Proving Coercion-Resistance of Scantegrity II

Ralf Küsters, Tomasz Truderung, and Andreas Vogt
University of Trier
{kuesters,truderun,vogt}@uni-trier.de

ABSTRACT. Recently, Küsters, Truderung, and Vogt have proposed a simple, yet widely applicable and formal definition of coercion-resistance for voting protocols, which allows to precisely quantify the level of coercion-resistance a protocol provides. In this paper, we use their definition to analyze coercion-resistance of Scantegrity II, one of the most prominent voting systems used in practice. We show that the level of coercion-resistance of Scantegrity II is as high as the one of an ideal voting system, under the assumption that the workstation and the PRNG used in Scantegrity II are honest.

1 Introduction

In the last few years many paper-based voting protocols have been proposed that are designed to achieve (various forms of) verifiability [8] and receipt-freeness/coercion-resistance [3], with protocols by Chaum [9], Neff [26], and Prêt à Voter [31, 11, 33, 32, 23] being the first such protocols; other protocols include Scratch&Vote [1], PunchScan [7, 28], ThreeBallot, VAV, Twin [30], Split Ballot [25], BingoVoting [4], a protocol by Riva and Ta-Shma [29], and Scantegrity II [10]. Scantegrity II is among the most successful protocols in that it has been used in practice for several elections. While this protocol is designed to provide coercion-resistance, this property has not been formally stated and analyzed for Scantegrity II so far. Providing such an analysis is the main goal of this paper.

Coercion-resistance is one of the most important and intricate security requirements for voting protocols [17, 27, 3]. Intuitively, a voting protocol is coercion-resistant if it prevents vote buying and voter coercion. Several definitions of coercion-resistance have been proposed in the literature (see, e.g., [17, 24, 12, 34, 15, 16, 14, 2, 19]), both based on cryptographic and symbolic models, where symbolic models take an idealized view on cryptography. However, in the cryptographic setting, only very few voting protocols have been analyzed rigorously w.r.t. coercion-resistance (see Section 5 for the related work). A major obstacle has been that the above definitions tend to be complex and limited in scope: They are often tailored to a very specific class of protocols or are too demanding; some otherwise reasonable protocols are deemed insecure or can be shown to be secure only under stronger assumptions or using stronger cryptographic primitives. Even some relatively simple voting protocols were out of the scope of most cryptographic definitions. The recently proposed definition by Küsters et al. [20] overcomes these problems and allows a precise and meaningful analysis of coercion-resistance of a wide range of voting protocols. (The paper by Küsters et al. contains a more detailed comparison of their definition with other definitions.) In this paper, we therefore use the definition by Küsters et al. to analyze Scantegrity II. Indeed other cryptographic definitions would not be applicable (see Section 4).

Contribution of this Paper. In this paper, we show that Scantegrity II provides an optimal level of coercion-resistance, i.e., the level of coercion-resistance Scantegrity II provides coincides with an ideal voting protocol, which only reveals the result of the election, but where a coercer can force voters to abstain from voting: We assume that the coercer can see the receipts of all voters, and hence, a coercer can force voters to abstain from voting (forced abstention). Our analysis also assumes that the workstation and the PRNG used in Scantegrity II are honest, i.e., these devices follow exactly the prescribed programs. This assumption is indeed necessary for Scantegrity II to provide coercion-resistance.

Structure of this Paper. In the next section, we restate the definition of coercion-resistance from [20]. In Section 3, we recall the Scantegrity II voting system and present a formal specification. The analysis of Scantegrity II is then presented in Section 4. Related work is discussed in Section 5. We conclude in Section 6. Some details of proofs are presented in the appendix.

2 Coercion-Resistance

In this section, we recapitulate the definition of coercion-resistance from [20]. We also recall from [20] the level of coercion-resistance an ideal voting protocol has, as this is used in Section 3. First, we introduce some notation and terminology.

2.1 Preliminaries

As usual, a function f from the natural numbers to the real numbers is *negligible* if for every $c > 0$ there exists ℓ_0 such that $f(\ell) \leq \frac{1}{\ell^c}$ for all $\ell > \ell_0$. The function f is *overwhelming* if the function $1 - f(\ell)$ is negligible. Let $\delta \in [0, 1]$. The function f is *δ-bounded* if f is bounded by δ plus a negligible function, i.e., for every $c > 0$ there exists ℓ_0 such that $f(\ell) \leq \delta + \frac{1}{\ell^c}$ for all $\ell > \ell_0$.

The modeling is based on a computational model similar to models for simulation-based security (see, e.g., [5, 18]), in which *interactive Turing machines (ITMs)* communicate via tapes. More concretely, Küsters et al. [20] use (a fragment of) the model proposed in [18]. In this model, at every time only one ITM is active, all other ITMs in the system wait to receive input. The active ITM may send a message to another ITM, which is then activated. If no message is sent, the so-called master ITM, of which a system has only one, is activated. The run of a system stops if, after being activated, the master ITM does not send a message. Every ITM can be activated an unbounded number of times and in every activation, an ITM may perform probabilistic polynomial-time computations in the length of the security parameter and the input received so far. However, only systems that have an overall polynomial runtime in the security parameter are considered. The details of the model are not essential for the rest of the paper. However, we fix some notation.

A *system* $\mathcal{S}$ of ITMs is a multi-set of ITMs, which we write as $\mathcal{S} = M_1 \parallel \cdots \parallel M_l$, where $M_1, \ldots, M_l$ are ITMs. If $\mathcal{S}_1$ and $\mathcal{S}_2$ are systems of ITMs, then $\mathcal{S}_1 \parallel \mathcal{S}_2$ is a system of ITMs, assuming that the systems are connectible w.r.t. their interfaces (external tapes). Clearly, a run of a system is uniquely determined by the random coins used by the ITMs in $\mathcal{S}$.

We assume that a system of ITMs has at most one ITM with a special output tape decision. For a system $\mathcal{S}$ of ITMs and a security parameter ℓ, we write $\mathsf{Pr}[S^{(\ell)} \mapsto 1]$ to denote the probability that $\mathcal{S}$ outputs 1 (on tape decision) in a run with security parameter ℓ.

A *property* of a system $\mathcal{S}$ is a subset of runs of $\mathcal{S}$. For a property γ of $\mathcal{S}$, we write $\mathsf{Pr}[\mathcal{S}^{(\ell)} \mapsto \gamma]$ to denote the probability that a run of $\mathcal{S}$, with security parameter ℓ, belongs to γ.

2.2 Voting Protocols

A *voting protocol* P specifies the programs (actions) carried out by honest voters and honest voting authorities, such as honest registration tellers, tallying tellers, bulletin boards, etc.

A voting protocol P, together with certain parameters, induces an *election system* $S = P(k, m, n, \vec{p})$. The parameters are as follows: k denotes the number of choices, an honest voter has in the election apart from abstaining from voting. In the simplest case, these choices can be the candidates the voter can vote for. Choices can also be preference lists of candidates, etc. In what follows, we often use the terms "candidate" and "choice" interchangeably. By m we denote the total number of voters and by n, with $n \leq m$, the number of honest voters. Honest voters follow the programs as specified in the protocol. The actions of dishonest voters and dishonest authorities are determined by the coercer, and hence, these participants can deviate from the protocol specification in arbitrary ways. We make the parameter n explicit since it is crucial for the level of coercion-resistance a system guarantees. One can also think of n as the minimum number of voters the coercer may not corrupt. The vector $\vec{p} = p_0, \ldots, p_k$ is a probability distribution on the possible choices, i.e., $p_0, \ldots, p_k \in [0, 1]$ and $\sum_{i=0}^{k} p_i = 1$. Honest voters will abstain from voting with probability p_0 and vote for candidate i with probability p_i, $1 \leq i \leq k$. We make this distribution explicit, because it is realistic to assume that the coercer knows this distribution (e.g., from opinion polls), and hence, uses it in his strategy, and because the specific distribution is crucial for the level of coercion-resistance of a system.

An election system $S = P(k, m, n, \vec{p})$ specifies (sets of) ITMs for all participants, honest voters and authorities, the coercer, subsuming dishonest voters and dishonest authorities, and the coerced voter: (i) There are ITMs, say $S_1, \ldots, S_l$, for all honest voting authorities. These ITMs run the programs as specified by the voting protocol. (ii) There is an ITM S_{v_i}, $i \in \{1, \ldots, n\}$ for each of the honest voters. Every such ITM first makes a choice according to the probability distribution $\vec{p}$. Then, if the choice is not to abstain, it runs the program for honest voters according to the protocol specification with the candidate chosen before. (iii) The coercer is described by a set C_S of ITMs. This set contains all (probabilistic polynomial-time) ITMs, and hence, all possible coercion strategies the coercer can carry out. These ITMs are only constrained in their interface to the rest of the system. Typically, the ITMs can directly use the interface of dishonest voters and authorities. They can also communicate with the coerced voter and have access to all public information (e.g., bulletin boards) and possibly (certain parts of) the network. The precise interface of the ITMs in C_S depends on the specific protocol and the assumptions on the power of the coercer. (iv) Similarly, the coerced voter is described by a set V_S of ITMs. Again, this set contains all (probabilistic polynomial-time) ITMs. This set represents all the possible programs the coercer can ask the coerced voter

to run as well as all counter-strategies the coerced voter can run (see Section 2.3 for more explanation). The interface of these ITMs is typically the interface of an honest voter plus an interface for communication with the coercer. In particular, the set V_S contains what we call a *dummy strategy* dum which simply forwards all the messages between the coercer and the interface the coerced voter has as an honest voter.

Given an election system $S = P(k, m, n, \vec{p})$, we denote by e_S the system of ITMs containing all honest participants, i.e., $\mathsf{e}_S = (S_{\mathsf{v}_1} \parallel \ldots \parallel S_{\mathsf{v}_n} \parallel S_1 \parallel \ldots \parallel S_l)$, where, as explained above, $S_{\mathsf{v}_1} \parallel \ldots \parallel S_{\mathsf{v}_n}$ are the ITMs modeling honest voters and $S_1 \parallel \ldots \parallel S_l$ are the honest authorities. A system $(c \parallel v \parallel \mathsf{e}_S)$ of ITMs, with $c \in C_S$ and $v \in V_S$, is called an *instance of* S. We often implicitly assume a scheduler (modeled as an ITM) to be part of a system. Its role is to make sure that all components of the system are scheduled in a fair way, e.g., all voters get a chance to vote. For simplicity of notation, we do not state the scheduler explicitly. We define a *run of* S to be a run of some instance of S.

For an election system $S = P(k, m, n, \vec{p})$, we denote by $\Omega_1 = \{0, \ldots, k\}^n$ the set of all possible combinations of choices made by the honest voters, with the corresponding probability distribution μ_1 derived from $\vec{p} = p_0, p_1, \ldots, p_k$. All other random bits used by ITMs in an instance of S, i.e., all other random bits used by honest voters as well as all random bits used by honest authorities, the coercer, and the coerced voter, are uniformly distributed. We take μ_2 to be this distribution over the space Ω_2 of random bits. Formally, this distribution depends on the security parameter. We can, however, safely ignore it in the notation without causing confusion. We define $\Omega = \Omega_1 \times \Omega_2$ and $\mu = \mu_1 \times \mu_2$, i.e., μ is the product distribution obtained from μ_1 and μ_2. For an event φ, we will write $\mathsf{Pr}_{\omega_1,\omega_2 \leftarrow \Omega}[\varphi]$, $\mathsf{Pr}_{\omega_1,\omega_2}[\varphi]$, or simply $\mathsf{Pr}[\varphi]$ to denote the probability $\mu(\{(\omega_1, \omega_2) \in \Omega : \varphi(\omega_1, \omega_2)\})$. Similarly, $\mathsf{Pr}_{\omega_1 \leftarrow \Omega_1}[\varphi]$ or simply $\mathsf{Pr}_{\omega_1}[\varphi]$ will stand for $\mu_1(\{\omega_1 \in \Omega_1 : \varphi(\omega_1)\})$; analogously for $\mathsf{Pr}_{\omega_2 \leftarrow \Omega_2}[\varphi]$.

A *property* of an election system $S = P(k, m, n, \vec{p})$ is defined to be a class γ of properties containing one property γ_T for each instance T of S. We will write $\mathsf{Pr}\,[T \mapsto \gamma]$ to denote the probability $\mathsf{Pr}\,[T \mapsto \gamma_T]$.

2.3 Defining Coercion-Resistance

We can now present the definition of coercion-resistance from [20]. Here, we concentrate on the case that only a single voter is coerced. As discussed in [20], the definition in fact also covers the case of multiple coerced voters and our results also hold for the case of multiple coerced voters. In what follows, let P be a voting protocol and $S = P(k, m, n, \vec{p})$ be an election system for P.

The definition of coercion-resistance assumes that a coerced voter has a certain goal γ that she would try to achieve in absence of coercion. Formally, γ is a property of S. If, for example, γ is supposed to express that the coerced voter wants to vote for a certain candidate, then γ would contain all runs in which the coerced voter voted for this candidate and this vote is in fact counted. We note that in some cases such a goal cannot be achieved, e.g., in case ballots are sent over an unreliable channel or an election authority misbehaves in an observable way (e.g., fails to provide a valid proof of compliance) and as a result the election process is stopped. A more realistic goal γ would then be that the coerced voter successfully votes for a certain candidate, provided that the voters ballot is delivered in

time and the election authority did not misbehave in an observable way.

In the definition of coercion-resistance the coercer demands full control over the voting interface of the coerced voter, i.e., the coercer wants the coerced voter to run the dummy strategy dum (that simply forwards all the messages between the coercer and the interface the coerced voter has as an honest voter) instead of the program an honest voter would run. If the coerced voter runs dum the coercer can effectively vote on behalf of the coerced voter or decide to abstain from voting. Of course, the coercer is not bound to follow the specified voting procedure; he can perform arbitrary coercion strategies: The coercer can send fake messages and depend his decisions on the information he has gathered so far. The intention of the coercer might even be to merely test whether the coerced voter follows his instructions, e.g., to find out whether this voter is "reliable", and hence, is a good candidate for coercion in later elections. Also, the coercer is not necessarily bound to use only the interface of the coerced voter in his coercion strategy. There may be other ways to vote on behalf of the coerced voter. However, for a protocol to be coercion-resistant, there will always be at least one step in the protocol that the coercer cannot do all by himself, e.g., register, perform operations on a security token, or vote in a voting booth. For such actions, the coercer has to consult the coerced voter.

Now, for a protocol to be coercion-resistant the definition requires that there exists a *counter-strategy* $\tilde{v}$ that the coerced voter can run instead of dum such that (i) the coerced voter achieves her own goal γ, with overwhelming probability, by running $\tilde{v}$ and (ii) the coercer is not able to distinguish whether the coerced voter runs dum or $\tilde{v}$. More precisely, the ability of the coercer to distinguish between these two cases is measured. Hence, $\tilde{v}$ has to simulate dum while at the same time make sure that γ is achieved. If such a counter-strategy exists, then it indeed does not make sense for the coercer to try to influence a voter in any way, e.g., by offering money or threatening the voter, at least not from a technical point of view:* Even if the coerced voter tries to sell her vote, the coercer is not able to tell whether she is actually following the coercer's instructions or just trying to achieve her own goal by running the counter-strategy. For the same reason, the coerced voter is safe, even if she wants to achieve her goal and therefore runs the counter-strategy.

The formal definition of coercion-resistance is the following:

DEFINITION 1. Let P be a protocol and $S = P(k, m, n, \vec{p})$ be an election system. Let $\delta \in [0, 1]$, and γ be a property of S. The system S is *δ-coercion-resistant w.r.t. γ*, if there exists $\tilde{v} \in V_S$ such that for all $c \in C_S$ we have:

(i) $\mathsf{Pr}[(c \parallel \tilde{v} \parallel \mathsf{e}_S)^{(\ell)} \mapsto \gamma]$ is overwhelming, as a function of the security parameter.

(ii) $\mathsf{Pr}[(c \parallel \mathsf{dum} \parallel \mathsf{e}_S)^{(\ell)} \mapsto 1] - \mathsf{Pr}[(c \parallel \tilde{v} \parallel \mathsf{e}_S)^{(\ell)} \mapsto 1]$ is δ-bounded, as a function of the security parameter.

Condition (i) says that by running the counter-strategy $\tilde{v}$ the coerced voter achieves her goal with overwhelming probability, no matter which coercion-strategy the coercer performs.

Condition (ii) captures that the coercer is unable to distinguish whether the coerced voter runs dum or $\tilde{v}$. More precisely, the coercer accepts a run (i.e., outputs 1 on tape decision)

*Of course, voters can be influenced psychologically.

with almost the same probability no matter whether the coerced voter performs dum or $\tilde{v}$, where "almost the same" is formalized as δ-bounded, for some reasonably small δ. Remark that requiring the difference in (ii) to be negligible instead of δ-bounded would be too strict: The difference, even for an ideal protocol, which merely reveals the result of the election, does not decrease with an increasing security parameter, but may depend on the number of choices, the distribution $\vec{p}$ on these choices, and the number of honest voters: Imagine for example that a candidate did not get any vote in an election. Now, if the coercer asked the coerced voter to vote for this candidate, it is clear that the coerced voter did not follow the coercer's instruction. The probability for this to happen is non-negligible and depends on $\vec{p}$ and the number of voters; the larger the number of voters is, the more likely it is that a candidate gets a vote. In fact, in Scantegrity II, δ will depend on the number of candidates, $\vec{p}$, and the number of honest voters. Such a δ provides for a precise measure of the level of coercion-resistance, which is of practical relevance: It might, for example, indicate that a voting protocols does not have a sufficient level of coercion-resistance if the number of voters is below a certain threshold, the number of candidates is too big, or the probability distribution of the choices (e.g., according to opinion polls) is problematic in terms of coercion-resistance (see also [20]).

2.4 Level of Coercion-Resistance of the Ideal Protocol

We briefly recall from [20] the level of coercion resistance of the *ideal protocol*, which just collects the inputs of the voters and outputs the correct result of the election.

We consider the goal γ_i of the coerced voter, for $i \in \{1, \ldots, k\}$, defined as follows: γ_i is satisfied in a run, if whenever the coerced voter has sent her candidate to the voting authority, she has successfully voted for the i-th candidate. This implies that if the coerced voter is not instructed by the coercer to vote, i.e., the coercer does not send his candidate to the coerced voter, and hence, effectively wants the coerced voter to abstain from voting, the coerced voter does not have to vote in order to fulfill γ_i. In other words, by γ_i abstention attacks are not prevented.

For the ideal protocol, one could consider abstention to be a goal of the coerced voter. But this goal cannot be achieved in most practical protocols in which a voter is given a receipt, as in Scantegrity, as such receipts can be used by the coercer to verify that the voter has actually voted.

The stronger and simpler goal γ_i' which requires the coerced voter to vote for i, even if the coercer wants the coerced voter to abstain is too strong for Scantegrity II, since we assume that the coercer can see all receipts of voters who voted, the coerced voter can be forced to abstain from voting. For reasons of uniformity, we therefore restrict ourselves to the goal γ_i.

Since we assume that the coercer knows the votes of dishonest voters, he can simply subtract these votes from the final result and obtain what we will call the *pure result* of the election. The pure result only depends on the votes by the n honest voters and the coerced voter. Hence, a pure result is a tuple $\vec{r} = (r_0, \ldots, r_k)$ of non-negative integers such that $r_0 + \cdots + r_k = n + 1$, where r_i, for $i \in \{1, \ldots, k\}$, is the number of votes for the i-th candidate and r_0 denotes the number of voters who abstained from voting. The coercer has to base

his decision—accept or reject—solely on such a pure result $\vec{r}$. We will denote the set of pure results by *Res*.

In the following definition of constants $\delta_{min}(k, n, \vec{p})$ expressing the level of coercion resistance of the ideal protocol, we will use the probability $A^i_{\vec{r}}$ that the choices made by the honest voters and the coerced voter yield the pure result $\vec{r} = (r_0, \ldots, r_k)$, given that the coerced voter votes for the i-th candidate. Let $r'_j = r_j$ for $j \neq i$ and $r'_j = r_i - 1$. It is easy to see that

$$A^i_{\vec{r}} = \frac{n!}{r'_0! \cdots r'_k!} \cdot p_0^{r'_0} \cdots p_k^{r'_k} = \frac{n!}{r_0! \cdots r_k!} \cdot p_0^{r_0} \cdots p_k^{r_k} \cdot \frac{r_i}{p_i}.$$

The intuition behind the definition of $\delta_{min}(k, n, \vec{p})$ is the following: If the coercer wants the coerced voter to vote for j and the coerced voter wants to vote for i, for some $i, j \in \{1, \ldots, k\}$, then, as it is shown in [20], the best strategy of the coercer to distinguish whether the coerced voter has voted for j or i is to accept a run if the pure result $\vec{r}$ of the election in this run is such that $A^i_{\vec{r}} \leq A^j_{\vec{r}}$. Let $M^*_{i,j} = \{\vec{r} \in Res : A^i_{\vec{r}} \leq A^j_{\vec{r}}\}$ be the set of those results, for which—according to his best strategy—the coercer should accept the run. Now, we are ready to define the constant δ^i_{min}, which is shown to be optimal in [20]:

$$\delta^i_{min}(n, k, \vec{p}) = \max_{j \in \{1,\ldots,k\}} \sum_{\vec{r} \in M^*_{i,j}} (A^j_{\vec{r}} - A^i_{\vec{r}}).$$

In the definition of this constant, all possible candidates $1, \ldots, k$ that the coercer can wish the coerced voter to vote for are taken into account, excluding abstention, as in this case the counter-strategy coincides with the dummy strategy. We take the worst possible case, i.e., the index j for which the sum in the expression above is maximal.

Figure 2.4 shows $\delta = \delta_{min}$ for some selected cases, see [20] for more values. The calculations show that the level of coercion-resistance heavily depends on the number of honest voters, the number of candidates, and the probability distribution on the choices.

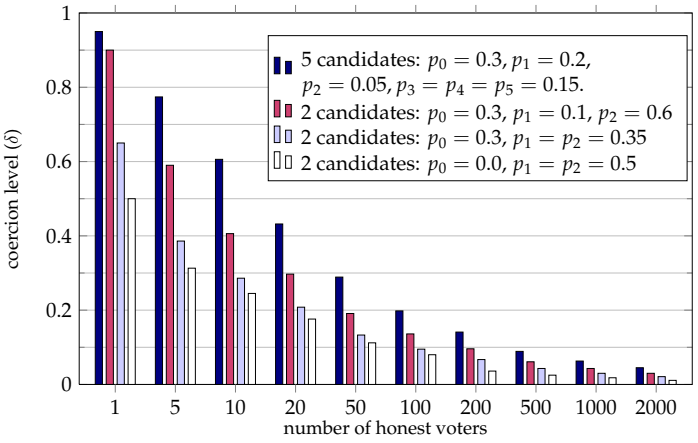

Figure 1: Level of coercion-resistance (δ) for the ideal protocol. The goal of the coerced voter is, in each case, to vote for candidate 1.

3 Scantegrity II

In this section, we first give an informal description of the Scantegrity II system [10]. We then provide a formal specification as an election system, as introduced in Section 2.2.

3.1 Informal Description

We shortly describe the Scantegrity II system as proposed in [10], which we will denote by $\mathsf{P}_{\mathsf{Sct}}$.

In addition to the voters, the participants in this system are the following: (i) A *workstation (WSt)*, which is the main component in the voting process. The workstation uses a *bulletin board*, that everybody has read-access to, for broadcasting messages. (ii) A *pseudo random number generator (PRNG)*, which is an independent source of randomness and which is connected to the workstation. (iii) Some number of *auditors* $aud_1, \ldots, aud_t$ who will contribute randomness in a distributed way used for randomized partial checking (RPC). (iv) A number of clerks $cl_1, \ldots, cl_r$ that share a secret seed as input to the workstation.

The election consists of three phases described below: initialization, voting, and tallying.

Initialization phase. In this phase, the election officials $cl_1, \ldots, cl_r$ secret-share a seed to a pseudo-random number generator and input this seed to a workstation that creates so called P-table that is, $k \cdot s$ pseudo-random confirmation codes $\{c_i^j\}_{\substack{i=1,\ldots,s \\ j=1,\ldots,k}}$ (where s is at least twice as high as the number of voters, and k is the number of candidates). This table will never be published. For every row $i = 1, \ldots, s$ in the P-table, a ballot is printed with the serial number i and the confirmation codes c_i^j written in invisible ink next to the respective candidate name j, $(j = 1, \ldots, k)$. The workstation also creates a Q-table $\{c_i^{\pi_i^{-1}(j)}\}_{\substack{i=1,\ldots,s \\ j=1,\ldots,k}}$ obtained from the P-table by permuting cells with a random permutation π_i in each row i. Next, the, so called, S-table of size $k \cdot s$ is created. This table is initially empty and will be used to mark positions corresponding to the candidates chosen by the voters. Further, another table, the R-table is created. The R-table consists of two columns, one column for Q-pointers p_k^Q and one column for so called S-pointers p_k^S, for $k = 1, \ldots, (s \cdot k)$. These pointers are just indices of the respective table and are (supposed to be) pseudo-randomly generated in a way that for every cell $(i, j) \in \{1, \ldots, s\} \times \{1, \ldots, k\}$ of the Q-table, there is exactly one Q-pointer $p_k^Q = (i, j)$ and, similarly, for every cell $(i, j) \in \{1, \ldots, s\} \times \{1, \ldots, k\}$ of the S table, there exists exactly one S-pointer p_k^S pointing to that cell. Moreover, for every k, if $p_k^Q = (i, j)$ and $p_k^S = (i', j')$, then $j' = \pi_i^{-1}(j)$, i.e. the S-pointer next to a Q-pointer pointing to a cell with confirmation code $c_i^{\pi_i^{-1}(j)}$ for candidate $j' = \pi_i^{-1}(j)$, points to a cell in the j'-th column of the S-table. The workstation commits on every value in the Q- and R-table and publishes these commitments.

Voting phase. In this phase the voter asks for either one or two ballots and a decoder pen which she can use to reveal the codes written in invisible ink. If she takes two ballots, she chooses one ballot to audit, which means that all codes are revealed and the workstation has to open all the corresponding commitments in the Q- and R-table. Because of this check, a

workstation that cheats by producing wrong ballots or wrong tables is detected with high probability.

The other ballot is used for voting: The voter unveils exactly the code next to the candidate she wants to vote for and notes down that code somewhere. Unveiling the code darkens the space next to the candidate, which can be detected by a conventional optical scanner. The choice of the voter, together with the ballot serial number is recorded.

Tallying phase. In this last phase, the election officials publish a list of all voters that voted and the tally given by the optical scanners. Furthermore, the P-table is used to reconstruct, for every voter, the confirmation code from the voters choice j and the ballot serial number i. The commitment to that code in the Q-table is then opened, i.e. the commitment on the value of the cell (i, j') of the Q-table, with $j' = \pi_i(j)$. Further, the corresponding cells in the R- and S-table are flagged: the election officials flag (publish) the index k of the R-table such that $p_k^Q = (i, j')$ and the index p_k^S of the S-table. In the end, for each row k of the R-table, either the commitment on the Q-pointer p_k^Q or on the S-pointer p_k^S is opened, depending on a publicly verifiable coin flip.

Now, the result can be easily computed from the publicly available information: the number of votes for candidate i is the number of flagged cells in the i-th column of the S-table. As the workstation does not know a priori the result of the coin flips, if it tries to flag several different cells, this is detected with high probability.

3.2 Modeling and Security Assumptions

The formal specification of Scantegrity II as an election system in the sense of Section 2.2 is straightforward. However, we highlight some modelling issues and, most importantly, state our security assumptions.

Voting Authorities. We assume that the workstation, the pseudo random number generator and at least one clerk cl_i $(i = 1, \ldots, r)$ is honest; all others may be dishonest. This assumption, that may seem to be rather strong, is in fact necessary for the Scantegrity II system to be coercion-resistant:

- Clearly, as the workstation learns the votes of the voters, it is necessary that the workstation is honest.
- If the PRNG were dishonest, it could leak the information which code it produces for which candidate to the coercer which would give the coercer the possibility to check a voters vote by demanding the receipt.
- If all clerks are dishonest, they may leak the secret seed that is used as input for the PRNG, and hence, it may be clear for the coercer which code is produced for which candidate.

Remarks: We could drop the assumption that the PRNG and one clerk is honest, if the workstation secretly permuted the codes such that only the (honest) workstation knows, which code belongs to which candidate.

Honest voters. As described in Section 2.2, an honest voter first makes a choice according to the probability distribution $\vec{p}$. If the choice is to abstain from voting, she abstains, otherwise follows the procedure described for the voting phase. After the voting phase is finished, an honest voter reveals her (paper) receipt, e.g., mails it to an organization to ask it to verify the correctness of the voting process w.r.t. her receipt or to publish it on some bulletin board. In particular, the coercer will get to see all receipts of honest voters, and hence, know whether a voter voted or not. The assumption that the paper receipts are revealed after the voting phase is reasonable. Also, the (presumably small) fraction of honest voters for which the coercer manages to get hold of the receipt earlier, could be considered to be dishonest. In any case, the assumption helps in the proof and we believe that our results also hold without this assumption.

The coerced voter. A coerced voter, running the dummy strategy or emulating it by running a counter-strategy, can communicate with the coercer and send her candidate on an untappable channel to the voting authority.

The coercer. The coercer can freely communicate with dishonest participants (voters and authorities) as well as with the coerced voter; in fact, dishonest participants are considered to be part of the coercer program. In a run of the system the coercer can see the following: (v1) his random coins, (v2) all messages published by the workstation, both in the initialization phase and the tallying phase, (v3) receipts of all honest voters, as already explained above, and (v4) the messages received from the coerced voter (and dishonest parties), including the receipt of the coerced voter. However, the coercer cannot directly see the information the coerced voter obtains in the voting booth. In particular, the coerced voter can lie about what she sees and does in the voting booth, such as the candidate she picked. So, while talking with the coercer on the phone would be allowed, taking pictures or videos should be prohibited (unless they can be manipulated on-the-fly, which, however, is unrealistic).

4 Analysis of Scantegrity II

In this section, we show that the Scantegrity II system, as specified in Section 3, enjoys the same level of coercion-resistance as the ideal protocol.

4.1 The Main Result

We prove the following theorem, where we will consider goals γ_i of the coerced voter, for $i \in \{1, \ldots, k\}$, as described in Section 2.3.

THEOREM 2. *Let $S = \mathsf{P}_{\mathsf{Sct}}(k, m, n, \vec{p})$. Then S is δ-coercion-resistant with respect to γ_i, where $\delta = \delta^i_{min}(n, k, \vec{p})$.*

Because, as it is shown in [20] and as we have already mentioned, $\delta^i_{min}(n, k, \vec{p})$ is the level of coercion-resistance the ideal voting protocol (i.e. the protocol that just outputs correctly the tally) achieves, Theorem 2 states the best possible δ for this protocol; for any $\delta' < \delta$ the system $\mathsf{P}_{\mathsf{Sct}}(k, m, n, \vec{p})$ is not δ'-coercion-resistant w.r.t. γ_i.

Remarks: None of the other definitions of coercion resistance proposed in the literature is suitable for the analysis of Scantegrity II (see also Section 5):

- Juels et al. [17] propose a definition specifically tailored towards voting in a public-key setting, with protocols having a specific structure. Scantegrity II does not fall into the class of protocols considered by Juels et al. Similarly for the definition proposed by Gardner et al. [15], which is also tailored to the protocol considered by the authors.
- The definition by Moran and Naor [24] is simulation-based, and hence, it suffers from the commitment problem. Due to the intensive use of commitments in Scantegrity II, the definition rejects the protocol as insecure, only due to the commitment problem. The definition might be possibly used to prove coercion-resistance of the protocol, if one would weaken the security assumptions, namely assume that *all* auditors are honest. In this case a simulator can simulate these auditors, which allows it to fake the proof that the R-table connects correctly the Q- and S-table, as it "knows" the challenges.
- The definition by Teague et al. [34] is intended to be used for protocols that have been reduced to ideal functionalities. As the authors suggest, this definition should be combined with a simulation-based definition, such as the one by Moran and Naor, and hence, it suffers from the same problem as that definition.

4.2 Proof of the Main Result

The remainder of this section is devoted to the proof of Theorem 2. First, we define the counter-strategy $\tilde{v}$ of the coerced voter: $\tilde{v}$ coincides with the dummy strategy dum, with the exception that $\tilde{v}$ votes for candidate i, i.e., the coerced voter reveals the code next to candidate i, if the coercer instructs the coerced voter to vote for some candidate j.

Clearly, if the coerced voter runs the counter-strategy $\tilde{v}$, then condition (i) of Definition 1 is satisfied for every $c \in C_S$. Note that if the coercer does not instruct the coerced voter to vote for some candidate j (abstention attack), then following the counter-strategy the coerced voter abstains from voting, which is in accordance with γ_i.

It remains to prove condition (ii) of Definition 1. For this purpose, let us fix a program c of the coercer. We need to prove that $\mathsf{Pr}[T \mapsto 1] - \mathsf{Pr}[\tilde{T} \mapsto 1] \leq \delta$, where $T = (\mathsf{dum} \parallel c \parallel \mathsf{e}_S)$ and $\tilde{T} = (\tilde{v} \parallel c \parallel \mathsf{e}_S)$. The rest of the proof consists of two parts, a cryptographic and a combinatorial part. The cryptographic part is Lemma 3. Using Lemma 3, the combinatorial part is merely a reduction to the ideal case (see Section 2.4); it does not have to be redone.

As introduced in Section 2.2, by $\omega_1 \in \Omega_1$ we denote a vector of choices made by the honest voters and by $\omega_2 \in \Omega_2$ we denote all the remaining random coins of a system. We denote by ρ a view of the coercer, as described in Section 3.2, (v1)–(v4). We use the notion of a *pure result* $\vec{r} = (r_0, \ldots, r_k)$ which is the result of the election not including the dishonest voters. In particular, it holds that $r_0 + \cdots + r_k = n + 1$ and the coercer can compute this result from his view, by subtracting the votes of dishonest voters from the result of the election. We will denote the pure result determined by a view ρ of the coercer by $\mathrm{res}(\rho)$. A pure result determined by ω_1 and the choice j of the coerced voter will be denoted by $\mathrm{res}(\omega_1, j)$.

As mentioned before, the coercer can derive from his view which voters abstained from voting. Given a view ρ of the coercer, we denote by $\text{abst}(\rho)$ the set of voters who abstained from voting, among the honest voters and the coerced voter; the number of such voters is referred to by $r_0(\rho) = |\text{abst}(\rho)|$. As this set/number depends only on ω_1, we will sometimes write $\text{abst}(\omega_1)/r_0(\omega_1)$.

For a coercer view ρ in a run of the system, we denote by $f(\rho)$ the candidate the coercer wants the coerced voter to vote for; if the coercer does not instruct the coerced voter to vote, then $f(\rho)$ is undefined. Note that the coercer has to provide the coerced voter with $f(\rho)$ before the end of the election. Consequently, all messages the coercer has seen up to this point only depend on ω_2 and are independent of the choices made by honest voters, which are determined by ω_1. Therefore, we sometimes write $f(\omega_2)$ for the candidate the coercer wants the coerced voter to vote for in runs that use the random coins ω_2.

For a coercer view ρ, let φ_ρ be a predicate over Ω_1 such that $\varphi_\rho(\omega_1)$ holds iff $\text{res}(\omega_1, f(\rho)) = \text{res}(\rho)$ and $\text{abst}(\omega_1) = \text{abst}(\rho)$, i.e., the choices ω_1 of the honests voter are consistent with the view of the coercer, as far as the result of the election and the set of abstaining voters is concerned, in case the coerced voter runs the dummy strategy. Analogously, for the counter-strategy, we define that $\tilde{\varphi}_\rho(\omega_1)$ holds iff $\text{res}(\omega_1, i) = \text{res}(\rho)$ and $\text{abst}(\omega_1) = \text{abst}(\rho)$.

For a coercer view ρ, by $T(\omega_1, \omega_2) \mapsto \rho$, or simply $T \mapsto \rho$, we denote the fact that the system T, when run with ω_1, ω_2, produces the view ρ (similarly for $\tilde{T}$). For a set M of views, we write $T(\omega_1, \omega_2) \mapsto M$ if $T(\omega_1, \omega_2) \mapsto \rho$ for some $\rho \in M$.

The following lemma is the key fact used in the proof of Theorem 2 (see Appendix A for the proof). It constitutes the cryptographic part of the proof of Theorem 2.

LEMMA 3. *Let ρ be a coercer view such that $f(\rho)$ is defined. Let ω_1^ρ and $\tilde{\omega}_1^\rho$ be some fixed elements of Ω_1 such that $\varphi_\rho(\omega_1^\rho)$ and $\tilde{\varphi}_\rho(\tilde{\omega}_1^\rho)$, respectively. Then, the following equations hold true:*

$$\mathsf{Pr}[T \mapsto \rho] = \mathsf{Pr}_{\omega_1}[\varphi_\rho(\omega_1)] \cdot \mathsf{Pr}_{\omega_2}[T(\omega_1^\rho, \omega_2) \mapsto \rho] \tag{1}$$

$$\mathsf{Pr}[\tilde{T} \mapsto \rho] = \mathsf{Pr}_{\omega_1}[\tilde{\varphi}_\rho(\omega_1)] \cdot \mathsf{Pr}_{\omega_2}[\tilde{T}(\tilde{\omega}_1^\rho, \omega_2) \mapsto \rho] \tag{2}$$

$$\mathsf{Pr}_{\omega_2}[T(\omega_1^\rho, \omega_2) \mapsto \rho] = \mathsf{Pr}_{\omega_2}[\tilde{T}(\tilde{\omega}_1^\rho, \omega_2) \mapsto \rho] \ . \tag{3}$$

Intuitively, the lemma says that the view of the coercer is information-theoretically independent of the choices of honest voters and the coerced voter as long as these choices are consistent with the result of the election given in this view.

Now, using this lemma, we can link the level of coercion-resistance the Scantegrity II system provides with the optimal bound δ_{min} established in [20]. Clearly, if $f(\rho)$ is defined, we have:

$$\begin{aligned}\mathsf{Pr}_{\omega_1}[\varphi_\rho(\omega_1)] &= \mathsf{Pr}_{\omega_1}[\text{res}(\omega_1, f(\rho)) = \text{res}(\rho)] \cdot \mathsf{Pr}_{\omega_1}[\text{abst}(\omega_1) = \text{abst}(\rho) \mid \text{res}(\omega_1, f(\rho)) = \text{res}(\rho)] \\ &= A^{f(\rho)}_{\text{res}(\rho)} \cdot \mathsf{Pr}_{\omega_1}[\text{abst}(\omega_1) = \text{abst}(\rho) \mid \text{res}(\omega_1, f(\rho)) = \text{res}(\rho)]\end{aligned}$$

and similarly

$$\mathsf{Pr}_{\omega_1}[\tilde{\varphi}_\rho(\omega_1)] = A^{i}_{\text{res}(\rho)} \cdot \mathsf{Pr}_{\omega_1}[\text{abst}(\omega_1) = \text{abst}(\rho) \mid \text{res}(\omega_1, i) = \text{res}(\rho)].$$

Furthermore, we have

$$\begin{aligned}\mathsf{Pr}_{\omega_1}[\mathsf{abst}(\omega_1) = \mathsf{abst}(\rho) \mid \mathsf{res}(\omega_1, f(\rho)) = \mathsf{res}(\rho)] = \\ = \mathsf{Pr}_{\omega_1}[\mathsf{abst}(\omega_1) = \mathsf{abst}(\rho) \mid r_0(\omega_1) = r_0(\rho)] \\ = \mathsf{Pr}_{\omega_1}[\mathsf{abst}(\omega_1) = \mathsf{abst}(\rho) \mid \mathsf{res}(\omega_1, i) = \mathsf{res}(\rho)],\end{aligned}$$

as the set of abstaining voters depends only on the number of abstaining voters.

Together with Lemma 3, we immediately obtain for all ω_1^ρ with $\varphi_\rho(\omega_1^\rho)$:

$$\mathsf{Pr}[T \mapsto \rho] - \mathsf{Pr}[\tilde{T} \mapsto \rho] = \left(A^{f(\rho)}_{\mathrm{res}(\rho)} - A^{i}_{\mathrm{res}(\rho)}\right) \cdot \mathsf{Pr}_{\omega_2}[T(\omega_1^\rho, \omega_2) \mapsto \rho] \cdot \mathsf{Pr}_{\omega_1}[\mathsf{abst}(\omega_1) = \mathsf{abst}(\rho) \mid r_0(\omega_1) = r_0(\rho)$$

Note that we do not assume here that there exists $\tilde{\omega}_1^\rho$ such that $\tilde{\varphi}_\rho(\tilde{\omega}_1^\rho)$. In this special case we have $A^i_{\mathrm{res}(\rho)} = 0$ and $\mathsf{Pr}[\tilde{T} \mapsto \rho] = 0$.

Let M be the set of views that are accepted by the program c of the coercer, i.e., for which the coercer outputs 1. In what follows, let j range over the set of candidate names $\{1, \ldots, k\}$, $\vec{r} = (r_0, \ldots, r_k)$ over all the pure results and S over all subsets of $\{1, \ldots, n\}$. Let $M_j^{\vec{r},S} = \{\rho \in M : f(\rho) = j,\ \mathsf{abst}(\rho) = S \text{ and } \mathsf{res}(\rho) = \vec{r}\}$. In the following, we can assume without loss of generality that M contains only views ρ such that $\mathsf{Pr}[T \mapsto \rho] > 0$ (this is because, by removing from M views that fail to satisfy this condition, we only make the expression $\mathsf{Pr}[T \mapsto 1] - \mathsf{Pr}[\tilde{T} \mapsto 1]$ bigger). Therefore, for all j, $\vec{r}$, and S such that $M_j^{\vec{r},S}$ is non-empty, there exists $\omega_1^{j,\vec{r},S}$ such that $\mathsf{res}(\omega_1^{j,\vec{r},S}, j) = \vec{r}$ and $\mathsf{abst}(\omega_1^{j,\vec{r},S}) = S$. Clearly, we have $\varphi_\rho(\omega_1^{j,\vec{r},S})$ for all $\rho \in M_j^{\vec{r},S}$. We have

$$\begin{aligned}\Phi &= \mathsf{Pr}[T \mapsto 1] - \mathsf{Pr}[\tilde{T} \mapsto 1] \\ &= \mathsf{Pr}[T \mapsto M] - \mathsf{Pr}[\tilde{T} \mapsto M] \\ &= \sum_j \sum_{\vec{r}} \sum_S \sum_{\rho \in M_j^{\vec{r},S}} \left(\mathsf{Pr}[T \mapsto \rho] - \mathsf{Pr}[\tilde{T} \mapsto \rho]\right) \\ &= \sum_j \sum_{\vec{r}} \sum_S \sum_{\rho \in M_j^{\vec{r},S}} (A^j_{\vec{r}} - A^i_{\vec{r}}) \cdot \mathsf{Pr}_{\omega_2}[T(\omega_1^{j,\vec{r},S}, \omega_2) \mapsto \rho] \cdot \mathsf{Pr}_{\omega_1}[\mathsf{abst}(\omega_1) = S | r_0(\omega_1) = r_0] \\ &= \sum_j \sum_{\vec{r}} (A^j_r - A^i_r) \sum_S \sum_{\rho \in M_j^{\vec{r},S}} \mathsf{Pr}_{\omega_2}[T(\omega_1^{j,\vec{r},S}, \omega_2) \mapsto \rho] \cdot \mathsf{Pr}_{\omega_1}[\mathsf{abst}(\omega_1) = S | r_0(\omega_1) = r_0].\end{aligned}$$

Let $M^*_{i,j} = \{\vec{r} : A^j_{\vec{r}} \geq A^i_{\vec{r}}\}$. Then, we obtain

$$\Phi \leq \sum_j \sum_{\vec{r} \in M^*_{i,j}} (A^j_{\vec{r}} - A^i_{\vec{r}}) \sum_S \sum_{\rho \in M_j^{\vec{r},S}} \mathsf{Pr}_{\omega_2}[T(\omega_1^{j,\vec{r},S}, \omega_2) \mapsto \rho] \mathsf{Pr}_{\omega_1}[\mathsf{abst}(\omega_1) = S \mid r_0(\omega_1) = r_0].$$

Next, we use that, by the definition of $M_j^{\vec{r},S}$, for $\rho \in M_j^{\vec{r},S}$ we have $f(\rho) = j$ and, because $f(\rho)$ depends only on ω_2, $T(\omega_1^{j,\vec{r},S}, \omega_2) \mapsto \rho$ implies $f(\omega_2) = j$. With this, we obtain

$$\mathsf{Pr}_{\omega_2}[T(\omega_1^\rho, \omega_2) \mapsto \rho] = \mathsf{Pr}_{\omega_2}[f(\omega_2) = j] \cdot \mathsf{Pr}_{\omega_2}[T(\omega_1^{j,\vec{r},S}, \omega_2) \mapsto \rho \mid f(\omega_2) = j]$$

for $\rho \in M_j^{\vec{r},S}$. Now, we can conclude

$$\begin{aligned}
\Phi &\leq \sum_j \sum_{\vec{r} \in M^*_{i,j}} (A^j_{\vec{r}} - A^i_{\vec{r}}) \sum_S \mathsf{Pr}_{\omega_1}[\mathsf{abst}(\omega_1) = S | r_0(\omega_1) = r_0] \sum_{\rho \in M_j^{\vec{r},S}} \mathsf{Pr}_{\omega_2}[f(\omega_2) = j] \cdot \mathsf{Pr}_{\omega_2}[T(\omega_1^{j,\vec{r},S}, \omega_2) \mapsto \rho \mid f(\omega_2) \\
&\leq \sum_j \mathsf{Pr}_{\omega_2}[f(\omega_2) = j] \sum_{r \in M^*_{i,j}} (A^j_r - A^i_r) \sum_S \mathsf{Pr}_{\omega_1}[\mathsf{abst}(\omega_1) = S | r_0(\omega_1) = r_0] \\
&\leq \sum_j \mathsf{Pr}_{\omega_2}[f(\omega_2) = j] \sum_{\vec{r} \in M^*_{i,j}} (A^j_{\vec{r}} - A^i_{\vec{r}}) \\
&\leq \sum_j \mathsf{Pr}_{\omega_2}[f(\omega_2) = j] \cdot \delta^i_{min}(n,k,\vec{p}) \ \leq \delta^i_{min}(n,k,\vec{p}) = \delta.
\end{aligned}$$

This concludes the proof of Theorem 2.

5 Related Work

As mentioned in the introduction, only very few voting protocols have been analyzed rigorously in cryptographic models w.r.t. coercion-resistance.

In [17], Juels et al. presented one of the first definitions of coercion-resistance, which is tailored towards voting in a public-key setting, with protocols having a specific structure. This definition is applied to the author's voting protocol.

A definition of coercion-resistance within the simulation-based approach was presented by Moran and Naor [24], based on a definition of coercion-resistance for multi-party computation by Canetti and Gennaro [6]. Moran and Naor apply their definition to two of their protocols, including their Split-Ballot protocol [25]. As further discussed in [20], and also mentioned in Section 4.1, among others due the so-called commitment problem, the definition of Moran and Naor is quite demanding and rules out many seemingly reasonable voting protocols.

Teague et al. [34] proposed a definition of coercion-resistance which takes a quantitative approach. However, this definition has the following limitations: (i) It is intended to be used for ideal protocols, combined, as the authors suggest, with a simulation-based definition. (ii) The coercer may only use a specific strategy to decide whether to punish the coerced voter or not. Also, the class of counter-strategies available to the coerced voter is limited. (iii) Only the probability that a cheating voter gets punished is considered, ignoring the possibility that a voter might try to sell her vote by following the instructions of the coercer. Teague et al. apply their definition to a tallying procedure for STV voting.

A recent definition of coercion-resistance by Gardner et al. [15] is specifically tailored to the protocol considered by the authors. It also considers only a very restricted part of an election process, denying, for example, the coercers access to information in the tallying phase.

As mentioned in the introduction, our work is based on the definition of coercion-resistance proposed in [20] by Küsters et al. They apply their definition to ThreeBallot [30] and Bingo Voting [4]. As Küsters et al. pointed out in their work, these protocols could not have been analyzed based on other cryptographic definition of coercion-resistance. As

meniontedin Section 4.1, the analysis of Scantegrity II is also outside the scope of other definitions of coercion-resistance.

As already mentioned in the introduction, several definitions of coercion-resistance were proposed in symbolic models [14, 2, 19]. The definitions in [14, 2] were, among others, applied to the protocol proposed in [17]. The authors of [19] applied their definition to the voting system Civitas [13], a voting protocol by Lee et al. [22], and one by Okamoto [27].

6 Conclusion and Future Work

In this paper, we have shown that Scantegrity II provides an optimal level of coercion-resistance, i.e., the same level of coercion-resistance as an ideal voting protocol, under the (necessary) assumption that the workstation and the PRNG used in Scantegrity II are honest. Since we assume that the coercer can see the receipts of all voters, and hence, can see whether or not a voter voted, Scantegrity II is not resistant to forced abstentation attacks.

Besides coercion-resistance, Scantegrity II is also designed to provide verifiability. We leave it to future work to analyze Scantegrity II w.r.t. this property. It seems possible to use a very recently proposed definition of verifiability by Küsters, Truderung, and Vogt [21] for this purpose. In the same paper, Küsters et al. also provide a definition of accountability, which would be interesting to apply to Scantegrity II.

References

[1] B. Adida and R. L. Rivest. Scratch & vote: self-contained paper-based cryptographic voting. In *Workshop on Privacy in the Electronic Society (WPES 2006)*, pages 29–40, 2006.

[2] M. Backes, C. Hritcu, and M. Maffei. Automated Verification of Remote Electronic Voting Protocols in the Applied Pi-Calculus. In *Proceedings of the 21st IEEE Computer Security Foundations Symposium (CSF 2008)*, pages 195–209. IEEE Computer Society, 2008.

[3] J. C. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing (STOC 1994)*, pages 544–553. ACM Press, 1994.

[4] J.-M. Bohli, J. Müller-Quade, and S. Röhrich. Bingo Voting: Secure and Coercion-Free Voting Using a Trusted Random Number Generator. In A. Alkassar and M. Volkamer, editors, *E-Voting and Identity (VOTE-ID 2007)*, volume 4896 of *Lecture Notes in Computer Science*, pages 111–124. Springer, 2007.

[5] R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. Technical report, Cryptology ePrint Archive, December 2005. Online available at `http://eprint.iacr.org/2000/067.ps`.

[6] R. Canetti and R. Gennaro. Incoercible Multiparty Computation (extended abstract). In *37th Annual Symposium on Foundations of Computer Science (FOCS '96)*, pages 504–513. IEEE Computer Society, 1996.

[7] D. Chaum. `http://punchscan.org/`.

[8] D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Commun. ACM*, 24(2):84–88, 1981.

[9] D. Chaum. Secret-Ballot Receipts: True Voter-Verifiable Elections. *IEEE Security & Privacy*, 2(1):38–47, 2004.

[10] D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. L. Rivest, P. Y. A. Ryan, E. Shen, and A. T. Sherman. Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems using Invisible Ink Confirmation Codes. In *USENIX/ACCURATE Electronic Voting Technology (EVT 2008)*. USENIX Association, 2008.

[11] D. Chaum, P.Y.A. Ryan, and S. Schneider. A practical, voter-verifiable election scheme. In *Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS 2005)*, volume 3679 of *Lecture Notes in Computer Science*, pages 118–139. Springer, 2005.

[12] B. Chevallier-Mames, P.-A. Fouque, D. Pointcheval, J. Stern, and J. Traoré. On Some Incompatible Properties of Voting Schemes. In *IAVoSS Workshop On Trustworthy Elections (WOTE 2006)*, 2006.

[13] M. R. Clarkson, S. Chong, and A. C. Myers. Civitas: Toward a Secure Voting System. In *2008 IEEE Symposium on Security and Privacy (S&P 2008)*, pages 354–368. IEEE Computer Society, 2008.

[14] S. Delaune, S. Kremer, and M.D. Ryan. Coercion-Resistance and Receipt-Freeness in Electronic Voting. In *Proceedings of the 19th IEEE Computer Security Foundations Workshop (CSFW'06)*, pages 28–39. IEEE Computer Society Press, 2006.

[15] R. Gardner, S. Garera, and A. D. Rubin. Coercion Resistant End-to-end Voting. In Roger Dingledine and Philippe Golle, editors, *13th International Conference on Financial Cryptography (FC 2009)*, volume 5628 of *Lecture Notes in Computer Science*, pages 344–361. Springer, 2009.

[16] H.L. Jonker and W. Pieters. Receipt-Freeness as a special case of Anonymity in Epistemic Logic. In *IAVoSS Workshop On Trustworthy Elections (WOTE 2006)*, 2006.

[17] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In *Proceedings of Workshop on Privacy in the Eletronic Society (WPES 2005)*, pages 61–70. ACM Press, 2005.

[18] R. Küsters. Simulation-Based Security with Inexhaustible Interactive Turing Machines. In *Proceedings of the 19th IEEE Computer Security Foundations Workshop (CSFW-19 2006)*, pages 309–320. IEEE Computer Society, 2006.

[19] R. Küsters and T. Truderung. An Epistemic Approach to Coercion-Resistance for Electronic Voting Protocols. In *2009 IEEE Symposium on Security and Privacy (S&P 2009)*, pages 251–266. IEEE Computer Society, 2009.

[20] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. A Game-based Definition of Coercion-Resistance and its Application. In *23th IEEE Computer Security Foundations Symposium, CSF 2010*. IEEE Computer Society, 2010. to appear.

[21] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Accountabiliy: Definition and Relationship to Verifiability. Technical Report 2010/236, Cryptology ePrint Archive, 2010. `http://eprint.iacr.org/`.

[22] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. Providing receipt-freeness in mixnet-based voting protocols. In *Proceedings of Information Security and Cryptology (ICISC 2003)*, volume 2971 of *Lecture Notes in Computer Science*, pages 245–258. Springer, 2003.

[23] D. Lundin and P. Y. A. Ryan. Human Readable Paper Verification of Prêt à Voter. In *European Symposium on Research in Computer Security (ESORICS 2008)*, pages 379–395, 2008.

[24] T. Moran and M. Naor. Receipt-Free Universally-Verifiable Voting With Everlasting Privacy. In C. Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 373–392. Springer, 2006.

[25] T. Moran and M. Naor. Split-ballot voting: everlasting privacy with distributed trust. In P. Ning, S. De Capitani di Vimercati, and P. F. Syverson, editors, *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007*, pages 246–255. ACM, 2007.

[26] C. A. Neff. Practical High Certainty Intent Verification for Encrypted Votes. `http://www.votehere.com/old/vhti/documentation/vsv-2.0.3638.pdf`.

[27] T. Okamoto. Receipt-Free Electronic Voting Schemes for Large Scale Elections. In B. Christianson, B. Crispo, T. M. A. Lomas, and M. Roe, editors, *Proceedings of the 5th International Workshop on Security Protocols*, volume 1361 of *Lecture Notes in Computer Science*, pages 25–35. Springer, 1997.

[28] S. Popoveniuc and B. Hosp. An introduction to PunchScan. In *IAVoSS Workshop on Trustworthy Elections (WOTE 2007)*, 2007.

[29] B. Riva and A. Ta-Shma. Bare-Handed Electronic Voting with pre-processing. In *USENIX/ACCURATE Electronic Voting Technology (EVT 2007)*, 2007.

[30] R. L. Rivest and W. D. Smith. Three Voting Protocols: ThreeBallot, VAV and Twin. In *USENIX/ACCURATE Electronic Voting Technology (EVT 2007)*, 2007.

[31] P. Y. A. Ryan. A variant of the Chaum voter-verifiable scheme. In *Water, Innovation, Technology & Sustainability (WITS 2005)*, pages 81–88, 2005.

[32] P. Y. A. Ryan. Prêt à Voter with Paillier Encryption-extended journal version. Technical report, University of Newcastle upon Tyne, 2008.

[33] P. Y. A. Ryan and S. A. Schneider. Prêt à Voter with Re-encryption Mixes. In *European Symposium on Research in Computer Security (ESORICS 2006)*, pages 313–326. Springer, 2006.

[34] V. Teague, K. Ramchen, and L. Naish. Coercion-Resistant tallying for STV voting. In *IAVoSS Workshop On Trustworthy Elections (WOTE 2008)*, 2008.

A Proof of Lemma 3

The core of Lemma 3 is stated in the following lemma.

LEMMA 4. *Let ρ be an arbitrary view such that $f(\rho)$ is defined. Let $\omega_1, \omega_1', \omega_1'', \omega_1'''$ be arbitrary, fixed elements of Ω_1 with $\varphi_\rho(\omega_1)$, $\varphi_\rho(\omega_1')$, $\tilde{\varphi}_\rho(\omega_1'')$, and $\tilde{\varphi}_\rho(\omega_1''')$. Then the sets*

$$A = \{\omega_2 : T(\omega_1, \omega_2) \mapsto \rho\}, \qquad C = \{\omega_2 : \tilde{T}(\omega_1'', \omega_2) \mapsto \rho\},$$
$$B = \{\omega_2 : T(\omega_1', \omega_2) \mapsto \rho\}, \qquad D = \{\omega_2 : \tilde{T}(\omega_1''', \omega_2) \mapsto \rho\}.$$

have the same cardinality, and hence, $\mu_2(A) = \mu_2(B) = \mu_2(C) = \mu_2(D)$.

To prove this lemma, we use Lemma 5. To state Lemma 5, we use the following notation. By $\tilde{T}_j$ we denote the system $(\tilde{v}_j \parallel c \parallel \mathsf{e}_S)$, where $\tilde{v}_j$ is defined like $\tilde{v}$ but votes for j instead of i. So we have $\tilde{v} = \tilde{v}_i$ and $\tilde{T} = \tilde{T}_i$. Moreover, for each view ρ of the coercer, for which $f(\rho)$ is defined, we clearly have: $T(\omega_1, \omega_2) \mapsto \rho$ iff $\tilde{T}_{f(\rho)}(\omega_1, \omega_2) \mapsto \rho$. A permutation σ on a tuple $(v_0, \ldots, v_n) \in \{0, 1, \ldots, k\}^{n+1}$ is a permutation on the set of indices $\{0, \ldots, n\}$. We write $\sigma(v_0, \ldots, v_n)$ for the tuple $(v_{\sigma(0)}, \ldots, v_{\sigma(n)})$. For simplicity of notation, we sometimes write $\sigma(v_i)$ instead of $v_{\sigma(i)}$. We say that σ does not change the abstaining votes of $(v_0, \ldots, v_n)$ if $\sigma(j) = j$ for every $j \in \{0, \ldots, k\}$ with $v_j = 0$. For $j \in \{1, \ldots, k\}$ and $\omega_1 \in \Omega_1 (= \{0, 1, \ldots, k\}^n)$, we consider (j, ω_1) to be an $(n+1)$-tuple over $\{0, 1, \ldots, k\}$. If σ is a permutation on (j, ω_1), we may apply σ to ω_1, written $\sigma(\omega_1)$, with the obvious meaning. With this and the above conventions, we have that $\sigma(j, \omega_1) = (\sigma(j), \sigma(\omega_1))$.

LEMMA 5. *For every $j \in \{1, \ldots, k\}$, every $\omega_1 \in \Omega_1$ and every permutation σ^0 on (j, ω_1) that does not change the abstaining votes, there is a bijective function $h = h^{j, \omega_1, \sigma^0}$ from Ω_2 to Ω_2 such that for all ω_2 we have that $\tilde{T}_j(\omega_1, \omega_2)$ yields the same view as $\tilde{T}_{\sigma^0(j)}(\sigma^0(\omega_1), h(\omega_2))$.*

We postpone the proof of this lemma to the end of this section. Now, Lemma 4 follows directly from Lemma 5: Given the assumptions of Lemma 4, there are permutations σ_1^0, σ_2^0, and σ_3^0 such that $(f(\rho), \omega_1) = \sigma_1^0(f(\rho), \omega_1') = \sigma_2^0(i, \omega_1'') = \sigma_3^0(i, \omega_1''')$. Moreover, $T(\omega_1, \omega_2) \mapsto \rho$ iff $\tilde{T}_{f(\rho)}(\omega_1, \omega_2) \mapsto \rho$ and $\tilde{T}(\omega_1, \omega_2) \mapsto \rho$ iff $\tilde{T}_i(\omega_1, \omega_2) \mapsto \rho$. From this and Lemma 5 we obtain that the functions $h^{f(\rho), \omega_1, (\sigma_1^0)^{-1}}$, $h^{f(\rho), \omega_1, (\sigma_2^0)^{-1}}$, and $h^{f(\rho), \omega_1, (\sigma_3^0)^{-1}}$, are bijections between A and B, A and C, and A and D, respectively.

Now with Lemma 4 we can easily complete the proof of Lemma 3:

$$\begin{aligned}
\mathsf{Pr}[T \mapsto \rho] &= \mathsf{Pr}_{\omega_1,\omega_2}[\varphi_\rho(\omega_1), T(\omega_1,\omega_2) \mapsto \rho] \\
&= \sum_{\omega_1' : \varphi_\rho(\omega_1')} \mathsf{Pr}_{\omega_1,\omega_2}[\omega_1 = \omega_1', T(\omega_1',\omega_2) \mapsto \rho] \\
&= \sum_{\omega_1' : \varphi_\rho(\omega_1')} \mu_1(\omega_1') \cdot \mathsf{Pr}_{\omega_1,\omega_2}[T(\omega_1',\omega_2) \mapsto \rho \mid \omega_1 = \omega_1'] \\
&= \sum_{\omega_1' : \varphi_\rho(\omega_1)} \mu_1(\omega_1') \cdot \mathsf{Pr}_{\omega_2}[T(\omega_1',\omega_2) \mapsto \rho] \\
&= \sum_{\omega_1' : \varphi_\rho(\omega_1)} \mu_1(\omega_1') \cdot \mathsf{Pr}_{\omega_2}[T(\omega_1^\rho,\omega_2) \mapsto \rho] \\
&= \mathsf{Pr}_{\omega_1}\left[\varphi_\rho(\omega_1)\right] \cdot \mathsf{Pr}_{\omega_2}\left[T(\omega_1^\rho,\omega_2) \mapsto \rho\right].
\end{aligned}$$

This proves (1). The proof for (2) is analogous. Statement (3) follows immediately from Lemma 4.

Proof of Lemma 5. To prove Lemma 5, we first introduce notation for the components (cryptographic operations, random numbers, etc.) of the Scantegrity II protocol.

The cryptographic components. We first describe in detail the structure of the sequence $\omega_2 \in \Omega_2$ of random coins. In the following, by $\mathsf{comm}(a)^r$ we denote the commitment on a with randomness r.

(a) α — the random coins of the coercer.
(b) c_i^j for $i \in \{0,\ldots,s\}$ and $j \in \{1,\ldots,k\}$ — the codes.
(c) π_i for $i \in \{0,\ldots,s\}$ — the permutations used to create the Q-table from the P-table (row i of P is permutated by π_i).
(d) π, — a permutation of $1,\ldots,s\cdot k$ representing the Q-pointers of table R.
(e) π', — a permutation of $1,\ldots,s\cdot k$ representing the S-pointers of table R, where we assume that π' is consistent with π and π_i, i.e. a every element of the P-table is mapped to the right column in the S-table.
(f) r_i^j for $i \in \{0,\ldots,s\}$ and $j \in \{1,\ldots,k\}$ — the random numbers used for the commitments of the entries in the Q-table, $cc_i^j = \mathsf{comm}(\pi_i(c_i^j), r_i^j)$.
(g) R_t^1, R_t^2 for $t \in \{1,\ldots,s\}$ — the random numbers used for the commitments of the entries in the R-table, $cR_t^1 = \mathsf{comm}(\pi(t), R_t^1)$ and $cR_t^2 = \mathsf{comm}(\pi'(t), R_t^2)$.
(h) Random values S_i for $i = 1,\ldots,n$ — determining whether the honest voter v_i takes two ballots and which of the ballots to audit.
(i) Random challenges $s_t \in \{1,2\}$ for $t \in \{1,\ldots,s\}$ contributed by the auditors.

A view ρ of the coercer, depending on ω_2 and the choices $v_0,\ldots,v_n$ taken by the voters, consists of the following parts:

(B1) α — random coins of the coercer.
(B2) The commitments $cc_i^j = \mathsf{comm}((c_i^{\pi_i(j)}), r_i^j)$, $cR_t^1 = \mathsf{comm}(\pi(t), R_t^1)$, and $cR_t^2 = \mathsf{comm}(\pi'(t), R_t^2)$.

(B3) $c_i^{v_i}$ — the code of the i-th voter, for every non-abstaining voter i (voting for candidate v_i).

(B4) The opened commitments in the Q table.

(B5) The challenges s_l and the corresponding opened commitments of the R-table.

(B6) The flagged entries in the R- and S-table.

Because every permutation is the finite composition of permutations that switch only two successive positions, it suffices to consider the case where σ flips the positions l and $l+1$; the rest follows from composing permutations and bijections. Let $\tilde{v}_0, \ldots, \tilde{v}_n$ be such that

$$\sigma(v_0, \ldots, v_n) = (\tilde{v}_0, \ldots, \tilde{v}_n) = (v_0, \ldots, v_{l+1}, v_l, \ldots, v_n.)$$

Further, we assume that $v_l = y \neq z = v_{l+1}$, as the case that $\sigma(v_0, \ldots, v_n) = (v_0, \ldots, v_n)$ is trivial. Recall that, by assumption, we have that $y, z \neq 0$.

Let ω_2 be any element of Ω_2 and let α, c_i^j, π_i, π, π', r_i^j, R_t^1, R_t^2, S_i, and s_t be the parts of ω_2 defined as above. Here, i ranges over $0, \ldots, s$ and j over $1, \ldots, k$ and t over $1, \ldots, k \cdot s$. We will denote the corresponding parts of $h(\omega_2)$ by $\tilde{\alpha}$, $\tilde{c}_i^j$, and so on. We define $h(\omega_2)$ as follows:

- $\tilde{\alpha} = \alpha$. As one can see, (B1) remains unchanged.
- $\tilde{c}_i^j$ are defined like c_i^j, except for that c_l^y and c_l^z as well as c_{l+1}^z and c_{l+1}^y are swapped. By that, (B3) remains unchanged.
- $\tilde{\pi}_i$ is defined as π_i, except for $\tilde{\pi}_l$ and $\tilde{\pi}_{l+1}$. These two permutations differ from π_l and π_{l+1} in that the positions of c_l^y and c_l^z and of c_{l+1}^z and c_{l+1}^y are swapped.
- $\tilde{r}_i^j$ are defined like r_i^j. By that we get that (B2) remains unchanged.
- The rest of ω_2 remains unchanged. By that, (B4), (B5), and (B6) remain unchanged.

This concludes the description of $h(\omega_2)$. As we have noted, all the parts (B1)–(B6) of the views in both cases—for ω_2 and $h(\omega_2)$—are exactly the same. As h just flips two times two codes and changes two corresponding permutations, h is a bijection from Ω_2 to Ω_2.