Election Security and Economics: It's all about Eve

David Basin[1], Hans Gersbach[2], Akaki Mamageishvili[2], Lara Schmid[1], and Oriol Tejada[2]

[1] Institute of Information Security ETH Zurich
[2] Chair of Macroeconomics: Innovation and Policy ETH Zurich

Abstract. A system's security must be understood with respect to the capabilities and behaviors of an adversary Eve. It is often assumed in security analysis that Eve acts as maliciously as possible. From an economic perspective, Eve tries to maximize her utility in a game with other participants. The game's rules are determined by the system and its security mechanisms, but Eve can invent new ways of interacting with participants. We show that Eve can be used as an interface to explore the interplay between security and economics in the domain of elections. Through examples, we illustrate how reasoning from both disciplines may be combined to explicate Eve's motives and capabilities and how this analysis could be used for reasoning about the security and performance of elections. We also point to future research directions at the intersection of these disciplines.

1 Introduction

Election security is an important societal problem as attacks on elections put democracy at risk. When establishing that an election system is secure, one must reason about the adversarial environment in which the system is used. This requires specifying the capabilities of the adversary, henceforth called Eve.

In the security community, one provides an adversary model that specifies Eve's capabilities and assumes she will exploit these capabilities, independent of the costs. For election security, one typically assumes the existence of reasonably strong adversaries when designing the system, for example adversaries that may compromise the client's platform but not the voting server or the postal channel. Such assumptions are usually made without detailed economic justifications. In economics, one considers what Eve is rationally motivated to do and one looks at the entire range of sophisticated mechanisms available to her to exploit the humans that use the system. For example, a wealthy adversary might try to buy votes in elections, with adverse consequence; see e.g. [14]. Moreover, economists may consider the scenario where a majority of citizens base their voting decisions on false assumptions about their decisions' effects, with adverse long-term societal consequences [6].

In this paper, we outline these two perspectives of Eve. We show that the perspective used in one discipline can sharpen the assumptions, models, and results used in the other discipline. Hence, both disciplines together can best ensure election security and the quality of election outcomes.

First, security analysis is central to economic models of elections since these models always depend implicitly on security properties such as integrity or coercion resistance, as we will illustrate in this paper. Hence, trust in an election's outcome depends on whether such security properties can be proven to hold. Moreover, when harmful adversarial behavior cannot be ruled out, an analysis of the adversary's capabilities provides a guide to constructing economic models involving these adversaries. One can then calculate the expected election outcome in the presence of the modeled adversary.

Second, economic analysis is important for security analysis in order to determine what a rational adversary will do. On the one hand, Eve may never undertake certain actions and thus these actions can be omitted from the security analysis. On the other hand, Eve may invent entirely new games to interact with a system's participants, which can undermine the system's security properties. This may necessitate modeling Eve or other participants differently in the security analysis. We illustrate this with two examples in this paper. In the first example, we show that the use of decoy ballots, which are fake ballots that are introduced to avoid vote buying, are much less secure than assumed so far. In the second example, we explain why the authenticity of voting-related information must be considered to be a central security property since, otherwise, an adversary could spoof a trusted information source and send biased information to voters, which could lead to undesirable voting outcomes.

Most research in security analysis and economics has been carried out independently. In recent times, research straddling these two disciplines has emerged. For example, malware researchers [8,25] have investigated the behavior of real-life adversaries and how this behavior relates to their economic goals. Other researchers [1,11,15] have modeled (coercible) users and security providers as rational agents and used this to investigate the adequacy of different security measures. Game-theoretic models have been employed [24,27] to analyze the security of physical environments, such as airports and harbors, and to determine the best strategies to protect them against adversaries. Recently, researchers in elections have started investigating this interplay too, for example, in the context of vote buying [18]. We see our work in line with this trend, explicating the interplay between security and economics and highlighting Eve's use as an interface between these disciplines.

We proceed as follows. In Section 2, we review how (voting) protocols are generally formalized in information security and economics, highlighting Eve's special role. In Section 3, we describe two voting protocols, a simple voting protocol and Chaum's [9] *random sample elections*, which we use in Sections 4 and 5 to illustrate how information security researchers and economists analyze voting protocols and to investigate the interplay between these two disciplines. Finally, in Section 6, we draw conclusions and provide a perspective on the scope and some of the challenges of this important interdisciplinary research area.

2 General approaches

2.1 Information security

To analyze a system in information security, one must specify the system P, the adversary (alias "Eve") A, and the desired security properties *Prop*. The system's security is established by proving an assertion of the form $P, A \vDash \mathit{Prop}$, which states that all possible system behaviors satisfy the property *Prop*, when P is executed in an environment with the adversary A. When the system is distributed, such as (voting) protocols are, this essentially means that all possible behaviors arising from agents executing the protocol, operating in parallel with the adversary, satisfy the property *Prop*. Rigorously establishing this requires precise specifications of P, A, and *Prop* and constructing proofs, ideally, using theorem provers or model checkers. For security protocols, the specifications are often given using symbolic models, and proofs are constructed using model checkers like ProVerif [7] or Tamarin [17,20]. See [4] for more on this.

We now further describe P, A, and *Prop*, focusing on the distributed setting. Here, P specifies the protocol that honest agents follow. For example, P is defined by *role specifications* that describe the behavior of honest agents in terms of which messages they send and receive and in which order. The protocol's execution semantics is defined by all possible interleavings of instantiated roles, also interleaved with actions of the adversary A.

A property *Prop* is intended to hold in every possible execution of the protocol. What *Prop* specifies depends on the system under consideration. For voting protocols, we are typically interested in the following properties. *Integrity* demands that information, e.g., votes, cannot be changed by an unauthorized entity. *Verifiability* additionally stipulates that integrity can be verifiably established, e.g., by individuals who check that their own votes are recorded as cast (*individual verifiability*) or that all votes are counted as recorded (*universal verifiability*). *Secrecy* and *privacy* guarantee that it is indistinguishable who voted for what. Finally, *coercion resistance* states that a voter cannot prove to an adversary how he voted, even if he actively collaborates with the adversary.

Eve, the adversary A, is the focus of this paper. We emphasize that a system's security can only be analyzed meaningfully with respect to a class of adversaries. For example, a system P that keeps data secret (*Prop*) in the face of a network adversary A, may be insecure against a stronger adversary with physical access to the system, who can perform side channel attacks or even remove and copy the hard disk. For security protocols, a popular adversary is the *Dolev-Yao* adversary [10], who has full control over the network. This adversary can read and change everything sent over the network, and can also send messages herself. Furthermore, this adversary can compromise agents and learn their secrets. We will consider a number of other adversaries shortly in the context of voting.

2.2 Economics

Economic models of collective decision mechanisms help to analyze the design and goals thereof. In particular, they can be used to establish if a given voting protocol is based on principles of liberal democracies and whether it yields welfare gains.

Game-theoretical models, in particular, are best suited for assessing the properties of collective decision mechanisms. These models aim to explain the strategic interaction between agents with opposing interests and to discern why some agents may opt for particular behaviors. A game-theoretical model of a collective decision mechanism demands that we specify the following elements:

1. The player set *(Who)*: who are the agents that can participate in the game?
2. The game rules *(How)*: what is each agent allowed to do and what information is available to him when he takes his decisions?
3. The strategy set *(What)*: what strategies are available to the agents, where a strategy describes what the agent does in each game situation?
4. Utilities *(Why)*: what does each player want to achieve in such a game?

Each player aims to maximize his (expected) utility, given his observations about other players' past actions and his predictions about past and future actions. Given a game, one looks for its *equilibria*, i.e., for the situations where no player has an incentive to change his decision given the (expected) decisions of the remaining players. These equilibria are predictions about the outcome of collective decisions, and can be investigated with respect to the quality and costs of the game's outcome. Most game-theoretical models do not assume the existence of an adversary that can influence the outcome of the collective decision. There is however a strand of literature that explicitly incorporates an adversary as an active player of the game. In this paper we examine one instance of such a model.

3 Voting protocols

Numerous voting protocols have been proposed in the past. We introduce here two protocols that we will subsequently use to illustrate how voting protocols are analyzed from the information security and economic perspectives.

Voting protocols often involve players (or agents) acting in roles, which are called *principals*. These include a *voting server/election authority*, with a database that processes all the cast votes, stores them, and tallies them. Often, the election authority, who conducts the elections, and the voting server are considered to be one principal. The eligible *voters* are the principals who are legally entitled to vote. When voting electronically, they cast their vote using a *computing platform*. Usually, one considers a *public bulletin board* where votes are published in an authentic way and cannot be altered afterwards. Finally, *auditors* are the principals who check the published information for consistency. Auditors may be voters, party members, candidates, or independent parties.

3.1 Simple voting protocol

A simple voting protocol is shown in Figure 1. This protocol is overly simple; it merely serves to illustrate Eve's role in the following sections. The three

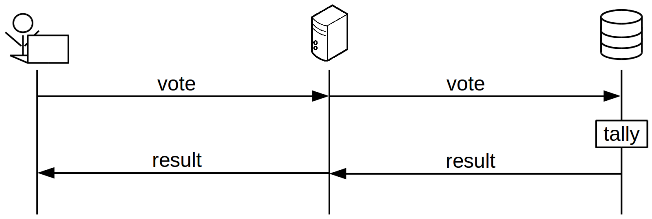

Fig. 1: A simple voting protocol.

involved principals, from left to right, denote a voter, a voting server, and a database where votes are collected. Here we explicitly separate the server from the database to model a traditional three-tier architecture with a presentation tier (browser on the client), a server tier, and a storage tier. In the protocol, a voter sends his vote to the server, which stores the vote in the database. After all votes have been collected, the votes in the database are tallied and the result is published on the server. A voter can read the published result from the server.

3.2 Random sample elections

A more complex protocol, but with stronger security guarantees, is random sample elections as introduced by Chaum [9]. The main novelty is that only a random sample of the electorate votes. The motivation is economic: this procedure should be less costly than voting by the entire electorate, and voters may be better informed when they vote less frequently.

In more detail, random sample elections partition the electorate into three sets. The first set consists of the randomly selected (real) voters, whose votes will be counted. The second set consists of *decoy voters* who can ask for, and receive, fake ballots, which they can sell to adversaries. The third set contains those members of the electorate who are not selected and do not ask for fake ballots. Votes cast with fake ballots will have no effect on the tally. Nevertheless, after a decoy voter has ordered a ballot, he behaves exactly as a normal voter when casting his vote. As we explain below, decoy votes are intended to prevent coercion. Additionally, there are auditors, who may be voters or other individuals.

Figure 2 illustrates some of the actions that can take place in random sample elections. As a preliminary step, decoy voters can actively order ballots; in contrast, selected real voters receive ballots without prior actions. This optional step for decoy voters is illustrated by the dashed arrow. Afterwards, the protocol for real voters and decoy voters is identical. First, each voter is provided a pair of ballots by mail. Each ballot has a serial number, $200a$ and $200b$ in the example, and two answers, yes/no, each with a unique code. A voter can choose either ballot for voting. Second, to cast his vote, the voter enters online the serial number of the chosen ballot and the code of his choice. Figure 2 depicts an example

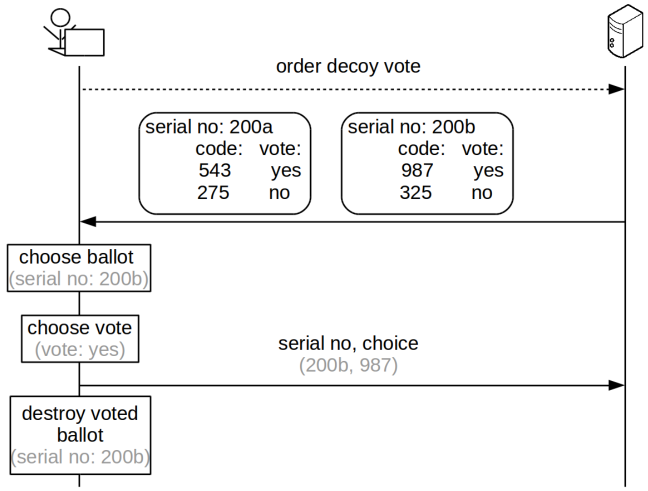

Fig. 2: The voting protocol for random sample elections, illustrated on an example. The dashed arrow indicates the message only sent by decoy voters.

of this in gray. Namely, the voter decides to vote with the ballot with the serial number 200b and the vote *yes*. Therefore, he looks up the code corresponding to *yes* on this ballot, which is 987, and he casts his vote by entering the serial number and this code online. Finally, the voter destroys the ballot with the serial number 200b so that no one can learn to which vote this code corresponds. He may write down the code 987 to help him remember later what he has sent.

During the voting procedure, the election authority posts information on the bulletin board to enable auditors to verify that the voting procedure was correctly followed. We explain next, on an example, the election authority's internal representation of this information.

Consider a random sample election with two voters, a real voter V_r and a decoy voter V_d. We assume that there are the two pairs of ballots given in Figure 3. The first pair (the two ballots on the left) is from Figure 2 and we assume that it was sent to the real voter V_r. The second pair (the two ballots on the right) is sent to the decoy voter V_d. Furthermore, we assume that, as in

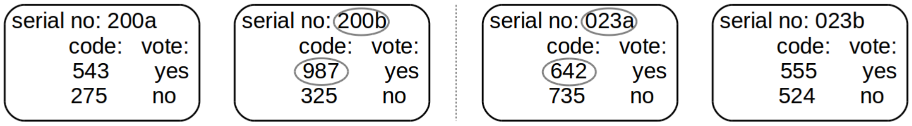

Fig. 3: Two pairs of ballots, where the left pair is from a real voter and the right pair is from a decoy voter. Choices are circled in gray.

code	**print**	**vote**	**voted**	**decoy**
200a, 543	200a, 543	yes	-	-
200a, 275	200a, 275	no	-	-
200b, 987	-	yes	x	-
200b, 325	-	no	-	-
023a, 642	-	yes	x	decoy
023a, 735	-	no	-	decoy
023b, 555	023b, 555	yes	-	decoy
023b, 524	023b, 524	no	-	decoy

Fig. 4: Internal representation of the information stored by the election authority in random sample elections (simplified).

Figure 2, V_r selects ballot 200b and votes *yes* and that V_d selects ballot 023a and votes *yes*.

Figure 4 illustrates the table that is known only to the election authority after the votes are cast. The first column denotes the serial numbers and the codes as appearing on the ballots. The second column indicates which ballots have not been used for casting votes and lists the serial number and codes of these ballots again. Recall that each voter receives two ballots, but only uses one for voting. In the example, the ballots 200a and 023b have not been used for casting votes. The third column indicates the vote that corresponds to the respective code in this column. For example, the first row indicates that on the ballot with serial number 200a, the code 543 represents the vote *yes*. The fourth column marks which votes have been chosen. For example, the third row indicates that on ballot 200b, the code 987, which encodes the choice *yes*, has been voted. Finally, the last column indicates whether the respective ballot was sent to a decoy voter, which is the case here for the ballots 023a and 023b.

We will explain in the next section how protocols for posting parts of this information enable verifiability.

4 Information security analysis

We first present the information security approach to analyzing security protocols. We start with the simple protocol from Section 3 and use it to highlight the importance of adversary models and also the relationship of these models to trust assumptions. Afterwards, we turn to random sample elections.

4.1 Adversary

Trust and compromised principals. In information security, one reasons about the adversary Eve, as formalized by an adversary model, or by trust assumptions. These notions are dual: if we trust a principal, for example a system component, to act in a certain way (e.g., to follow a specification), this is equivalent to assuming that Eve cannot compromise the component and thereby alter its behavior. For example, if we consider a trusted client and server in our simple voting protocol (Figure 1), we can send messages to the server over the Transport Layer Security protocol TLS (which establishes a secure channel) and hence an adversary who can only observe the network cannot eavesdrop or tamper with transmitted messages, such as votes or election results. However, if we consider a compromised client platform, the adversary can both learn and alter the votes sent. Similarly, if we do not trust the server, i.e., if it can be compromised, then it does not help to use a secure channel to send votes over the network. Eve can still learn and alter the votes because she can learn all the server's secrets.

The following example illustrates that considering different trust assumptions for different usage scenarios is commonplace.

Example 1. The Swiss regulations for electronic voting [22,23] dictate that if at least 30% of the electorate vote electronically, it is assumed that the platform is untrusted but the server is trusted. However, if at least 50% of the electorate vote electronically, it must be assumed that both the platform and the server are untrusted. Equivalently, in the first case, it is assumed that Eve can corrupt just the platform, whereas in the second case, she can corrupt the server as well. Hence two different adversary models are used for the two scenarios. ■

Channel assumptions. Continuing with our simple voting protocol, suppose the connection from the voter to the server is not secured by TLS but instead that the unencrypted votes are sent over the insecure network. The voting protocol then does not achieve vote secrecy, even with respect to a weak adversary such as a passive, eavesdropping adversary. It is thus crucial that we state for all principals whether they can be compromised and, moreover, for all communication channels, what Eve's capabilities are.

For online voting, many formalisms assume a Dolev-Yao adversary who can control the network. Assume now that in the simple protocol, votes are not cast online but that the postal service is used instead. Some voting schemes effectively introduce the postal service as an auxiliary (out-of-band) communication channel, which is assumed to be trustworthy, i.e., a secure channel. However, as the following example suggests, one must carefully examine whether such assumptions are justified and what the effects are when these assumptions fail.

Example 2. A reported case of voter fraud took place in the canton of Valais, Switzerland, in March 2017 [21,26]. Normally, ballots are sent to voters by the postal service, after which they are filled out and signed by the voters. The ballots are subsequently cast using the postal service or are hand-delivered to a polling station. In the reported case, some empty ballots were never received by the intended voters. The election protocol used allows voters to order new ballots in such situations. However, when casting their newly ordered ballots, the affected voters noticed that someone else had already voted in their name. The most likely explanation is that the ballots were stolen from their mail boxes and cast by an adversary. Hence, the postal channel did not provide a secure channel from the election authority to the voters, as an adversary had access to the ballots. ■

Summarizing, the adversary model must precisely define for each principal involved and each channel used how Eve can interact with and possibly compromise them. Otherwise security cannot be meaningfully established. See [4] for an account of how to formalize such models in general. [2,3,5] explain how to formalize channel models and adversaries with a wide range of capabilities.

4.2 Security properties

There are many security properties relevant for voting protocols. We concentrate on coercion resistance, integrity, and verifiability, and consider them in the context of random sample elections. We also present some additional properties specific to random sample elections.

Coercion resistance. In voting, Eve may try to coerce voters or pay them to vote as she wishes. Sometimes a distinction is made as to whether the voter is willing to collaborate with Eve, for example, for money. In such a context, a protocol where a voter cannot possibly prove that he voted as Eve demanded is more secure with respect to coercion than a protocol where the voter can prove how he voted if he chooses to collaborate with Eve.

In random sample elections, Chaum [9] suggests that coercion resistance can be achieved by employing decoy votes. These votes are indistinguishable from real votes, but they do not contribute to the tally. Since they can be sold, Eve may be less interested in buying votes because she cannot distinguish a real vote from a decoy vote. In terms of the adversary model, the security properties, and the protocol, this can be understood as follows: if decoy votes are effective, Eve will not buy votes and therefore we can exclude the action of vote buying from the adversary model. Of course, if we model an adversary that does not engage in vote buying, coercion resistance holds, independent of the protocol.

Whether or not Chaum's proposal is an adequate countermeasure to vote buying boils down to an economics question. Eve's problem, according to [19], is that she must offer a sufficiently high price for votes in order to attract real votes in addition to the decoy votes that will always be offered to her. Whether Eve engages in vote-buying in such a situation depends on two factors. First, as the share of decoy votes increases, Eve can buy fewer real votes with a given budget. However, an adversary with an extremely large budget might not be deterred by decoy votes. Second, Eve must know the distribution of the real voters' willingness to sell their votes. Otherwise, she risks buying mainly decoy votes if the price is low or, alternatively, vote-buying may be extremely expensive.

Current analysis of decoy votes [19] suggests that an appropriate design of decoy votes is a viable tool to achieve coercion resistance, however, never in an absolute sense. In Section 5.3, we will discuss new ways to buy votes when there are decoy votes, which cast doubt on whether decoy votes achieve their intended purpose. Furthermore, we demonstrate that they allow an adversary to distinguish real from decoy voters.

Finally, as a side remark, note that decoy votes may pose a challenge to the credibility of the entire voting process since the electorate is encouraged to interact with the adversary.

Integrity and verifiability. Integrity is the property that data cannot be changed in unauthorized ways, for example, the votes cannot be manipulated. Verifiability is the property that participants or outsiders can establish the integrity of the election results. Equivalently, it is verifiable that no one, including the election authority or even errors in the voting software, can alter the result without this being detected. Verifiability properties are often classified as either *individual verifiability* or *universal verifiability.* Individual verifiability states that each voter can verify that his vote was recorded as cast. Universal verifiability states that auditors, which can be anyone, can verify that the recorded votes were counted correctly by the server. To establish such a property, the election authority often publishes different stages of its computations. For example, it publishes the recorded votes in encrypted form and then publishes the decrypted votes as the final tally. Additionally, the authority proves that the tally corresponds to the encrypted votes.

Verification can be performed in different ways. Take, for example, the problem of showing that the decrypted votes correspond to the encrypted ones. A possible strategy is to verify this by a *cut and choose* argument. In cut and choose, the authority constructs several tables of intermediate results and cryptographically commit to them. Once committed, they cannot change the tables' entries. A random event then decides which columns of each table must be revealed. The revealed columns allow anyone to verify that the tables are consistent, without revealing anything secret. Note that at the time it commits to the tables, the election authority does not know which columns will later be revealed. Therefore, if the consistency checks are verified in many iterations of this procedure, all the computations must have been done correctly with high probability.

Example 3, at the end of this section, illustrates cut and choose on the example of random sample elections. Chaum does not explicitly formalize the considered adversary model in random sample elections. However, the presented mechanism establishes the verifiability of the voting tally even if the election authority is compromised.

If we assume that an adversary cannot compromise the election authority, we are usually not concerned with verifiability properties. If the election authority behaves according to the protocol, the result will not be manipulated. However, if we assume that the election authority can be compromised, then verifiability is important. Also, as the adversary can manipulate each part of the computation, we must ensure that we check all relevant parts, from ballot printing all the way to the fact that the ballots are recorded as cast and counted as recorded.

Other properties. Two other security properties specific to random sample elections are the *integrity* and the *verifiability* of the random selection. This means

code	**print**	**vote**	**voted**	**decoy**
200a, 543	200a, 543	yes	-	-
200a, 275	200a, 275	no	-	-
200b, 987	-	yes	x	-
200b, 325	-	no	-	-
023a, 642	-	yes	x	decoy
023a, 735	-	no	-	decoy
023b, 555	023b, 555	yes	-	decoy
023b, 524	023b, 524	no	-	decoy

(a) Full (internal) representation.

code	**print**	**vote**	**voted**	**decoy**
023b, 524	023b, 524	no	-	decoy
023a, 735	-	no	-	decoy
200b, 987	-	yes	x	-
023a, 642	-	yes	x	decoy
200b, 325	-	no	-	-
023b, 555	023b, 555	yes	-	decoy
200a, 275	200a, 275	no	-	-
200a, 543	200a, 543	yes	-	-

(b) Check individual verifiability.

code	**print**	**vote**	**voted**	**decoy**
200b, 325	-	no	-	-
200a, 275	200a, 275	no	-	-
023a, 735	-	no	-	decoy
023a, 642	-	yes	x	decoy
023b, 524	023b, 524	no	-	decoy
200a, 543	200a, 543	yes	-	-
200b, 987	-	yes	x	-
023b, 555	023b, 555	yes	-	decoy

(c) Check print auditing.

code	**print**	**vote**	**voted**	**decoy**
023b, 555	023b, 555	yes	-	decoy
023a, 735	-	no	-	decoy
200b, 987	-	yes	x	-
200b, 325	-	no	-	-
023b, 524	023b, 524	no	-	decoy
023a, 642	-	yes	x	decoy
200a, 543	200a, 543	yes	-	-
200a, 275	200a, 275	no	-	-

(d) Check final tally.

Fig. 5: Simplified version of cut and choose for random sample elections.

that the sampled voters are drawn *uniformly at random* from the set of possible voters, that the election authority cannot manipulate the sample group, and that everyone can verify this while still ensuring the anonymity of the real voters. Similarly to establishing the verifiability of the tally, the election authority publishes information on the bulletin board that allows such verification. In particular, the election authority commits to certain values before an unpredictable public random event produces the randomness for the random sampling.

Another important property for random sample elections is the *anonymity of the sample group*. This states that no one can learn who the real voters are. Random sample elections aim to achieve this with decoy voters that can interact with the election authority in exactly the same way as real voters. Hence they are indistinguishable from the perspective of an observing adversary. Interestingly, if the adversary can also interact with real and decoy voters, she can use this to her advantage as we explain in the following section.

Example 3. We present a simplified version of cut and choose for random sample elections, continuing the example from Section 3.2. For readability, in Figure 5a we present again the table that is only known to the election authority. We gray out this table's content to denote that the gray values are not visible on the bulletin board, but only known internally.

Of course, at the beginning of the election, some of these entries are not yet known. In a first phase, which takes place before the ballots are sent to the voters, the election authority fills in the first, third and fifth columns of the table in Figure 5a, while the second and fourth columns remain empty. The election authority then produces multiple copies of this table, $3k$ copies in this example, and randomly permutes their rows, resulting, for example, in the tables shown in Figures 5b–5d. Then, it encrypts each column of each table with a different secret key and publishes all the resulting encrypted tables on the bulletin board. At this stage, the bulletin board contains $3k$ tables where columns one, three, and five are filled in but the content is not yet readable by the auditors. The columns are encrypted in such a way that they hide the contents of the columns but they can later only be decrypted to the original plain text. With this mechanism, the election authority *commits* to the content without revealing it at this point.

Afterwards, the real voters are chosen, the ballots are sent to the real and decoy voters, and the voters cast their votes. Then, the second and fourth columns are filled into all $3k$ copies of the table, after the votes have been recorded. The resulting columns are again encrypted and published, such that the bulletin board now contains $3k$ full, but hidden tables; this concludes the "cut"-phase.

Next, in the "choose"-phase, the $3k$ tables are divided into three disjoint batches, each containing k tables, based on an unpredictable, random event. The membership of a table to a batch decides which of the table's columns must be revealed on the bulletin board for auditors to inspect. Each table in Figures 5b–5d represents one batch. The white columns depict which columns are revealed for all tables in this batch for the verifiability checks. The gray columns are never revealed. It is important that the event that determines which tables go into which batch is unpredictable so that the election authority cannot prepare the tables in such a way that all the checks go through even when the tables are inconsistent. Furthermore, it is crucial that the columns of all tables have already been committed to, since this allows an auditor to discover if the election authority has manipulated the tables after-the-fact. The following verifiability checks are used by this procedure.

In the first batch, depicted by the table in Figure 5b, the serial numbers and codes, their repetition in unused ballots, and the voted marks (white columns) are revealed on the bulletin board. This enables every voter to verify that his vote has been recorded as cast. For example, the voter V_r can verify that the ballot $200b$ was used to cast a vote (because the field in "print" is empty) and that the code 987 was marked as voted. However, no one else learns that the code 987 corresponds to the *yes* vote.

The published columns in Figure 5c enable voters to verify *print auditing*, that is that the ballots were printed correctly by the election authority. Each voter can check that the code-vote association of his unused ballot is correctly depicted by the table. For example, the voter V_r can check that for the ballot $200a$, the code 275 corresponds to *no* and 543 to *yes*, corresponding to the copy of the ballot he still has in his possession. This ensures that the election authority cannot forge votes by printing ballots incorrectly. In particular, because the authority cannot predict which ballot will be chosen by the voter, it cannot know which ballot must be revealed for the consistency check.

In the final batch, as depicted in Figure 5d, the last three columns of the tables are revealed. This enables all participants to verify the tally. In the example, everyone can see that there are two votes for *yes* and one of them has been sent by a decoy voter and will thus not be counted in the tally. [3] Note that because all tables have different row permutations, this procedure also ensures vote privacy. No auditor of the bulletin board can conclude, for example, that the voter with ballots 200a and 200b voted *yes* with code 987. ■

Note that although Chaum does not provide formal models, the protocol we have sketched (and his extensions) are sufficiently detailed that they can be appropriately formalized and verified from the information security perspective.

5 Economic perspective

In this section, we outline the economic analysis of random sample elections with decoy votes, explore the required security properties, and show that more sophisticated adversaries may violate some of the security properties of random sample elections with decoy votes.

5.1 Economic analysis

We illustrate the analysis of random sample elections. In the simplest setting with *private values* and *costly voting*, we consider a model that has the following features:

1. There are two alternatives ($\mathcal{S}$ and $\mathcal{P}$), representing candidates or issues.
2. The electorate is a given finite set N, which is randomly split into three subsets N_1, N_2 and N_3. Members of N_1 have the right to vote (henceforth called "sample group"), members of N_2 obtain decoy ballots (henceforth "decoy group"), and members of N_3 do not participate in the process. For any given set S, we use $|S|$ to denote its cardinality.
3. Voters $i \in N$ are of two types $t_i = \mathcal{S}$ and $t_i = \mathcal{P}$, that is, they either prefer $\mathcal{S}$ or $\mathcal{P}$.
4. A share $\lambda_{\mathcal{S}}$ prefers $\mathcal{S}$ and a share $\lambda_{\mathcal{P}}$ prefers $\mathcal{P}$, with $\lambda_{\mathcal{S}} + \lambda_{\mathcal{P}} = 1$.
5. Any voter i's utility is:

	t_i chosen	t_i not chosen
i votes	$1 - c$	$-c$
i does not vote	1	0

In this table, we have normalized the utility gain to 1 when the preferred alternative is chosen by the sample group. Voting is costly, as citizens need time to make up their minds and to vote. These costs are captured by the parameter c, $0 < c < 1$, which is assumed to be the same for all voters for illustrative purposes.

[3] The actual table in random sample elections is more involved and also includes information allowing one to ascertain that the right voters have been provided with ballots. We refer to [9] for further details, which are not relevant for this paper.

6. Real and decoy voters decide whether to abstain or to vote for one of the two alternatives. The votes of decoy voters are disregarded.

Finding the equilibria of the above game is the core of the economic analysis. For examples related to this game, see [16]. An immediate observation is that no voter will cast a vote against his preferred alternative. Building on equilibria outcomes, we can then make welfare comparisons relative to the standard voting system where all N citizens vote simultaneously, which serves as a benchmark. The equilibria can be used to assess whether the voting outcome will achieve a low quality or high quality of collective decisions and whether or not the election generates high costs for the citizens.

In the random sample elections game introduced before, we can immediately observe that the highest decision quality is achieved if and only if

$$\mathcal{S} \text{ is chosen} \Leftrightarrow \lambda_{\mathcal{S}} \geq \frac{1}{2}.$$

Regarding the costs, the best possible situation occurs when nobody votes. In this case, however, no democratic decision-making is possible. Accordingly, there is a trade-off between quality and costs.[4] Typically, this is resolved by a welfare function that incorporates these two objectives or, alternatively, by achieving a certain quality at minimal cost. In most of the well-established costly voting models, the voting outcome does not achieve particularly high quality and the margin between the votes cast for $\mathcal{S}$ and $\mathcal{P}$ is much smaller than the margin between the support for the two alternatives in the entire population. Intuitively, this can be explained as follows: If a voter is in favor of the same decision as most voters are, he will more likely not vote. He can save the cost of voting because it is probable that his favored choice wins anyways. The small difference between votes cast for $\mathcal{S}$ and $\mathcal{P}$ opens up great opportunities for Eve. By manipulating a small number of votes, Eve can arrange that her preferred alternative wins, even if the support for the other alternative is much larger in the entire population.

5.2 Implicit security properties

In the following, we review some standard assumptions that are typically taken for granted in the voting model in Section 5.1. We show that with the insights provided by information security analysis, these assumptions can be made explicit and can be proven to hold.

Economic models usually assume that the adversary does not interfere with the voting process. However, if one takes Eve seriously, it is easy to imagine different ways that she can affect the outcome of a collective decision directly. First, a small fraction of votes may be manipulated after they have been submitted by the voters, but before they have been made public. The severity of this problem increases the more a voting system tends to compress the vote

[4] In general, this does not hold for all citizens. A fraction of voters derives positive value from engaging in deliberation and voting.

margin, say by providing members of the majority with lower incentives to turn out than members of the minority. When margins are small, manipulating a few votes may suffice to change the outcome. As we have seen, the property of information security that denotes that no one can alter the votes after they have been cast is *integrity.* Additionally, one can require that everyone must be able to verify that this property holds. This is captured by the properties *individual* and *universal verifiability.*

Second, Eve may want to influence the selection of the voters in the sample group. To ascertain that a protocol is not vulnerable to such attacks, the sample group must be chosen *randomly*, and the *integrity* of the assignment of voters to the sample group must hold. Again, an additional requirement can be that these properties are *verifiable.*

Third, Eve may want to buy certain votes directly from the citizens. For this to be possible, she must have access to the voters' identities, who, in turn, need to prove to Eve that they have voted as agreed. Hence, both the *anonymity* of the sample group and *coercion resistance* are important properties.

Finally, Eve could try and send messages with political content to (targeted) voters to influence their evaluations of alternatives, and ultimately their decisions. This is related to the channel assumptions in the adversary model of information security. If we assume that there are only insecure channels from the election authority to the voter, then Eve could effectively influence voters by forging information as coming from the authority. If, however, the channels from the authority to the voter enable *message authentication*, then Eve cannot convincingly send messages as coming from the authority; this might decrease her chances to influence the voters.

For completeness, we summarize the security requirements needed for the successful implementation of random sample elections. They are: integrity and verifiability of the tally and the selection, random selection of the sample group, anonymity of the sample group, coercion resistance, and message authentication. We have just argued that economic models rely on these properties, which must be established by using the methods of information security. Conversely, as discussed in Section 4.2, information security sometimes assumes certain adversary capabilities that are based on economic reasoning, for example the argument that Eve will not buy votes if decoy votes are deployed because they make vote buying ineffective. Economic approaches can help to devise extremely sophisticated adversaries that exploit humans. We demonstrate that if we model a more sophisticated adversary, even with a very low budget she can break the anonymity of the sample group when decoy votes are used.

5.3 Vote buying

Decoy ballots have been advocated as a viable tool against vote buying. For instance, [19] analyze decoy ballots from a game-theoretic perspective and conclude that they are reasonably immune to vote-buying attempts by malicious adversaries facing budget constraints. In their analysis, they only consider simple attacks by the adversary: she sets a price at which she is willing to buy votes, both from real voters and decoy voters. With the help of a simple model, we briefly discuss how a more sophisticated adversary Eve can separate decoy votes from real votes in the process of vote-buying.[5]

Consider now that the electorate N is composed of risk-neutral citizens, which base their decision solely on expected gains. We also assume that $|N|$ is sufficiently large so that we can work with the law of large numbers, and we denote by p, for $0 < p < 1$, the percentage of citizens who have real votes. These voters are chosen randomly. The rest of the electorate obtains decoy votes.[6] We stress that the parameter p can be chosen by the election designer. Whether one's ballot is real or decoy is private information, and hence, there is no possibility for an outside agent (including Eve) to distinguish between the two types of ballots. For a voter i, let V_i be the utility he obtains from voting. If a voter i has a decoy ballot, his utility is $V_i = 0$. If a voter i has a real ballot, his utility is $V_i = V > 0$. The exact value of V is determined in equilibrium. We assume that the adversary's goal is to buy half of the real votes, which amount to a share $p/2$ of the population.

We consider two possible procedures employed by Eve. First, suppose that she offers each citizen a certain amount x in exchange for his vote. Clearly, if $x < V$, she will only obtain decoy ballots. Hence assume that $x = V$, so that all citizens who are offered the deal accept. In order for Eve to obtain half of the real votes, on average she then needs to offer x to a half of the population since decoy ballots and real ballots are indistinguishable. This means that Eve expects per-capita costs denoted by B where

$$B = \frac{V}{2}.$$

Second, suppose that Eve chooses an entirely different approach and uses so-called "Devil's Raffles", i.e. offering lotteries $L_k = (p_k, q_k), (k = 1, 2, ...)$ of the following kind: with probability p_k, the voter will receive a sure payoff q_k in exchange for his vote, and with probability $1 - p_k$ no transaction will occur and the voter (real or decoy) will keep his ballot. Consider now two lotteries L_1 and L_2 with

$$\begin{aligned} p_2 &:= \tfrac{1}{2} \\ q_1 &:= V - \varepsilon \\ q_2 &:= V + \varepsilon \end{aligned}$$

for some small value $\varepsilon > 0$. Moreover, let

$$p_1 := \frac{\varepsilon + p_2 q_2}{q_1} = \frac{\varepsilon + \frac{1}{2}(V + \varepsilon)}{V - \varepsilon}. \tag{1}$$

[5] The simple model we consider is different from, yet similar in spirit to, the one considered by [19].

[6] Thus we assume that $|N_3| = 0$. This is without loss of generality. Moreover, a full-fledged analysis reveals in our setting that all members of N_2 will apply for decoy votes.

Hence,

$$p_1 \cdot q_1 = \varepsilon + p_2 \cdot q_2 > p_2 \cdot q_2 = \frac{1}{2} \cdot (V + \varepsilon). \tag{2}$$

Thus, the expected payoff from choosing lottery L_1 is higher than that from choosing L_2.

Let us next examine the utilities of citizen i. On the one hand, if he accepts the lottery L_k, for $k \in \{1, 2\}$, he expects

$$\mathbb{E}[i \text{ sells his vote for } L_k] = p_k \cdot q_k + (1 - p_k) \cdot V_i. \tag{3}$$

If, on the contrary, citizen i does not sell his vote, he expects

$$\mathbb{E}[i \text{ does not sell his vote}] = V_i, \tag{4}$$

which is zero for decoy voters and V for real voters.

Since $V_i = 0$ for decoy voters, they will buy lottery L_1 since $p_1 q_1 > p_2 q_2$. For real voters $V_i = V$ and choosing lottery L_2 therefore yields the expected payoff

$$\frac{1}{2}(V + \varepsilon) + \frac{1}{2}V = V + \frac{1}{2}\varepsilon, \tag{5}$$

while selecting L_1 yields

$$p_1(V - \varepsilon) + (1 - p_1)V = V - p_1 \varepsilon. \tag{6}$$

Hence real voters will buy lottery L_2.

Eve will offer these lotteries to a share s of the population. In order to obtain, on average, half of the real votes again, s must satisfy

$$s \cdot (p \cdot p_2 + (1 - p) \cdot 0) = p/2 \Leftrightarrow s = \frac{1}{2p_2} = 1.$$

This calculation reflects that $p \cdot p_2$ is the probability that a real voter gives Eve his vote (in lottery two), whereas $(1 - p) \cdot 0$ is the probability that Eve receives a real vote from a decoy voter. The result makes sense: Real voters have a chance of $\frac{1}{2}$ to be able to sell their votes. Hence, the entire electorate must be invited to apply for the lotteries.

We next calculate Eve's expected aggregate costs. For this purpose, we make ε arbitrarily small and neglect it in the calculation. Then the expected budget amounts to

$$B = p \cdot p_2 \cdot q_2 + (1 - p) \cdot p_1 \cdot q_1 \approx p_2 \cdot q_2 = \frac{1}{2} \cdot q_2 = \frac{V}{2}.$$

We obtain two conclusions from an economics perspective. First, attacks with Devil's Raffles are useful to identify who has a decoy ballot and who does not have one because real and decoy voters choose the lottery L_2 and L_1 to sell their votes, respectively. Moreover, Eve can elicit p if it is not known to her with a small budget by selecting small values of p_1 and p_2. Second, regarding the budget needed to obtain half of the real votes: there is no improvement compared to the first procedure where a price is fixed at which a fraction of votes is bought. However, there are more sophisticated forms of Devil's Raffles that also lower the budget [13].

From the security perspective, we learn that a sophisticated adversary can buy votes, even in the presence of decoy ballots. Given this, a protocol using decoy votes is unlikely to provide coercion resistance unless other more effective mechanisms are in place. Repairing this problem would require a protocol redesign. Moreover, the economic analysis demonstrates that decoy votes violate the anonymity of the sample group. Thus even if coercion resistance can be established using decoy ballots, this mechanism should not be used when the anonymity of the sample group is important.

6 Outlook

Through examples, we have shown how the adversary Eve provides an effective interface between security and economics. In particular, information security focuses on what Eve can technically do in a system that incorporates security mechanisms with the aim of achieving security properties. In contrast, economic models investigate what Eve is rationally motivated to do in a self-designed game with the system's participants. We have illustrated how these two viewpoints can complement each other. Economic models implicitly assume security properties that can be made explicit and be proven by using the techniques of information security. Similarly, informal economic arguments motivating the adversary models used in information security must be analyzed with great care. The example of the decoy votes, which are supposed to avoid coercion, shows that sophisticated adversaries can design out-of-the box games that endanger other security properties, such as the anonymity of the sample group.

An important future research direction is certainly to investigate the wide spectrum of adversary models used in election research, their economic justifications, their effects on critical security properties, and as a consequence how voting protocols must be strengthened (or weakened). In addition there are serious concerns that go beyond the actual voting and tallying protocol. Free and fair elections [12] impose requirements before and after the election: including basic freedoms like those of free speech, free press, free movement and assembly, as well as more specialized rights like access to polls and protection from intimidation. Recent elections in America and France have shown that organizations and other countries can attempt to influence public opinion by propaganda or "fake news".

Such election hacking is a major challenge for democracy and an important research direction for both information security and economic research. We conclude with an illustration based on our example from Section 5.1. Suppose that Eve manages to send a message about the relative merits of the two alternatives $\mathcal{S}$ and $\mathcal{P}$ that is perceived to be from a trusted authority and affects through biased information ("fake news") individual evaluations of the alternatives. Assume in our random sample elections game that Eve can manipulate in this way a small fraction of the sample group's members. Two possibilities can occur. First, and less plausibly, assume that it is common knowledge among all voters that Eve has manipulated a fraction of voters who then vote as desired by Eve and that Eve's preferred alternative is also commonly known. Then, the other voters could adjust their decision whether to abstain or not and could—and would—neutralize this manipulation. Second, and more plausibly, assume that Eve's manipulation is hidden. Since vote margins are typically small in costly voting setups, such a hidden manipulation—even of a small fraction of voters—would affect the outcome significantly. This type of manipulation makes voting outcomes extremely vulnerable and developing adequate security countermeasures is a considerable challenge.

References

1. Anderson, R.: Why Information Security is Hard – An Economic Perspective. In: Proceedings of the 17th Annual Computer Security Applications Conference. pp. 358–365. ACSAC '01, IEEE Computer Society, Washington, DC, USA (2001), http://dl.acm.org/citation.cfm?id=872016.872155
2. Basin, D., Cremers, C.: Modeling and analyzing security in the presence of compromising adversaries. In: Computer Security - ESORICS 2010. Lecture Notes in Computer Science, vol. 6345, pp. 340–356. Springer (2010)
3. Basin, D., Cremers, C.: Know your enemy: Compromising adversaries in protocol analysis. ACM Trans. Inf. Syst. Secur. 17(2), 7:1–7:31 (Nov 2014), http://doi.acm.org/10.1145/2658996
4. Basin, D., Cremers, C., Meadows, C.: Model Checking Security Protocols, chap. 24. Springer-Verlag (2017)
5. Basin, D.A., Radomirovic, S., Schläpfer, M.: A complete characterization of secure human-server communication. In: 2015 IEEE 28th Computer Security Foundations Symposium. pp. 199–213. IEEE Computer Society (2015)
6. Beilharz, H.J., Gersbach, H.: Voting Oneself into a Crisis. Macroeconomic Dynamics 20(4), 954–984 (2016)
7. Blanchet, B.: An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In: Proceedings of the 14th IEEE Workshop on Computer Security Foundations. pp. 82–96. CSFW '01, IEEE Computer Society, Washington, DC, USA (2001), http://dl.acm.org/citation.cfm?id=872752.873511
8. Caballero, J., Grier, C., Kreibich, C., Paxson, V.: Measuring Pay-per-Install: The Commoditization of Malware Distribution. In: Proceedings of the 20th USENIX Conference on Security. pp. 13–13. SEC'11, USENIX Association, Berkeley, CA, USA (2011), http://dl.acm.org/citation.cfm?id=2028067.2028080
9. Chaum, D.: Random-Sample Voting, http://rsvoting.org/whitepaper/white_paper.pdf, accessed: 2017-07-07
10. Dolev, D., Yao, A.: On the Security of Public Key Protocols. IEEE Transactions on information theory 29(2), 198–208 (1983)
11. van Eeten, M.J., Bauer, J.M.: Economics of Malware: Security Decisions, Incentives and Externalities. OECD Science, Technology and Industry Working Papers 2008(1) (2008)
12. Elklit, J., Svensson, P.: What Makes Elections Free and Fair? Journal of Democracy 8(3), 32–46 (1997)

13. Gersbach, H., Mamageishvili, A., Tejada, O.: Sophisticated Attacks on Decoy Votes. Mimeo (2017)
14. Gersbach, H., Mühe, F.: Vote-buying and Growth. Macroeconomic Dynamics 15(5), 656–680 (2011)
15. Gordon, L.A., Loeb, M.P.: The Economics of Information Security Investment. ACM Transactions on Information and System Security (TISSEC) 5(4), 438–457 (2002)
16. Krasa, S., Polborn, M.K.: Is Mandatory Voting Better than Voluntary Voting? Games and Economic Behavior 66(1), 275–291 (2009)
17. Meier, S., Schmidt, B., Cremers, C., Basin, D.: The TAMARIN Prover for the Symbolic Analysis of Security Protocols. In: Proceedings of the 25th International Conference on Computer Aided Verification. pp. 696–701. CAV'13, Springer-Verlag, Berlin, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-39799-8_48
18. Oppliger, R., Schwenk, J., Helbach, J.: Protecting Code Voting Against Vote Selling. In: Sicherheit 2008: Sicherheit, Schutz und Zuverlässigkeit. Konferenzband der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), 2.-4. April 2008 im Saarbrücker Schloss. LNI, vol. 128, pp. 193–204. GI (2008)
19. Parkes, D.C., Tylkin, P., Xia, L.: Thwarting Vote Buying Through Decoy Ballots. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems. pp. 1679–1681. International Foundation for Autonomous Agents and Multiagent Systems (2017)
20. Schmidt, B., Meier, S., Cremers, C., Basin, D.: Automated Analysis of Diffie-Hellman Protocols and Advanced Security Properties. In: Proceedings of the 2012 IEEE 25th Computer Security Foundations Symposium. pp. 78–94. CSF '12, IEEE Computer Society, Washington, DC, USA (2012), http://dx.doi.org/10.1109/CSF.2012.25
21. Schweizer Radio und Fernsehen (SRF): Spurensuche nach dem Wahlbetrug im Wallis. https://www.srf.ch/news/schweiz/spurensuche-nach-dem-wahlbetrug-im-wallis, accessed: 2017-06-22
22. Schweizerische Bundeskanzlei: Anhang zur Verordnung der Bundeskanzlei über die elektronische Stimmabgabe. https://www.bk.admin.ch/themen/pore/evoting/07979/index.html?lang=de, Inkrafttreten: 2014-01-15, Accessed: 2017-06-16
23. Schweizerische Bundeskanzlei: Verordnung der Bundeskanzlei über die elektronische Stimmabgabe. https://www.admin.ch/opc/de/classified-compilation/20132343/index.html#app1, Inkrafttreten: 2014-01-15, Accessed: 2017-06-16
24. Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B., Meyer, G.: Protect: A Deployed Game Theoretic System to Protect the Ports of the United States. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. pp. 13–20. International Foundation for Autonomous Agents and Multiagent Systems (2012)
25. Stone-Gross, B., Holz, T., Stringhini, G., Vigna, G.: The Underground Economy of Spam: A Botmaster's Perspective of Coordinating Large-Scale Spam Campaigns. LEET 11, 4–4 (2011)
26. Tages Anzeiger: Wahlbetrug im Oberwallis – 30-jähriger Schweizer verhaftet. http://www.tagesanzeiger.ch/schweiz/standard/Wahlbetrug-im-Oberwallis--30jaehriger-Schweizer-verhaftet/story/14197130, accessed: 2017-06-22
27. Tambe, M.: Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned. Cambridge University Press (2011)