability. A voting machine delivers a paper ballot containing a plaintext vote as well as the encrypted version as a QR-code. The voter can check the correctness of the plaintext vote before putting it in a ballot box. The encrypted version is scanned and posted to a *BB* and the paper copy is kept by the voter as a receipt.

STAR-Vote Another polling station e-voting scheme is STAR-Vote [22] which combines electronic voting machines (DREs) with a paper trail to achieve end-to-end verifiability and allow for efficient risk limiting audits (RLAs) [82]. The correctness of the encryption of the vote, can be tested by the voter by a sort of Benaloh challenge, where discarded ballots are decrypted in public. Note that it was not a design goal of STAR-vote to have eligibility verifiability.

Tamarin A TAMARIN model for a simplified version of Selene has already been developed [29]. The authors used the equational theory developed in [43] for the commitments in Selene, which we will use here as well. Vote-Privacy and Receipt-Freeness were proved for a protocol running over untappable channels.

Basin et al. in [18] have developed a protocol for random sample voting with the associated proofs in TAMARIN, and proved Receipt-Freeness and Verifiability. We use their definition of encryption with randomness.

Some other examples of voting protocol models in TAMARIN can be found in [43] where the equational theory for trapdoor commitments have been developed and applied in Okamoto’s protocol [97] and the FOO protocol [51].

6.3 A paper-based version of Selene

In this section, we will detail the protocol as well as the voting experience. We refer to [107] for a complete version including the preliminary security analysis.

6.3.1 Participants and Primitives

The main participants of the protocols are:

- The voters V_i . We assume that they are provided with electronic ID cards², e.g. as part of a national electronic ID infrastructure like in Estonia [1].³ The card stores a secret signing key together with the ID which is associated with the corresponding public verification key vk_i . We assume that the card can

²See [2] for an example of a smartcard implementing ElGamal encryption with Elliptic Curves.

³See [92] for a recently found flaw in that system, demonstrating the importance of a secure implementation of this system.

perform an encryption of the ID with the election key and sign input using the secret signing key. Further the voter has public and secret key pair, pk_i, sk_i for the Selene mechanism, where the latter is stored in a Vote Supporting Device (VSD) e.g. a smartphone or a computer and perhaps also on the smartcard.

- The Election Authority EA is managing the election and protocol setup.
- The Tally Tellers TT create the public election key PK_T and threshold share the secret key. They also facilitate a Mixnet M which is used to ensure privacy, and performs parallel verifiable re-encryption mixes see e.g. [128].
- A public web Bulletin Board BB is used for verifiable communication, and will be assumed to be append-only and have a consistent public view.
- The Tracker Retrieval Authority (TRA) is responsible for relaying communication between the voters and TT . Specifically TRA will send the so-called α term to the voter, which can be turned into a tracker for her vote using the secret key sk_i .
- Registration Clerks and Talliers assisting at the polling station.
- Printers with Card Readers. These print a ballot code, bcode , onto the paper ballots in the form of a QR-code, containing the re-encrypted ID and digital signature of the voter.
- Optical Scanners with a Receipt Printer. The Scanner reads out the voter's choice on the paper-ballot and the bcode , and sends an encryption of the vote to BB together with a re-encryption of the ballot code and an encrypted receipt code. It delivers a *ballot proof* to the voter, that contains the receipt code in plain text together with a digital signature for accountability.

Some primitives used are

- Encryption. We assume an IND-CPA secure homomorphic encryption scheme allowing re-encryption and verifiable mixing. To be explicit we choose ElGamal encryption which was used in Selene, and the homomorphic properties are needed for the Selene mechanism. We denote encryption with the key PK $\{\cdot\}_{PK}$ and re-encryption is denoted $\{\cdot\}'_{PK}$. For some parts the homomorphic properties are not necessary and we use a RCCA secure scheme instead, i.e. the only malleability of the ciphertext is the ability to re-encrypt which is necessary for privacy and mixnets. To be explicit we can use the OAEP 3-round transformation [101, 100] of ElGamal. A single ciphertext then basically consists of two ElGamal ciphertexts and is RCCA secure under the Gap Diffie-Hellman assumption. We denote this encryption $\{\cdot\}_{\text{OAEP}, PK}$. The parallel mixing is easily adapted to this encryption scheme since it basically consists of two ElGamal ciphertexts.

- Zero-Knowledge Proofs. We use zero-knowledge proofs, and proofs of knowledge, as well as signatures to ensure universal verifiability. For non-malleability the strong form [25] of the Fiat-Shamir transform [49] is used for obtaining non-interactive proofs, and we further include election identifiers in the hash to avoid malleability across elections.
- Plaintext equivalence tests (PETs). A PET [64] produces a public verifiable test whether two ciphertexts are an encryption of a same plaintext message, without revealing the plaintexts to anybody. The test requires a threshold set of the Tellers TT .
- QR-codes. A QR-code is a matrix barcode containing information for reliable and easy scanning. The encryption schemes use here can be based on elliptic curves requiring in the order of 512 bit strings. A ciphertext could then e.g. be stored in a QR-code version 6 (up to 1088 bits) or a version 10 for two OAEP ciphertexts.

6.3.2 The Voting Experience

In this section we describe the protocol from the voter’s perspective. The vote-casting ceremony is close to a paper-ballot election with optical scanning of the ballots. The entire voting experience is described in figure 6.1 and more cryptographic details will be given in next section.

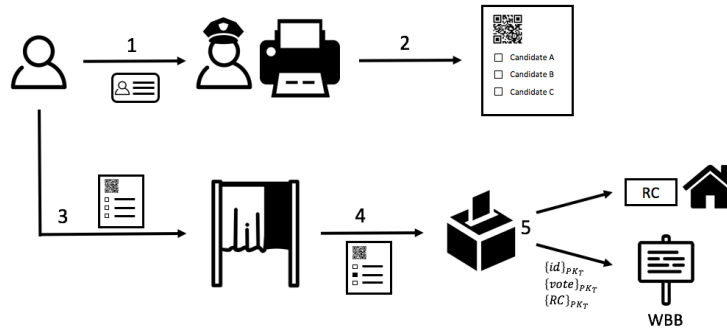


Figure 6.1: Description of the voting phase.

(1) The voter enters the polling station and goes to a registration clerk with her ID card to be identified. (2) Her ID card is read and the printer delivers the ballot with the encrypted ID contained in a QR-Code. (3) The voter goes to a booth to fill her ballot. (4) She puts her ballot into a ballot box containing the scanner, (5) that sends the encrypted vote to the bulletin board and prints the voter a take home receipt code.

6.3.2.1 Registration

We assume that all voters are in possession of ID smart-cards, e.g. as part of a national electronic ID infrastructure. The ID card can create signatures and will be used to authenticate voters. The registration of the voters could probably happen automatically if based on a national PKI, alternatively by a company etc.

Besides the ID-card, the Selene mechanism assumes that each voter V_i holds a secret key, sk_i . This may require a registration step by the voters, the details of which we omit, but note that these keys could potentially be used for multiple elections. The tracker authority TRA also needs to know where to contact the voter for the tracker retrieval phase, e.g. an email address. Such confirmed contact data is normal to have in an electronic ID infrastructure. However, for improved usability, we assume that the voter is using an app (e.g. authenticated via the sim card as in Estonia [1]) that accesses sk_i e.g. via the smartcard or, if properly authenticated, the program could be the creator of the Selene keys.

6.3.2.2 Voting phase

On the voting day, the voter presents her ID card to a poll worker to confirm ID and eligibility, as in standard elections the voters showing up can be recorded in a paper log. The printer is equipped with a smart card reader and interacts with the card to retrieve an encryption, by the smart-card, of the ID and signature. It prints an unfilled ballot with a QR-code which encodes a re-encryption of the voter ID and digital signature, confirming the voter was present.

Then the voter enters the booth to fill the ballot, and finally she heads to a ballot box with a scanner/printer. The latter delivers a receipt code RC_i on paper, without releasing it, before scanning the ballot. The scanner re-encrypts the ballot code, encrypts the vote and releases the receipt code to the voter. The data is stored and sent to BB after voting ends, and the paper ballot is retained in the ballot box.

6.3.2.3 Tracker retrieval

After the tally phase, cast votes and corresponding tracking numbers will appear on the BB . After a pause, allowing coerced voters to access this information, the voters will receive their α -term (see introduction) via their support device at randomised times, as in Selene. The device will calculate the voter's unique tracker using the received α -term, the public β -term and the trapdoor key sk .

However, in contrast to Selene the α term will only be sent to the voter if she at some point after election enters a correct receipt code RC_i in her device.

6.3.2.4 Voting in case of coercion

Coerced voters can take steps to mitigate the coercion. After the tally board is created with votes and corresponding trackers, the voter can choose a tracker pointing

to a candidate of the coercer’s choice. Further, the voter can calculate a fake α -term using \mathbf{sk} that opens to this tracker. The voter can now show the coercer this tracker and α -term, if required.

Further, for improved coercion-resistance, the coerced voter can also contact *TRA* with authentication and request to not receive the real α -term, but only the fake. Now, even in the case where a coercer or vote buyer controls the interface to receive the α -term, he does not receive any convincing evidence of the cast vote. As mentioned above the essential assumption here is that the voter has access to \mathbf{sk} , e.g. via multiple copies or the storage on the ID card.

6.3.2.5 Comments on usability

The voting experience is close to a standard optical-scan scheme. As with an optical-scan or STAR-Vote [22], the scanner and ballot box can be combined, so that the ballot will be read before being fed in automatically in the ballot box. The only aspect that might be a bit troubling for some voters, in addition of the one brought by the standard Selene mechanism, is the printing of the QR code on the ballot form. This does not affect usability, it is automatic as far as the voter is concerned, but might be worrying from a privacy perspective.

We avoid the verification steps such as Benaloh challenges [21] of encrypted data. Instead, we have the extra Selene verification phase with the receipt code and tracker check, which we believe is more understandable for voters.

For disabled persons, multi-lingual communities or generally complicated ballots, voting machines could also be used to fill out the ballots. Here the QR code created by the printer is scanned by the voting machine to produce the filled out ballot, which is kept as a paper record. A scanning step is not necessary in this case.

6.3.3 Protocol

We now describe the protocol in more technical detail including cryptography.

6.3.3.1 Pre-Election Setup

Let us recall Selene’s set-up phase that we will also follow here [109].

The Tally Tellers set up a secure group and create the threshold election key PK_T for ElGamal encryption (or another homomorphic encryption scheme).

We assume that all voters have PKs in the chosen group. Let $\mathbf{pk}_i = g^{x_i}$ be the public key of voter V_i , and $x_i = \mathbf{sk}_i$ their secret key. The Election Authority publishes on *BB* the set of tracking numbers n_i . These could just be $1, \dots, n$ with n the number of eligible voters. Using a verifiable re-encryption mix each voter is associated a unique secret encrypted tracker on *BB*: $(ID_i, \{g^{n_j}\}_{PK_T})$, where $j := \pi(i)$, and π is the secret permutation resulting from the mixes.

As described in detail in [109], the Tally Tellers TT_1, \dots, TT_t produce a trapdoor commitment $C_i = \text{pk}_i^{r_i} \cdot g^{n_j}$ where $r_i = \sum_{k=1}^t r_{i,k}$, along with an α -term $\alpha_i = g^{r_i}$ that will be kept secret under encryption.

Only TT_k knows $g^{r_{i,k}}$.⁴

Before vote casting BB displays

$$(\text{ID}_i, \text{vk}_i, \text{pk}_i, \{g^{n_j}\}'_{PK_T}, C_i)$$

Here vk_i is the verification key for voter V_i , and the corresponding secret key is stored along with ID_i on the voter's smartcard. The smartcard can produce signatures that can be verified via vk_i and we assume the signature scheme to be existentially unforgeable. Further, the smartcard can produce an encryption that can be used in the mixnet construction and decrypted by TT . We here use ElGamal encryption $\{\cdot\}_{PK_T}$, the OAEP version thereof discussed above, and e.g. Schnorr signatures, but other choices are possible, and since the smartcards are used across elections it might be preferable to use a separate key for this part.

6.3.3.2 Voting

Voter V_i goes to the polling station and is identified and registered by a clerk. If her identity is confirmed and if she has not voted yet, the clerk proceeds to the printing. The ID card is read and delivers an encryption of the voter's ID to the printer. The latter re-encrypts it (to avoid privacy attacks from a colluding ID card and scanner) and delivers a QR-code representing the ballot code $\text{bcode} = (\{\text{ID}_i\}_{\text{OAEP}, PK_T}, \{\text{sign}_i\}_{\text{OAEP}, PK_T})$.⁵ The signature is of the ID and the election ID, but can also include e.g. the date and the printer ID. The clerk should be screened from seeing the printed ballot, but can check that the correct ID card is read in the card reader.

After retrieving her ballot, the voter enters a booth and fills out the ballot with her vote vote_i by hand.

The voter now proceeds to a ballot box that contains a scanner. The scanner first prints a receipt code, that is not yet detachable from the ballot box. This ensures that the receipt code does not depend on the vote and thus cannot be used as a subliminal channel. The receipt code is a random short pin, e.g. five digits, with check digits. The voter then puts her ballot in the box, the scanner reads it and releases the receipt code. It processes the data and re-encrypts the ballot code elements, encrypts the vote and receipt code, and publishes on BB (after election, if offline):

$$\{\text{ID}_i\}'_{\text{OAEP}, PK_T}, \{\text{sign}_i\}'_{\text{OAEP}, PK_T}, \{\text{bcode}\}_{PK_T}, \{\text{vote}_i\}_{PK_T}, \{RC_i\}_{PK_T}, \Pi_i$$

⁴A difference to [109] is that we do not introduce separate Tracker Tellers, but instead let the Tally Tellers handle this, and we introduce a single separate Tracker Retrieval Authority TRA .

⁵Cryptographically it would suffice to leave out the encryption of ID_i , since it can be determined from testing different vk 's.

Here Π_i is a zero-knowledge proof of plaintext knowledge for the vote and receipt code and correct message space, for less malleability we suggest including an AND-proof, proving that the two first encryptions are re-encryptions of the **bcode** in the third ciphertext. We include the election identifier in the hash of the Fiat-Shamir transform. The proofs will prevent vote copy attacks also across elections. The reason to re-encrypt the ciphertexts in the ballot code is to prevent coercion attacks via taking a picture of the filled-in ballot as a proof of the cast vote. In this case, a coerced voter can fill out a ballot as required by the coercer, photograph it, and go back to the officials for a new ballot and hand the (photographed) one, which is destroyed. They now cast their intended vote using the new ballot form. The re-encryption means that the paper ballot won't be linkable to the public electronic record, which is also important in the RLAs.

Finally, $\{\text{bcode}\}_{PK_T}$ is an encryption of the ballot code which is here written in shorthand, but includes several ElGamal ciphertexts. If needed, these can be decrypted and allow crosschecking with the corresponding paper record.

6.3.3.3 Mix and decryption

These published tuples are now sent through a parallel mixnet (e.g. Verificatum [128]) on the *BB* after checking the proofs Π_i . After decryption of the first term we get back the ID and signature, i.e. we get mixed tuples of

$$ID_i, \text{sign}_i, \{\text{bcode}\}'_{PK_T}, \{\text{vote}_i\}'_{PK_T}, \{RC_i\}'_{PK_T}$$

Now the signature can be checked for eligibility verification and with the previous data on the *BB* we construct (suppressing re-encryption for clarity)

$$ID_i, \{g^{n_{\pi(i)}}\}_{PK_T}, C_i, \{\text{vote}_i\}_{PK_T}$$

As in Selene, the second and last term, the encrypted tracker and vote, are put through a verifiable parallel mix, after which the Tellers perform a verifiable decryption, to obtain the final tally board containing tracker/vote pairs:

$$(n_{\pi(i)}, \text{vote}_i)$$

6.3.3.4 Tracker notification

Receipt verification Before she can check her vote, the voter must enter the receipt code RC_i on her device (after log in). The app will encrypt the receipt code and a *TT* will do a PET between this encryption and the one displayed on the bulletin board. This verification is also done to ensure that the paper ballot and the corresponding electronic record are related to the same voter, i.e. to prevent an attack from the printer putting the wrong ID on the ballot.

Tracker retrieval The public commitment C_i and the corresponding α -term can be combined to form an encryption of the tracker under the voter’s public key:

$$(\alpha_i, C_i) = (g^{r_i}, \text{pk}_i^{r_i} \cdot g^{n_{\pi(i)}})$$

If the voter has entered the correct receipt code, the α_i term will be sent to the voter, and she can then compute the decryption using her secret key and retrieve $g^{n_{\pi(i)}}$, and hence her unique tracker $n_{\pi(i)}$. The Tracker Retrieval Authority will get the α_i shares from each Tally Teller (authenticated for accountability), multiply these together to obtain α_i and send this unauthenticated to the voter.

As described in Selene [109] it is computationally hard, without knowing sk_i , to calculate an alternative α -term that opens to a valid tracker. Thus, the α -terms can be transmitted unauthenticated to the voter. On the other hand the voter can efficiently calculate such a fake α -term for any tracker (see [109]), and thus shows this to a coercer in case of coercion.

6.3.3.5 Risk Limiting Audits

A comparison Risk Limiting Audit (RLA) [82] is a method to confirm (or refute) the outcome of an election to any required confidence, by random sampling of the paper ballots. The digital and paper records of the vote are compared. Typically, for reasonably large margins, a small sample will suffice to achieve a good level of confidence, e.g. 95%. This technique requires a link between the digital and paper copies for every ballot.

The RLA testing, can be used to monitor the behaviour of the scanners. The audit should be performed in both directions, i.e. first start from tuples on BB , decrypt the `bcode` and find the corresponding paper ballot and check the consistency. In the other direction we can also start from a paper ballot, and the corresponding encrypted ballot can be found via PET tests, or more efficiently via an obfuscation of a part of the ballot code by lifting to a secret power homomorphically and then decrypting.

6.4 Security proofs

We propose an initial model for the voting protocol Electryo. The additional feature of Electryo compared to Selene is the link between paper ballots and electronic ballots, allowing the possibility to perform (comparison) Risk Limiting Audits [82] efficiently. The voter casts a paper ballot printed at the polling station, which contains an encryption of her ID represented in a QR code. The ballots are scanned to create an encrypted digital version of the paper ballots on the bulletin board. When scanning the ballot, the ID on the QR code is re-encrypted, while the QR code is itself encrypted before being sent to the server. A receipt-code is also created at that

point that will be used later to allow the voter to verify his vote on presentation of the valid receipt-code. From the data on the bulletin board an anonymous tally list of plaintext votes is created, each associated with a tracking number. After the election ends, each voter will be able to retrieve their tracker and hence check their vote. A *Tracker Retrieval Authority* will take care of notifying the voters of their tracking number. All these aspects will be reflected in the model.

Among the available tools for formal verification, we used the TAMARIN prover [16] to develop our model. As already described in chapter 2, TAMARIN has an expressive language based on multiset rewriting rules. This lets us represent a symbolic model of the adversary’s knowledge and messages sent over the network. It also uses equational theories, that allow us to specify cryptographic operators, like encryption but also Pedersen commitments. We will detail give the equational theories we used below.

In this section, we provide a formal model of the Electryo protocol. We model the tracker commitments, encryption, signatures, and channels rules between entities. We provide proofs for ballot-secrecy and individual verification (see section 6.4.2).

6.4.1 Tamarin

We recall quickly the semantics and notations that we described already in chapter 2.

6.4.1.1 Semantics

In TAMARIN, messages are represented as terms. A term is an element t or a function $f(t_1, \dots, t_n)$ of arity n , where t_1, \dots, t_n are terms. We also define a set of operators, or functions, with their arities. An equation is a pair of terms s and t such as $s = t$. We define E as a set of equations. An equational theory is the smallest congruence closure containing all instances E .

Protocols are modeled through multiset rewriting rules. These rules use sets of Facts $F(t_1, \dots, t_n)$ of arity n . We denote fresh values with \sim and public values with $\$$. Facts are user-defined except: **Fr**, **In** and **Out** for inputs and outputs of a rule, and **K** is the attacker knowledge. An exclamation mark **!** before a Fact will define it as persistent and can be consumed many times, while a linear Fact can be consumed only once.

6.4.1.2 The Electryo model

In our model, we consider two voters **V1** and **V2**, an Election Authority **EA**, the Tracker Retrieval Authority **TRA** and a scanner **S**. The ID cards, used to perform the ID encryption, are not distinguished from the printer. This is discussed below in the trust assumptions paragraph.

Channel rules We denote by $\bullet \rightarrow$ an *authentic* channel, that means the adversary cannot modify the messages or their sender, but he can access this data. This ensures that a message is correctly delivered but can be seen and copied by an adversary. More details about TAMARIN channel rules can be found on the manual web page [16].

We also define the *untappable* channel by \boxRightarrow , which means that a message is not readable nor modifiable over the network like a secure channel [91], but also the message won't be persistent and replayable later.

Trust assumptions In TAMARIN, the adversary is a standard Dolev-Yao style [42], that is controlling the network and can apply all operators.

The adversary learns all messages sent by participants when they are output with the **Out** fact. He can send messages to the participants with the **In** fact, that is we assume that every input could be given by the adversary. The adversary can also generate fresh values and knows all public values. Finally, he can apply functions available in the set of operators. He will be provided with additional information depending on the trust assumptions below.

As a simplification in the current model, we merged the ID card and the printer into one entity. This means that the printer is reading this voter's ID card without changing the information. The adversary can still modify the information on the printed ballot which will correspond to the ID card and the printer colluding in the original model. We also assume that the voter is using her own ID card which is checked by polling station clerks in the Electryo protocol.

Equational theories To model Electryo, that is using the Selene mechanism, we need the trapdoor (td) commitments equations defined in [43], that we already mentioned in chapter 2.

We also use an asymmetric encryption scheme for the messages' encryption, and the built-in package *multiset* to model the shuffling of messages as described in [29].

6.4.1.3 Tamarin Model of Electryo

An overview of the model is given in Fig. 6.2. Compared to the existing implementation of Bruni et al. [29], this model considers more cryptographic primitives and provides more data to the adversary as we will detail below. The EA generates tracking numbers and together with the TRA, computes the commitments. The TRA keeps the α -terms secret. The EA publishes the commitments with a persistent fact. Then, voters retrieve their ballot with a ballot code computed from their identity and signature, $bcode = \langle cp(\$V, r, pkT), cp(sign(\$V, skV), s, pkT) \rangle$.

Voters input their ballot code and intended vote into the scanner, that computes an encryption of their ballot code $bcode$, an encryption of their vote, generates a

receipt-code RC and encrypts it. Finally, it calculates a re-encryption of the ciphertext buried in the ballot code.⁶ In particular, it computes $\text{cp}(\$V, r', \text{pkT})$ and $\text{cp}(\text{sign}(\$V, \text{skV}), s', \text{pkT})$. The scanner sends all of this data to the EA, and gives the plaintext receipt-code to the voter.

When the EA receives the data for both voters, it decrypts and publishes the votes on the bulletin board with the tracking number and the encrypted RC.

When the votes are published, the TRA can send the α -term to the voter. We use an authentic channel to notify the voters.⁷ Each voter can open the commitment and retrieve their tracker. A trace is written to provide a verifiability lemma, checking the validity of the receipt-code and of the vote (see section 6.4.2).

6.4.2 Results

6.4.2.1 Privacy

To prove privacy properties, we need to prove indistinguishability between two executions of a system. In TAMARIN, this is done through observational equivalence [17]. For this, the tool uses a multiset rewriting system where terms can be written using a special operator $\text{diff}(\cdot, \cdot)$. With this operator, we are able to instantiate two possible elements in one term. Then TAMARIN creates two systems, a left and a right, with identical rules where the difference is on the value of the term instantiated with diff .

To verify the observational equivalence, TAMARIN uses dependency graphs. A dependency graph represents the execution of a system. To each node corresponds one rule defined in the model, and there is a direct relation called edge from a rule r_1 to r_2 iff r_1 outputs a fact to r_2 input. The equivalence between two graphs depends on *mirroring*, that is: given a dependency graph, its mirrors contain all graphs on the other side (left or right) of the system defined with the diff operator, where nodes are instances of the same rules and edges are the same.

Vote-privacy First, we used the definition of Delaune et al. [39] for vote-privacy: an attacker cannot detect if voters V1 and V2 swap their votes. In our model, we use the diff operator during the setup phase when defining each entity knowledge: when defining the two voters in the setup rule, we swapped their intended vote. This is defined as follows:

$$\begin{aligned} & \text{St_V_1}('V1', \text{pkV1}, \sim\text{ltkV1}, \text{diff}('candA', 'candB'), \text{pkT}) \\ & \text{St_V_1}('V2', \text{pkV2}, \sim\text{ltkV2}, \text{diff}('candB', 'candA'), \text{pkT}) \end{aligned}$$

⁶For readability, this does not appear in figure 6.2.

⁷Sufficient for Vote-privacy. To prove Receipt-Freeness, we will need a stronger assumption on channels.

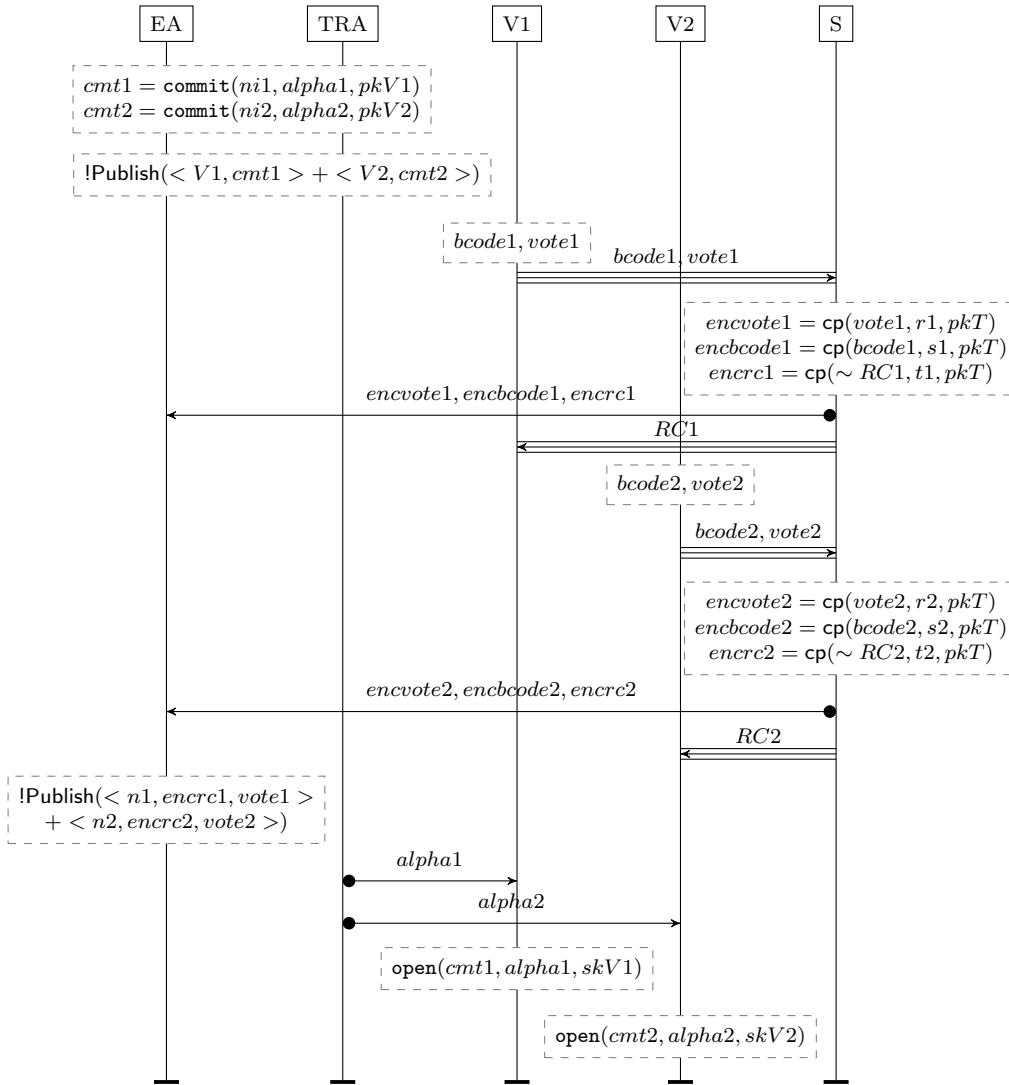


Figure 6.2: An overview of the model.

where $pkV\cdot$ is the voter's public key, $\sim\text{sk}V\cdot$ is the voter's secret key and pkT is the election key.

In Electryo privacy is guaranteed (for covert adversaries) unless the ID card, printer and the scanner collude. Indeed, we found a trivial attack when the card/printer and the scanner collude. On the other hand, using the above definition, we have a proof in TAMARIN, when neither the card/printer nor the scanner collude with the adversary. Proofs for privacy when only one entity is misbehaving are in progress.

6.4.2.2 Electryo verifiability

Verifiability is defined by individual verifiability, that is a voter can verify that her vote was really counted correctly, and universal verifiability, that is the outcome reflects the sum of all cast votes. In this model we only proved individual verifiability. To check verifiability properties, we can use *traces* and express properties as first order logic formulas. These formulas use the temporality of the protocol that let us use the order of events.

In this model, we defined individual verifiability as the ability of voters to correctly check their tracker and verifying that the recorded vote is correct. For Electryo, we also need to verify the correctness of the receipt-code. Given the action `Vote`, when the voter `V` casts his vote `vote` and receives his receipt-code `rc`, the action `Learn` when the voter `V` computes his tracker `n`, and the action `BB`, when votes, encrypted ballot-codes and trackers are published, we define individual verifiability as:

$$\begin{aligned}
 & \text{All } V \text{ vote } rc \ n \ \#i1 \ \#i2. \\
 & \text{Vote}(V, \text{vote}, rc) \ @i1 \ \& \ \text{Learn}(V, n) \ @i2 \\
 \implies & \text{Ex } \text{othervote } r \ \text{pkT} \ \#j. \\
 & \text{BB}(\langle n, \text{cp}(rc, r, \text{pkT}), \text{vote} \rangle + \text{othervote}) \ @j
 \end{aligned}$$

To verify this lemma, we used our model defined above, and we modeled a simple malicious behaviour either from the scanner or the card/printer allowing it to modify the ballot-code identity. The lemma remains proven for all traces.

Attack detection We modeled a simple malicious behaviour from the scanner: if the scanner is taking the adversary's input to replace the identity of a voter in the ballot code. Given the action `Detection` when the EA receives the ballot code for an identity `id1` with the signature of an identity `id2`, if those are not equal then there exists an action `ChangeBC` when the scanner modifies the ballot code, we have:

$$\begin{aligned}
 & \text{All } id1 \ id2 \ r \ s \ \text{pkT} \ \text{skV} \ \#i. \\
 & \text{Detection}(\text{cp}(id1, r, \text{pkT}), \text{cp}(\text{sign}(id2, \text{skV}), s, \text{pkT})) \ @i \ \& \ \text{not}(id1 = id2) \\
 \implies & \text{Ex } \#j. \ \text{ChangeBC}(id1, id2) \ @j
 \end{aligned}$$

In our model as defined above, this lemma has been verified.

6.5 A trusted protocol?

Electryo is a protocol trying to merge the usual paper-ballot procedure and the verifiability feature, here using the usable mechanism provided by Selene. In this chapter, we described the cryptographic setting of the protocol, and we have proved several security properties. Our previous work on Selene has shown that Selene is

promising regarding usability aspects, however our studies' participants seem to be committed to the ceremony of paper-based protocols. As they are also more familiar the paper procedures, they might be willing to trust more easily a verifiable scheme which is closer to what they are used to. Electryo, with risk limiting audits and a dispute resolution procedure, might be more convincing regarding voters' concerns. However, usability and trust of Electryo have never been tested, and it could be a hint for future research to evaluate trust in these electronic systems using physical ballots, as for Prêt-à-Voter.

Chapter 7

Conclusion

7.1 Summary

In this thesis, we have explored several aspects of voting. We studied technical aspects as well as usability elements that we summarize below.

First, we studied the security properties and their definitions in the symbolic model, in particular in the context of tracker-based verifiable protocols. We have proposed new definitions for Receipt-Freeness and Coercion-Resistance, and we have demonstrated those definitions with the TAMARIN prover, for the e-voting protocol Selene and the boardroom protocol from CNRS.

Then, we studied several aspects of user experience in voting: we started by proposing two usable interfaces for Selene, on mobile and browser, built using a user-centred approach with experts from the HCI and the security fields. We evaluated these interfaces through three user studies where several questionnaires were given to the participants. From the mobile application, we have learned that a good usability and user experience were not the only elements to take into consideration when designing a usable and secure voting app. While participants rated well their user experience, they were not convinced by the security of the system. Also, even though some security elements were clearly communicated to the participants, it was not enough to make them see or understand it. From the web application, we have learned that the coercion mitigation feature was hard to understand and the associated tasks were too complex for the voters.

To have more insights on their user experience and to understand those results, we evaluated the participants' mental models through interviews and analysis of drawings. From interviews, we compared the voters' feedback to the possible security properties in voting, in particular privacy and verifiability. We learned that the voters' expectations are not faraway from our desired voting properties, however those

are not always perceived as being part of the system by the voters. Also, the general workflow in online voting is perceived to be harder to understand than putting a paper in a ballot box by most voters. From drawings, we have learned that showing a clear risk to the participants increased their understanding. Also, not all voters need to see the same amount of details inside their app. Finally, we have seen that small interactions with security features help the voters to see it, rather than just communication and signs.

From those observations, we provided some recommendations for the design of future voting interfaces, such as the need for transparency and correctness, a good education on risk, easy security interactions and access to several levels of verification and of details.

To go further, we have tested a coercion-mitigation mechanism, and we evaluated the voters' experience through a game involving a vote-buyer. In this application, participants were more confused by the additional option available (conceal their vote) and the user experience was rated lower than the standard experience. The vote-buying game was stressful and participants mentioned their need of integrity in voting system. From this experience, we have highlighted a relation between trust and understanding, as well as trust and user experience. We gave additional recommendations for the design of future voting systems and user studies.

Finally, in this thesis, we provide a new design and security evaluation in the symbolic model of a voting system. Electryo has been designed to be close to a standard paper-voting protocol but including the verifiability and coercion mitigation mechanism of Selene. It also provides additional security features such as risk-limiting audits to ensure a correct counting. While the usability of such a protocol has not been evaluated yet, it might be that voters will more appreciate paper protocols where they can understand easily all steps during the voting phase.

The results in this thesis are bounded by the stage at which users were included. As mentioned in chapter 3, our user-centred process used an evaluation of users, where we chose to follow the results found in existing literature. In particular, the design of Selene was motivated by the users' feedback on previous e-voting protocols, such as Helios. Another method would be to involve and discuss with voters before any application or protocol design. Discussing deeper with users to know which features they would like to see at a very early stage would help to consider every point of view. As an example there might be a part of the electorate who want to protest, or perform some kind of disruption, when they disagree with their political system, and our security protocols would not enable this.

7.2 Future work

As future directions, we propose the following possible extensions:

- The TAMARIN prover being limited, it would be interesting to evaluate the security of tracker-based protocols with other verification tools. In TAMARIN, we could not implement more than 2 voters, and the number of available cryptographic primitives is limited.
Another track of research is to refine other definitions in voting, such as accountability or eligibility verifiability, and take into account the voters' mistakes in the model.
- Regarding usable security, we can refine our studies on coercion-resistance, and adapt our game design to other protocols. Coercion-resistance features have been rarely studied in the literature and it is also important to look at their limitations.
- So far, we have developed interfaces to improve the usability and the user experience of voters. The next step would be to implement a complete version of Selene, including all security features, for a broader use.
- Regarding the Electryo protocol, we did provide a partial security proof with the TAMARIN prover. A future direction would be to prove all properties of the protocol, using our new definitions for tracker-based protocols.
- In the thesis, we have proposed a new Trust questionnaire to evaluate the users in the context of voting. The next step would be to test this questionnaire with other voting protocols, and push it as a new standard for the evaluation of voting systems.

Bibliography

- [1] Estonia id card. <http://www.id.ee/>. Accessed: 2017-11-10.
- [2] An example of smart card implementing elgamal encryption with elliptic curves. https://www.nxp.com/docs/en/data-sheet/P40C040_C072_SMX2_FAM_SDS.pdf. Accessed: 2018-05-11.
- [3] German elections. <http://www.dw.com/en/german-election-volunteers-organize-the-voting-and-count-the-ballots/a-40562388>. Accessed: 2017-11-10.
- [4] Recommendation cm/rec(2017)5[1] of the committee of ministers to member states on standards for e-voting. https://search.coe.int/cm/Pages/result_details.aspx?ObjectID=0900001680726f6f.
- [5] Universal declaration of human rights, article 21. <https://www.un.org/en/universal-declaration-human-rights/index.html>.
- [6] A., N. D. Mental models. In *Human-computer Interaction*, D. GENTNER and e. A.L. STEVENS, Eds. Lawrence Erlbaum Associates Inc., 1983, ch. Some Observations on Mental Models, pp. 7–14.
- [7] ACEMYAN, C. Z., KORTUM, P. T., BYRNE, M. D., AND WALLACH, D. S. Usability of voter verifiable, end-to-end voting systems: Baseline data for helios, prêt à voter, and scantegrity II. In *2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections, EVT/WOTE '14* (2014).
- [8] ACEMYAN, C. Z., KORTUM, P. T., BYRNE, M. D., AND WALLACH, D. S. Users' mental models for three end-to-end voting systems: Helios, prêt à voter, and scantegrity II. In *Human Aspects of Information Security, Privacy, and Trust - Third International Conference, HAS 2015* (2015).
- [9] ADIDA, B. *Advances in Cryptographic Voting Systems*. PhD thesis, Massachusetts Institute of Technology, 2006.
- [10] ADIDA, B. Helios: Web-based open-audit voting. In *Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008* (2008).

- [11] ALI, S. T., AND MURRAY, J. An overview of end-to-end verifiable voting systems. *CoRR abs/1605.08554* (2016).
- [12] ARAPINIS, M., CORTIER, V., AND KREMER, S. When are three voters enough for privacy properties? In *Computer Security - ESORICS 2016 - 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26-30, 2016, Proceedings, Part II* (2016).
- [13] ARNAUD, M., CORTIER, V., AND WIEDLING, C. Analysis of an electronic boardroom voting system. In *E-Voting and Identify - 4th International Conference, VoteID 2013, Guildford, UK, July 17-19, 2013. Proceedings* (2013), J. Heather, S. A. Schneider, and V. Teague, Eds., vol. 7985 of *Lecture Notes in Computer Science*, Springer, pp. 109–126.
- [14] BADA, M., SASSE, A. M., AND NURSE, J. R. C. Cyber security awareness campaigns: Why do they fail to change behaviour?, 2019.
- [15] BARTSCH, S., AND SASSE, M. How users bypass access control - and why: the impact of authorization problems on individuals and the organization.
- [16] BASIN, D. A., CREMERS, C., DREIER, J., MEIER, S., SASSE, R., AND SCHMIDT, B. Tamarin prover manual. <https://tamarin-prover.github.io/manual/>, 2019.
- [17] BASIN, D. A., DREIER, J., AND SASSE, R. Automated symbolic proofs of observational equivalence. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015).
- [18] BASIN, D. A., RADOIROVIC, S., AND SCHMID, L. Alethea: A provably secure random sample voting protocol. In *31st IEEE Computer Security Foundations Symposium, CSF 2018* (2018).
- [19] BEN-NUN, J., FAHRI, N., LLEWELLYN, M., RIVA, B., ROSEN, A., TASHMA, A., AND WIKSTRÖM, D. A new implementation of a dual (paper and cryptographic) voting system. In *Electronic Voting* (2012), pp. 315–329.
- [20] BENALOH, J. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, Department of Computer Science, 1987.
- [21] BENALOH, J. Simple verifiable elections. *EVT 6* (2006), 5–5.
- [22] BENALOH, J., BYRNE, M., KORTUM, P. T., MCBURNETT, N., PEREIRA, O., STARK, P. B., AND WALLACH, D. S. Star-vote: A secure, transparent, auditable, and reliable voting system. *CoRR* (2012).

- [23] BENALOH, J. C., AND TUINSTRA, D. Receipt-free secret-ballot elections (extended abstract). In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing* (1994).
- [24] BERNHARD, D., CORTIER, V., GAUDRY, P., TURUANI, M., AND WARINSCHI, B. Verifiability analysis of chvote. *IACR Cryptology ePrint Archive* (2018).
- [25] BERNHARD, D., PEREIRA, O., AND WARINSCHI, B. How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios. In *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings* (2012), X. Wang and K. Sako, Eds., vol. 7658 of *Lecture Notes in Computer Science*, Springer, pp. 626–643.
- [26] BLANCHARD, E., AND SELKER, T. Origami voting: a non-cryptographic approach to transparent ballot verification. In *Financial Cryptography and Data Security - FC 2020 (Workshop on Voting)* (2020).
- [27] BLANCHET, B. Modeling and verifying security protocols with the applied pi calculus and proverif. *Foundations and Trends in Privacy and Security* (2016).
- [28] BORGMAN, C. L. The user’s mental model of an information retrieval system: An experiment on a prototype online catalog. *International Journal of man-machine studies* 24, 1 (1986), 47–64.
- [29] BRUNI, A., DREWSSEN, E., AND SCHÜRSMANN, C. Towards a mechanized proof of selene receipt-freeness and vote-privacy. In *Electronic Voting - Second International Joint Conference, E-Vote-ID 2017* (2017).
- [30] CHAUM, D. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM* 24, 2 (1981), 84–88.
- [31] CHAUM, D. Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings* (1988), C. G. Günther, Ed., vol. 330 of *Lecture Notes in Computer Science*, Springer, pp. 177–182.
- [32] CHAUM, D., CARBACK, R., CLARK, J., ESSEX, A., POPOVENIUC, S., RIVEST, R. L., RYAN, P. Y., SHEN, E., AND SHERMAN, A. T. Scantegrity ii: End-to-end verifiability for optical scan election systems using invisible ink confirmation codes. *EVT* 8 (2008), 1–13.

- [33] CHEVAL, V., KREMER, S., AND RAKOTONIRINA, I. The DEEPSEC prover. In *Computer Aided Verification - 30th International Conference, CAV 2018* (2018).
- [34] COHEN, J., AND FISCHER, M. A robust and verifiable cryptographically secure election scheme (extended abstract). pp. 372–382.
- [35] COOPER, A., REIMANN, R., CRONIN, D., AND NOESSEL, C. *About face: the essentials of interaction design*. John Wiley & Sons, 2014.
- [36] CORTIER, V., EIGNER, F., KREMER, S., MAFFEI, M., AND WIEDLING, C. Type-based verification of electronic voting protocols. In *Principles of Security and Trust - 4th International Conference, POST 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015, Proceedings* (2015).
- [37] CORTIER, V., GALINDO, D., KÜSTERS, R., MÜLLER, J., AND TRUDERUNG, T. Sok: Verifiability notions for e-voting protocols. In *IEEE Symposium on Security and Privacy, SP 2016* (2016).
- [38] CULNANE, C., RYAN, P. Y. A., SCHNEIDER, S., AND TEAGUE, V. vvote: a verifiable voting system (DRAFT). *CoRR abs/1404.6822* (2014).
- [39] DELAUNE, S., KREMER, S., AND RYAN, M. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security* (2009).
- [40] DISTLER, V., ZOLLINGER, M.-L., LALLEMAND, C., RØNNE, P. B., RYAN, P. Y., AND KOENIG, V. Security–visible, yet unseen? how displaying security mechanisms impacts user experience and perceived security. In *CHI Conference on Human Factors in Computing Systems (CHI '19)* (2019).
- [41] DODIER-LAZARO, S., ABU-SALMA, R., BECKER, I., AND SASSE, A. From paternalistic to user-centred security: Putting users first with value-sensitive design.
- [42] DOLEV, D., AND YAO, A. C. On the security of public key protocols. *IEEE Trans. Information Theory* (1983).
- [43] DREIER, J., DUMÉNIL, C., KREMER, S., AND SASSE, R. Beyond subterm-convergent equational theories in automated verification of stateful protocols. In *Principles of Security and Trust - 6th International Conference, POST 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017* (2017).

- [44] DREIER, J., HIRSCHI, L., RADOMIROVIC, S., AND SASSE, R. Automated Unbounded Verification of Stateful Cryptographic Protocols with Exclusive OR. In *31st IEEE Computer Security Foundations Symposium, CSF 2018* (2018).
- [45] DREIER, J., LAFOURCADE, P., AND LAKHNECH, Y. Defining privacy for weighted votes, single and multi-voter coercion. In *Computer Security - ESORICS 2012 - 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10-12, 2012. Proceedings* (2012).
- [46] DRIZA MAURER, A. Updated european standards for e-voting. pp. 146–162.
- [47] Estonian national electoral committee. <https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia>, 2019.
- [48] EVERETT, S. P., BYRNE, M. D., AND GREENE, K. K. Measuring the usability of paper ballots: Efficiency, effectiveness, and satisfaction. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 50*, 24 (2006), 2547–2551.
- [49] FIAT, A., AND SHAMIR, A. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO* (1986).
- [50] FUGLERUD, K. S., AND RØSSVOLL, T. H. An evaluation of web-based voting usability and accessibility. *Universal Access in the Information Society 11*, 4 (2012), 359–373.
- [51] FUJIOKA, A., OKAMOTO, T., AND OHTA, K. A practical secret voting scheme for large scale elections. In *Advances in Cryptology - AUSCRYPT '92, Workshop on the Theory and Application of Cryptographic Techniques* (1992).
- [52] GARFINKEL, S., AND RICHTER LIPFORD, H. Usable security: History, themes, and challenges. *Synthesis Lectures on Information Security, Privacy, and Trust 5* (09 2014), 1–124.
- [53] GIBSON-ROBINSON, T., ARMSTRONG, P., BOULGAKOV, A., AND ROSCOE, A. FDR3 — A Modern Refinement Checker for CSP. In *Tools and Algorithms for the Construction and Analysis of Systems* (2014), E. Ábrahám and K. Havelund, Eds., vol. 8413 of *Lecture Notes in Computer Science*, pp. 187–201.
- [54] GJØSTEEN, K. The norwegian internet voting protocol. In *E-Voting and Identity - Third International Conference, VoteID 2011, Tallinn, Estonia, September 28-30, 2011, Revised Selected Papers* (2011), A. Kiayias and H. Lipmaa, Eds., vol. 7187 of *Lecture Notes in Computer Science*, Springer, pp. 1–18.

- [55] GLASS, A., MCGUINNESS, D. L., AND WOLVERTON, M. Toward establishing trust in adaptive agents. In *Proceedings of the 13th International Conference on Intelligent User Interfaces, IUI 2008, Gran Canaria, Canary Islands, Spain, January 13-16, 2008* (2008), J. M. Bradshaw, H. Lieberman, and S. Staab, Eds., ACM, pp. 227–236.
- [56] GOGGIN, S. N., BYRNE, M. D., AND GILBERT, J. E. Post-election auditing: Effects of procedure and ballot type on manual counting accuracy, efficiency, and auditor satisfaction and confidence.
- [57] GREENE, K. K., BYRNE, M. D., AND EVERETT, S. P. A comparison of usability between voting methods. In *2006 USENIX/ACCURATE Electronic Voting Technology Workshop, EVT'06* (2006).
- [58] HAINES, T., AND BOYEN, X. VOTOR: conceptually simple remote voting against tiny tyrants. In *Proceedings of the Australasian Computer Science Week Multiconference, Canberra, Australia, February 2-5, 2016* (2016), ACM, p. 32.
- [59] HANEL, P., AND VIONE, K. Do student samples provide an accurate estimate of the general public? *PLoS ONE* 11 (12 2016).
- [60] HAO, F., AND RYAN, P. Y. *Real-World Electronic Voting*. CRC Press, 2017, ch. 1 - Software independence revisited.
- [61] HERRNISON, P. S., NIEMI, R. G., HANMER, M. J., BEDERSON, B. B., CONRAD, F. G., AND TRAUOGOTT, M. The importance of usability testing of voting systems. In *2006 USENIX/ACCURATE Electronic Voting Technology Workshop, EVT'06* (2006).
- [62] HUANG, D., RAU, P. P., SALVENDY, G., GAO, F., AND ZHOU, J. Factors affecting perception of information security and their impacts on IT adoption and security practices. *Int. J. Hum.-Comput. Stud.* 69, 12 (2011), 870–883.
- [63] IOVINO, V., RIAL, A., RØNNE, P. B., AND RYAN, P. Y. A. Using selene to verify your vote in JCJ. In *Financial Cryptography and Data Security - FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA* (2017).
- [64] JAKOBSSON, M., AND JUELS, A. Mix and match: Secure function evaluation via ciphertexts. In *International Conference on the Theory and Application of Cryptology and Information Security* (2000), Springer, pp. 162–177.
- [65] JOHNSON-LAIRD, P. N. *Mental models: Towards a cognitive science of language, inference, and consciousness*. No. 6. Harvard University Press, 1983.

- [66] JUELS, A., CATALANO, D., AND JAKOBSSON, M. Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, WPES 2005* (2005).
- [67] KANG, R., DABBISH, L., FRUCHTER, N., AND KIESLER, S. “my data just goes everywhere:” user mental models of the internet and implications for privacy and security. In *Proceedings of the 2015 SOUPS Symposium on Usable Privacy and Security* (Berkeley, CA, USA, 2015), USENIX Association, pp. 39–52.
- [68] KARAYUMAK, F., KAUER, M., OLEMBO, M. M., VOLK, T., AND VOLKAMER, M. User study of the improved helios voting system interfaces. In *1st Workshop on Socio-Technical Aspects in Security and Trust, STAST 2011* (2011).
- [69] KIRLAPPOS, I., PARKIN, S., AND SASSE, A. "shadow security" as a tool for the learning organization. *ACM SIGCAS Computers and Society* 45 (02 2015), 29–37.
- [70] KIRLAPPOS, I., AND SASSE, A. What usable security really means: Trusting and engaging users. 69–78.
- [71] KREMER, S., RYAN, M., AND SMYTH, B. Election verifiability in electronic voting protocols. In *Computer Security - ESORICS 2010, 15th European Symposium on Research in Computer Security* (2010).
- [72] KULESZA, T., STUMPF, S., BURNETT, M., YANG, S., KWAN, I., AND WONG, W. Too much, too little, or just right? ways explanations impact end users’ mental models. In *Proceedings of the IEEE Symposium on Visual Languages and Human Centric Computing* (Sep. 2013), pp. 3–10.
- [73] KULYK, O., NEUMANN, S., BUDURUSHI, J., AND VOLKAMER, M. Nothing comes for free: How much usability can you sacrifice for security? *IEEE Security & Privacy* (2017).
- [74] KULYK, O., AND VOLKAMER, M. Usability is not enough: Lessons learned from ‘human factors in security’ research for verifiability. *IACR Cryptology ePrint Archive* (2018).
- [75] KÜSTERS, R., MÜLLER, J., SCAPIN, E., AND TRUDERUNG, T. select: A lightweight verifiable remote voting system. In *IEEE 29th Computer Security Foundations Symposium, CSF 2016* (2016).
- [76] KÜSTERS, R., TRUDERUNG, T., AND VOGT, A. Accountability: definition and relationship to verifiability. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010* (2010).

- [77] KÜSTERS, R., TRUDERUNG, T., AND VOGT, A. Verifiability, privacy, and coercion-resistance: New insights from a case study. In *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA* (2011).
- [78] LALLEMAND, C., AND GRONIER, G. *Méthodes de design UX : 30 méthodes fondamentales pour concevoir des expériences optimales*. France, Paris : Eyrolles, 2018.
- [79] LALLEMAND, C., AND KOENIG, V. Lab testing beyond usability: Challenges and recommendations for assessing user experiences. *Journal of Usability Studies* (2017).
- [80] LAZAR, J., FENG, J., AND HOCHHEISER, H. *Research Methods in Human-Computer Interaction, 2nd Edition*. Morgan Kaufmann, 2017.
- [81] LEVITT, S. D., AND LIST, J. A. What do laboratory experiments tell us about the real world. *Journal of Economic Perspectives* (2007).
- [82] LINDEMAN, M., AND STARK, P. B. A gentle introduction to risk-limiting audits. *IEEE Security & Privacy* 10, 5 (2012), 42–49.
- [83] LLEWELLYN, M., SCHNEIDER, S., XIA, Z., CULNANE, C., HEATHER, J., RYAN, P. Y. A., AND SRINIVASAN, S. Testing voters’ understanding of a security mechanism used in verifiable voting. In *2013 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, EVT/WOTE ’13, Washington, D.C., USA, August 12-13, 2013* (2013), USENIX Association.
- [84] LUHMANN, N. *Trust and Power*, 3 ed. Polity Press, The address, 11 2017.
- [85] LUNDIN, D., AND RYAN, P. Y. A. Human readable paper verification of prêt à voter. In *Computer Security - ESORICS 2008, 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings* (2008), S. Jajodia and J. López, Eds., vol. 5283 of *Lecture Notes in Computer Science*, Springer, pp. 379–395.
- [86] MAC NAMARA, D., SCULLY, T., AND GIBSON, P. Dualvote addressing usability and verifiability issues in electronic voting systems, 2011. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.399.7284>.
- [87] MALHEIROS, M., JENNETT, C., SEAGER, W., AND SASSE, A. Trusting to learn: Trust and privacy issues in serious games. 116–130.
- [88] MARKY, K., KULYK, O., RENAUD, K., AND VOLKAMER, M. What did I really vote for? On the usability of verifiable e-voting schemes. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)* (New York, NY, USA, 2018), ACM, pp. 176:1–176:13.

- [89] MARKY, K., ZIMMERMANN, V., FUNK, M., DAUBERT, J., BLECK, K., AND MÜHLHÄUSER, M. Improving the usability and ux of the swiss internet voting interface. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)* (New York, NY, USA, 2020), ACM, pp. 640:1–640:13.
- [90] MARKY, K., ZOLLINGER, M.-L., FUNK, M., RYAN, P. Y., AND MÜHLHÄUSER, M. How to assess the usability metrics of e-voting schemes. In *Financial Cryptography and Data Security - FC 2019* (2019).
- [91] MAURER, U. M., AND SCHMID, P. E. A calculus for secure channel establishment in open networks. In *Computer Security - ESORICS 94, Third European Symposium on Research in Computer Security* (1994).
- [92] NEMEC, M., SYS, M., SVENDA, P., KLINEC, D., AND MATYAS, V. The return of coppersmith’s attack: Practical factorization of widely used rsa moduli. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), ACM, pp. 1631–1648.
- [93] NESTAS, L., AND HOLE, K. Building and maintaining trust in internet voting. *Computer* 45, 5 (May 2012), 74–80.
- [94] NETO, A., LEITE, M., ARAUJO, R., MOTA, M., NETO, N., AND TRAORÉ, J. Usability considerations for coercion-resistant election systems. In *IHC 2018: 17th Brazilian Symposium on Human Factors in Computing Systems, Belém Brazil, October 2018* (10 2018), pp. 1–10.
- [95] NORMAN, D. A. *The Design of Everyday Things*. Basic Books, 2013.
- [96] NORMAN, D. A., AND DRAPER, S. W. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1986.
- [97] OKAMOTO, T. An electronic voting scheme. In *Advanced IT Tools, IFIP World Conference on IT Tools* (1996).
- [98] OLEMBO, M., RENAUD, K., BARTSCH, S., AND VOLKAMER, M. Voter, what message will motivate you to verify your vote? In *Workshop on Usable Security (USEC) 2014* (2014).
- [99] OLEMBO, M. M., BARTSCH, S., AND VOLKAMER, M. Mental models of verifiability in voting. In *E-Voting and Identify - 4th International Conference, Vote-ID 2013* (2013).
- [100] PEREIRA, O., AND RIVEST, R. L. Marked mix-nets. In *International Conference on Financial Cryptography and Data Security* (2017), Springer, pp. 353–369.

- [101] PHAN, D. H., AND POINTCHEVAL, D. Oaep 3-round: A generic and secure asymmetric encryption padding. In *International Conference on the Theory and Application of Cryptology and Information Security* (2004), Springer, pp. 63–77.
- [102] PIETERS, W. Explanation and trust: what to tell the user in security and AI? *Ethics and Information Technology* 13, 1 (nov 2010), 53–64.
- [103] PROLIFIC. Prolific. <https://www.prolific.co/>.
- [104] RIEGELSBERGER, J., SASSE, A., AND MCCARTHY, J. The mechanics of trust: A framework for research and design. *International Journal of Human-Computer Studies* 62 (03 2005), 381–422.
- [105] RIVEST, R. On the notion of ‘software independence’ in voting systems. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* 366 (11 2008), 3759–67.
- [106] RIVEST, R., AND SMITH, W. Three voting protocols: Threeballot, vav, and twin. *EVT* (08 2007).
- [107] RØNNE, P. B., RYAN, P. Y., AND ZOLLINGER, M.-L. Electryo, in-person voting with transparent voter verifiability and eligibility verifiability. In *Third International Joint Conference on Electronic Voting E-Vote-ID 2018, TUT Press Proceedings* (2018).
- [108] RYAN, P. Y., BISMARCK, D., HEATHER, J., SCHNEIDER, S., AND XIA, Z. Prêt à voter: a voter-verifiable voting system. *IEEE transactions on information forensics and security* 4, 4 (2009), 662–673.
- [109] RYAN, P. Y. A., RØNNE, P. B., AND IOVINO, V. Selene: Voting with transparent verifiability and coercion-mitigation. In *Financial Cryptography and Data Security - FC 2016* (2016).
- [110] RYAN, P. Y. A., AND SCHNEIDER, S. A. Process algebra and non-interference. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop, CSFW 1999, Mordano, Italy, June 28-30, 1999* (1999), IEEE Computer Society, pp. 214–227.
- [111] SAKO, K., AND KILIAN, J. Receipt-free mix-type voting scheme - A practical solution to the implementation of a voting booth. In *Advances in Cryptology - EUROCRYPT ’95, International Conference on the Theory and Application of Cryptographic Techniques* (1995).
- [112] SALLAL, M., SCHNEIDER, S., CASEY, M., DRAGAN, C., DUPRESSOIR, F., RILEY, L., TREHARNE, H., WADSWORTH, J., AND WRIGHT, P. Vmv: Augmenting an internet voting system with selene verifiability, 2019.

- [113] SCHMIDT, B., MEIER, S., CREMERS, C. J. F., AND BASIN, D. A. Automated analysis of diffie-hellman protocols and advanced security properties. In *25th IEEE Computer Security Foundations Symposium, CSF 2012* (2012).
- [114] SCHNEIDER, S., LLEWELLYN, M., CULNANE, C., HEATHER, J., SRINIVASAN, S., AND XIA, Z. Focus group views on prêt à voter 1.0. In *2011 International Workshop on Requirements Engineering for Electronic Voting Systems, REVOTE 2011* (2011).
- [115] SCHREPP, M. User experience questionnaire handbook. <https://www.ueq-online.org/>, 2018.
- [116] SELKER, T., ROSENZWEIG, E., AND PANDOLFO, A. A methodology for testing voting systems. *Journal of Usability Studies* 2, 1 (Nov. 2006), 7–21.
- [117] SERDULT, U., GERMANN, M., MENDEZ, F., PORTENIER, A., AND WELLIG, C. Fifteen years of internet voting in switzerland [history, governance and use]. In *2015 Second International Conference on eDemocracy eGovernment (ICEDEG)* (2015).
- [118] SERDÜLT, U., AND KRYSSANOV, V. Internet Voting User Rates and Trust in Switzerland. In *Third International Joint Conference on Electronic Voting: E-Vote-ID 2018* (October 2018), R. Krimmer, M. Volkamer, V. Cortier, et al, and U. Serdült, Eds., E-Voting.CC GmbH, pp. 211–212.
- [119] SHELDON, K., ELLIOT, A., KIM, Y., AND KASSER, T. What is satisfying about satisfying events? testing 10 candidate psychological needs. *Journal of personality and social psychology* 80 (03 2001), 325–39.
- [120] SHELDON, K., ELLIOT, A., KIM, Y., AND KASSER, T. What is satisfying about satisfying events? testing 10 candidate psychological needs. *Journal of personality and social psychology* 80 (03 2001), 325–39.
- [121] STANDARDIZATION, I. O. F. Iso 9241-11: Ergonomics of human system interaction – part 11: Guidance on usability, 1998.
- [122] STANDARDIZATION, I. O. F. Iso 9241-210: Ergonomics of human system interaction – part 210: Human-centred design for interactive systems, 1999.
- [123] TULLIO, J., DEY, A. K., CHALECKI, J., AND FOGARTY, J. How it works: A field study of non-technical users interacting with an intelligent system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), CHI '07, Association for Computing Machinery, p. 31–40.

- [124] TULLIS, T., AND ALBERT, W. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics: Second Edition*. 01 2008.
- [125] WARKENTIN, M., SHARMA, S., GEFEN, D., ROSE, G. M., AND PAVLOU, P. Social identity and trust in internet-based voting adoption. *Government Information Quarterly* 35, 2 (2018), 195–209.
- [126] WEBER, J.-L., AND HENGARTNER, U. Usability study of the open audit voting system helios. <http://www.jannaweber.com/wp-content/uploads/2009/09/858Helios.pdf>, 2009.
- [127] WIKSTRÖM, D. How to implement a stand-alone verifier for the verificatum mix-net, 2012.
- [128] WIKSTRÖM, D. User manual for the verificatum mix-net.
- [129] ZENG, E., MARE, S., AND ROESNER, F. End user security & privacy concerns with smart homes. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)* (Berkeley, CA, USA, 2017), USENIX Association, pp. 65–80.
- [130] ZOLLINGER, M., DISTLER, V., RØNNE, P. B., RYAN, P. Y., LALLEMAND, C., AND KOENIG, V. User experience design for e-voting: How mental models align with security mechanisms. In *Fourth Joint International Conference on Electronic Voting, E-Vote-ID 2019, TalTech Proceedings* (2019).
- [131] ZOLLINGER, M.-L. Selene explained to our study’s participants. https://youtu.be/ziTAM_ZKFRw, 2020.

Appendix A

Questionnaires

A.1 System Usability Scale Questionnaire

	Strongly disagree		Strongly agree		
I think that I would like to use this system frequently	1	2	3	4	5
I found the system unnecessarily complex	1	2	3	4	5
I thought the system was easy to use	1	2	3	4	5
I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
I found the various functions in this system were well integrated	1	2	3	4	5
I thought there was too much inconsistency in this system	1	2	3	4	5
I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
I found the system very cumbersome to use	1	2	3	4	5
I felt very confident using the system	1	2	3	4	5
I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

A.2 User Experience Questionnaire

	1	2	3	4	5	6	7	
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	understandable
creative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull
easy to learn	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn
valuable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	exciting
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	interesting
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	predictable
fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow
inventive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	supportive
good	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasing
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	leading edge
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasant
secure	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure
motivating	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating
meets expectations	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	efficient
clear	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	practical
organized	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered
attractive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive
friendly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	innovative

A.3 Psychological Needs Questionnaire

During this interaction, I felt...

	1	2	3	4	5
... that my actions were based on my interests.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... free to do things my own way.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... free from any pressure or influence.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... free of having to make meaningful choices.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... that I was successfully completing tasks.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... I mastered complex situations.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... very capable in what I did.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... I could achieve my goals.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... I performed well.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... a sense of contact with other people in general.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... close and connected with people who are important to me.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... cared for.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... aware of others' emotions, activities, or mood.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... that I was experiencing new activities.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... that I experienced enjoyable sensations.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... physical or emotional pleasure.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... that I discovered new sources and types of stimulation.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... that things were structured and predictable.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... that I could frequently apply my routines and habits.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... that I could act in a safe and secure way.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... I understood how things worked.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... in control.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... that I was a person whose opinion counts for others.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... that I influenced others.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... as someone that others take as a person who can give guidance.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... I an a liable person	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... my actions were with purpose.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... my actions conformed to my values.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... a sense of fulfillment.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... being a person of value.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Appendix B



Mental Models



B.1 Drawing material

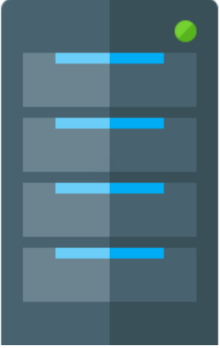


The following material was given to the participants for the drawing sessions.

- Paper and pens,
- a non-exhaustive list of entities on post-its (see below),
- the screenshots of the app.

The following icons were provided to help them in the drawing session.

Application	
Device	

Ballot	<p style="text-align: center;">Stimmzettel für die Wahl zum Deutschen Bundestag am 24. September 2017 im Wahlkreis 186 – Darmstadt</p> <p style="text-align: center;">Sie haben 2 Stimmen</p> <p>hier 1 Stimme für die Wahl eines Ihrer Wahlkreisabgeordneten</p> <p>hier 1 Stimme für die Wahl einer Landesliste (Partei) – möglichen Stimmkreis für die Verteilung der Sitze insgesamt auf die einzelnen Parteien –</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: left;">Erststimme</th> <th colspan="2" style="text-align: left;">Zweitstimme</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Dr. Müssen, Adolf Mitglied des Bundestages</td> <td>CDU</td> <td>Christlich Demokratische Union Deutschlands</td> </tr> <tr> <td>2</td> <td>Spillner, Christof Mitglied des Bundestages</td> <td>SPD</td> <td>Sozialdemokratische Partei Deutschlands</td> </tr> <tr> <td>3</td> <td>Wegmann, Christian Landtagsabgeordneter</td> <td>GRÜNE</td> <td>Grüne Partei für eine bessere Welt</td> </tr> <tr> <td>4</td> <td>Friedrich, Michael Landtagsabgeordneter</td> <td>DIE LINKE</td> <td>Die Linke – Partei der Arbeit</td> </tr> <tr> <td>5</td> <td>Katzenbach, Frank Landtagsabgeordneter</td> <td>AfD</td> <td>Alternative für Deutschland</td> </tr> <tr> <td>6</td> <td>Mühlhölzer, Michael Landtagsabgeordneter</td> <td>FDP</td> <td>Freie Demokratische Partei</td> </tr> <tr> <td>7</td> <td></td> <td>PIRATEN</td> <td>Piratenpartei Deutschland</td> </tr> <tr> <td>8</td> <td>Baumann, Friedrich Landtagsabgeordneter</td> <td>FRD</td> <td>Freie Demokratische Partei</td> </tr> <tr> <td>9</td> <td></td> <td>FRD</td> <td>Freie Demokratische Partei</td> </tr> <tr> <td>10</td> <td></td> <td>FRD</td> <td>Freie Demokratische Partei</td> </tr> <tr> <td>11</td> <td></td> <td>FRD</td> <td>Freie Demokratische Partei</td> </tr> <tr> <td>12</td> <td>Schäpe, Axel Landtagsabgeordneter</td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>13</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>14</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>15</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>16</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>17</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>18</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>19</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>20</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>21</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>22</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>23</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>24</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>25</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>26</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>27</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>28</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>29</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> <tr> <td>30</td> <td></td> <td>MLPD</td> <td>Marxistische-Leninistische Partei Deutschlands</td> </tr> </tbody> </table>	Erststimme		Zweitstimme		1	Dr. Müssen, Adolf Mitglied des Bundestages	CDU	Christlich Demokratische Union Deutschlands	2	Spillner, Christof Mitglied des Bundestages	SPD	Sozialdemokratische Partei Deutschlands	3	Wegmann, Christian Landtagsabgeordneter	GRÜNE	Grüne Partei für eine bessere Welt	4	Friedrich, Michael Landtagsabgeordneter	DIE LINKE	Die Linke – Partei der Arbeit	5	Katzenbach, Frank Landtagsabgeordneter	AfD	Alternative für Deutschland	6	Mühlhölzer, Michael Landtagsabgeordneter	FDP	Freie Demokratische Partei	7		PIRATEN	Piratenpartei Deutschland	8	Baumann, Friedrich Landtagsabgeordneter	FRD	Freie Demokratische Partei	9		FRD	Freie Demokratische Partei	10		FRD	Freie Demokratische Partei	11		FRD	Freie Demokratische Partei	12	Schäpe, Axel Landtagsabgeordneter	MLPD	Marxistische-Leninistische Partei Deutschlands	13		MLPD	Marxistische-Leninistische Partei Deutschlands	14		MLPD	Marxistische-Leninistische Partei Deutschlands	15		MLPD	Marxistische-Leninistische Partei Deutschlands	16		MLPD	Marxistische-Leninistische Partei Deutschlands	17		MLPD	Marxistische-Leninistische Partei Deutschlands	18		MLPD	Marxistische-Leninistische Partei Deutschlands	19		MLPD	Marxistische-Leninistische Partei Deutschlands	20		MLPD	Marxistische-Leninistische Partei Deutschlands	21		MLPD	Marxistische-Leninistische Partei Deutschlands	22		MLPD	Marxistische-Leninistische Partei Deutschlands	23		MLPD	Marxistische-Leninistische Partei Deutschlands	24		MLPD	Marxistische-Leninistische Partei Deutschlands	25		MLPD	Marxistische-Leninistische Partei Deutschlands	26		MLPD	Marxistische-Leninistische Partei Deutschlands	27		MLPD	Marxistische-Leninistische Partei Deutschlands	28		MLPD	Marxistische-Leninistische Partei Deutschlands	29		MLPD	Marxistische-Leninistische Partei Deutschlands	30		MLPD	Marxistische-Leninistische Partei Deutschlands
Erststimme		Zweitstimme																																																																																																																											
1	Dr. Müssen, Adolf Mitglied des Bundestages	CDU	Christlich Demokratische Union Deutschlands																																																																																																																										
2	Spillner, Christof Mitglied des Bundestages	SPD	Sozialdemokratische Partei Deutschlands																																																																																																																										
3	Wegmann, Christian Landtagsabgeordneter	GRÜNE	Grüne Partei für eine bessere Welt																																																																																																																										
4	Friedrich, Michael Landtagsabgeordneter	DIE LINKE	Die Linke – Partei der Arbeit																																																																																																																										
5	Katzenbach, Frank Landtagsabgeordneter	AfD	Alternative für Deutschland																																																																																																																										
6	Mühlhölzer, Michael Landtagsabgeordneter	FDP	Freie Demokratische Partei																																																																																																																										
7		PIRATEN	Piratenpartei Deutschland																																																																																																																										
8	Baumann, Friedrich Landtagsabgeordneter	FRD	Freie Demokratische Partei																																																																																																																										
9		FRD	Freie Demokratische Partei																																																																																																																										
10		FRD	Freie Demokratische Partei																																																																																																																										
11		FRD	Freie Demokratische Partei																																																																																																																										
12	Schäpe, Axel Landtagsabgeordneter	MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
13		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
14		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
15		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
16		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
17		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
18		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
19		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
20		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
21		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
22		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
23		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
24		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
25		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
26		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
27		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
28		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
29		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
30		MLPD	Marxistische-Leninistische Partei Deutschlands																																																																																																																										
Ballot Box																																																																																																																													
Internet																																																																																																																													

Server	
Encryption	
Voter	

B.2 Mental Models

In table B.3, we give the distribution of the levels of sophistication over the participants in column 2 “Level”. We also give the number of occurrences of a given mental model per participant, denoted as follows:

- TU: Technology Understanding
- MV: Meaning of the Verification Phase
- SC: Security Concerns
- US: Unnecessary steps

For example, the participant P10 had a structural advanced level of sophistication, mentioned 15 times a mental model about Technology Understanding, 6 times about the Meaning of the Verification Phase, 8 times mentioned Security Concerns and 3 times that there was Unnecessary steps.

Participant	Level	Mental Models			
		TU	MV	SC	US
P01	Structural advanced	7	11	16	
P02	Structural basic	15	5	10	
P03	Structural basic	5	2	17	4
P04	Structural advanced	6	8	5	
P05	Structural advanced	12	5	14	6
P06	Functional advanced	11	4	4	2
P07	Structural basic	8	2	4	
P08	Functional advanced	9	6	7	4
P09	Structural advanced	13	8	15	7
P10	Structural advanced	15	6	8	3
P11	Functional advanced	12	7	4	3
P12	Structural advanced	22	8	10	6
P13	Functional advanced	4	3	3	
P14	Structural basic	11	2	6	1
P15	Structural basic	12	4	5	
P16	Structural basic	23	2	15	1
P17	Structural advanced	14	5	9	3
P18	Structural basic	14	10	7	
P19	Structural basic	15	2	15	1
P20	Structural basic	11	8	18	
P21	Functional advanced	6	7	4	
P22	Functional basic	5	6	6	2
P23	Functional advanced	9	7	12	
P24	Functional advanced	14	3	5	3

Table B.3: Distribution of mental models.

Appendix C

Trust and Understanding

C.1 Study Description

This figure below is the study description given to the participants in the online study described in chapter 5.

The aim of this study is to evaluate a new online voting system. This new system provides a feature to let you verify your vote, by giving you a tracking code. In addition, to preserve your privacy against someone asking for your tracking code, a feature allows you to hide your vote.

1/ Click on the link to start the study. First you will receive a consent form to sign and a demographic questionnaire. In the questionnaire, we ask about your opinion to configure the study (the same voting question will be asked in step 5). This opinion is used only for the configuration, we won't use it in the research.

Then you can watch a 4-minutes video (included in the allocated time), that explains the voting system. In case this is not possible for you, you can find out the information in the application. It is very important that you take note of this information. Be aware that we will ask easy questions in the questionnaire regarding this available information.

2/ After step 1, you will be redirected on the e-voting system. Your credentials to connect are:

- Login: your Prolific ID
- Pwd: F5t8k09!

3/ You must cast a vote.

4/ After your vote has been sent, you must assume that the elections are over and you can go to the verification page. From there, we have included instructions in the app, as a tutorial, to help you test the application. After having followed all instructions, you can go to a questionnaire to continue the study.

5/ After having filled the questionnaire, you will enter the last phase of the user study, and you must go again through the application. Other instructions will be given directly in the application, representing the will of a vote buyer. This is a game developed to evaluate an additional scenario. After the verification, you can finish the study.

How to opt out

At any moment, you can decide to stop the study. If you do so, please be informed that you won't be paid. If you want to opt out after your submission, please send an email to marie-laure.zollinger@uni.lu with your prolific ID and your data will be removed.

Contact information

If you have any question or issue for the payment, please send an email to marie-laure.zollinger@uni.lu

Ethical approval has been obtained for this study, you can visit the following link for more information: https://wwen.uni.lu/research/researchers_research/standards_policies

C.2 Post-hoc Tukey between feelings and UEQ scores

A Welch ANOVA has been run for the items Attractiveness, Perspicuity and Efficiency. We found a significant relation between attractiveness and the feelings reported by participants ($F(7, 56.04) = 10.22, p = 0$). Similarly, we found a significant relation between perspicuity and feelings ($F(7, 55.78) = 14.79, p = 0$) and between efficiency and feelings ($F(7, 57.11) = 11.45, p = 0$).

A standard 1-way ANOVA has been run for the items Dependability, Stimulation and Novelty and we found the following relations with the feelings' feedback: for dependability $F(7, 291) = 8.64, p = 0$; for stimulation $F(7, 291) = 8.57, p = 0$; and for novelty $F(7, 291) = 3.55, p = 0.001$.

The following table will report the significant results after running a post-hoc tukey test.

		Difference between the means	P value
Attractiveness	Good over Overwhelmed	1.17	0.008
	Good over Stressed	0.88	0.036
	Good over Offended	1.16	0.001
	Good over Confused	1.16	0
	Interested over Overwhelmed	1.52	0.001
	Interested over Stressed	1.24	0.005
	Interested over Offended	1.51	0
	Interested over Confused	1.52	0
	Strong over Offended	1.06	0.017
	Strong over Confused	1.07	0.001
Perspicuity	Good over Overwhelmed	2.18	0
	Good over Stressed	1.35	0.001
	Good over Offended	1.18	0.004
	Good over Confused	1.69	0
	Interested over Overwhelmed	2.02	0
	Interested over Stressed	1.19	0.033
	Interested over Confused	1.52	0
	Strong over Overwhelmed	1.80	0
	Strong over Confused	1.30	0
Efficiency	Good over Overwhelmed	1.54	0
	Good over Stressed	1.00	0.022
	Good over Offended	1.39	0
	Good over Confused	1.36	0

	Interested over Overwhelmed	1.72	0
	Interested over Stressed	1.18	0.023
	Interested over Offended	1.57	0
	Interested over Confused	1.54	0
	Strong over Overwhelmed	1.41	0.006
	Strong over Offended	1.26	0.005
	Strong over Confused	1.23	0
Dependability	Good over Overwhelmed	1.34	0
	Good over Stressed	0.85	0.007
	Good over Offended	0.97	0.001
	Good over Confused	0.95	0
	Interested over Overwhelmed	1.19	0.003
	Interested over Confused	0.8	0.009
	Strong over Overwhelmed	1.2	0.001
	Strong over Offended	0.83	0.033
	Strong over Confused	0.80	0.003
Stimulation	Good over Stressed	0.84	0.01
	Good over Offended	1.2	0
	Good over Confused	1.02	0
	Interested over Overwhelmed	1.13	0.027
	Interested over Stressed	1.11	0.01
	Interested over Offended	1.46	0
	Interested over Confused	1.29	0
	Strong over Offended	1.09	0.006
	Strong over Confused	0.9	0.003
Novelty	Good over Confused	0.58	0.025
	Interested over Offended	1.00	0.016
	Interested over Confused	0.83	0.012

Table C.2: Post-Hoc Tukey significant results between feelings and UEQ scores.