Realizing Security Requirements with Physical Properties: A Case Study On Paper Voting

André van Cleeff, Trajce Dimkov, Wolter Pieters, Roel Wieringa

Computer Science Department, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
{a.vancleeff,t.dimkov,w.pieters,r.j.wieringa}@utwente.nl

KEYWORDS : integrated security, physical security, security requirements engineering, KAOS, paper-voting, electronic-voting

Well-established security models exist for testing and proving the logical security of IT systems. For example, we can assert the strength of cryptographic protocols and hash functions that prevent attackers from unauthorized changes of data. By contrast, security models for physical security have received far less attention. This situation is problematic, especially because IT systems are converging with physical systems, as is the case when SCADA systems are controlling industrial processes, or digital door locks in apartment buildings are replacing physical keys. In such cases, it is necessary to understand the strengths, weaknesses and combinations of physical and digital security mechanisms. To realize this goal, we must first learn how security requirements are realized by the physical environment alone and this paper presents a method for analyzing this, based on the KAOS requirements engineering framework. We demonstrate our method on a security-critical case, namely an election process with paper ballots. Our analysis yields a simple ontology of physical objects used in this process, and their security-relevant properties such as visibility, inertness and spatial architecture. We conclude with a discussion of how our results can be applied to analyze and improve the security in other processes and perform trade-off analysis, ultimately contributing to models in which physical and logical security can be analyzed together.

1. Introduction

The automation of business processes is the replacement of physical events by digital events, and the replacement of physical entities by digital entities.[1] We automate because some properties of software, such as high speed, low cost and high accuracy, are more desirable than the corresponding properties of physical events and entities. Unfortunately, digital objects do not have uniformly better security properties than physical objects regarding the confidentiality, integrity and availability of information. For example, sharing information has become much easier with the Internet, making it more difficult to ensure the confidentiality of information. However there is no extensive and structured body of knowledge about what these physical and digital security properties are. This makes it difficult for system architects to perform the trade-off between physical and digital components and create optimally secure combinations. Understanding these trade-offs is becoming more important as IT systems are converging with the physical environment: smart buildings (with movement and temperature sensors), or door locks operable by mobile phone cannot be seen as purely digital or physical. This problem of understanding the trade-offs between physical and digital is nowhere more clear than in the context of voting systems, where the security of electronic voting systems has been heavily debated since their introduction. Can they -in any form- satisfy common voting security requirements such as vote secrecy and integrity of the process? The preliminary outcome of this debate as summarized in an ACM statement [1] is that completely electronic voting, without using any paper, is not capable of meeting those requirements sufficiently. Voters should be provided with a paper trail that they can inspect independently from the voting machines they used. Such arguments have led countries like Germany to abolish electronic voting altogether, favoring a complete paper-voting process. In some sense this is surprising, because the security of electronic voting has been studied extensively, whereas there is little scientific knowledge regarding the security of paper voting [2]. Indeed many countries including the United States and South Korea [3] continue to use e-voting, or are performing trials.

We will use paper voting as a case study and investigate its security characteristics, to draw conclusions about the impact that physical entities and events have on the realization of security requirements. Our contribution in this paper is threefold. We provide (i) a method for finding relevant security properties of the physical domain, (ii) an elaborated and refined explanation of the security of the paper-voting process and (iii) an ontology of physical items and their security-relevant properties. The ontology can help system architects to choose between physical and digital mechanisms for realizing security.

Section 2 presents a deeper analysis of the problem and of related work. We explain our research method for systematically identifying those properties of the physical domain that have a positive or negative impact on security requirements in Section 3. Sections 4 and 5 present the results of our case study. Our ontology is presented in Section 6. Finally, results are discussed and summarized in Sections 7 and 8.

[1] By a "digital X" we mean an X realized in software.

2. Related work

Investigating the security of physical processes has been done before and through various means. First, formal methods exist that researchers can use to build models of physical processes, of which they can prove certain properties. Secondly, if exact modeling is not possible, simulations can be performed, taking into account the uncertainty that comes along with physical processes. Third, actual elections can be studied for strength and weaknesses, and these results can be generalized to other cases. We first summarize this literature and then comment on it.

2.1 Formal models

As for formal models, a first body of related work concerns modeling procedures that span across the physical, digital and social domain. Probst et al. [4] and Dimkov et al. [5] have developed such models, which allow modeling the mobility of objects. Threats can span different domains, for example an employee receives an USB stick from a friend (social domain), and plugs it into a computer (physical domain), causing a security breach in the server (digital domain). The models allow formal verification of certain security characteristics, to find out whether certain attacks are possible, taking into account existing security policies.

In the area of e-voting security, Weldemariam and Villafiorita propose a method for analyzing procedural security [6]. Procedures are actions executed by agents on assets that can belong to both to the digital and the physical domains. To this effect, they create UML activity models to represent procedures and describe possible actions on assets. These assets are classified according to their mobility, evolution and number of instances, and can be either digital or physical. They define threat actions such as replacement and removal on these assets. Next, they extend the model with threats and asset flows and define the security objectives. Finally, a model checking approach based on NuSMV is used to assess the security of the procedures.

Bryl et al. also evaluate procedural alternatives [7]. Their objective is to mitigate the risk of introducing new security threats in new electronic procedures, and use the existing paper procedures as a point of departure. To this end, they combine process modeling in UML (use case, activity and object diagrams) with goal driven reasoning in the agent-oriented modeling tool Tropos. In particular, UML is used to model both existing “as is” and proposed “to be” processes, while Tropos is used in between to reason about design alternatives, both for providing a rationale for the chosen solutions, and for investigating security issues. The Tropos model is then transformed into a formal Datalog model, to automatically verify model properties.

2.2 Simulations

A different approach is proposed by Pardue et al., who advocate using simulation methods to determine the security of voting systems [8]. The first step is to create a threat tree: a hierarchical structure displaying the various means (starting with the leaves of the three) that a threat (the root node) can be realized. Nodes are connected by special *AND* nodes (all leaves must be realized) or *OR* nodes (one of the leaves must be realized). Experts then estimate the likelihood of the leaves (called *TERMINAL* nodes), which are tied to the motivation of attackers and the complexity of the attack itself. In turn, this information is used to run simulations using a Monte Carlo method for doing risk assessment and performing trade-off analysis between specific systems, both paper and electronic.

2.3 Election observation

A third source of information is studies of actual elections, of the attacks that took place, and defenses against them. A thorough discussion of paper voting was made by Harris [9] in his book on election administration in the United States. Knowledge about elections is also codified in manuals for election observers. These follow a checklisting approach, where officials assess the security of an election process by checking a long list of variables [10-12].

2.4 Analysis

Model checking and simulation provide insight in the properties of election procedures, but in order to do so they abstract away from the enormous complexity of physical processes. It is not known which properties we can safely abstract away from, and which are important. Election observers also make abstractions, because they typically list the threats and mitigations, but fail to explain why these threats are actually possible. What is missing is a systematic investigation of what physical properties are important to the voting process.

As an example, ballot theft is a well-known security problem of paper voting, as these ballots can be used to stuff the ballot box. Monitoring of the ballot box as well as strict voting ballot security can prevent this attack. However, this does not explain why such an attack is possible. Intuitively the attack is explainable by pointing out that the ballots must be manufactured prior to the voting, and lack any form of access control. In this paper we will apply a more precise form of argumentation, to arrive at more well-founded conclusions.

From that perspective, our research is inspired by the usage of Toulmin arguments for security requirements engineering by Haley et al. [13]. At the heart of a Toulmin argument is a specific claim, for example about the security of a system. This claim is supported on certain grounds and a warrant provides further arguments about the support of the grounds to the claim. This structure is naturally recursive as a warrant can also have grounds. In a similar way, we are interested in deepening the understanding of the security of physical processes, beyond normal argumentations about threats and mitigations. In the next section we will explain our approach towards achieving this goal.

3. Research method

This section presents the research method that we used to understand the impact of the physical environment in a paper-based election.

3.1 Steps in the analysis

We performed four analyses as indicated in Figure 1. Based on an analysis of the literature of voting in general we identified a tree of security goals for voting, which we represent as a KAOS goal tree. We then analyzed a particular case of paper voting, modeling the physical entities and agents that play a role in the voting process following the KAOS method. In addition, we modeled security threats to this process as KAOS obstacles. Finally, we modeled the role that physical entities play in realizing security goals as well as in posing security threats. In particular, we identified the properties of physical entities that contribute to their role as security mechanisms or as security threats. This resulted in an ontology of security-relevant physical entities and properties.

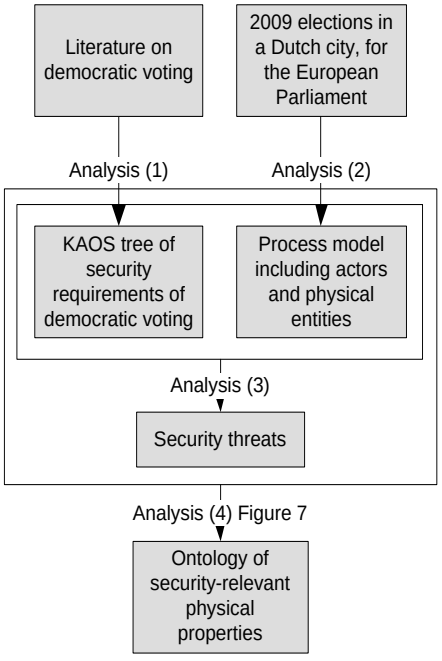

Figure 1: Steps in the analysis

3.2 Case description

To collect data about the paper voting process, we performed a case study. We chose to examine the election for the European Parliament, as held in the Netherlands in June 2009. In this election, over 12 million people were allowed to vote and 4.5 million actually voted. The reasons for selecting this case were threefold: first, the process is completely paper-based (exception for some software for the final tabulation). Second, independent reports about the election process security were available [14]. Third, we had access to election officials in one municipality, who supplied us with material used for training election officials [15] and answered questions about the process.

4. KAOS model of voting security requirements

4.1 The KAOS Requirements engineering method

We used the KAOS requirements engineering method [16] for modeling the physical entities and agents in the voting process. The main motivations for choosing KAOS were that it is not biased towards software (as we modeled a physical process), and has the notion of domain properties: characteristics of the domain that are relevant for the system because they contribute to (or detract from) goal realization. Properties can be domain variants (immutable characteristics such as physical laws), but also hypothesis that are to some extent context dependent. These and other KAOS concepts are listed in Figure 2.

A KAOS requirements model starts from goals, the objectives to be met. Goals are decomposed in subgoals, resulting in a tree-structure.[2] A goal is achieved if either all subgoals are achieved (*AND* decomposition) or one subgoal is achieved (*OR* decomposition). Goal realization can depend on the environment, which has characteristics called domain properties. Goals conflict when the achievement of one goal makes the realization of another goal harder. Obstacles hinder the realization of goals and can be resolved by setting other goals that prevent obstacles from occurring. (In security terminology, the term threat is similar to obstacle, and the term mitigation to resolution.) Finally, agents execute operations on entities.

KAOS term	Explanation
Object	Thing of interest in a composite system.
Entity	Identifiable and independent object.
Agent	Active participant in a process (an agent is a special type of entity).
Conflict	Situation when the realization of one goal hinders another goal's realization.
Domain Property	Descriptive assertion about objects in the environment of the system.
Goal	Objective to be met by cooperation of agents.
Obstacle	Condition (other than a goal) of which satisfaction may prevent another goal.
Operation	State transitions of objects performed by agents.

Figure 2: KAOS terminology (adapted from Van Lamsweerde [17]).

4.2 Identifying security requirements

Next, we reverse engineered the paper voting process, starting with top-level goals, and continuing with obstacles, agents and operations.[3]

Goals

To build the KAOS model, we first investigated the goals that a general voting process should realize [17-20]. We included goals as far as they are relevant and realized on Election Day itself and are not related to specific e-voting or paper voting procedures. We grouped goals by information security properties: confidentiality, integrity and availability. Added was the goal of assurance, the confidence that these properties actually hold, which is crucial for accepting the election result. Figure 3 shows the goal tree. Ultimately, voting processes contribute to the top-level goal of allowing citizens to take part in the government, either directly or by being represented through a representative (T1) [10]. Two subgoals realize this together: an election that satisfies all security goals (T2), and eligible voters actually voting in the election (T3). Except for the assurance goal (S1), all goal decompositions are *AND* decompositions. We will briefly explain a number of other goals.

To allow voters to cast their vote, resources such as the ballot box and the voting booth need to be available (A1). The process between the start of the election and the announcement of the results must be executed correctly, such that all legitimate votes are represented in the end result. This is the goal of integrity (I1). Among other goals, integrity requires legitimate votes (I4). A separate goal is that of accuracy (I2): the extent to which the transformation of votes into final results occurs without errors.

Confidentiality (C1) requires that the link between the vote and the voter is kept secret. Confidentiality is realized in two ways: first, voters should keep their vote private, which makes it impossible for others to buy their votes, or can coerce them to vote for a specific candidate, because voters cannot present a proof of how they voted.

[2] For brevity, we will not elaborate on the distinction between KAOS goals and requirements.

[3] Concerning the scope of the research, we focus on democratic voting inside a polling station using a voting booth. For a detailed investigation of remote voting (including postal voting) we refer to Puiggali and Morales-Rocha [20] and Krimmer and Volkamer [21].

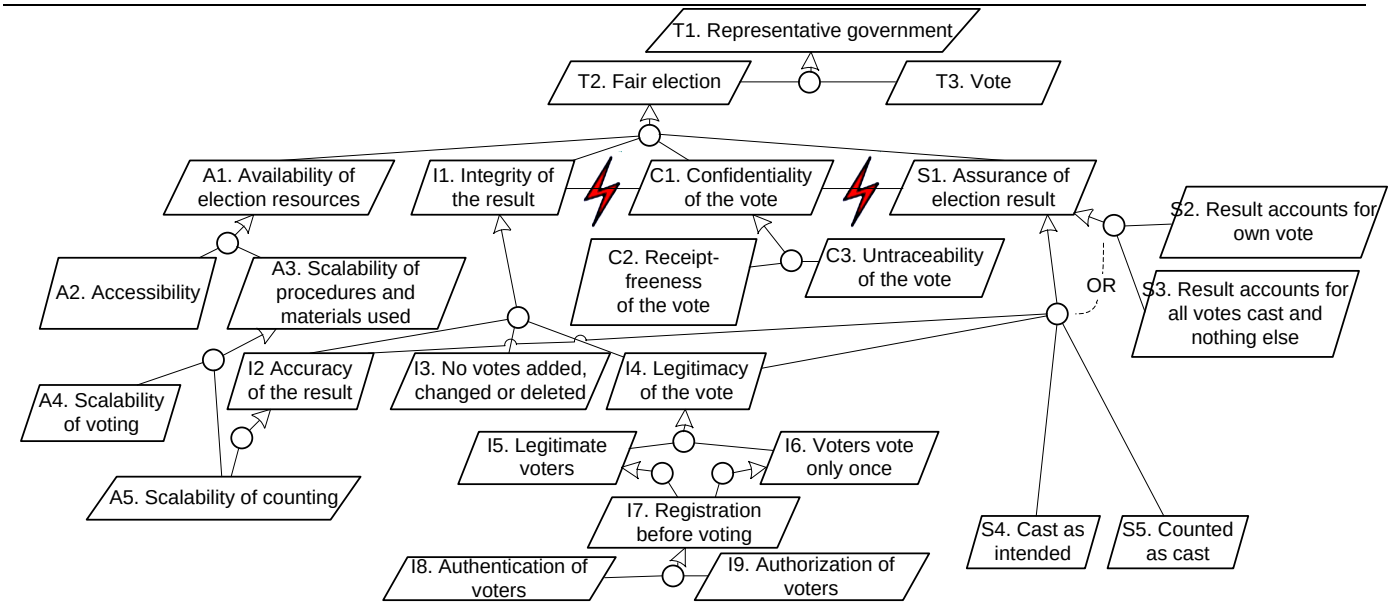

Figure 3: Top-level goal tree. The lightning symbol indicates a goal conflict.

We call this "receipt-freeness of the vote" (C3), i.e. the voter does not have a "receipt" to show how she voted. Second, others should not be able to deduce that a voter voted in a particular way. We call this "untraceability of the vote" (C2).

A key security problem in every election is that all agents in the process (voters, candidates and election officials) have an interest in the election result, being citizens subjected to the election outcome; *in elections there are no trusted third parties*. Thus, all parties must be assured (S1) that the election achieves the security goals. One approach is by ensuring confidence in each step:

- Legitimate voters: only eligible voters can vote (I4).
- Cast as intended: the votes are not changed after casting (S4).
- Counted as cast: the vote count reflects all votes cast (S5) and the counting is accurate (I2).

Goal conflicts

Solving goal conflicts is by definition not trivial, as one goal's realization makes it harder to satisfy another goal. In our requirements model, the goal of confidentiality (C1) is at odds with the goals of integrity (I1) and assurance (S1) of the vote. If each voter gets a signed receipt of her vote, and these receipts are publicly made available, the integrity of the election is easily assured, at the cost of confidentiality loss.

5. The voting process and security threats in the case study

After the creation of a general KAOS model of voting security requirements, we investigated a particular paper voting process (the 2009 European election in the Netherlands) and identified the physical entities and operations in it. Figure 4 shows some of the relevant entities in the election process such as pencils. Figure 5 shows the steps in the voting process as operations in KAOS, how they contribute to security, and their relations to the actors and entities.

ID	Name
B1	Voter
B2	Official
B3	Polling station
B4	Voting booths
B5	Ballot box

ID	Name
B6	Red pencil
B7	Ballots
B8	Election report
B9	Voting manuals
B10	Voter IDs

Figure 4: Entities used in paper voting (B1 and B2 are also agents).

ID	Operation	Goal	Agent	Entities
P1	Enter polling station	S1	Voter	Polling station
P2	Hand over ID	I8	Voter	Voter ID
P3	Authenticate voter	I8	Official	Voter ID
P4	Receive ballot	T3	Voter	Ballot
P5	Enter voting booth	C2	Voter	Voting booth
P6	Inscribe ballot	T3	Voter	Ballot
P7	Fold ballot	C1	Voter	Ballot
P8	Deposit ballot	S4	Voter	Ballot
P9	Exit voting booth	T3	Voter	Voting booth

Figure 5: Operations performed by agents on entities, and their contribution to goals.

Paper voting threats and mitigations

We also investigated threats (obstacles in KAOS terms) against the paper voting process described above, and considered mitigations (resolutions in KAOS terms) of them. Threats can be found in many sources, among those literature on e-voting [22-25], and election manuals [12, 15]. Specific paper voting threats are listed by Jones [26] and Harris [9]. We only examined threats that satisfied three criteria: (i) they occur on Election Day, (ii) they concern the paper voting process (and not e-voting), and (iii) threats are non-violent. Similarly to the goal tree, we group these threats based on the top-level security properties. Several key threats are summarized in Figure 6.

ID	Threat	Goal
T1	Marking ballots	C3
T2	Recording the vote	C2
T3	Chain voting	C2
T4	Unauthorized voting	I8, I9
T5	Adding, removing and changing ballots	I2, I3
T6	Inability to observe the process	S1

Figure 6: Threats to voting goals.

The first threat to the confidentiality of the vote is to mark the ballots (T1) such that they are traceable to a voter, for example by leaving fingerprints, which is mitigated by securely storing and destroying the ballots. Still, voters can either mark the ballots themselves (such as by voting in a unique pattern [27]), or others can pre-mark ballots, such as by having unique serial numbers.

The voting itself can also be recorded (T2), for example by using a cellphone camera [26] or by forcing voters to accept "assistance" in the voting booth. Chain voting (T3) is a specific threat in which a vote buyer hands a pre-filled ballot over to a voter, who casts it and delivers the blank ballot (which the voter received from the officials) in return, allowing the vote buyer to start a new "chain" [28]. Mitigating chain voting is done by marking ballots on handout and checking the mark on deposit.

Integrity threats include tampering with the vote registry, not performing authentication and authorization and not keeping track of who voted (T4). Mitigations are distributing unique authorization documents to voters, which are taken in by the officials on ballot handout.

An attacker can further stuff the ballot box with votes, or even swap the whole box (T5). Other threats are that voters receive more than one ballot, that cast votes are altered or that votes are removed from the ballot box, or substituted by others. Sealing the votes, having observers in the polling station and comparing records (how many people registered, how many voted) mitigate these threats. Observing the election process gives confidence in the election result: being denied access to the polling station, being unable to observer the voting and counting threatens the ability to observe the voting process (T6).

6. Finding security-relevant physical domain properties

Formally, in KAOS, a goal is achieved if all its subgoals, as well as domain properties and other assumptions are achieved:

(i) {Subgoals,Domain Properties,Assumptions}⊨ Goal

Obstacles can threaten the realization of goals. For a given goal, the list of obstacles is complete when the goal is realized if they do not occur:

(ii) {Obstacles,Domain.Properties}⊨ Goal

Finally, actors (persons, software programs) execute operations that contribute to the realization of the requirements:

(iii) {Specifications(Operations)}⊨ Goal

Because in our context no formal correctness proofs are possible, we need a systematic way to examine how the goals are satisficed and what the contribution is of the physical environment, namely the physical domain properties. In total, we used five steps to understand the effects of physical entities.

1. determine possible states of entities;
2. examine why entities help to realize operations;
3. examine why entities contribute to realizing goals;
4. examine why entities help to resolve related conflicts;
5. examine why entities play a part in the occurrence of related threats and mitigations;

Figure 7 shows these sub-steps in a schematic KAOS diagram. For all entities combined, we found 64 properties [21].

6.1 Applying the analysis

We show how we have applied these steps to one specific entity, namely the paper ballot. To begin, paper can be in several states (1), including "written" and "folded". Paper can also help to realize operations (2): because it is "portable", it can be deposited in the ballot box. Concerning the goals (3), the "folding" contributes to the confidentiality of the vote, and the paper's visibility contributes to the integrity of the voting process (depositing multiple ballots is detectable). As for resolving conflicts (4), the conflict between confidentiality and integrity is partly resolved because the ballot remains "unchanged" after the voter marks it in an anonymous way and deposits it into the ballot box. Finally, the threat of chain voting (5) is made possible because a person can "conceal" a ballot. Figure 8 shows a schematic overview of these characteristics.

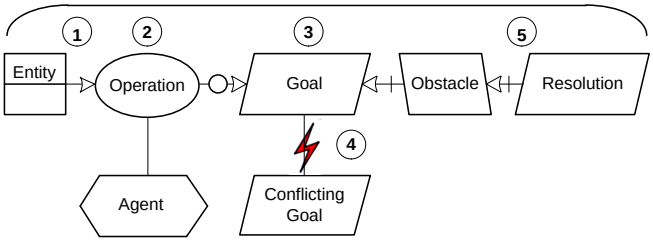

Figure 7: Schematic overview of a KAOS model and relation with the steps of our reverse engineering method.

Step	Characteristics	Impact
1. Possible states	Written, Folded	Confidentiality
2. Realize operations	Movable	Integrity
3. Realize goals	Folding	Confidentiality
	Visible	Integrity
4. Resolve conflicts	Inert	Confidentiality vs. integrity
5. Impact threats	Concealed	Integrity (Chain voting)

Figure 8: Effects of the paper ballot on security goals.

6.2 Resulting ontology and its application for explaining paper voting security

In the next step we combined the results for each entity in one ontology. Figure 9 summarizes our results and shows the complete taxonomy that we developed for the paper voting case using our method. All the properties are listed in Figure 10, next to their effect on security properties. Figure 11, which is specific for this case, shows the type of each object in our paper voting case and for each object how one of its properties mitigates a threat. For brevity we only list one such mitigation per entity, more are mentioned in our technical report [22].

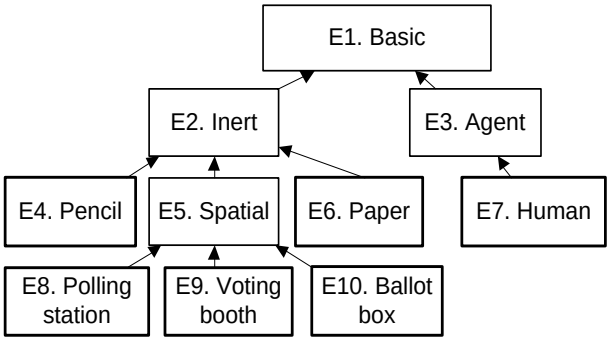

Figure 9: Taxonomy of physical entity types. E1, E2, E3 and E5 are abstract types.

Type	ID	Property	Impact
Basic	D1	Manufacturable	IS
	D2	Cohesive	I
	D3	Minimum size	S
	D4	Visible	CIAS
	D5	Destroyable	A
Inert	D6	Inertness	CIAS
Agent	D7	Active	
Person	D8	Move	
	D9	Non-deterministic	
	D10	Carry physical entities	
	D11	Modify physical entities	
	D12	Observe physical entities	
	D13	Convey information	
Spatial	D14	Architecture of opening (entry, exit, bandwidth)	CIAS
	D15	Architecture of containment	CIS
	D16	Architecture of internal arrangement	CIS
	D17	Architecture of observation	CIS
Paper	D18	Foldable	C
	D19	Writable from nearby	I
	D20	Writable with pen	I

Figure 10: Entity types, their properties and positive effects on security (C=confidentiality, I=integrity, A=availability, S=assurance).

ID	Entity	Type	Conflict resolution
B1	Voter	E7	D4→T5
B3	Polling station	E8	D16, D17, →T2, T5
B4	Voting booths	E9	D15 →T2
B5	Ballot box	E10	D14 →T5
B7	Ballots	E6	D6 →T2
B10	Voter IDs	E6	D1 →T4

Figure 11: Entities, their types and impact on threats.

These figures contain conclusions from our case analysis, of which we hypothesize that they are general: properties such as visibility and inertness are by no means limited to entities in the paper voting process. Furthermore, the effects of these properties on security can also be reproduced in a different context. Consider a hospital where a doctor stores a medical file. If no one has access to the location where it is stored (it is inert), the file will remain confidential. We provide additional arguments for this hypothesis of generalizability in the analysis below. Next, each entity type and property is described in further detail.

Properties of basic physical entities

Basic physical entities are created from matter and this process requires special equipment (D1). After creation they are cohesive, cannot change easily (D2). The entities also have a certain minimum size (D3) and combined with their cohesiveness, it makes them visible for persons (D4). Finally they can be destroyed but only using special equipment (D5). As an example, a ballot box cannot be duplicated easily, and retains shape during the election. Due to its size, it is visible for everyone, and is hard to destroy.

Properties of inert entities

Inert entities are a subtype of basic entities, and their characteristic is inertness (D6): they are inanimate, incapable of active behavior. Inertness has a positive effect on assurance.

As mentioned before, elections lack trusted third parties, and there is mutual distrust between all participants. However, they all trust the polling station, and its contents, such as ballot boxes and ballots to be inert, not under active control by another participant. Thus, the polling station functions as a neutral terrain, a "trusted third space", which solves the trust problem.

As only inert entities are used in the writing, the writings (using pen, pencil or stamp) remain unchanged on the paper and only recorded in the voter's memories and on the paper itself (D19, D20). Paper does not offer physical restrictions against writing and erasing marks, but the act of writing and the writings themselves are visible. The most specific contribution of the paper ballot to confidentiality is that it can be folded (D18), hiding the vote. Because the voter's fingerprints can be recovered, it can be argued that ballots are personally identifiable. However, this also requires equipment, which is observable.

Properties of agent and person entities

In our typology, persons are basic physical entities; they cannot be created easily and are observable. As agents, they are active participants that can influence and observe the environment (D7). Persons act non-deterministically (D9), their actions cannot precisely be predicted or controlled. They can move (D8), carry (D10) modify (D11) and observe physical entities. Finally, they are able to interpret a situation (D12) and communicate these observations to other persons (D13). Persons are observable, and can thus be held accountable for their actions. Because the ability of a person for action is limited, there is also a limit to the malicious impact one person can have on the whole process that runs for a limited time.

Access control is implemented using three key mechanisms:

1. By keeping people away from the paper - the ballot box helps in this fact, ballots cast into the ballot box cannot be touched.
2. By limiting the amount of tools that people have - no erasers should be allowed during counting.
3. By observing the process of the counting by other people - and because it is clear when people are deviating from the procedure.

Furthermore, persons' limited capacity for observation contributes to the confidentiality of the vote: with improved vision persons could observe too many details, threatening confidentiality. On the other hand, if election officials would be visually impaired, integrity and assurance are threatened: it would be much harder to detect someone dropping two ballots in the box. Thus, the conflict between integrity, assurance and confidentiality also exists in the physical domain, and physics solves it by allowing sufficient observations for assurance and integrity, but not too much to threaten confidentiality.

As for the importance of communication, no single person can observe the whole voting procedure. Participants must rely on other's observations, and communicate these. Other capacities of persons are also limited: they walk only slowly through the polling station, increasing the chance to observe their wrongdoing. Likewise, they can carry a limited amount of material such as paper, but cannot easily move a ballot box or voting booth, making it likely that these entities remain in place. Persons can also inscribe and fold paper, but very crudely, limiting the possibilities for covert channels.

Properties of spatial entities

The fourth physical type, inheriting all the properties of physical entities is the spatial entity. It consists of inert matter. Spatial entities have a specific architecture, which determines what entities it can contain (D15), how they can enter through an opening and how they can exit (D14), how they are organized inside the space (D16), and how they can be observed (D17). The entrance and exit properties determine the bandwidth of the space: how many entities can enter

and exit at the same time.
The voting booth can contain one or two persons at most, who enter through one side. The person stands in the booth facing the inside, with the paper in front of her, such that the process of writing cannot be seen clearly.
The bandwidth of the ballot box lid allows only a couple of papers ballots to pass at the same time, and it is very difficult to remove them if the lid is on. The contents of the ballot box is completely inert, thus the event of ballot reception is not recorded, but the voter (and observers) are assured that the ballot is cast as intended. Inside the box, ballots are piled on top of each other and they exit in more or less random order which further anonymizes the contents.
The explanation of the spatial entity type concludes our analysis of the security of paper voting. In the final sections we will discuss and summarize our results, point out the implications for e-voting and provide an outlook for future voting systems.

7. Discussion

Our ontology does not only help to understand known threats and mitigations; we can also discover new threats by considering how properties of entities such as inertness (on which voting security depends) are violated. For example voters cannot only violate the inertness assumption of the voting booth by using a smartphone and recording videos of their vote to sell it. RFID chips also can violate the inertness of the paper and communicate the vote to others. As such technologies become ubiquitous, we can predict that assuring the inertness property will be problematic in the future. How realistic these scenarios are is open for discussion, but they are technically feasible.
As for the application of our results beyond the current case, we first believe that our ontology can be used to assess and design the security of other voting processes. There are many mixed forms between fully automated voting processes and completely paper-based processes. For example an optical scan machine can automatically count paper ballots and other e-voting systems print and fill in the ballot for the voter, or even allow the voter to cast both an electronic and a paper ballot [29]. Designers of voting procedures can use our analysis to decide which parts should be automated (which positive security impacts of the physical environment can be discarded or can be improved upon) and which should remain physical.

8. Summary

In this paper we investigated how the physical environment contributes to the realization of security goals using a method based on the KAOS requirements engineering methodology. In a case study on paper voting, we examined how entities affect goals, obstacles, conflict and obstacle resolutions. This resulted in an ontology of physical entities with specific properties. Our analysis gives insight into common wisdom such as the importance of visibility for the paper voting process; it is *limited* visibility that helps paper voting security. We discovered that there are no trusted third parties in voting processes and that the *inert nature* of entities plays a key role in assuring security. Concerning paper voting, our results can be first used to better understand how to model and simulate voting processes described in Section 2.
As for generalizing the results, although we do not claim that our set of properties is complete, we do claim that these properties are general, in the sense that physical entities have these properties in other contexts too. Ultimately, our results should contribute to the development of a detailed integrated security model in which we can assess the security of integrated systems and perform trade-off analysis between different logical and physical components.
As future work, we intent to investigate physical and digital security further by examining virtualized systems. Virtualization introduces a software layer that decouples applications from the underlying hardware. This replaces physical protection mechanisms (such as physical separation) with digital mechanisms. We are interested to understand the security differences between virtualized and non-virtualized systems to further test and improve our ontology.

ACKNOWLEDGEMENT

This research is supported by the research program Sentinels (www.sentinels.nl). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

REFERENCES

1. J. Grove, “ACM statement on voting systems,” Comm. of the ACM, vol. 47, no. 10, pp. 69–70, 2004.
2. A. Yasinsac and M. Bishop, “The dynamics of counting and recounting votes,” IEEE Security and Privacy, vol. 6, no. 3, pp. 22–29, 2008.
3. K. Y. Kim, D. J. Kim, and B. G. Lee, “Pre-test analysis for first experiences of korean e-voting services,” in Future Information Technology, ser. Communications in Computer and Information Science, J. J. Park, L. T. Yang, and C. Lee, Eds. Springer Berlin Heidelberg, 2011, vol. 185, pp. 272–279.
4. C. Probst, R. Hansen, and F. Nielson, “Where can an insider attack?” in Formal Aspects in Security and Trust, ser. LNCS, vol. 4691. Springer Berlin / Heidelberg, 2007, pp. 127–142.
5. T. Dimkov, W. Pieters, and P. Hartel, “Portunes: representing attack scenarios spanning through the physical, digital and social domain,” in ARSPA-WITS, 2010.
6. K. Weldemariam and A. Villafiorita, “Procedural security analysis: A methodological approach,” Journal of Systems and Software, vol. 84, no. 7, pp. 1114 – 1129, 2011.
7. V. Bryl, F. Dalpiaz, R. Ferrario, and A. Mattioli, “Evaluating procedural alternatives: a case study in e-voting,” Electronic Government, an International Journal, vol. 6, no. 2, pp. 213–231, 2009.
8. H. Pardue, J. Landry, and A. Yasinsac, “A Risk Assessment Model for Voting Systems using Threat Trees and Monte Carlo Simulation,” in Proceedings of the 2009 First International Workshop on Requirements Engineering for e-Voting Systems. IEEE Computer Society, 2009, pp. 55–60.
9. J. Harris, Election administration in the United States. The Brookings Institution, 1934.
10. OSCE Office for Democratic Institutions and Human Rights (ODIHR), Guidelines for Reviewing a Legal Framework for Elections. ODIHR, 2001.
11. ------, Handbook for Domestic Election Observers. ODIHR, 2003, ISBN 83-912750-8-6.
12. ------, Election Observation Handbook, 5th ed. ODIHR, 2005, ISBN 83-60190-00-3.
13. C. Haley, R. Laney, J. Moffett, and B. Nuseibeh, “Security requirements engineering: A framework for representation and analysis,” IEEE Transactions on Software Engineering, vol. 34, no. 1, pp. 133–153, 2008.
14. OSCE/ODIHR Expert Group Report 11 – 30 May 2009, Elections to the European Parliament 4 - 7 June 2009.ODIHR, September 2009.
15. Ministery van Binnenlandse Zaken en Koninkrijksrelaties,

"Werkmap voor stembureauleden - versie 'stemmen in een willekeurig stemlokaal'," 2009, (in Dutch).
16. A. van Lamsweerde, Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley, 2009.
17. H. Jonker, "Security Matters: Privacy in Voting and Fairness in Digital Exchange," Ph.D. dissertation, University of Luxembourg/Technische Universiteit Eindhoven, 2009.
18. C. Kalloniatis, E. Kavakli, and S. Gritzalis, "Addressing privacy requirements in system design: the PriS method," Requirements Engineering, vol. 13, no. 3, pp. 241–255, 2008.
19. L. Langer, A. Schmidt, J. Buchmann, and M. Volkamer, "A taxonomy refining the security requirements for electronic voting: Analyzing helios as a proof of concept," in 2010 International Conference on Availability, Reliability and Security. IEEE, 2010, pp. 475–480.
20. J. Puiggali and V. Morales-Rocha, "Remote voting schemes: A comparative analysis," in E-Voting and Identity, ser. LNCS, vol. 4896. Springer Berlin / Heidelberg, 2007, pp. 16–28.
21. R. Krimmer and M. Volkamer, "Bits or paper? comparing remote electronic voting to postal voting," in EGOV (Workshops and Posters), 2005, pp. 225–232.
22. A. van Cleeff, T. Dimkov, W. Pieters, and R. J. Wieringa, "The security of paper voting," Universiteit Twente, Tech. Rep., Oct. 2011, in preparation.
23. California Institute of Technology and The Massachusetts Institute of Technology Corporation, "VOTING: What Is; What Could Be," 2001.
24. J. van Eerden and R. de Jong, Eds., Fraude en ongewenste beïnvloeding bij verkiezingen. Kiesraad, 2008, (in Dutch).
25. L. Norden, The Machinery of Democracy: Voting System Security, Accessibility, Usability, and Cost. Brennan Center for Justice at NYU School of Law, 2006.
26. D. Jones, "Threats to voting systems," in NIST Workshop on Threats to Voting Systems, 2005, http://vote.nist.gov/threats/papers/threats_to_voting_systems.pdf, Retrieved 2011-08-20.
27. D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. Rivest, P. Ryan, E. Shen, and A. Sherman, "Scantegrity ii: End-to-end verifiability for optical scan election systems using invisible ink confirmation codes," in Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop, 2008.
28. D. Jones, "Chain voting," 2005, http://vote.nist.gov/threats/papers/ChainVoting.pdf, Retrieved 2011-08-20.
29. O. Spycher, R. Haenni, and E. Dubuis, "Coercion-resistant hybrid voting systems," in 4th International Workshop on Electronic Voting, R. Krimmer and R. Grimm, Eds., Bregenz, Austria, 2010.