Robust Electronic Voting: Introducing Robustness in Civitas

Fateme Shirazi  
CASED \ TU Darmstadt  
Darmstadt, Germany  
fateme.shirazi@cased.de

Stephan Neumann  
Saarland University  
Saarbrücken, Germany  
s9neumann@stud.uni-saarland.de

Ines Ciolacu  
Saarland University  
Saarbrücken, Germany  
s9ciolacu@stud.uni-saarland.de

Melanie Volkamer  
CASED \ TU Darmstadt  
Darmstadt, Germany  
melanie.volkamer@cased.de

Abstract—Civitas is a remote electronic voting system, providing verifiability and some coercion resistance. It is a refinement of a cryptographic voting scheme proposed by Juels, Catalano, and Jakobsson in 2005. In this paper we analyze the robustness of Civitas. In electronic voting, robustness has different interpretations. Tally availability is the most common interpretation. In addition to this interpretation, we also consider the availability of the election for every willing voter (voting availability). For both criteria a formal definition is provided. It is shown, that Civitas does not comply with this definition. Therefore, we extend Civitas in order to overcome this shortcoming. This extension also tackles a coercion resistance vulnerability which was identified by Küsters and Truderung in 2009.

Keywords—electronic voting; internet voting, robustness, formal definition, Civitas

I. INTRODUCTION

Cryptographic primitives and protocols are becoming more and more important for a wide variety of distributed computing tasks where the processing agents are either unreliable or untrustworthy. One of the important applications are governmental elections. Electronic voting refers to classical voting with the help of some electronic means and it can be applied either remotely over the internet or in polling stations. Cryptographers have been proposing constructions for electronic voting since 1980s with a first proposal by David Chaum [4]. After three decades of research effort, we also see some real remote electronic voting systems being used over the world like in Estonia and Switzerland.

Numerous functional and security requirements for electronic voting systems have been defined on various levels: from a legal point of view like in [16], in an informal manner like in [9] and [11], in a semi formal manner like in [24] and [21], and in a formal manner like in [22], [14], [20] and [16]. The most popular requirements are eligibility, fairness, vote-privacy, receipt-freeness, coercion resistance, individual and universal verifiability.

Even though there is an intense interest in these properties, there are still others which have not been analyzed with such great care. Among them we mention the robustness aspect. A general definition of robustness would be that it ensures the quality of being able to withstand stresses, pressures, or changes in procedure or circumstance. A system, organism or design may be said to be “robust” if it is capable of coping well with variations (sometimes unpredictable variations) in its operating environment with minimal damage, alteration or loss of functionality. As these aspects of robustness do mainly address the operational environment, they are often formalized as assumption like in the Common Criteria Protection Profile for internet voting systems [24] and, therefore, not focused on in electronic voting protocol research and papers.

In our opinion the robustness property should be broader then the above one and should also be part of the protocol analysis. Electronic voting schemes should ensure that even given a distributed and faulty environment, the final tally is outputted. Furthermore, the election process should be available for every voter who is willing to cast a vote.

We review in this paper existing literature on robustness and availability definitions and extend them by own ones both in informal and formal manner. We apply this robustness definition to Civitas [6] one of the most popular and examined voting schemes. We show that it does not comply with our definition of robustness which shows that it is important to take all aspects of robustness into account.

Therefore, we present an improvement of Civitas to overcome these shortcomings. The extension we propose has two design options. The design options have different influence on usability and coercion resistance. In an analysis we show that one of the improved version does not violate any security requirements while the other one solves the coercion resistance problem of Civitas identified in [15] while decreasing user friendliness.

This paper is organized as follows. Section II, offers a brief review on the related work concerning Civitas extensions. Section III is dedicated to the robustness definitions from the literature and to the adaption of that definition with respect to our needs. We proceed by describing the framework of Civitas, its design phases, and trust assumptions. In Section V, we analyze Civitas with respect to robustness and propose an extension for Civitas to improve the robustness properties of Civitas in Section VI. Section VII investigates the proposed extension of Civitas regarding security and usability issues. Finally, concluding remarks on the extension and future work are outlined.
II. RELATED WORK

In this section, we will briefly discuss other relevant work in the context of Civitas and improvements of the protocol while we deal with related work on requirement definition for robustness in Section III. Küsters and Truderung [15] proposed a coercion resistance definition in a symbolic way. Based on that definition, they analyzed Civitas and discovered two coercion resistance flaws. Corresponding on these flaws, they suggested proper improvements. In [2] the authors present a formalization and security proof for JCJ, the protocol underlying Civitas, in applied Pi-calculus [1]. Smyth et al. adopted the approach of [2] to the Civitas protocol in their work [19].

III. ROBUSTNESS DEFINITIONS

In this section we will first provide existing definitions and aspects of robustness in electronic voting and then extend these definitions by our own ones, first in an informal then in a formal manner.

A. Existing Definitions

There exists a couple of different ways and intentions to define robustness in the context of electronic voting.

A typical requirement of a voting system as defined for instance in [24] and [21] is that it should be available during the whole vote casting period, i.e. voters intending to vote should have access to the voting system at any time. This covers protecting against denial of service attacks which is addressed in [3]. Here the authors propose to use Peer-to-Peer web caches to achieve a reliable messaging system and thus being resistant against distributed denial of service attacks. We call this aspect of robustness service and network availability. This includes according to [21], being robust "against power outage at the voting server, unexpected user activity, environmental effects (for instance, mechanical, electromagnetic, and climatic) to the voting server, and network problems".

Joaquim et al. motivated their scheme called REVS proposed in [13] by the necessity of a fault-tolerant remote electronic voting system. REVS uses replication as the basic mechanism to tolerate system failures in communications, servers and voters applications. A similar requirement to the environment in which a remote electronic voting system is used is defined in [24]. This aspect of robustness is called fault-tolerance. It includes that no valid votes can get lost due to storage problems.

Note, these two issues are not part of what an electronic voting scheme can accomplish but only the environment in which it is used by techniques like redundancy and appropriate back up strategies.

Another aspect of robustness is the availability of the election results based on the stored encrypted votes, i.e. in particular it should be possible to decrypt votes and to produce the final tally. In a simple definition this should hold in general and in an extended definition - according to [8] - this should even hold in the presence of an adversary. Here, procedures are required to ensure that all or in the second case enough keys in terms of shares of the decryption key are available to decrypt votes or the encrypted sum. This aspect of robustness is called tallying availability.

The authors of [12] investigate another aspect of robustness that is in the context of mix cascades used to anonymize encrypted votes before decrypting and tallying them. The authors define the term robustness as providing strong evidence that the mixing procedure has been carried out correctly even in the presence of malicious participants. This approach addresses the accuracy of the tallying process in the presence of an adversary. Another aspect is addressed in [5]. Here, the authors interpret robustness as the possibility of detecting unauthorized votes in the electronic ballot box even if authorities collude. Since from our point of view accuracy is not the intended purpose of robustness we rule it out from the aspects that define robustness in this paper. However, we want to emphasize that robustness is a culmination of completeness and accuracy in an electronic voting scheme.

Note, while the two first aspects of robustness address external attackers and faults that are not necessary caused by an attacker, the last two aspects of robustness address the fact that electronic voting systems should be robust against single malicious authorities or entities involved in the electronic voting system set up, e.g. as key holder or as one MIX node. Determining the robustness against either single malicious entities or even collaboration ones is also discussed in [23]. Here, a so called $k$-resilience value is used to express the number of entities that need to cooperated maliciously in order to violate violate a particular requirement like secrecy of the vote or the integrity of the electoral roll.

B. Extended Robustness Definition and Corresponding Trust Model

All the above listed robustness definitions are important and should be considered by any electronic voting system and/or its operational requirements. For this paper we only concentrate on (remote) electronic voting schemes. As a scheme itself can only provide tallying availability while the other aspects needs to be ensured by the operational environment, we will only take this aspect into account for this paper.

In addition to the tallying availability, we propose to extend the robustness definition for electronic voting schemes by the aspect of voting availability by adding the following definition.

Definition 1 (Voting Availability).
All willing voters are able to submit their votes, thereby finishing their process (even in the presence of an attacker).

The motivation to extend tallying availability by voting
availability is based on the fact that the participation of voters is the main goal of electronic voting schemes. However, blocking authorized voters from voting has not been obviated sufficiently so far. As voting availability tends to overcome this shortcoming we propose to integrate it in the list of requirements in the category of robustness properties.

To sum up, a remote electronic voting scheme is only robust if it ensures tallying availability, and voting availability. To analyze these two aspects of robustness we adapt the threat model of Civitas: We assume that the adversary may corrupt a subset of election authorities carrying out the election process. More precisely, we allow the adversary to have the following abilities according to the protocol scheme:

- The adversary can analyze and synthesize messages. Not only might messages between protocol participants be blocked, but they might also be analyzed or composed respecting computation restrictions.
- The adversary can corrupt a threshold of the parties. Sometimes, the adversary might corrupt up to a certain number of parties carrying out an election for different reasons. Thereby, he may ask secrets of the parties or even force them to act on behalf of itself.
- The adversary is a probabilistic polynomial time machine. We restrict the adversary’s computation power in a reasonable way.

C. Formal Definition of Robustness

In this subsection we provide a fundamental definition of robustness which can be used in rigorous mathematical reasoning. It is based upon fundamental ideas of both, complexity theoretical concepts as well as cryptographic techniques. Therefore, we refresh the concept of a negligible function.

Definition 2 (Negligible Function).
A function \( \mu : \mathbb{N} \to [0,1] \) is called negligible iff for all polynomials \( p(\cdot) \), there exists \( N \), such that for all \( n > N \),

\[
\mu(n) < \frac{1}{p(n)}.
\]

In the following we denote by \( \text{tally}^{\pi(r_1,\ldots,r_n,A)}(k) \) the produced tally by carrying out protocol \( \pi \) using a set of participating machines \( r_1,\ldots,r_n \) in the presence of an adversary \( A \) under security parameter \( k \). We denote by \( \text{term}(p) \) that a process \( p \) terminated, hence reached the end of its algorithmic description.

These basics allow us to define the robustness in electronic voting schemes.

Definition 3 (Robust Electronic Voting Scheme).
Let \( \pi(R_1,R_2,\ldots,R_n) \) be a \( n \)-party e-voting protocol where \( R_i \) corresponds to the \( i \)-th protocol role and let \( \mu \) be a negligible function. We consider a protocol run \( \pi(r_1,r_2,\ldots,r_n,A) \) where \( r_i \) denotes the set of agents operating according to role \( R_i \) in the protocol and \( A \) denotes the adversary. Protocol \( \pi(R_1,R_2,\ldots,R_n) \) is robust if for all PPT adversaries \( A \) after the protocol run \( \pi(r_1,r_2,\ldots,r_n,A) \) the following holds:

\[
\Pr[\bigwedge_{v \in \text{term}} \text{tally}^{\pi(r_1,r_2,\ldots,r_n,A)}(v) \neq 0] \geq 1 - \mu(n).
\]

This definition covers both aspects voting and tallying availability in the presence of an attacker and is used to analyze Civitas in Section IV-B.

IV. CIVITAS VOTING SCHEME

Our work is based on the Civitas voting scheme [6]. Civitas is a refinement of JCJ [14] which provides security proofs for verifiability and coercion resistance. Although the basic structure of Civitas is based JCJ, it varies from JCJ in several ways. Such as improvements on coercion resistance, scalability, and availability of votes. Since Civitas inherits security properties of JCJ, it provides verifiability and coercion resistance.

Other than most of the proposed electronic voting schemes which are just designed at an abstract level, Civitas is actually implemented (from a functional point of view while there are no graphical user interfaces). From a practical point of view, Civitas is implemented by the type-based language Jif, a derivation of Java, allowing to enforce information flow policies.

A. Sketch of the Voting Protocol.

Civitas consists of five participating roles each represented by a polynomial time algorithm. Every participant incorporates one corresponding instantiation of these algorithms. The participating roles are as follows.

- Supervisor. A supervisor \( S \) administers an election. We consider as in the original paper that the supervisor is a non-corruptible party.
- Registrar. A registrar \( R \) authorizes the voters and also posts the list of authorized voters.
- Voter. A voter \( V \) is an entity that is authorized to vote. The voter’s algorithm consists of two parts, including credential generation process and vote submission.
- Registration Teller. A registration teller \( RT \) is responsible for distributing the private credential share of each voter and posting the corresponding public credential share on the bulletin board.
- Tabulation Teller. The tabulation teller’s \( TT \) algorithm consists of several parts including mixing, decrypting and tallying the votes.

Two storage services are used as follows:

- Ballot Boxes. The ballot boxes are used to store cast votes. It is an intermediate channel of the votes before they get retrieved by the tabulation tellers. The content of the ballot boxes is not public.
• Bulletin Board. A bulletin board \( BB \) is used which is a public storage channel to store all the information needed for verifiability of the election, therefore it is a publicly verifiable channel.

A protocol run of Civitas consists of the following phases: Setup, voting and tabulation. In the voting phase, the voter has to authenticate himself and receive the required credential for voting from the registration tellers. The voter proceeds then by sending his encrypted vote to (preferably more then one) ballot box(es). The tabulation phase is the phase in which the votes get tallied by the tabulation tellers. At this level the votes get retrieved by the tabulation tellers from the ballot box(es). Afterwards the votes are processed in order to eliminate the invalid ones and in order to anonymize remaining votes (shuffling them as in the mixnet algorithm [18]) such that they are ready to be decrypted, and tallied. The final tally then is posted on the bulletin board. Figure 1 illustrates the protocol structure of Civitas.

![Figure 1. Sketch of the Civitas electronic voting protocol](image)

B. Security Properties and Trust Assumptions

Civitas is one of the most powerful voting protocols around which in particular ensures integrity by a strong verifiability and confidentiality including coercion resistance in the sense of the following definitions and under the trust assumptions below.

**Definition 4** (Verifiability according to [6]).
The final tally is verifiable correct. Each voter can check that their own vote is included in the tally (voter verifiability). Anyone can check that all votes cast are counted, that only authorized votes are counted, and that no votes are changed during counting (universal verifiability).

**Definition 5** (Coercion Resistance according to [6]). Voters cannot prove whether or how they voted, even if they can interact with the adversary while voting.

The specification of Civiticas involves the following seven assumptions considering cryptographic assumptions as well as further restrictions on the threat model:

1) The adversary is not capable of simulating the voter throughout the entire registration process. If the adversary would be able to simulate a voter during the entire phase, there would be no difference between adversary and the voter as both entities would terminate with the same knowledge and both could use their knowledge to trigger a valid vote.

2) Each voter trusts at least one registration teller, and the channel from the voter to the voter’s trusted registration teller is untappable. Civitas provides a way of producing fake credentials in order to assure coercion resistance. Therefore, each voter has to trust one registration teller which does not output the generated credential share to the adversary.

3) Voters have to trust their machines. As otherwise the adversary could theoretically read all processing steps carried out on that machine (e.g. authentication, encryption, decryption, packet sending/receiving).

4) The channels on which voters cast their votes are anonymous.

5) The ballot boxes to which the voters submit their votes can not all be corrupted. We assume that each voter’s vote will be handled by at least one correct ballot box and that the channels to ballot boxes are anonymous (refer to the prior assumptions) as the adversary could otherwise verify if a voter abstains from the election.

6) There exists at least one honest tabulation teller. Within the set of tabulation tellers, there is at least one that is not corrupted. Given the opposite, the tabulation tellers could easily collaborate in order to decrypt credentials and votes, thereby violating coercion resistance.

7) The Decisional Diffie-Hellman and RSA assumptions hold, and SHA-256 implements a random oracle.

Based on these so called trust assumptions, Civitas claims to guarantee the above proposed security properties.

V. ANALYSIS OF CIVITAS TOWARDS ROBUSTNESS

In this section, we first describe previous robustness improvements of Civitas to ensure tallying availability and afterwards we show that even the improved version does not provide voting availability.

A. Previous Robustness Improvements of Civitas to ensure Tallying Availability

In the original Civitas scheme [6] no high availability is assured, although the authors mentioned that they had it in mind while designing Civitas. However, the authors of Civitas investigated the robustness of Civitas in [7]. They improved the Civitas tabulation phase by replacing the distributed decryption with a threshold decryption. In the original version of Civitas, all the tabulation tellers have to participate in the decryption process and behave honestly, since for the decryption all tabulation tellers had to act according to the protocol in order to carry out the distributed decryption. If one of the tabulation tellers would refuse to do so the whole decryption process would fail to proceed and the election has to be restarted.
In the improved version with threshold scheme, just $k$ out of $n$ tabulation tellers have to be honest in order to carry out the decryption successfully. Therefore the robustness in terms of tallying availability is guaranteed in the improved version.

B. Analysis Regarding Voting Availability

Prior improvements of Civitas [7] tackle only the tallying availability aspect of robustness. In this section, we show that Civitas in neither of the two versions ensures the voting availability aspect of robustness and is, therefore, not robust in terms of Definition 3.

An attacker can corrupt a registration teller $rt_i$ and force him to abstain from distributing its credential shares to voters. The output of the protocol run between the remaining agents and the voters will be empty as no voter is capable of computing its credential according to the registration phase of Civitas. As consequence, not even one voter succeeded to receive his credential and consequently submit his vote. Hence, no final tally can be computed.

$$Pr[tally^{\pi(rt_1,rt_2,\ldots,rt_n)} \neq \emptyset] = 0$$

Even considering the scenario, where the attacker allows normal operation apart from the fact, that $rt_i$ is not allowed to output $v_j$’s private credential, we end up with

$$Pr[\bigwedge_{\pi} term(v)] = 0$$

which means that not all the voters completed their voting process, in contradiction to Definition 3. Although the voter notices the misbehavior of the registration teller, he can not prove that to any authority without violating coercion resistance as stated in the original Civitas.

VI. EXTENSION OF CIVITAS TOWARDS ROBUSTNESS

From a restrictive point of view we could extend the second trust assumption of Civitas as posed in Section IV-B, stating that: Each voter trusts at least one registration teller, and the channel from the voter to the voter’s trusted registration tellers is untappable by:

- The voter trusts all registration tellers, and the channel from the voter to all registration tellers is untappable. However, this assumption is contrary to the concept of distributed trust and does not comply with our trust model in Section III-B. Assuming that all registration tellers can be trusted boils down to one central trusted registration authority.

Therefore, we propose to slightly modify the Civitas scheme itself. The modification is based on the idea to integrate threshold schemes not only in the tabulation phase but in the registration process as well. To do so, we first show how to modify the credential generation process and then how to determine trusted registration tellers.

A. Modification to the Credential Generation Process

Civitas currently requires that a voter obtains credential shares from all registration tellers, otherwise the voter can not compute the secret credential and thus can not cast a vote. Threshold schemes tend to be the appropriate tool to weaken the assumption that all involved registration tellers have to operate correctly. However, registration tellers operate independently of each other, thus the generation of a secret and the distribution of secret shares would imply a new assumption on Civitas, as threshold schemes assume a trusted dealer.

![Figure 2. Attack scenario in Civitas vs. extension of Civitas. The extension prevents the attack.](image)

Our idea is that the voter determines a subset of registration tellers he trusts and he is willing to cooperate with, e.g. trustworthy institutions such as universities. The protocol can be designed in two ways. The voter has either to decide on the set of trusted registration tellers before sending the request for his credential share or upon receiving credential shares. In case of determining the set before requesting the shares, he will only send requests to trusted registration tellers. In the latter case, the voter contacts an arbitrary subset of registration tellers and based on the response, the voter commits to a subset of trusted registration tellers. In both cases based on the determined subset (which we encode in a set of indexes), the registration phase proceeds as the original Civitas protocol. Under the assumption that trusted registration tellers operate properly towards the asking voters, we lower the risk of selectively preventing voters from getting their credentials (compare to Figure 2).

In order to apply our extension, we suggest to replace the second trust assumption of Civitas as posed in Section IV-B, stating that: Each voter trusts at least one registration teller, and the channel from the voter to the voter’s trusted registration teller is untappable by the following one:

- The voter trusts a set of registration tellers, such that the size of that set is greater than one half of the total number of registration tellers. In addition, the channel from the voter to one of the voter’s trusted registration tellers is untappable.

Independent of which option is chosen, with this proposal the attacker can not corrupt a registration teller in order
to prevent a voter from participating in the election. In case of determining them beforehand, all the participating registration tellers are honest toward their voter. In case of determining them based on the received valid shares, the attempt of the attacker fails since the corrupted registration tellers will not be counted as participating registration tellers for that voter. We will discuss the procedure of determining the subset of trusted registration tellers thoroughly in the following section.

B. Determining Trusted Registration Tellers

The voter’s selection of trusted registration tellers plays a central role in our proposal. In this subsection we show how they can be determined.

To carry out the subsequent distributed decryption and tallying phase, each voter’s trusted subset has to be known in order to generate their public credentials. The tabulation tellers need the public credential in order to validate a vote, and also to maintain verifiability, public credentials have to be available on the bulletin board. If the voter however would have to publish his set of trusted registration tellers himself on the bulletin board (i.e. committing to the set by signing its decision), coercion resistance would be violated. To overcome this drawback, we propose two possible adaptions of the voting phase.

If we assume, that a registrar is completely trusted by the voters, then each voter could submit the indexes of trusted registration tellers to the registrar who then publishes the list of trusted tellers on the bulletin board, the registration tellers post the public credential shares on the bulletin board, and there the public credential is generated. However, adding an assumption about the trustworthiness of the registrar would render the distributed registration unnecessary.

We consider a more convenient adaption thereby relying on reasonable assumptions. Based on the point in time the voter commits to his trusted subset of registration tellers the protocol proceeds as follows:

- If the voter commits to the trusted subset before requesting the credential shares: Initially, each voter selects a set of trusted registration tellers which will be contacted in the following manner. Within the communication protocol between each voter and a registration teller, the voter discloses the index set of trusted registration tellers to each trusted registration teller.

- If the voter commits to the subset after receiving the credential shares: The voter contacts an arbitrary subset of registration tellers to request the credential shares. After receiving the credential shares based on our initial protocol he checks the correctness of the shares. Depending on the validity check of the shares the voter commits to the trusted subset of registration tellers. Subsequently, the voter contacts the trusted registration tellers once more and discloses the index set of trusted registration tellers to them.

Registration tellers then publish the received index set together with the public credential share according to the Civitas specification. Finally, the bulletin board contains for each voter a set of public credential shares together with a set of index sets. The tabulation phase relies on the voters’ trusted index set. Therefore, we restate the second trust assumption, i.e. we assume that each voter trusts more than one half of the registration tellers and a valid credential is generated from more than one half of the registration tellers. Each tabulation teller determines the set of public credentials as follows: For each credential, the tabulation teller discovers the absolute majority of occurring index sets. Since the number of indexes in the index set has to be more than half, we prevent a successful collusion in order to produce a majority result. The resulting index set contains the trusted registration tellers for the corresponding voter. The public credential is generated by combining the shares corresponding to these indexes.

VII. ANALYSIS OF EXTENDED VERSION

In this section we analyze the improved version of Civitas regarding security and usability issues. Besides showing that the improved version of Civitas ensures our robustness definition, we will show that the vulnerability regarding coercion resistance as detected in [15] does not exist anymore in the first design option of our extension.

A. Security Analysis - Robustness

The extension of Civitas according to Section VI allows to state the following theorem.

**Theorem 1 (Robustness of extended Civitas).** The extended Civitas protocol \( \pi(V, RT, TT, S) \) where \( V, RT, TT, S \) denotes the corresponding roles \( R_1, R_2, \ldots, R_n \) of Civitas, satisfies Definition 3.

Civitas embodies four individual roles (supervisor, registration teller, voter and tabulation teller) which entraps to split up the robustness proof into three distinct phases based
on their interaction. However, a closer inspection of Civitas revealed that \( RT \) and \( V \) interfere with each other in a way that termination can not be achieved individually. Thus, the analysis is split up into two phases, the voting (including registration and voting phase) and the tabulation phase.

**Lemma 1** (Robustness of extended Civitas voting phase). Let \( \pi(V, RT, TT, S) \) be the Civitas protocol and let \( \mu \) be a negligible function. For all PPT adversaries \( A \) after the protocol run \( \pi (\pi, \pi, \pi, s, A) \) the following holds:

\[
Pr[\bigwedge_{\pi} \text{term}(v)] \geq 1 - \mu(n).
\]

**Lemma 2** (Robustness of extended Civitas tabulation phase). Let \( \pi(V, RT, TT, S) \) be the Civitas protocol and let \( \mu \) be a negligible function. For all PPT adversaries \( A \) after the protocol run \( \pi (\pi, \pi, \pi, s, A) \) the following holds:

\[
Pr[\text{tally}^{\pi(\pi, \pi, \pi, s, A)} \neq \emptyset] \geq 1 - \mu(n).
\]

**Sketch of the Proof:** The proof of the robustness property could be carried out by indistinguishability of the protocol runs in the presence/absence of the adversary. But a goal-oriented approach is rather preferred, which relates correctness of participants and termination guarantees. Therefore the proof of both lemmas concerning the robustness definition can be carried out by case analysis, taking the robustness critical milestones into account. Starting from the termination of the supervisor which is equivalent to the termination of the whole protocol one can move backwards in order to cancel out all possible non robust behavior of each step of the protocol. It can be trivially shown that if having the predefined assumption in mind, there can be no non robust behavior observed at any step.

**B. Security Analysis - Coercion Resistance**

Civitas might be extended in a number of ways to improve the overall robustness of the scheme. The proposed extension, where the voter has to commit to the trusted registration tellers before contacting them, also restates a crucial attack on coercion resistance, which was discovered by Küsters and Truderung in [15].

In their attack scenario, voter \( v_i \) asks the adversary to assure himself that \( v_i \) did not ask his credential at registration teller \( rt_j \). In case \( v_i \) did not ask the credential at \( rt_j \), the adversary is convinced that \( v_i \) did not vote. The coercion resistance is no more guaranteed.

This scenario allows a voter to convince the adversary that he abstains the election in case the adversary corrupts at least one registration teller. Due to our first proposed adaption, voters cooperate only with trusted registration tellers. Therefore, any conspiracy between such registration tellers and the adversary is ruled out as follows:

- If there exists a registration teller under the influence of an attacker and if the voter does not trust him, then the latter will not ask for his secret share from this corrupted registration teller, thus the attacker will not learn whether the voter will vote or not.
- The attacker cannot gather the necessary \( t \) secret shares needed to compute the secret credential, because he does not know in advance the set of registration tellers for a certain voter.

Thereby, the extended version of Civitas restates the attack identified in [15].

The second option of our proposal where the voter commits after contacting the registration tellers allows for a more flexible voter process, which makes the use of Civitas in real-life scenarios more possible. Hence, in such a scenario coercion resistance can no longer be assured, and is therefore no improvement regarding coercion resistance compared to the original Civitas specification.

**C. Usability Analysis**

Usability has rarely been taken into consideration lately in electronic voting schemes. It appears the more secure an electronic voting scheme gets the more its usability is hurt. There are different issues on this matter. People do not trust cryptographic primitives in real-life issues and these are also quite complex to use for an average user. The other issue is also that some attacks on real-life use of electronic voting are hard to be formalized in formal security models such as attempts to confuse the voters.

In case of Civitas one usability challenge is managing the credentials. The user has to generate the real credential and also the fake credential which might be not trivial for an average user, on the other hand the registration phase could be done once and the credential can be reused for several elections in order to decrease these complex efforts. But on the other hand in this case, the recovery of lost credential also becomes a usability challenge. For both cases, performing for each election an registration phase or performing it once for several election, the voter has to be able to distinguish real and fake credentials, which might be quite difficult for an average user.

The first design option where the voter commits to the trusted subset before requesting the credential shares, the usability is obviously weakened since the voter has to decide on the trusted registration tellers himself. The additional effort to the voters procedure for choosing the trusted registration tellers weakens the usability but increases coercion resistance of Civitas on the other side. The second design option in which the voter commits after receiving the credential shares, usability is not hurt. In this case, the voter commits to the subset of trusted registration tellers depending on the credential shares they have submitted. In the registration phase the subset of trusted registration tellers is determined as a part of the protocol and is performed automatically by the machine of the voter. Hence, relying on this approach, the usability is not affected.
VIII. CONCLUSION AND FUTURE WORK

Modern electronic voting schemes, such as Civitas, meet the requirements with respect to many well established security properties, i.e. coercion resistance, universal and voter verifiability. Nevertheless, the establishment and the success of electronic voting schemes may depend on other, less noted requirements. We consider as crucial requirement for each electronic voting scheme the robustness in terms of voting and tallying availability even in the presence of adversaries. Therefore, the paper provides informal and formal definitions of robustness. The term robustness is not new to the cryptographic community, but has only been applied to other cryptographic structures, such as encryption schemes or mix networks. To the best of our knowledge, we are the first to propose a definition considering the robustness of electronic voting schemes in terms of both voting and tallying availability.

In addition, we analyzed the robustness of the Civitas voting scheme and discovered that the system is vulnerable to adversarial attacks. We identified a weakness of the scheme, in the registration phase. We proposed an extension of Civitas which prevents the identified attacks on robustness. The extension consists of remodeling the registration phase in order to guarantee robustness with respect to our definition. We also showed that the extended version of Civitas complies with our definition. In addition the first proposed design option solves the problem with coercion resistance stated in [15] but decreases the usability of the protocol. On the other hand, the second design option of our proposal preserves the usability of the protocol, but does not tackle the coercion resistance issue.

Within this work, we only provide a sketch of the proof for the modified version of Civitas to be robust. Due to the importance of Civitas, we consider a strict mathematical robustness proof as a further work of substantial relevance. Based on a formal specification of Civitas, one can formalize robustness and prove it.

Furthermore, for future work we also intend to apply a dealer free secret sharing approach to solve the robustness issue in the registration phase. In addition, we also propose to adapt the registration phase of Civitas by incorporating trust models [17] in order to allow voters to evaluate the trustworthiness of registration tellers according to direct and indirect evidence. Thereby, the overall trustworthiness and the robustness of Civitas can be increased significantly.

REFERENCES


