UNIVERSITY OF TARTU

INCOERCIBILITY IN E-VOTING

A SURVEY ON AVAILABLE TECHNIQUES

Research Seminar in Cryptography

Supervised by Prof. Helger Lipmaa

SERGIO A. FIGUEROA

Contents

Chapter 1

Introduction to Incoercibility

Voting is intended to be a simple and powerful tool to aggregate the opinion of the individuals of a society in the form of a consensual decision. Be it the election of a president, the decision about a controversial law or the decision about a new marketing campaign (just to pose some examples), it is a tool intended to be as powerful as simple.

The creation of a system that properly captures these opinions and produces a reliable output is by itself a complex engineering problem. However, the problem of voting acquires an even bigger dimension when it is considered in an adversarial environment. Voters, voting authorities, stakeholders and even external adversaries have frequently strong motivations to affect the result of a voting process, posing a strong threat for the process and everyone involved in it.

In an adversarial scenario of that nature, the security requirements are often hard to define, and in some cases contradictory. One common trade-off is the one between ballot secrecy and the verifiability of the tally. Some of the most popular and critical voting examples require that the specific vote of a person is kept secret from his peers. On the other hand, voters are usually interested in making sure that their vote was counted and the tally accurately reflects the content of all the votes casted (and *only* the votes casted).

This document will describe in detail these difficulties and trade offs in relation with one major requirement: incoercibility.

1.1 What is coercion and why it matters

There is a common saying among information security practitioners that states that human beings are the weakest link in the security chain. In the scenario of an election, that assertion becomes particularly critical. The most noticeable aspect is the fact that, during the process of casting a vote, the voter is an input system: he is introducing his opinion into the system.

It always makes sense to target the weakest link in the chain. It will break the chain with the least amount of effort and will cause the same effect: the chain will be broken.

The attempt of an adversary to influence the input of a voter into the voting system is called *coercion*. In turn, the property of a system that protects a voter from being coerced is called *incoercibility*.

1.2 Classification of coercion

Intuitively, coercion could take different shapes, posing a different threat level on the voter. A subtle implication (a boss suggesting that the voter is less eligible for a promotion if the latter votes according to his own mind), a persuasive gift (a politician offering a lunch to everyone who votes for him), an emotional blackmail (a patriarchal figure who dictates the way the whole family should vote) or a physical threat (an armed group threatening the lives of those who vote against their candidate) could be just some of the examples of how wide an diverse coercion can be.

From the point of view of a voting system, protecting the mind of the voter from the psychological effects of coercion is definitely out of scope. Coercion is also a social problem and should be also addressed with non-technical controls. However, it is essential to ensure that the system does not *enable* coercion. If the voter can **lie convincingly** about the way he voted, **without the system proving him wrong**, the voting system is *incoercible*.

The next sections will describe the different levels of *power* that an adversary can achieve, interpreting the classification suggested by [2].

In that text were proposed three categories of coercion: *Input/Output (I/O) coercion*, *receipt coercion* and *active coercion*

The most basic case: I/O coercion

In this scenario, the coercer sees the voting process as a black box. He gives the voter an input to the protocol and can see the output produced. Any additional interaction remains secret to him. In other words, the coercer tells the voter which strategy to follow, but the first only has access to the output of the process (i.e. the tally).

Example of success

In some instances of legislative power (e.g. senate, congress, etc.), the votes about certain law projects are published at the end of the session. The reasoning behind this is to assure their electors that their representative is proceeding according to his campaign promises. This is (possibly by design) an I/O coercible system: the representative cannot vote according to his own mind, but is expected to act as a proxy of the opinion of his electors.

Example of failure

Even a default process with paper ballots can protect against this kind of coercion. The voter is told to cast a vote in one direction, but his vote is private, and the output of the process is just the final tally, with no way to calculate the original way the specific voter vote.

Introducing verification of success: receipt coercion

The system allows the voter to create a *receipt* that allows him to prove the way he voted. That is: a trace that allows the coercer to verify how the voter voted.

Trivial example of success

In an unprotected Internet voting scenario, the voter can store a traffic capture of his interaction with the voter server. If the message is sent as plaintext, the traffic capture works trivially as a receipt.

Another easy example of success

In an Internet voting scenario where the communication is protected via HTTPS, it could be possible for the voter to take a screenshot of his accepted vote as an easy receipt of the process.

A bad attempt to correct

Bad cryptography can arise as an option at this point. For instance, the voting authority could decide not to send the option selected back to the voter. Instead, to provide a way for the voter to verify its choice, they decide to send the hash `h(id_voter||id_option_selected)`, signed to ensure authenticity. Even if the identification of the voter is not known by the coercer, or a salt is added to the process, **it would be easy for the coercer to verify the value of the voting option in polynomial time.**

Active coercion

In *active coercion*, the coercer has total access and control to the messages exchanged. This means that he can determine the input to the protocol and verify that the voter actually send it.

1.3 Scope of the document

This document aims to describe and compare different approaches that have been proposed to achieve incoercibility both theoretically and practically. Since the definitions of some of the core concepts vary among the authors, it will also attempt to convey the common aspects of these concepts and the implications of the different interpretations, if they are relevant.

Chapter 2

Approaches to incoercibility

One of the main challenges in the process of achieving any security feature is to precisely define what it means. It is very easy to say "the system should be secure", "this server should be highly available" or "this mobile application cannot be hacked". However, every person will have a very different opinion of what it means and, furthermore, it will be impossible to implement measures that fulfill said requests.

This chapter explores different approaches to the problem of incoercibility: how to define it? How to measure it? How to achieve acceptable levels?

2.1 Theoretical limit

It is easy to think of this problem in an ideal scenario where not even the total computational power of the universe would be able to get any clue about the individual way in which a voter cast his ballot under any circumstance. However, even if the protocol was a perfectly secure black box, the tally is supposed to be a function of the votes, and consequently it might leak information about the votes. This section aims to discuss some scenarios as an illustrative example of this leakage, yet it is not intended to be exhaustive.

Only one voter

If only one person votes, it is not possible to hide his vote without not taking it into account for the election. This might sound like an unlikely scenario, but it is not impossible. For instance, some distributed voting processes, such as a countrywide presidential election, calculate local tallies for each precinct before adding them up altogether. In countries with high abstinence levels (i.e. where very few of the possible voters do vote), having few voters at a single precinct is far from seldom.

Unanimity

If all the voters vote in the same way, it is not possible for any of the voters to claim that they have voted otherwise. In small groups of voters, it is possible to reach unanimity with a non-negligible probability.

Knowledge of some of the votes

A generalization from the unanimity case can be reached if some of the voters make their votes public (or, at least, known to the attacker). Let us suppose there are 10 voters, out of which 5 are very predictable and will choose Alice as a candidate. If the final tally is $50\% - 50\%$, it will not be possible for any of the other 5 voters to claim they have voted for Alice.

In this case, the identification of the vote depends on the certainty with which we know the votes of the first 5 voters. For instance, if one of them lies, the probability distribution changes. However, it gives an idea of how the distribution of votes among the voters is not as independent as it can be thought in the first place.

2.2 Formalizing a voting protocol

After understanding what cannot be achieved in an ideally incoercible system, it becomes possible to approach to realistic definitions of an incoercible system.

In previous sections, the incoercible system has been described as a system that allows the voter to *lie convincingly* about the way he has voted. The first step towards a definition of incoercibility that can be measured and achieved is to formalize what does to "lie convincingly" means in terms of a voting protocol.

The variables involved in a voting protocol can be represented in terms of the actors participating and the information they generate and exchange. A group of people allowed to provide an input to the voting process are potential voters, and the set of their unique IDs is defined as L. An ordered subset $V \subseteq L$, represents the set of voters who participated in the election, and an ordered set v of the same length as V contains the votes that were cast.

For each voter V_i, there exists an entry $v_i \in v$ that represents his vote. The goal of the election is to calculate a tally function $f(v)$.

These variables are enough to produce a simple generic but functional voting process, yet do not provide enough tools to analyze all the information that can be learned by an adversary.

The protocol π executed between each voter and the voting authority is expected to be public. The set of all the messages exchanged by the parts are stored and considered public, and this set is commonly known as a *bulletin board* B[1]. The specific part of the bulletin containing information about V_i is referred as B_i. The bulletin board is assumed to be public.

Once the basic components of a voting system have been specified, the definition of incoercibility can be defined in more specific terms: the system should not allow an adversary to calculate any specific pair (V_i, v_i) from the public information of the system: the tally $f(v)$ and the bulletin board B.

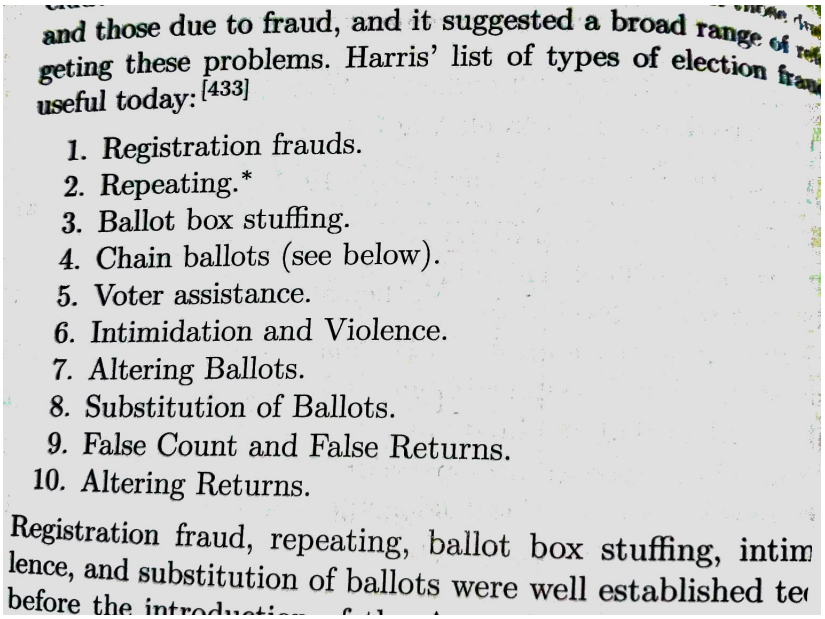
[1]The description of a bulletin board is often vague and it is hard to grasp its structure, which cannot be described in clear with terms as "list" or "count". Since a protocol is usually complex, verbose and may have a complicated flow, describing the structure that stores one specific instance of it is going to add complexity to a definition that would be otherwise simple. As a way to try to grasp the concept more clearly, think of a traffic capture at a central node of the network or a very detailed server log. The purpose of this definition is that, since the voting process could be held over the Internet, the channel should not be assumed to be secure, and therefore every message of the protocol can be assumed to be public.

2.3 Why is it a challenge?

This semi-formal definition is incomplete and isolated: it only considers external adversaries, ignores the effect of other security requirements and does not take into account the viability of the system (i.e. how viable is it to implement a solution that upholds the definition). The purpose of this section is to strengthen the definition by pointing out the obstacles posed by these aspects.

Verifiability

There is no purpose in the secure capture and storage of votes if the tally can be calculated without using **all** the votes captured and **only** the votes captured. In paper ballot voting processes, it is extremely hard to offer a reasonable guarantee for this. Stuffed ballots (ballots added surreptitiously to the ballot box), tampered ballots and deliberate miscount are just some examples. In electronic voting, however, it is possible to introduce technical ways to strengthen the reliability of the output. The property of a voting system that refers to the possibility of proving the correctness of the calculation of the tally is called *verifiability*.



and those due to fraud, and it suggested a broad range of
geting these problems. Harris' list of types of election fra
useful today: [433]

1. Registration frauds.
2. Repeating.*
3. Ballot box stuffing.
4. Chain ballots (see below).
5. Voter assistance.
6. Intimidation and Violence.
7. Altering Ballots.
8. Substitution of Ballots.
9. False Count and False Returns.
10. Altering Returns.

Registration fraud, repeating, ballot box stuffing, intim
lence, and substitution of ballots were well established te
before the introduction

Figure 1: *Common types of frauds with paper ballots*[2]

Intuitively, there is a clash between the expectations of incoercibility and verifiability. The voter wants to make sure that his ballot is stored as he cast it and counted as it was stored, but the association between his vote and his identity must remain securely isolated.

That intuition is not wrong: the main purpose of [1] is to prove that it is not possible to build a system that provides simultaneously verifiability and *receipt freeness* (an instance of incoercibility that will be covered in section 2.4).

[2] Available at [4]

Colluding adversaries and universal composability

There are several parties interested in a voting process. The boundaries between them are often diffuse, they all have different ideas about what warranties should the process offer and different powerful interests in attacking the process. For instance, if the voting authority colludes with a coercer, they could give the latter more coercive power in most protocols (for instance, by giving him access to the sets V and v).

If we change the word *voter* for *user* or *part*, that is the description of a *multi-party protocol*. The concept of securing e-voting and *secure multi-party computation* grow together and the literature about them frequently converges.

In particular, there is one concept that "captures most security guarantees that one would expect from a multi-party protocol" ([2]). The idea that a protocol can be secure even when used as part of a bigger insecure protocol and the environment is adversarial is known as *universal composability*.

One of the features that are not covered my universal composability is incoercibility. Furthermore, one of the promises of an universal composable protocol, universal verifiability, will be an obstacle for satisfying strong definitions of incoercibility.

2.4 What should an incoercible protocol achieve?

The previous sections have avoided to define formally what an incoercible protocol achieves, focusing on the limitations and obstacles that make this definition a non-trivial task. In fact, both the definition and terminology associated to incoercibility varies through the available literature. This section will present two broadly accepted and powerful definitions of incoercibility, and discuss if and how can they be achieved.

Receipt freeness

One of the most popular definitions of incoercibility is *receipt freeness*. Expanding on the idea that a voting system should not leak the way in which a voter voted, a stronger notion suggests that the voter should not be able to generate a *receipt*, that is, a verifiable proof of the value of v_i.

Following the definition in [1], where receipt freeness is the only kind of incoercibility considered, a receipt could be defined as a *witness* w or a *trapdoor*: a value that makes easy a problem that would be hard otherwise.

In a voting process with receipt, given a bulletin board B, a voter list V and a vote list v, there is a function $R(B, V_i, v_i, w)$ that outputs 1 if the pair (V_i, v_i) can be derived from B. R should be hard (computationally infeasible, likely based on an NP-hard problem) to compute without a valid *witness* w. "A voting scheme achieves the receipt-freeness property if there is no such a relation R, or the witness w is hard to compute.", claim the authors.

Receipt freeness is, however, not enough to protect against active coercion.

Building active-coercion resistance

Active coercion is invasive. The coercer has full control of the messages that are exchanged in the protocol. The extent of that control is neutralized by [2] with the introduction of a *trusted* hardware token that allows the voter to communicate with the voting authority trespassing the total control of the coercer over the channel: the coercer may have full control of the protocol, but building layers out if its reach could help to achieve protection against his attacks. Nevertheless, "building layers out of its reach" is not straightforward.

Viability and assumptions

Regarding the possibility of achieving these definitions, the works in [2] and [1] show that it is possible to reach very different conclusions (even contradictory) in similar scenarios where the only difference is the set of security assumptions used as a starting point.

In particular, [1] shows that in order to achieve receipt freeness and universal verifiability (that is, verifiability of the correctness of the entire tally) it is required to have all voters casting their votes and or private channels between the voters and the voting authority. Both assumptions are ruled out as unrealistic by the paper.

The introduction of trusted hardware by [2] provides additional cryptographic possibilities, but it is unclear how could it be implemented, which capabilities it should have, where does the control of the coercer end, how can it protect against a collusion involving the voting authority and how can it be implemented at a larger scale.

In other words, the lack of sensible assumptions is currently limiting what can be done to provide incoercibility through cryptographic mechanisms.

2.5 Re-voting: introducing process security

While cryptography is often regarded as the strongest information security control possible, non-cryptographic controls are frequently useful or fundamental to achieve security objectives.

The Estonian Internet voting system[3] is an example of this complementary approach. While supported on a set of different cryptographic protocols to assure other security requirements, such as ballot secrecy or verifiability, the main control of the system against coercion is procedural: in Estonia, if the voter has been coerced or just changed his mind, **he can vote again, overriding the original vote**. Furthermore, the voter can approach to a physical precinct and vote by paper ballot. If he does so, **the physical vote is the only one that counts**.

Re-voting is only available during the *advance voting* period, which spans for a week and ends three days before the *election day*. During the election day, voting is only possible via physical ballot (either at a precinct or at home), available exclusively in the case of not having voted at all during the advance voting period.

[3] Often referred to as i-voting.

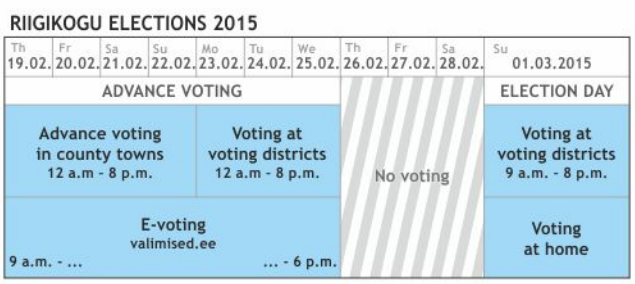


Figure 2: *The timetable for the Riigikogu (Estonian parliament) election 2015*[4]

The rationale

If the coercion occurs during a very short timespan, re-voting can be an effective and simple tool. The voter will be able to vote according to his mind later, once he is no longer under coercion.

Considering the physical approach to coercion, re-voting is a very effective measure: if the coercer needs to be physically near to the voter during the entire time of the election, **he will have to dedicate a huge amount of effort to effectively coerce a small amount of voters**.

On the other hand, if the coercer has access to the channel between the voter and the server, he might have the ability to detect subsequent attempts of re-voting. This can be prevented by voting physically at some point of the process.

Deniable re-voting

Even though additional layers could generate traces that would make evident a re-vote, it is still possible to prevent the voting system from being a source of leaked information. In particular, the research of [5] shows that it is possible to leverage homomorphic encryption algorithms (tipically used to calculate the tally without decrypting the votes) to prevent the coercer from finding out the event of a re-vote, even under active coercion.

Security implications

Under re-voting, coercion is still possible. **However, it requires a large amount of effort in two fronts to be effective** (i.e. to produce a noticeable impact in the outcome of the election). In one hand, the coercer would need to compromise the network in order to intercept the communication between the voters and the server. And even in that scenario, the interception would likely produce only metadata and encrypted data. On the other hand, the coercer would need to compromise also the physical environment (e.g. by bribing an election officer), in order to be notified when a specific voter attempts to vote.

Note also that allowing re-voting adds new requirements for voter authentication. If an attacker tries to impersonate a voter in a scenario where only one vote can be cast, the attack is likely to be detected, even if the authentication protocol is faulty. If re-voting is possible, the authentication protocol needs to be capable of rejecting any impersonation at any point. There is no point to incoercibility if identity theft is possible and effectively exploitable.

[4] Available at [3]

In general, the channel redundancy scenario (i-voting plus paper ballot) adds complexity to the activities of the coercer. However, process security can open alternative questions that otherwise would remain closed. For instance, could the coercer disenfranchise voters in order to change the outcome of the election? These could be some scenarios for additional thought: the coercer forces the voter to vote online and then retains the identification card of the voter to prevent his re-voting. A coercer could attempt to reduce the availability of the channel between the voter and the server (would a malicious ISP be in advantage for this case?). One answer leads to a thousand new questions.

There is one scenario that seems to be favorable for coercers in the Estonian implementation of re-voting: *last minute coercion*. Since re-voting is an unlimited feature, it does not matter how a voter cast his vote beforehand, or if he did it at all: coercing effectively the voter during the last minutes of the advance voting period is enough to succeed in the coercion strategy. One possible approach could be to say that, if the voter knows he is under the risk of being coerced, he can vote via paper ballot, but that totally defeats the purpose of e-voting. Why bother about the design of online voting if the claim is that paper ballot is safer? Would it be reasonable to assume that the voter knows the risk beforehand and can act accordingly? Furthermore, does this decision increase the risk of foreign interference? If all it takes to rig the elections is to coerce the right amount of voters for just five minutes, the challenge does not seem impossible.

Chapter 3

Final remarks

Procedural security controls, and particularly re-voting, are a pragmatic solution, but one that makes coercion annoying instead of unfeasible. Even when it has worked seamlessly for the Estonian process for several years, it might not be a solution for every conceivable democratic process.

One solution could be to secure the process: to model the way in which the different actors could attack the voting process and implement controls to mitigate or transfer the risk. After several iterations, this approach could produce a solid process, but most likely not one that could be reproducible. Likewise, network, host and software security considerations could harden one specific instance of the problem, but leave the general problem unresolved.

The most universal approach would be cryptography, but the current schemes are insufficient against realistic coercion scenarios within an universally composable protocol. If researchers can find a way to relax the assumptions, the definitions or, otherwise, offer a viable scheme, incoercibility (at least with respect to some kinds of coercion) would become provable.

Appendix A

Notation definitions

Symbol	Meaning
B	Bulletin board
B_i	Part of the bulletin board corresponding to V_i
L	Set of eligible voters
V	Ordered set of voters that voted. $V \subseteq L$
V_i	Unique id of the i-th voter
v	Ordered set of votes
v_i	Vote of V_i
w	Witness value (trapdoor)

Bibliography

[1] Benoît Chevallier-Mames, Pierre-Alain Fouque, David Pointcheval, Julien Stern, Jacques Traor é. *On Some Incompatible Properties of Voting Schemes*. D. Chaum, R. Rivest, M. Jakobsson, B. Schoenmakers, P. Ryan, and J. Benaloh. Towards Trustworthy Elections, 6000, springer, pp.191-199, 2010, lncs. <inria-00539539>

[2] Joël Alwen, Rafail Ostrovsky, Hong-Sheng Zhou, and Vassilis Zikas. *Incoercible Multi-Party Computation and Universally Composable Receipt-Free Voting*. Advances in Cryptology - CRYPTO 2015, pp 763-780, 2015.

[3] *Electoral Management - Estonian National Electoral Committee*. Available at `http://www.vvk.ee/?lang=en`. Retrieved on October 2015.

[4] Jones, D., and Simons, B. *Broken Ballots: Will Your Vote Count?* Stanford, CA: CSLI Publications, 2012

[5] Achenbach, D. et al. *Improved Coercion-Resistant Electronic Elections through Deniable Re-Voting.* USENIX Journal of Election Technology and Systems (JETS) Volume 3, Number 2. August 2015