Minimal Models for Receipt-Free Voting

Juraj Skripsky

March 2002

Abstract

Most of today's voting protocols are vulnerable to coercion and vote-buying because they allow the voter to carry away a "receipt" which proves what vote he has cast. We take a look at what it means for a protocol to be receipt-free. Unfortunately, no uniform definition for this concept exists today. Even worse, definitions used in literature are often far too simple to capture and treat all relevant aspects adequately. Important in this respect is the precise relationship of voter, adversaries and authorities as well as the inherent differences between coercion and vote-buying. We argue why we consider vote-buying to be the more dangerous attack.

We show why existing protocols offering receipt-freeness are not useful for real-life elections. A short overview of the most prominent protocols is given.

Finally two practical models where receipt-freeness is possible proposed. We sketch ideas of how protocols for these models could be constructed.

Contents

1 Introduction

With the Internet soon reaching every single home, digital voting - or "e-voting" - is becoming an attractive means for conducting elections. The voter will be able to cast his vote quickly and comfortably from his PC (or in fact any computer connected to the internet). The collection and tallying of all cast votes will be a fully automated process. Already we see early implementations of such voting systems appearing, as is the case in Geneva.

Still, traditional systems with voting offices and ballots sent by mail won't disappear in the near future. E-Voting will only be an alternative way of casting a vote. But in the long run e-voting systems are almost certain to replace today's paper-based systems.

1.1 Security Requirements

Just as with the traditional systems, we expect their digital counterparts to offer a reasonable degree of security. The three key requirements we expect any fair election to meet are correctness, privacy and availability. The first two requirements are of particular interest for this project. They can be broken down into the following list of desirable properties (taken from [Hir01]):

- Eligibility. Only entitled voters are able to submit a vote.
- No Double Voting. Each voter can cast only one single vote.
- Validity. Only valid votes are counted.
- Local Verifiability. Each voter can verify whether his vote is included in the published tally.
- Global Verifiability. Anyone can verify that all valid votes have been counted, and that the published tally is correct.
- Secrecy. It is infeasible to find out which voter has submitted which vote.
- Anonymity. It is infeasible to find out whether or not a particular voter has participated in the voting.

1.2 Coercion and Vote-Buying

Two attacks left unconsidered in classical voting protocols are coercion and vote-buying. A coercer forces someone to cast a particular vote by threatening him. A vote-buyer pays money to anyone casting the desired vote.

Unfortunately coercion and especially vote-buying can be much more dangerous attacks in a digital election than a traditional one. In both attacks the adversary wants to be sure that the voter did what he was told to. If the voter can somehow generate a digital proof, which shows he actually cast the "right" vote, verification of this fact is simple. Then the attack can be vastly automated, in a way not possible in a traditional election. The voter downloads a program, which is able to generate this proof. He casts his vote. He runs the program and transfers the resulting proof by email to the adversary, whose computer automatically verifies its validity. If indeed the voter "obeyed" he's on the safe side or - in the case of vote-buying - will have the promised money transfered to his bank account.

1.3 Receipt-Freeness

There are various ways how these attacks can be dealt with. One elegant and (at least theoretically) simple approach is the concept of receipt-freeness as introduced by Benaloh and Tuinstra [BT94]. Coercion and vote-buying owe much of their power to the generation of a vote's proof - or "receipt" for short - and its automated verification. To solve or greatly alleviate this problem we make it impossible for the voter to construct such a receipt. If no receipt can be constructed or equivalently the voter can always construct a fake receipt (which strictly speaking is no receipt at all), the attacks lose their attractiveness.

1.4 Standard Model

A wealth of voting protocols exists today. They follow different approaches on how the votes are being encrypted, posted and tallied. But most of them share the set of assumptions made about the communication infrastructure. This standard model is the minimal model still sufficient for most protocols. It consists of the following assumptions:

- all channels used are insecure (but synchronous)
- adversaries see all channels all the time
- a PKI has already been set up
- we have multiple authorities
- we have a bulletin board

The bulletin board can be thought of as an authenticated public channel with memory. It can be jointly simulated by the authorities. Voters can post their ballots to the bulletin board. Once a ballot has been posted, it can't be changed or deleted from the board.

2 Definition

To be able to properly design and analyse protocols and prove their receipt-freeness it is crucial to have a precise definition. It should cover all aspects related to the following questions:

- What is the adversary able and allowed to do?
- What is the voter allowed to do?
- What is considered to be a receipt and what is not?

Unfortunately, no broadly accepted definition exists so far. Various flavors of receipt-freeness are implicitly considered in the literature. Sometimes even rather unrealistic definitions are used (e.g. some implicitly include the assumption of "honest" voters deviating from the protocol).

2.1 Degrees of Freedom

Let's first take a look at the relationships of the three main entities in our scenario: voters, authorities and the adversary.

It's essential to have a clear understanding of the extent and timing of all communication between the adversary and the voter. Do we allow communication before, during or only after the voting procedure?

Another interesting question is whether some of the authorities might be corrupted. How many might be corrupted and how far does the cooperation with the adversary go? Or is it even possible that the adversary has complete control over an authority (meaning they're conceptually one "player")? Does the voter know how many or which authorities are or might be corrupted?

We want to keep the definition general enough to allow for construction of fake receipts that will be accepted by an adversary with a probability slightly smaller than one. This might help us substantially in designing a useful protocol. Again we must decide on what we mean by "slightly smaller than one".

2.2 Vote-Buying vs. Coercion

Another difficulty for finding a useful definition stems from the fact that there are subtle differences in what needs to be done to prevent vote-buying as opposed to coercion.

In vote-buying we have the voter "conspiring" with the adversary. They don't necessarily trust each other. But they will try their best to receive the offered money and have the favorite vote cast. As a consequence the voter is even willing to deviate from the protocol in order to construct a receipt. So in the end, we're looking at two separate cooperating adversaries.

Coercion is much simpler to analyse in this respect. In this scenario, the voter is just a poor guy following the protocol, casting the requested vote. Here the focus can rest exclusively on the coercer.

While it is often stated in literature that coercion is the "stronger notion", I tend to disagree. Neither notion is stronger than the other. The following two very rough sketches of voting protocols illustrate this. The first one is "immune" to coercion, but not to vote-buying. In the second protocol it's the other way round.

Protocol 1: The voter encrypts his vote using the authorities' public key and a locally generated random number. Next he erases all information about the random number and posts the vote to the bulletin board. Without the random number the voter has no way of opening his vote. A vote-buyer could of course equip the voter with a modified version of the voting software. So before being erased, the random number would be sent to the vote-buyer for vote-verification.

Protocol 2: The voter sends his vote unencrypted to one of the authorities (there's no bulletin board). Assume the adversary can only tap the channel to one authority at the time (and we don't know which one it is). If the vote-buyer intercepts a ballot then he knows who voted and for whom. Otherwise he knows nothing about this particular ballot. Assume we use ten authorities. Then from the perspective of the vote-buyer, the money is probably better spent in an advertising campaign. If the same protocol is used with the presence of a coercer, things look darker for the voters. With all authorities receiving roughly the same number of ballots, one out of ten cheating voters will be caught. Most of them won't be willing to take their chances and will vote as requested.

Looking at the first protocol, one could argue that the voter could deviate from the protocol here as well, making it susceptible for coercion. The coercer might after all command him to do so. But then we would have to choose the voter's allowed actions arbitrarily. Otherwise he might just as well be allowed to give away all his secrets to the coercer and let him cast the vote, turning coercion into a useless abstraction (because it would be impossible to prevent).

2.3 Focus on Vote-Buying

When comparing coercion and vote-buying in the context of a real-life election, I clearly consider the later to be of higher importance. This opinion is based on two closely related observations.

In a large-scale digital election with a voting protocol vulnerable to vote-buying, such "attacks" will happen for sure. Receipt-generation and verification can be performed by software making it an extremely high scalable attack. Risks are moderate, as data transfers on the Internet can be made essentially anonymous and hard to trace. People possessing the necessary amount of money will try to influence the outcome of elections.

Coercion on the other hand scales badly. To convince him that the threat is real, the coercer has to know his victim. But once he has his attention, the attack can be very effective. With a gun pointed to your head, there is no doubt how you vote. The problem with coercion is that it takes place mainly outside the controllable voting system. This means it can never be completely prevented and reasonably good protection will always be hard to achieve.

2.4 Receipt-Freeness in the Standard Model

Most voting protocols have been designed for the standard model. Unfortunately it is always possible to construct a receipt in this model. The standard model inherently precludes receipt-freeness, as will be shown.

With a voter claiming to have cast a particular vote, it will be possible for the adversary to find out whether he is lying or not.

Proof (sketch): The adversary has complete knowledge about the transcript of all the messages that were sent back and forth between the voter and all authorities. Moreover he knows all secrets of the voter (an argument why this is a reasonable assumption follows). It's easy to see that the in- and outgoing communication of the voter's computer must determine the vote. It might be very difficult to calculate the vote, but it is nevertheless contained in this transcript. Having all his secrets and the claimed vote, the adversary can now simulate the voter's computer. All ingoing communication is taken from the recorded transcript. All resulting outgoing communication of the simulation must match the transcript if indeed the claimed vote was cast.

Why is it plausible to assume that the adversary has access to the voter's secret keys? A coercer simply forces him to reveal all his secrets. In the case of vote-buying the situation is slightly more complicated. It can be shown that the vote-buyer actually doesn't have to know the secret keys to verify the claim. The simulation mentioned in the proof above can be transformed into a related predicate. This predicate is satisfied if the claim is true and we are entering a matching secret key as parameter. Using a zero-knowledge proof this predicate can be verified without the voter giving away any of his secrets. So even the notion of a "valuable secret key" mentioned in the literature doesn't bring any advantage.

3 Existing Protocols

One might be tempted to think that it is possible to come up with receipt-free versions of existing protocols by applying a few simple modifications. Using the right encryption and replacing critical proofs by their zero-knowledge siblings should be all it takes. But this is not the case. As we have seen before, the use of encryption and zero-knowledge proofs doesn't help in the standard model. We need additional assumptions about the communication infrastructure. But generally these assumptions will make it difficult to keep properties in the areas of availability and correctness of the original protocol.

Let's take a look at existing protocols, which have been explicitly designed to offer receipt-freeness. They all suffer from a common problem. To overcome the limitations of the standard model, additional, rather unrealistic assumptions have to be made and relied upon in the protocol. Unfortunately, most papers introducing them omit the explicit stating of important assumptions. This makes it difficult to properly analyse and compare the protocols.

3.1 Voting Booths and Untappable Channels

The two most prominent assumptions used are the presence of a virtual voting booth and the availability of untappable channels. The use and behaviour of the virtual voting booth follows its real-life counterpart. During the protocol the voter enters and leaves the voting booth at carefully chosen points in time. While he's inside the booth his capabilities are restricted in the following way. He can't write to any channels, neither public nor private. But he can still read from all public channels and from a private channel provided by the booth. Virtual voting booths cannot be constructed using cryptographic tools alone.

Untappable channels primarily offer deniability of the transfered data. No matter what was sent over such a channel, the sender as well as the receiver can later separately lie about the data without being caught. Here too, the offered security cannot be achieved by cryptographic means, it has to be physical. Untappable channels can be one-way or two-way and don't necessarily offer authenticity. Note that a direction of a one-way untappable channel can be turned around effectively creating a two-way untappable channel. By utilizing a one-time-pad sent in the channel's untappable direction, subsequent messages going the other way can be unconditionally and deniably encrypted. As long as the sender of the original channel can lie about sent data, he can lie about the decryption of received data as well. Of course the parties still have to 'coordinate their stories' if they both want to lie about the transfered information.

It is interesting to compare the two constructs. The fundamental characteristic they share is that they allow the voter to lie about the received data. So what are the differences and can one be transformed into another? Is one of the notions stronger than the other? Intuitively even one-way untappable channels would seems to be "superior" to voting booths in terms of what can be accomplished by using them. But the single-authority protocol in [BT94] wouldn't work in its present form with a one-way untappable channel because the implicit timing restrictions imposed by the entering and leaving of the booth can't be enforced with "bare" channels. Obviously untappable channels are more powerful in certain situations by letting both parties freely use other channels. This means there is no general reduction from one construct to the other. No notion is stronger.

3.2 The Protocols

To satisfy the receipt-freeness property, it's always necessary to "decouple" the voter from his vote or ballot at some point. He mustn't have enough information to prove what vote is hidden inside the final ballot. Different approaches to this "decoupling" are being followed in the presented protocols. As the protocols owe their receipt-freeness primarily to this "decoupling", we classified them according to the concepts used for this purpose.

Knowledge of homomorphic encryption and existing voting schemes is assumed (for a description, see [Hir01]). With the exception of the last one, all protocols use homomorphic functions for the underlying encryption of the votes.

3.2.1 Ballots Generated by Authorities: [BT94]

Two protocols are presented in this paper. They're the first protocols to offer receipt-freeness. Both assume the presence of a voting booth. The first protocol uses one single authority. Ballots containing ordered pairs of 0s and 1s are generated by the authority. The authority proves publicly that the ballots have been generated correctly. While in the voting booth, the voter receives a proof showing for every ballot which vote is which. All he has to do now is announce which part of the first ballot he wants to cast. Note that the ordering of steps 4 and 5 of the protocol cannot be swapped. Otherwise it would be possible for the voter to "commit" himself to the information given to him privately. This commitment could be announced before the publishing of the following random bits, giving him a means of constructing a convincing receipt.

The second protocol uses multiple authorities. It has been broken (it is shown in [HS00] how a receipt can be constructed). Here, the voter generates an initial ballot and uses the encryptions supplied by the authorities as "blinding factors" for the final ballot.

3.2.2 Ballot Shuffling: [SK95], [HS00], first Protocol of [Hir01]

These protocols all assume the availability of untappable channels from the authorities to every voter. They share the following pattern. The authorities construct the initial ballots. There is a ballot for every possible vote. Over the untappable channel, the voter receives the information about the ordering of the encrypted votes. Then the ballots are shuffled by the first authority. The ordering is changed and every ballot is reencrypted (using a new randomness). A proof is given, showing that the reordering as well as the reencryption has been done correctly. Subsequently, the next authority takes the new ballots and shuffles them again. After several such steps the voter picks the ballot containing the vote he wants to cast. As he doesn't know the resulting randomness of the encryption, he can't open the chosen ballot.

3.2.3 Randomizer: [BPP$^+$01], second Protocol of [Hir01]

We assume that there is a special entity we call "randomizer". This can be a designated authority or a tamper-resistant device (e.g. smartcard). Additionally we have an untappable channel leading from this entity to every voter.

The voter constructs an initial ballot containing his vote. He knows everything about his ballot, including the randomness used in the encryption. He also has to provide a proof of the validity of his ballot, showing that he used a valid vote. The ballot is transfered to the randomizer, which reencrypts it using new randomness. The voter makes sure the vote hasn't been changed by verifying the respective proof. Finally, the proof of validity has to be adapted to the ballot's new randomness. This is done jointly by the voter and the randomizer.

3.2.4 Blind Signatures: [Oka97]

This protocol uses the voting scheme with anonymous channels and blind signatures. We assume the presence of a voting booth with anonymity (i.e. the authorities do not know who the voter is). While this is fine for traditional elections, this will be very hard to achieve in the digital setting.

4 Deniable Encryption, Incoercible MPC

Two cryptographic primitives and protocols, which might be of interest in the context of receipt-freeness, are deniable encryption ([CDNO97]) and incoercible multiparty computation ([CG96]). It has to be noted that both constructs are rather expensive with respect to the amount of communication necessary.

Two parties communicate by transfering encrypted messages. A message is intercepted by an adversary who later asks one of the parties to reveal the random choices (and the secret key, if one exists) used in the encryption. Deniable encryption allows either of them to present fake random choices to the adversary. This will make the ciphertext look like an encryption of a different cleartext. Different protocols are described in the paper offering deniability for symmetric and asymmetric encryption (with the case of encryption with one-time-pads trivially enjoying deniability).

Incoercible MPC (which makes use of deniable encryption) extends this idea to the MPC setting. Here the protocol enables us to generate fake inputs, outputs and random choices that match the public transcript of the performed MPC.

While these protocols don't help us in the standard model, they might turn out to be useful in a relaxed version of the same. Moreover they can only be applied in the case of coercion. As soon as the voter is not interested - as is the case in vote-buying - in taking advantage of the deniability, they lose most of their power. As I've chosen to concentrate mainly on the treatment of vote-buying, the relevance of these constructs is only small.

5 Practical Models

Clearly the assumptions made in receipt-free protocols so far make them unsuitable for real-life elections (what could be considered the minimum requirement here are one-way untappable channels from the authorities to every voter). Yet at the same time there's no hope for receipt-freeness in the standard model.

Could there be something in the wide area between always-tapped and never-tapped channels, offering the desired properties but still being practical?

5.1 Modified Standard Model

When looking at the standard model there is one assumption, which seems to be far too restrictive from the point of view of the protocol designer. How reasonable is it to assume that an adversary can tap the channels between all voters and authorities of the Internet during an entire election? Is it possible to slightly relax this restriction, just enough so we can be sure nobody will be interested in buying votes anymore?

5.1.1 Partially Tappable Networks

It would be very helpful to have a model of a network where tapping is allowed only for a subset of all connections as well as only for a limited time. Unfortunately I've not been able to come up with a useful model for such a "partially tappable network" which would allow me to deduce or prove statements beyond trivial observations.

A model should cover following elements:

- when and which connections are tapped
- (similarly) probability of a particular connection being tapped at a certain point in time
- description of the adversaries' tapping capabilities (how many connections or channels can he tap simultaneously, how much information can he store)
- how to take the network's topology into account

Intuitively the topology of the network under scrutiny is very important (the Internet for example roughly consists of a multitude of hierarchically arranged networks and subnetworks). To illustrate its importance just consider all the direct connections from the voters or the authorities to the Internet. Tapping all those connections would seem - although minimal in extent - to be worst attack one can think of, basically leading us back to the standard model situation.

Given such a network, how do we exploit its new features?

5.1.2 Bundling and Splitting of Ballots

If a voter finds an untapped path to one of the authorities, he should be able to vote in a receipt-free manner. But we're facing two problems here. How does he find this path? And even if he knew a method to find it, would he be interested to do so? Someone wanting to sell his vote certainly wouldn't. If the ballot were to take a path completely outside the control of the voter, its probability of traveling "safely" would be much higher. Splitting the ballot and transfering the separate pieces over multiple paths would again raise this probability.

The following idea makes use of these observations. We partition the set of voters into small groups. To vote, every group first engages in a joint intra-group computation. As a result, everyone holds a piece of everyone else's ballot. Then every voter transfers the pieces he holds to the authorities. One single untapped path used by any member of the group should be enough to ensure receipt-freeness for all ballots of the group. The bigger the group, the higher the chance of there being a member who is honest (meaning he doesn't want to sell his vote) and lucky enough to have a untapped voting channel. If a large portion of all connections in a network is tapped, the probability for a randomly chosen path being untapped from source to destination is rather low. Intuitively, applying the described bundling and splitting of ballots will "strengthen" the weak network. Of course the partitioning must be done by the authorities to prevent "bought" voters from getting together.

Still, there are a lot of obstacles to overcome. For example, the communication between the members of a group will be partially tapped as well. Under which circumstances will receipt-freeness still be satisfied? If a group member rejects to cooperate, how do the others resolve the situation?

5.2 Hybrid Systems

No country is going to switch from traditional elections to digital ones overnight. There will be a gradual transition. Voting offices and paper-based ballots will certainly coexist with their digital alternative for some time to come. So it might be worth thinking about whether and how we could take advantage of the "tools" they offer. If we don't treat them as strict alternatives, maybe the combination of both toolsets might bring us nearer to the desired properties we're aiming for.

5.2.1 Geneva

One example where a combination of systems is being used today is Geneva. There, you may use the mail or go to the voting office directly to the cast your vote. But it is also possible to use the Internet for this purpose. At present these systems are mostly separated. They only actively "share information" (besides the tallying of course) to efficiently prevent double voting. The ballot sent by mail to every voter contains a field with a hidden individual secret number. To uncover this number, the field's surface has to be scratched off. This opening process is irreversable. Once it has been done, the ballot can only be used for Internet voting.

5.2.2 Last Ballot

A simple idea illustrates how a hybrid system can be much more powerful than a digital one alone.

Without altering the voting protocol itself, we could allow the voter to sent multiple ballots, or more precisely voting multiple times. All but the chronologically last ballot would be discarded. Multiple ways of vote casting (Internet, in persona) could be allowed. So anyone selling his vote today could still go to the voting office the next day and "change" his ballot by casting a new one. This renders receipts useless.

5.2.3 Additional Primitives

Looking at today's voting systems, what "tools" do they offer? We now have additional ways of casting a vote. The ballot can reach the authorities by...

- being sent by mail
- being brought to the voting office personally
- combinations of Internet and mail

The voter receives a letter in the mail containing his ballot. Various elements could be added to this paper-based ballot with the goal of making the vote receipt-free:

- separate and easily fakeable piece of paper with a secret number printed on
- stack of papers in a special ordering (cast ballot depends on this initial ordering)
- circular sequence of papers in a calendar-like fashion (similarly to the previous idea)
- secret numbers hidden in fields whose surface can be scratched off (irreversibly revealing the number)

What is common to all these elements? If we want to transfer secret information to the voter, it has to be done in a deniable way. The information is contained in some sort of "initial configuration". This initial configuration must display following characteristics:

- random
- different for every voter (i.e. every voter gets his personal initial configuration)
- known to authorities
- easily changable by the voter
- (ideally it would be) jointly generated by multiple authorities, without any single one (especially the last) knowing the complete result

Additionally, all possible configurations have to be "on the same level", meaning every configuration can always be reached from every other configuration. This seems to rule out all ideas using hidden secret numbers, as their opening is irreversible.

6 Open Problems

It would be interesting to have a detailed comparison of the primitives offered by traditional and digital voting systems (e.g. voting booth, untappable channels, mail). Knowing which primitive are equal to or a real subset of others would allow us to transform existing and future protocols partially or completely from paper-based to digital voting systems and vice versa. This would of course also require a detailed description of the circumstances or restrictions under which it is possible to replace one primitive with another without changing any of the protocol's properties.

To what extent can universal verifiability and receipt-freeness be satisfied at the same time? These two properties seem to be inherently conflicting. Universal verifiability is the combination of global verifiability and an extended version of local verifiability. Any voter can verify whether everyone else's vote is included in the published tally. In general, universal verifiability implies use of a bulletin board. If all ballot were published on the bulletin board in cleartext, verification would be simple. Receipt-freeness on the other hand works best if as little information as possible is known about the every single ballot.

7 Conclusions

There is no general recipe to turn a protocol into a receipt-free version of the same. Receipt-freeness cannot be added as an afterthought.

It's important to have a precise definition of the concept, which also takes into account the fundamental differences of the two attacks.

Existing receipt-free protocols are based on rather unrealistic assumption, which often aren't stated explicitly. Many have been partially or completely broken.

Receipt-freeness is a subtle property. It's very hard to get right. The inherent conflict of privacy and correctness as well as the fact the we're dealing with two adversaries (in the case of vote-buying) complicates the construction of protocols.

Still, the combination of mail-based and digital voting systems as well as communication models resembling the Internet in a more suitable fashion than the standard model promise virtually endless possibilities to construct voting protocols. So there's a good chance of inventing a protocol offering a reasonably good form of receipt-freeness.

A Project Description

Semester Project
Assistant: Martin Hirt
Start: October 23, 2001
End: March 15, 2002

A.1 Introduction

Secret-ballot voting schemes belong to the most significant applications of cryptographic protocols. They allow the computation of the tally of the cast votes, while preserving privacy of each particular ballot. More precisely, the following properties must be satisfied:

- Secrecy. It is infeasible to find out which voter has submitted which vote.
- Eligibility. Only entitled voters are able to submit a vote.
- no double-voting. Every entitled voter can cast only one single vote.
- Validity. Only valid votes are counted, e.g., "yes" and "no" votes.
- Correctness. The published tally is the correct sum of all valid votes.
- Local verifiability. Every voter can verify whether his vote is included in the tally.
- Global verifiability. Anyone can verify the correctness of the tally.

An important concept that was neglected in the classical literature is *receipt-freeness*. The goal of receipt-freeness is to thwart vote-selling and coercion. More formally, receipt-freeness requires that the voter is not able to construct a receipt (a witness) of the cast vote.

The concept of receipt-freeness was introduced by Benaloh and Tuinstra [BT94]. Later, many receipt-free protocols were proposed [SK95, Oka96, Oka97, HS00, LK00, Hir01, BPP$^+$01]. All these protocols make additional (rather unrealistic) assumptions on the communication model, e.g., existence of a voting booth or of untappable channels.

Receipt-freeness is a very subtle property. The problem is that the voter *wants* to prove his vote, and hence he might even deviate from the protocol in order to construct a receipt. Even worse, there are no sharp definitions of receipt-freeness, and many different flavors of this property are implicitly considered in the literature. Several of the proposed protocols were partially or completely broken [MH96, Oka97, Sch99, HS00].

A.2 Description

The goal of this work is to bring light into the definitions and models of receipt-freeness. So far, many different (and incompatible) notions are used, and comparison of proposed protocols and models is very difficult. Several conjectures and propositions on impossibility or optimality of certain models can be found in the literature, where formal definitions and proofs are missing. Special focus should be given to practical models, leading to the question what type of receipt-freeness is still possible with realistic assumptions.

A.3 Tasks

The following is an (incomplete) list of tasks:

- study of the literature,
- find exact definitions of receipt-freeness,
- propose realistic models for receipt-free voting protocols,
- proof sufficiency and necessity of assumptions.

References

[BPP^{+}01] Olivier Baudron, David Pointcheval, Guillaume Poupard, Jacques Stern, and Pierre-Allain Fouque. Practical multi-candidate election system. In *Proc. 20th ACM Symposium on Principles of Distributed Computing (PODC)*, 2001.

[BT94] Josh Cohen Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *Proc. 26th ACM Symposium on the Theory of Computing (STOC)*, pages 544–553. ACM, 1994.

[CDNO97] Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In *Advances in Cryptology — CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 90–104, 1997.

[CG96] Ran Canetti and Rosario Gennaro. Incoercible multiparty computation. In *Proc. 37th IEEE Symposium on the Foundations of Computer Science (FOCS)*, pages 504–513, 1996.

[Hir01] Martin Hirt. *Multi-Party Computation: Efficient Protocols, General Adversaries, and Voting.* PhD thesis, ETH Zurich, 2001. Reprint as vol. 3 of *ETH Series in Information Security and Cryptography*, ISBN 3-89649-747-2, Hartung-Gorre Verlag, Konstanz, 2001.

[HS00] Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. In *Advances in Cryptology — EUROCRYPT '00*, volume 1807 of *Lecture Notes in Computer Science*, pages 539–556, 2000.

[LK00] Byoungcheon Lee and Kwangjo Kim. Receipt-free electronic voting through collaboration of voter and honest verifier. In *Japan-Korea Joint Workshop on Information Security and Cryptology (JW-ISC2000)*, pages 101–108, 2000.

[MH96] Markus Michels and Patrick Horster. Some remarks on a receipt-free and universally verifiable mix-type voting scheme. In *Advances in Cryptology — ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 125–132, 1996.

[Oka96] Tatsuaki Okamoto. An electronic voting scheme. In *Proc. of IFIP '96, Advanced IT Tools*, pages 21–30. Chapman & Hall, 1996.

[Oka97] Tatsuaki Okamoto. Receipt-free electronic voting schemes for large scale elections. In *Proc. of Workshop on Security Protocols '97*, volume 1361 of *Lecture Notes in Computer Science*, pages 25–35, 1997.

[Sch99] Berry Schoenmakers, 1999. Personal communication.

[SK95] Kazue Sako and Joe Kilian. Receipt-free mix-type voting scheme – A practical solution to the implementation of a voting booth. In *Advances in Cryptology — EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 393–403. Springer-Verlag, 1995.