Formalising Receipt-Freeness

H.L. Jonker[1] and E.P. de Vink[1,2]

[1] Dept. of Math. and Comp. Sc., Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, the Netherlands
[2] LIACS, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, the Netherlands

Abstract. Receipt-freeness is the property of voting protocols that a voter cannot create a receipt which proves how she voted. Since Benaloh and Tuinstra introduced this property, there has been a large amount of work devoted to the construction of receipt-free voting protocols. This paper provides a generic and uniform formalism that captures the notion of a receipt. The formalism is then applied to analyse the receipt-freeness of a number of voting protocols.

Keywords: online voting schemes, receipt-freeness, formal methods, eGovernment.

1 Introduction

In 2005, the Estonian government held elections for local office. Voters could cast their votes online or in a traditional voting booth. This constituted the first application of online voting on a national scale. Prior to the Estonian 2005 local office elections, a large amount of research has been conducted into voting and online voting. Voting is a means to establish consensus for a group of people regarding a certain set of candidates or choices. There are various methods to conduct an election, such as collecting one vote per voter (1V1V for short), e.g. elections for a governmental office, or collecting a list for each voter which orders the candidates by voter's preference (e.g. the scheme used to determine the Dutch *Top 2000* music list). This paper focuses upon 1V1V online voting protocols, in which all votes have equal weights.

Online voting schemes can offer an important advantage over traditional elections involving paper ballots in voting booths: voting is not restricted to a limited number of locations which have limited opening hours, but can be done anywhere a network connection is available, for as long as the elections are open. However, there is also a danger: strengths of the paper ballot system may be lost and new flaws can be introduced. To prevent this, electronic voting schemes should be designed to comply with a complete set of requirements.

Over the years, several desirable properties of electronic elections have been distinguished (see e.g. [15, 10]). In this paper we focus on the property of *receipt-freeness*. A receipt allows a voter to prove how she voted. If receipts cannot be constructed, the voting protocol is said to be receipt-free.

Most other established properties of voting protocols, such as universal verifiability (any party can verify that the result is compromised of all legitimately cast votes, and nothing else), democracy (only eligible voters can vote, and all voters can vote once), accuracy (no vote can be altered, duplicated or forged without being detected), robustness (the system can resist faulty behaviour of any reasonably sized coalition of participants) and integrity (the result of a vote consists precisely of all valid votes, and nothing more), are orthogonal to the concept of receipt-freeness. However, the notion of individual verifiability can be in conflict with it. Individual verifiability is the property that an individual voter can verify that her vote is correctly taken into account in computing the final tally. If this verification is transferable, the verification and the keys used to encrypt and/or blind the vote together would constitute a receipt. One method to reconcile the two properties is to use designated-verifier proofs [13].

This paper formalises the notion of receipt-freeness, which was introduced in the seminal work of Benaloh and Tuinstra [3]. In the early nineties, this property was violated by virtually all online voting schemes of that time. (Note that paper ballot 1V1V voting systems usually are receipt-free, which means that an online voting system which allows for receipts introduces a new flaw.) The lack of receipt-free schemes prompted the development of voting schemes, which would account for receipt-freeness in addition to other requirements.

In [3], the concept of receipt-freeness was introduced for a particular protocol (which we name BT). The definition does not lend itself for a convenient generalisation, however. It relies on distinguishing votes for candidate '0' from votes for candidate '1'. The main contribution of our paper is a constructive, generic formalisation of the notion of receipt-freeness in a uniform framework for 1V1V voting schemes. The process algebraic setting adopted here facilitates reasoning about and verification of receipt-freeness and enables detection and identification of receipts. The proposed formalism is then applied to the voting protocols BT of Benaloh and Tuinstra [3], SK95 of Sako and Kilian [18], HS of Hirt and Sako [11], ALBD of Aditya et al. [1] (which are receipt-free) and the RIES protocol as discussed in [12] (which is not).

Since Benaloh and Tuinstra's work there has been much research into receipt-freeness. In some works, e.g. [17], this notion is refined into two notions: receipt-freeness, which holds if a protocol does not require receipts to function, and uncoercibility which holds if receipts cannot be constructed. However, in this paper receipt-freeness means that voters cannot construct a proof of how they voted. These two concepts seem closely related, but as Chaum shows in [4], a voting protocol may give tickets (which seem like receipts) to voters without the voter being able to use these to prove how she voted. The reverse is also possible: a voting protocol need not supply voters with receipts in order for voters to be able to construct proofs of how they voted. An example of this is given by the attack on Benaloh and Tuinstra, mentioned in [11].

We note that this attack is not targeted at the core of BT, but exploits an auxiliary procedure in which voters prove that their encrypted vote is either a '0' or a '1'. This auxiliary problem can be mitigated without breaking receipt-freeness, e.g. by applying the technique used in [2, Section 10.3.2]. We conclude that receipt-freeness of the core of BT remains unchallenged. Below we will argue that the essence of the BT protocol indeed satisfies the receipt-freeness requirement.

The remainder of this paper is organised as follows: Section 2 describes related work. In Section 3, the notion of receipts in online voting systems is examined in depth. Section 4 formalises the notion of receipts, and expresses receipt-freeness in a process-algebraic setting. In Section 5, the application of this formalism to various protocols (BT, SK95, HS, ALBD and RIES) is discussed. Section 6 presents conclusions and possible directions for future research.

Acknowledgements The authors would like to thank Jos Baeten, Berry Schoenmakers, Wolter Pieters, Cas Cremers and Anna Zych for their insightful remarks on this subject and Simona Orzan for proofreading this paper.

2 Related work

In [8], an earlier formalisation of receipt-freeness is proposed. This formalisation does not define what constitutes a receipt, but instead relies on observational equivalence between a system with a colluding voter and a system without a colluding voter. A disadvantage of this formalism is that while it may be used to establish receipt-freeness of a protocol, it offers little aid to identify receipts when these are present. Our approach focuses on establishing what can constitute a receipt. This enables the identification of receipts, and provides a heuristic to take receipts into consideration in the early stages of designing a protocol.

Many voting protocols have been developed over the years. FOO [9] is well-known and has served as a base for various implementations (e.g. [7]). Its main goal is to allow voters to vote anonymously. RIES [12] has been used in two water management board elections in the Netherlands, handling over 70,000 votes in one case (making it one of the largest online elections). Where FOO and RIES are geared towards implementations, CFSY [5] and CGS [6] are aimed at providing desirable properties. The protocol described in CFSY provides information-theoretic privacy, universal verifiability and robustness. The main difference with CGS is that CGS allows a significant reduction in workload per voter, but offers only computational privacy.

Since Benaloh's and Tuinstra's work, various receipt-free voting protocols have been proposed, such as [18, 1, 11]. Other research has been aimed at providing receipt-freeness in more generic settings (as opposed to a single protocol), such as [16], which focuses on mix-nets, and [17], which proposes the use of an external trusted source of randomness.

The mechanisms to prove receipt-freeness given in these works are aimed at the specific topics covered. There seems to be little work towards a general formalisation of receipt-freeness that provides a uniform method to establish the absence or presence of receipts. It is the aim of the present paper to provide such a formalisation.

In the typical set-up adopted here, the adversary under consideration is an outside observer. Juels et al. mention several privacy-related attacks on voting protocols in [14]. Privacy and receipt-freeness are closely related, however, as stated in [14], receipt-freeness does not prevent these attacks. Juels et al. offer a stronger concept, which they call coercion-resistance, that offers receipt-freeness and prevents these attacks. Like [8], the work of Juels et al. enables establishing coercion-resistance, but their computational adversary model provides no indiciation of what constitutes a violation of this property when it is succesful.

This paper focuses on receipt-freeness (hence, it does not consider these privacy-related attacks) and on providing a constructive approach to determining receipt-freeness.

3 Receipts in online voting schemes

Intuitively, a receipt r is an object that proves that a voter v cast a vote for candidate c. This means that a receipt r has the following properties:

(R1) r can only have been generated by v.
(R2) r proves that v chose candidate c.
(R3) r proves that v cast her vote.

The difference between property (R2) and proerty (R3) is highlighted by the following example. Consider an election with paper ballots, where the ballots authenticate voters. In this setting, a voter could fill in a ballot, leave the voting booth with the ballot and show the ballot to anyone. This would satisfy properties (R1) and (R2), while clearly the voter did not cast a vote! Thus, a receipt must also prove that the voter cast her vote, which provides the justification for introducing (R3).

Although rather intuitive, the above properties do not catch receipt-freeness in non-1V1V settings. Consider, for example, the cheating in Italian elections for multiple posts (that are not 1V1V) as mentioned by Benaloh and Tuinstra in [3]. The authentication in that attack is provided by assigning each voter a permutation of posts. The voter authenticates herself by listing her choices in the assigned order on the ballot. Since anyone can construct such a permutation, this is usually not sufficient to satisfy (R1) because of the small number of permutations involved.

Receipts are regarded as undesirable objects in voting systems, for the following reason: if receipts can be created, an attacker can coerce (e.g. threaten) or entice (e.g. bribe) voters to vote in a manner of his choosing, and the voters can prove that they complied with this manner. This would mean that the result of an election no longer reflects the consensus of the voters, which is the primary goal of a voting system.

For many existing voting systems generation of receipts is possible. We illustrate this for the FOO protocol [9]. The main goal of the FOO protocol is to provide anonymity for the voters. This is achieved by the use of anonymous channels and blind signatures. Blind signatures are a cryptographic mechanism that allows a party to encrypt a message, have the encrypted message signed by another party and then remove the encryption, whilst preserving the signature.

The FOO protocol works as follows: First, a voter has her blinded, encrypted vote signed by a registration authority. Next, she removes the blinding and sends the signed, encrypted vote anonymously to the counter. After all votes have been received, the counter publicises an enumerated list of all received votes. The voter then sends her decryption key (and associated list number) to the counter. After the counter has received all keys, he updates the public list to include the keys.

There is an obvious receipt here, given the public list: the key k used by the voter and its number ℓ in the list. This pair (ℓ, k) matches the requirements for receipts: k authenticates a specific voter (satisfying (R1)); using k, the ℓth entry can be shown to be a vote for a specific candidate (satisfying (R2)); and since the pair (ℓ, k) is on the list, this refers to a cast vote (satisfying (R3)).

It should be noted that if the above receipt is shown to have been generated before the keys are published, it remains a valid proof after the publication of the keys. One way to prove that the receipt was constructed before publication is to use timestamping. As timestamping can be used in general, we adopt the view that the protocol is not receipt-free if at any time a voter can create a receipt.

4 Formalisation

In this section the notion of receipts will be formalised and subsequently used to specify what receipt-freeness means formally.

The architecture of 1V1V voting protocols is usually built from the following fundamentals:

- agents taken from the set $\mathcal{A}$, voters $\mathit{Vot} \subseteq \mathcal{A}$
- choices or candidates in Can, ballots in $\mathcal{B}$, and results (multisets of choices) in $\mathcal{M}(\mathit{Can})$
- received ballots in $\mathcal{RB}$, from which the result will be computed
- a choice function $\Gamma\colon \mathit{Vot} \to \mathit{Can}$, which specifies how the voters vote
- a function $\mathit{tally}\colon \mathcal{P}(\mathcal{RB}) \to \mathcal{M}(\mathit{Can})$, which gives the election result for a set of received ballots.

To denote receipts, the following notation is used:

- the set of receipts Rcpt
- $\mathit{Terms}(v)$, the set of all terms that a voter $v \in \mathit{Vot}$ can generate
- authentication terms $\mathcal{AT}(v)$ for each voter, such that

$$t \in \mathcal{AT}(v) \implies \forall w \neq v\colon t \notin \mathit{Terms}(w)$$

  we put $\mathcal{AT} = \{\mathcal{AT}(v) \mid v \in \mathit{Vot}\}$

– a function $auth \colon \mathcal{AT} \rightarrow Vot$, which returns the unique voter that created an authentication terms.

A voting system is a system which takes a choice function as input and returns a set of received ballots as output.

Definition 1 *Given the above ingredients, a voting system $\mathcal{VS}$ is a mapping: $\mathcal{VS} \colon (Vot \rightarrow Can) \rightarrow \mathcal{P}(\mathcal{RB})$.*
The result of $\mathcal{VS}$ for choice function Γ is given by: $tally(\mathcal{VS}(\Gamma))$.

We have occasion to use the following auxiliary receipt decomposition functions:

– $\alpha \colon Rcpt \rightarrow \mathcal{AT}$, which extracts the authentication term from a receipt
– $\beta \colon Rcpt \rightarrow \mathcal{B}$, which extracts the ballot from a receipt
– $\gamma \colon Rcpt \rightarrow Can$, which extracts the candidate from a receipt.

Note that these functions depend on the structure of receipts, and thus vary from voting system to voting system. For example, for the FOO protocol discussed above, receipts are of the form (ℓ, k) if $\mathcal{RB}$ was known. Assuming that elements of $\mathcal{RB}$ are of the form $(num, \{vote\}_k)$, this means that in FOO (where $\mathcal{B} = \mathcal{RB}$):

– $\alpha(\mathcal{RB}, (\ell, k)) = k$,
– $\beta(\mathcal{RB}, (\ell, k)) = (\ell, e) \in \mathcal{RB}$, for some encrypted vote e, and
– $\gamma(\mathcal{RB}, (\ell, k)) = c$, such that $\{c\}_k = e$.

Using the above ingredients, the notions (R1) to (R3) can be expressed as follows.

Definition 2 *A receipt r for voter v concerning candidate c has the following properties:*

(r1) $\alpha(r) \in \mathcal{AT}(v)$
(r2) $\gamma(r) = \Gamma(v)$
(r3) $\beta(r) \in \mathcal{RB}$.[3]

Thus, a valid receipt r of voter v for candidate c can now be characterised as follows: $auth(\alpha(r)) = v \implies \gamma(r) = \Gamma(v)$ (satisfying (R1) and (R2)) and $\beta(r) \in \mathcal{RB}$ (satisfying (R3)).

Since $\gamma = \Gamma \circ auth \circ \alpha$ satisfies $auth(\alpha(r)) = v \implies \gamma(r) = \Gamma(v)$, we can now formulate when a voting system $\mathcal{VS}$ allows receipts.

[3] This assumes that $\mathcal{RB} \subseteq \mathcal{B}$. In case there are additional operations on the ballot after the voter sends her ballot (such that $\mathcal{RB} \not\subseteq \mathcal{B}$), which the voter cannot apply to her ballot herself, an auxiliary partial function $\phi \colon \mathcal{B} \rightarrow^* \mathcal{RB}$ is introduced; the requirement then is formulated as $\phi(\beta(r)) \in \mathcal{RB}$. Unless otherwise mentioned, $\mathcal{RB} \subseteq \mathcal{B}$.

Definition 3 *A protocol $\mathcal{VS}(\Gamma)$ grants receipts in Rcpt with derived functions α, β, γ iff*

$$\gamma = \Gamma \circ auth \circ \alpha \ \wedge \ \exists \varrho \in Vot \rightarrow Rcpt \colon \forall v \in Vot \colon \beta(\varrho(v)) \in \mathcal{RB}$$

As is obvious from the definition, γ can be seen as an abbreviation for $\Gamma \circ auth \circ \alpha$. Therefore, to determine presence of a receipt in a given protocol (for given $\Gamma, auth$), only α and β need to be defined.

The advantage of the above generic formalism is that it covers all receipts, also ones constructed by side-channels. The obvious disadvantage is that it does not provide us with a procedure to verify receipt-freeness of a given protocol. Note that a receipt is derived from a particular execution of a voting protocol, also referred to as a run. Only the public and private information exchanged during the protocol run can be considered a building block of an associated receipt. Information not present in the run does not qualify for this.

In our approach to abstract voting protocols, all data is represented by terms. If we limit the notion of receipts to terms (i.e. $Rcpt \subseteq Terms$), procedural verification of receipt-freeness becomes possible by examining the suitability of terms as receipts. However, by limiting the formalism to terms, the ability to detect receipts constructed using side-channels (e.g. the interval between sending messages) is lost. In the remainder of this paper, we only consider receipts that are terms.

The exact syntax of terms depends on the specific protocol, however, in general terms it can be described by the following:

Definition 4 *Given*

- *a set of choices or candidates Can*
- *a set of plaintexts PT (PT_i for specific agent i)*
- *a set of keys Keys ($Keys_i$ for specific agent i)*
- *a set of functions Func ($Func_i$ for specific agent i)*

the class of terms of agent i is defined by

$$\begin{aligned} m_i ::= c \in Can \mid pt_i \in PT_i \mid k_i \in Keys_i \mid \\ (m_1, m_2)_i \mid \{m\}_k^i \mid f(m)_i, f \in Func_i \end{aligned}$$

which denotes respectively a candidate, a plaintext, a key, tupling of two terms, encryption of a term or function application to a term.
We write $t \in t'$ if t is a subterm of t'.

We can now formulate what extraction of a term means: For all functions F which extract one subterm from a term, it obviously holds that $F(t) \in t$. Therefore, we have the following observation.

Lemma 1. *(SUBTERMS) For all terms $t \in Rcpt$: $\alpha(t) \in t \ \wedge \ \beta(t) \in t$.*

The definition of $\mathcal{AT}(v)$ directly leads to

$$t \ \in \ t' \ \wedge \ t \ \in \ \mathcal{AT}(v) \ \Longrightarrow \ t' \ \in \ \mathcal{AT}(v)$$

which in turn leads to the following.

Corollary 1 *It holds that $auth(\alpha(r)) = v$ iff $r \in \mathcal{AT}(v)$.*

5 Application

The purpose of this section is to illustrate how to apply the formalism from the previous section to existing protocols. We provide a high-level analysis of the receipt-freeness of various voting protocols.

5.1 Application to Benaloh-Tuinstra

Since we are particulary interested in receipt-freeness, we restrict our examination of BT to just those parts related to receipt-freeness. As secret-sharing is not related to receipt-freeness, we focus on the 'scaled-down election protocol' as opposed to the protocol for multiple authorities. The proof of receipt-freeness of this specific protocol has not been disputed as far as we are aware. BT uses probabilistic encryption, which we denote by $x \in E(m)$ if x is an encryption of m.

The protocol distinguishes the following parties: voters $v \in Vot$; an authority A, who instantiated a probabilistic encryption scheme with parameter n, encryption function $E()$ and decryption function $D()$, and a beacon BC. The role of the beacon is to provide random bits, which are used in zero knowledge proofs (ZKP). Public communications between A and B are denoted by $s_{a \to b}$ (send) and $r_{a \to b}$ (receive), private communication (communication over a private, untappable channel) is denoted by $p_{a \to b}$. A superscript '*' denotes communication of BT's ZKP of the message instead of the actual message, so $p^*_{a \to b}(m)$ means A sends a ZKP of m over a private channel to B. As proven in [3], no terms of this ZKP can identify the voter's ballot (i.e., for all such terms t, $\forall b \in \mathcal{RB}\colon b \notin t$). Processes can be guarded: $b \to P$ denotes the process that can only execute if b is true.

The voting authority A takes the following steps per voter v: A sends V an encryption of '0' and an encryption of '1' in random order, and, over a private channel, a ZKP of which is which. The voter then casts her vote by returning the received value corresponding to her choice. This is modelled as follows (note that $min(), max()$ are used here to provide a random order of the two encryptions).

$$\begin{aligned} A(v) = \textstyle\sum_{x \in E(0),\ y \in E(1)}\ & s_{a \to v}(min(x,y), max(x,y)) \cdot \\ & p^*_{a \to v}(x \in E(0) \wedge y \in E(1)) \cdot \big(\ r_{v \to a}(x) + r_{v \to a}(y)\ \big) \end{aligned}$$

A voter v thus receives two encryptions over a public channel and a ZKP over a private channel proving which ciphertext denotes '0'. The voter then sends the encryption corresponding to the vote of her choice (given by $\Gamma(v)$) over a public channel. This is modelled by

$$\begin{aligned} V = \textstyle\sum_{x,y}\ r_{a \to v}(x,y) \cdot \textstyle\sum_{i \in \{0,1\}}\ & p^*_{a \to v}(x \in E(i) \wedge y \in E(1-i)) \cdot \\ & \big(\Gamma(v) = i \to s_{v \to a}(x) \quad + \quad \Gamma(v) = 1-i \to s_{v \to a}(y)\ \big) \end{aligned}$$

In examining the possible terms for voter v, note that v performs three actions: a receive over a public channel; a non-transferable receive over a private, untappable channel; and a send of a subterm of the message received in the first action over a public channel. It is clear that the first receive action does not provide an authentication term (communications over public channels can never constitute authentication terms). Hence, the last send action cannot communicate an authentication term. These actions thus cannot provide a term to satisfy the first conjunct of Definition 3 (since these terms are not in $\mathcal{AT}$, they are not in the domain of α).

The only possible terms which could supply a receipt are therefore the terms used in the ZKP. Since for all these terms t, we have that $\nexists b \in \mathcal{B}\colon b \in t$, the SUBTERMS lemma, Lemma 1, cannot be satisfied by any of these terms. Hence, none of the terms used in the protocol can act as a receipt.

5.2 Receipt-free voting using mixnets

The well-known SK95 protocol [18] uses a mixnet to provide anonymity. The protocol shuffles all possible votes and uses secure, untappable channels to the voter to prove that the mixnet functions correctly. Due to the untappability, there are no side-channel attacks on these proofs. Once again, these proofs are ment to provide the voter with the ability to prove different ordenings of the shuffled vote. This is done using "chameleon blobs" – zero knowledge bit commitments that can be opened by a verifier in both ways. Because of this our analysis will use the assumption that these proofs cannot be exploited to provide receipts.

SK95 works as follows: The mixnet shuffles all possible votes and provides a proof of correctness of shuffling and a proof of the ordening of the votes to the voter, over secure, untappable channels. The voter submits the vote corresponding to her choice to the mixnet for the counter.

It is obvious from the above description that, aside from the proofs, voters possess no distinguishing knowledge. This means that no terms outside the proofs provide a means to distinguish voters, i.e. no term outside the proofs is in the set $\mathcal{AT}(v)$. Since we assumed that the proofs cannot be exploited to provide receipts, this lack of authenticating terms leaves us with no means to construct an appropriate function $\alpha\colon Rcpt \to \mathcal{AT}$, since $\mathcal{AT} = \emptyset$. Thus, the protocol is receipt-free.

Note that to fully prove receipt-freeness of SK95, obviously the transmitted proofs have to be considered as well. These proofs do provide authentication terms, but these terms are not linked to ballots, i.e.

$$\forall t \in \mathit{Terms}(v)\colon t \notin \mathcal{AT}(v) \vee \nexists b \in \mathcal{B}\colon b \in t$$

which violates the SUBTERMS lemma.

5.3 Receipt-free voting using homomorphic encryption

In the work by Hirt and Sako [11] the shuffling of SK95 is applied to protocols based on homomorphic encryption. In this respect, the HS protocol resembles the SK95 protocol. They propose a voting protocol which works as follows: each valid vote is encrypted in a deterministic way. Then, each authority takes the set of encrypted votes supplied by the previous authority, reencrypts each vote and outputs the votes in a random order. Again, each authority transmits a proof of correctness of this shuffle is transmitted publicly and the permutation is privately (untappably) communicated to the voter. The correctness of the permutation is privately proven to the voter using a designated-verifier proof.

Due to the conceptual similarities with SK95, we will not delve deeply into the analysis of HS here. We note that the analysis of SK95 applies to HS as well: voters cannot be distinguished except by terms used in proving the shuffle, and assuming these proofs do not provide receipts, the protocol is receipt-free. Since the proofs are for a designated verifier, these proofs cannot be used to convince anyone but the voter of the order of the votes. Thus, again,

$$\forall t \in \mathit{Terms}(v) \colon t \notin \mathcal{AT}(v) \vee \nexists b \in \mathcal{B} \colon b \in t$$

and thus the proposed protocol is receipt-free.

Note that Schoenmakers proposed an attack on this scheme, as mentioned in [14]: The coercer can force a voter to vote randomly. However, as mentioned in the introduction, this does not violate receipt-freeness and therefore it is not further considered.

5.4 Receipt-freeness despite signing

The voting protocol ALBD, proposed by Aditya et al. in [1], differs from BT, HS and SK95 because it requires voters to sign votes. This means that there exist deliberate authentication terms for this protocol. On top of that, voter-supplied randomness is used to encrypt their vote in violation of the analysis in [17, Section 2.3]. However, receipt-freeness is attainable as these authenticating terms are only used in communications over an untappable channel. Assuming the administrator never discloses any knowledge of these terms, they cannot be linked to cast ballots.

We use the following notation in the process description: $E(m, k)$ denotes a random encryption of message m using key k, and $R(m, k')$ denotes re-encryption of m with key k'. $\{m\}_{SK(v)}$ denotes the signing of message m by agent v. Again, $p_{v \to a}(m)$ denotes communications over a private, untappable channel and $p^*_{a \to v}(m)$ denotes a communication over the private channel consisting of a designated-verifier proof of m.

ALBD uses a two-way untappable channel. Vote casting works as follows: A voter encrypts her vote ($m = E(\Gamma(v), k)$), signs it ($s = \{m\}_{SK(v)}$), and sends it over the untappable channel to the administrator. After the vote has closed, the administrator publishes re-encryptions ($r = R(s, k')$) of all received votes in random order. The administrator sends each voter a designated-verifier proof of the correctness of the re-encryption of their vote (denoted by $\exists k' \colon r = R(E(\Gamma(v), k), k')$). This is modelled as follows:

$$\begin{aligned} V(v) = {} & \textstyle\sum_k p_{v \to a}(\{E(\Gamma(v), k)\}_{SK(v)}) \cdot \\ & \qquad \textstyle\sum_r s_{a \to v}(r) \cdot p^*_{a \to v}(\exists k' \colon r = R(E(\Gamma(v), k), k')) \\ A(v) = {} & \textstyle\sum_y p_{v \to a}(\{y\}_{SK(v)}) \cdot \sum_{k'} s_{a \to v}(R(y, k')) \cdot \\ & \qquad\qquad p^*_{a \to v}(\exists k \colon R(y, k') = R(y, k)). \end{aligned}$$

The designated-verifier proof only involves communication from the administrator to the voter. Because of this and that it is a *designated-verifier* proof it cannot be exploited to acquire a receipt.

Receipt-freeness of ALBD holds, because $\beta(r)$ cannot be constructed, as

$$\forall t \in \mathcal{AT}(v) \colon \nexists b \in \mathcal{B} \colon b \in t$$

This means that there is no term that authenticates a voter *and* points to a ballot.

5.5 Application to RIES

The RIES protocol [12] deserves mentioning since it was used in two instances of the Dutch regional water management board elections. RIES, however, has trivial receipts. The purpose of its discussion here is as a more detailed example of how receipts are caught by the proposed formalism. The protocol uses a central administrator.

RIES works in three stages: pre-election, vote casting and post-election. In the pre-election stage, voters are registered, voter keys are handed out, the total number of voters is announced and for each voter, a ballot is constructed. The ballot is created using a keyed hash-function ($H(k, m)$ for key k and message m) and a keyless hash function ($G(m)$ for message m) and lists first the hashed voter-id ($G(H(k_v, \mathit{election\_id}))$, for a given voter v) and then in the specified order a hash of each candidate ($G(H(k_v, c))$, for $c \in \mathit{Can}$). Both hash-functions are publicly known. The ballots are published as a list B. Note that, due to the imposed order, it is known to which candidate each item on B belongs. This is captured by the function *can*, defined as $\mathit{can}(G(H(a, b))) = b$.

A voter v casts a vote for candidate c by sending her voter-id $H(k_v, \mathit{election\_id})$ and her vote $H(k_v, c)$ via an anonymous, encrypted channel to the administrator. After the elections close, a list of all received ballots $\mathcal{RB}$ is published. Tallying is done by, for each $r \in \mathcal{RB}$, counting one vote for $\mathit{can}(G(r))$.

Obviously, the voter-id together with the cast vote constitutes a receipt. However, due to the application of hash function G, $\mathcal{RB} \nsubseteq \mathcal{B}$. Since all parties know G, a voter can also hash his cast vote with G. Usually, it is assumed that the used hash-functions are collision-free and that all voter keys are different. In this case the hash of the cast vote would be sufficient to act as a receipt. Given a tuple of voter-id a and hashed cast vote $b = G(H(k_v, \mathit{can}))$, α and β can be defined as:

- $\alpha((a, b)) = a$ (or b)

– $\beta((a, G(H(k_v, can)))) = G(H(k_v, can))$.

Concluding In case of the RIES scheme an explicit construction of receipts invalidates the receipt-freeness of the protocol. For the other schemes, the absense of receipts was argued based on requirements on the receipt structure.

6 Conclusion and future work

The SUBTERMS lemma, as exploited above, establishes a method to prevent receipts: check that for all terms, the property of receipts does not hold. As seen in the analysis of voting protocols, this is done by assuring that all terms that can be used to authenticate a voter cannot be used to identify a ballot. The reverse is also true: all terms which can be used to identify a ballot cannot authenticate a voter.

The analysis of BT, SK95, HS and ALBD shows that these protocols are receipt-free. The example of FOO and the analysis of RIES demonstrate that these protocols are not receipt-free. The receipt in RIES remains valid indefinately. Although the keys used in FOO are revealed, this does not prevent construction of a receipt that remains valid by using timestamping.

A further line of research is to extend the intruder model to include the possibility of one or more authorities colluding with the intruder. This can be extended to the case where all authorities, but not the voter, cooperate with the intruder.

On a final note, the notion of receipt-freeness also has its applicability in online auctions, which we did not investigate for this paper. A formalisation of receipt-freeness for use in auction protocols is another possible line of future work.

References

1. R. Aditya, B. Lee, C. Boyd, and E. Dawson. An efficient mixnet-based voting scheme providing receipt-freeness. In S. Katsikas, J. Lopez, and G. Pernul, editors, *TrustBus2004*, volume 3184 of *LNCS*, pages 152–161. Springer, 2004.
2. O. Baudron, P.-A. Fouque, D. Pointcheval, J. Stern, and G. Poupard. Practical multi-candidate election system. In *PODC*, pages 274–283, Newport, Rhode Island, 2001.

3. J. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC'94*, pages 544–553. ACM, 1994. Montreal, Canada.
4. D. Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security and Privacy*, 02(1):38–47, 2004.
5. R. Cramer, M.K. Franklin, B. Schoenmakers, and M. Yung. Multi-autority secret-ballot elections with linear work. In U.M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 72–83. Springer, 1996.
6. R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In W.Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 103–118. Springer, 1997.
7. L.F. Cranor and R.K. Cytron. Sensus: A security-conscious electronic polling system for the internet. In *HICSS '97*, page 561. IEEE, 1997. Maui, Hawaii.
8. S. Delaune, S. Kremer, and M.D. Ryan. Receipt-freeness: Formal definition and fault attacks (extended abstract). In *Proceedings of the Workshop Frontiers in Electronic Elections (FEE 2005)*, Milan, Italy, September 2005.
9. A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In J. Seberry and Y. Zheng, editors, *ASIACRYPT'92*, volume 718 of *LNCS*, pages 244–251, 1992. Gold Coast, Australia.
10. D. Gritzalis. Principles and requirements for a secure e-voting system. *Computers & Security*, 21:539–556, 2002.
11. M. Hirt and K. Sako. Efficient receipt-free voting based on homomorphic encryption. In Bart Preneel, editor, *EUROCRYPT '00*, volume 1807 of *LNCS*, pages 539–556. Springer, May 2000.
12. E. Hubbers, B. Jacobs, and W. Pieters. RIES – internet voting in action. In R. Bilof, editor, *COMPSAC'05*, pages 417–424. IEEE, 2005.
13. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In U.M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 143–154. Springer, 1996.
14. A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In V. Atluri, S. De Capitani di Vimercati, and R. Dingledine, editors, *WPES'05*, pages 61–70. ACM, 2005.
15. C. Lambrinoudakis, D. Gritzalis, V. Tsoumas, M. Karyda, and S. Ikonomopoulos. Secure electronic voting: the current landscape. In D. Gritzalis, editor, *Secure Electronic Voting*, volume 7 of *Advances in Information Security*, pages 101–122. Kluwer, 2003.
16. B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. Providing receipt-freeness in mixnet-based voting protocols. In J.I. Lim and D.H. Lee, editors, *ICISC'03*, volume 2971 of *LNCS*, pages 245–258, 2003.
17. E. Magkos, M. Burmester, and V. Chrissikopoulos. Receipt-freeness in large-scale elections without untappable channels. In B. Schmid, K. Stanoevska-Slabeva, and V. Tschammer, editors, *I3E'01*, volume 202 of *IFIP conference proceedings*, pages 683–694. Kluwer, 2001.
18. K. Sako and J. Kilian. Receipt-free mix-type voting scheme – a practical solution to the implementation of a voting booth. In L.C. Guillou and J.-J. Quisquater, editors, *EUROCRYPT'95*, volume 921 of *LNCS*, pages 393–403. Springer, 1995.