Journal of Computer Security 0 (0) 1
IOS Press

Modelling Human Threats in Security Ceremonies

Giampaolo Bella [a], Rosario Giustolisi [b,*] and Carsten Schürmann [b]
[a] *Dipartimento di Matematica e Informatica, Universitá degli Studi di Catania, Italy*
E-mail: giamp@dmi.unict.it
[b] *CISAT, IT University of Copenhagen, Denmark*
E-mails: rosg@itu.dk, carsten@itu.dk

Abstract. Socio-Technical Systems (STSs) combine the operations of technical systems with the choices and intervention of humans, namely the users of the technical systems. Designing such systems is far from trivial due to the interaction of heterogeneous components, including hardware components and software applications, physical elements such as tickets, user interfaces, such as touchscreens and displays, and notably, humans. While the possible security issues about the technical components are well known yet continuously investigated, the focus of this article is on the various levels of threat that human actors may pose, namely, the focus is on *security ceremonies*. The approach is to formally model human threats systematically and to formally verify whether they can break the security properties of a few running examples: two currently deployed Deposit-Return Systems (DRSs) and a variant that we designed to strengthen them. The two real-world DRSs are found to support security properties differently, and some relevant properties fail, yet our variant is verified to meet all the properties.

Our human threat model is *distributed and interacting*: it formalises all humans as potential threatening users because they can execute rules that encode specific threats in addition to being honest, that is, to follow the prescribed rules of interaction with the technical system; additionally, humans may exchange information or objects directly, hence practically favour each other although no specific form of collusion is prescribed. We start by introducing four different human threat models, and some security properties are found to succumb against the strongest model, the addition of the four. The question then arises on what meaningful combinations of the four would not break the properties. This leads to the definition of a lattice of human threat models and to a general methodology to traverse it by verifying each node against the properties. The methodology is executed on our running example for the sake of demonstration. Our approach thus is modular and extensible to include additional threats, potentially even borrowed from existing works, and, consequently, to the growth of the corresponding lattice. STSs can easily become very complex, hence we deem modularity and extensibility of the human threat model as key factors. The current computer-assisted tool support is put to test but proves to be sufficient.

Keywords: security protocols, formal methods, attacker models, Tamarin

1. Introduction

The use of technology in our daily activities is pervasive. Relevant examples exceed smartphones and include online food ordering services, unmanned bike-rental systems and automatic Deposit-Return Systems (DRSs) for cans and bottles through reverse vending machines, which provide the running examples for this article.

It is difficult to state precisely what the security of such systems means and what it would imply, as demonstrated by innumerable real-world attacks intertwining features of the technical systems with

*Corresponding author. E-mail: rosg@itu.dk.

specific human activity; for example, a train ticketing system can become insecure if passengers are dishonest and controllers are lazy [1].

Security ceremonies [2] extend technical systems with models of human users. They consider the possible combinations of human-and-system and human-to-human interactions so that the attack surface becomes prone to a broader range of threats which are deriving not just from vulnerabilities of the technical systems but also from the people who use these. While the term *Socio-Technical System* (STS) is broad, given to any instantiation of socio and technical elements engaged into any goal-directed behaviour, security ceremonies pose particular focus on the security goals, hence on the threat model of the system.

This article addresses the general challenge of how to model human threats in security ceremonies with the aim of studying the security properties more realistically than it has been possible so far, that is, in front of users who may raise threats beyond making errors, namely not only by disclosing information or passing objects but also by forging physical elements. While threats that humans raise against the technical system directly, namely by individual interactions with it, have already been investigated [3], the focus of this article is on their extension with threats that humans raise against the technical system indirectly, namely by interacting with other threatening humans without explicit collusion. In other words, we define and formalise threats both on human-to-system channels and on human-to-human channels. Real-world examples include the mentioned threat to the train ticketing system and, more in general, the threat that sees a human disclosing the contents of a ticket and another human exploiting them to forge a ticket with the same contents and feed this to a machine.

This article leverages two Danish DRSs as running examples. Both use a paper-based voucher system, generated by a reverse vending machine and refunded by the cashier. In order to define appropriate formal models for them, we had to reverse-engineer the DRS technologies in use because no design documents, implementation or process definition details were accessible. As a result, this article contributes to the formal analysis of security ceremonies both by theoretical advances and by applied, computer-assisted examples over DRSs using the Tamarin tool [4]. More precisely, this article extends our preliminary work on the security analysis of the Danish DRSs [5] with the following contributions.

(1) *Definition of distributed and interacting human threats in security ceremonies in epistemic modal logic, and their encoding in Tamarin*. Humans will be enabled to be *chatty* and reveal information, *cocky* and give out objects, *forger* and fabricate paper receipts or objects. Each will be modelled by appropriate rules in epistemic modal logic and encoded in Tamarin.

(2) *Definition of security properties of DRSs in metric first-order logic, and their encoding in Tamarin*. Properties *cash for voucher*, *cash for container*, *cash for purchase* say that, if a cashier cashes out a voucher then, respectively, a corresponding voucher has been printed, a corresponding container has been returned, or purchased. In turn, the last two properties can be strengthened through a bound to a specific customer, and all can be faithfully represented in metric first-order logic and encoded in Tamarin.

(3) *Formal analysis of DRSs against distributed and interacting human threats in Tamarin*. The DRSs deployed in Danish supermarkets Kvickly & Coop and Netto are analysed, and the latter is found to be stronger to some extent. However, both subvert the customer version of cash for container and both versions of cash for purchase. We also introduce a variant by leveraging a non-forgeable, electronic receipt and verify that it effectively supports all the given security properties.

(4) *Definition of a lattice of human threat models*. We found out that a lattice of human threat models exists, where each node stands for one or a combination of human threats. In particular, because

we allow for 4 different threat models to be combined, the cardinality of the nodes in the lattice is 16, and it is clear that not all pairs of nodes can be related to each other.

(5) *Definition of a search methodology, within the lattice of human threat models, for the maximal threat models not breaking the security properties*. When a property is found to fail against the strongest human threat model that includes all possible threats, a relevant question arises: what weaker threat models would the property withstand? And, in particular, what would be the maximal threat models that the property withstands? To answer these, it is necessary to traverse the lattice and reiterate the formal analysis of the properties in each node, namely against the threat model that the node represents. We have devised an efficient methodology to do this.

The structure of what follows is simple. Section 2 introduces the distributed and interacting human threat model. Section 3 explains how to derive models for DRSs. Section 4 unfolds the formal analysis. Section 5 discusses the related work and Section 6 ends with a few concluding remarks.

2. A Threat Model for Interacting Humans

2.1. Security ceremonies

The term *ceremony* has been originally introduced as a network protocol whose nodes are not limited to computers but also include humans [2]. Thus, links are not limited to traditional computer-to-computer communication channels but also encompass human-and-computer and human-to-human interactions. In short, nothing is out-of-band into a ceremony.

Ceremonies are of special interest for the security community because they introduce two novel challenges to the well-established area of security protocol modelling and analysis. First, they pose the problem of modelling human nodes. Ellison observes that while human nodes can have state and state machine like computer nodes, they are difficult to model because messages processed by human nodes are often subject to error thus human nodes cannot be programmed as done with computer nodes. Second, ceremonies introduce the problem of modelling an adversarial environment that takes into account human nodes as well as their interactions with other human nodes and computer nodes. While this enables a richer and detailed analysis of a ceremony, it might introduce inconsistent scenarios. For example, an adversary able to modifying a "speech" packet in a human-to-human interaction is unrealistic when such interaction happens in person [6].

In this paper, we assume that all technical components work as intended, namely, computer nodes are honest. However, we do not exclude from our analysis a Dolev-Yao adversary [7], which is allowed to attack such nodes. We focus on how to model the human nodes and their interactions with other human and computer nodes as well as how to specify the threats that originate from such human nodes.

2.2. Threat Modelling

The human participant in a security ceremony can be easily understood as follows:

Honest is a human who follows the rules of a given ceremony precisely, hence without posing any threat.

We assume that not all humans are honest, hence the quest to model malicious deviations from honesty. Our model of human threat rests on the observation that information as well as objects may be abused to subvert the ceremonies. Threats are captured by four main models, as described below:

Chatty is a human who discloses their own information, including personal data and any other relevant details available to the specific ceremony under scrutiny, such as the contents of a ticket. It is relevant because it enables the analyst to assess the extent to which information alone can be sufficient to break a security ceremony. For example, a chatty human breaks password-based authentication, a finding that would clearly call for stronger measures, such as multi-factor authentication.

Cocky is a human who gives out own objects that are relevant to the given ceremony, such as specific vouchers or other objects that would constitute an advantage in executing the various steps. By modelling a cocky human, the analyst can verify whether a security ceremony relies on object possession appropriately or excessively. For example, while a cocky human breaks authentication based upon possession of some physical token, augmenting the latter with more defences in depth, such as some liveness and freshness detection measures, may resolve the issue.

Receipt forger ("Rforger" in brief) is a human who counterfeits printouts out of known information. While an Rforger cannot invent relevant information, they can translate the information they have gathered, for example from a Chatty, into the relevant piece of paper that the given ceremony would accept, for example, as in the case of a traditional train ticket. This type of threat is easy to embody by purchasing a thermal printer. In fact, an Rforger may succeed when owning the same printout as the legitimate owner is the only security measure, but not against ceremonies that record the number of times that the printout is used.

Object forger ("Oforger" in brief) is a human who counterfeits objects out of known information about them. So, an Oforger can build a passport once they know all relevant data, including an eID if they know the cryptographic material that it ought to contain. Once more, while an Oforger may succeed in breaking some ceremonies, an Oforger may not do so when the target ceremony prescribes stronger authentication measures, for example based upon knowledge of information that the Oforger does not necessarily have.

A few remarks are due. First of all, at the typical symbolic level, the models do not need to capture the human's goal, namely to express whether the disclosure of information that a Chatty operates is intentional or not, or whether Cocky, Rforger or Oforger are driven by fun or profit — unless such features explicitly characterise the analyst's research questions.

The four models of human threat are, in general, unrelated to each other, so that each human may follow just one or a subset of them without any limitation. More precisely, each human acts as Honest and, additionally, may follow some of the threat models, all of them at an extreme. We shall see below that the various combinations of threat models form a lattice, and it is interesting to study which precise combinations, if not the strongest possible, break or not the given security properties.

Our overarching assumption sees *every* human potentially malicious, hence able to execute some of the threat models. In consequence, a security ceremony will have to withstand scenarios that combine two or more humans who are malicious, for example, a human who is Chatty with another who is Rforger, or a human who is Cocky with another who is Oforger. Also, one may consider a more general forger who yields the capabilities of both Rforger and Oforger. While it should be possible to specify a more general definition of forger, we advocate for a neat separation of the forging capabilities. As we shall see later, such separation would yield a more precise tool analysis as per the lattice of threat models. Moreover, a more general forger can be already obtained by considering the threat combination Rforger *and* Oforger.

It is clear that a human may fetch additional objects either because he or she is an Oforger and manages to forge the target object or, equally, because someone else is Cocky with them and gives them the same object. However, our models allow for a variety of scenarios, for example where the Oforger only has some of the necessary information to forge the object, hence they find it more effective to exploit another human who is Cocky and provides the object straight away.

2.3. A primer on epistemic modal logic

This section introduces a formulation of epistemic modal logic to describe formally the human threat models introduced above distinguishing between *knowledge* and *possession*: principals may have knowledge of the truth of facts, they might be in possession of physical objects, or both. Appropriate rewrite rules capture the laws by which principals can learn, create, or destroy. The idea of using epistemic logic in modelling principals is not new. It is easy to think about the Dolev-Yao model in this logic, and it has been used, for example, to express security policies in a proof-carrying file system [8].

Formally, our version of epistemic logic is based on (multiplicative) linear logic, which is sometimes called the logic of resources. In linear logic, assumptions are consumed whenever they are used, and therefore also called *resources*. Linear implication, written as $F \multimap G$ is similar to classical implication, except that the resource F can only be used once and only once in the proof of G. Linear conjunction is written as $F \otimes G$, and it requires that all available resources are split into two parts, one part is consumed in the proof of F and the other in the proof of G. While 1 is the multiplicative unit (truth) in linear logic, the aforementioned modalities for knowledge and possession must be added. Therefore, $[\![K]\!]F$ expresses that *principal K knows fact F*, and $[K]G$ to be read as *principal K is in possession of physical object G*. In this article, principals can refer to humans as well as to systems. In this logic, we write $P\ (t_1, \ldots, t_n)$ for predicates, where t denote terms, x variables, $(t_1, \ldots, t_n)$ tuples, and $f\ (t_1, \ldots, t_n)$ for n-ary function symbols. If $n = 0$ then we call a predicate a logical constant and omit the empty trailing parentheses. We write $\forall$ and $\exists$ for the first-order quantifiers ranging over individuals, and Π to quantify over principals. These constituents can be formalised as follows:

$$\begin{array}{lll} \text{Terms} & t ::= x \mid (t_1, \ldots, t_n) \mid f\ t \\ \text{Formulas} & F, G ::= 1 \mid P\ t \mid F \multimap G \mid F \otimes G \mid [\![K]\!]F \mid [K]F \\ & \qquad \mid \forall x.\, F \mid \exists x.\, F \mid \Pi K.\, F \end{array}$$

2.4. Formalisation of the threat model

The different *human threats* can be formalised now by declaring the adversarial traits as logical constants that may be assigned to a principal, which could either be a human or a system:

$$K ::= \mathit{chatty} \mid \mathit{cocky} \mid \mathit{Rforger} \mid \mathit{Oforger}.$$

For example, we would write $[\![\mathit{chatty}]\!]\mathit{info}(o)$ to express that a chatty principal knows some information about an object o. Differently from the standard Dolev-Yao model, there is not one single attacker, such as the network attacker, but principals are given adversarial capabilities according to their categorisation. For each rule defining the human threat model, we specify, using the quantifier for principals, the specific threats. Similarly to the Dolev-Yao model, these models will have the capabilities to interact with the environment, not to encrypt or decrypt messages, but to learn and possibly replicate physical artefacts useful to achieve individual goals.

We distinguish between three unary predicates. We write *object(o)* for a physical object that is defined by o. To simplify the presentation, we leave the object information abstract. However, it is possible to enrich it with as many details as are necessary, depending on the specific application. The object itself aims to be paradigmatic and can be replaced as appropriate for the given system. Receipts are modelled by the predicate *receipt(r)*. For example, in a DRS, r may contain a QR-code with information about the object being returned. Here, we will use *QR(o)* as a 1-ary function symbol whose input is the information of a particular object. Thus, we use function symbols, such as *QR*, to express constraints on the general object. We follow the principle of being as general as possible, but when we are too general, we express constraints. In fact, distinguishing between object and receipts yield a more domain-specific threat model (e.g. see Cocky's rules) and a more fine-grained analysis as per Section 4. The methodology that we propose can be used in other context as well, although rules might need to express different constraints than the one advanced in this article to capture a different application domain.

Learner's rules. Honest humans must be augmented with additional rules if we want to enable them to exercise some threats in security ceremonies. The essential augmentation allows each human to learn information from physical objects that they may have. However, this can be hardly considered a threat because every human may learn by looking at objects and reading receipts, yet there may be systems that work smoothly even without prescribing their users to learn anything, such as car park payment systems based upon tokens. The learner's general rules are the following *Look* and *Read* rules, which allow a human to learn the defining characteristics of the *object* or the information stored within a QR-code, both abbreviated by o. It can be easily imagined how these rules could be refined to model a more sophisticated threat

$$(\mathit{Look}) : \Pi P.\, \forall o.\, [P]\mathit{object}(o) \multimap ([\![P]\!]\mathit{info}(o) \otimes [P]\mathit{object}(o))$$

$$(\mathit{Read}) : \Pi P.\, \forall o.\, [P]\mathit{receipt}(\mathit{QR}(o)) \multimap ([\![P]\!]\mathit{info}(o) \otimes [P]\mathit{receipt}(\mathit{QR}(o)))$$

Chatty's rule. The *Chatty* rule models a participant who shares part of his or her knowledge with someone else. Predicate *info(o)* models all information about object o monolithically, but an application-driven refinement of this would be easy to introduce:

$$(\mathit{Chatty}) : \Pi P.\, \forall o.\, [\![\mathit{Chatty}]\!]\mathit{info}(o) \multimap [\![P]\!]\mathit{info}(o)$$

Cocky's rules. Next, we define two rules for passing physical objects from one principal to another, in line with what we have dubbed "cocky" human behaviour. The *give* rule is conveniently defined for *objects* whereas the *hand* rule is defined for receipts. This level of detail may favour trace inspection once a formal analysis effort finds a relevant scenario. Of course, these rules may have to be adjusted if our human threat model is applied to other settings. The linear nature of epistemic logic ensures that principal P no longer has access to the *object* after the *give* rule was applied, and similarly for the *hand* rule:

$$(\mathit{Give}) : \Pi P.\, \forall o.\, [\mathit{Cocky}]\mathit{object}(o) \multimap [P]\mathit{object}(o)$$

$$(\mathit{Hand}) : \Pi P.\, \forall r.\, [\mathit{Cocky}]\mathit{receipt}(r) \multimap [P]\mathit{receipt}(r)$$

Forger's rules. Finally, we present two additional rules that allow a human forger to replicate physical objects, for example, by means of a regular printer, a 3D printer or other more sophisticated techniques. The first rule allows the human to build new objects, solely from the description o of an object that the human *Oforger* needs to know. This models the threat of an adversary being able to replicate any kind of object. The second rule allows an *Rforger* to *print out* receipts from the knowledge of object o, for example, a QR code that contains information from o. Note how the knowledge modality only refers to predicate *info*, whereas the possession modality refers to predicates *object* and *receipt*, which refer to physical entities:

$$(Build) : \Pi P.\, \forall b.\, [\![Oforger]\!] info(b) \multimap [P] object(b)$$

$$(Print) : \Pi P.\, \forall o.\, [\![Rforger]\!] info(o) \multimap [P] receipt(QR(o))$$

Epistemic modal logic is mainly concerned with reasoning about knowledge and possession. However, there exist extensions that consider time, location, and belief. These extensions are relevant for modelling more complex attackers, such as an attacker that can take advantage of the location of the victim.

Having seen a formalisation of a threat model for humans, we study our threat model against two different Danish DRSs, which are introduced in the next section.

3. Modelling the Deposit Return Systems

A DRS allows one to get rewarded for returning a container through Reverse Vending Machines (RVMs). It typically sees a set of human principles (e.g. customers and cashiers) and a set of technical principals (e.g. RVMs and servers).

In Denmark, the deposit return scheme is typically implemented by supermarket chains through RVMs. The customer experience is the same independently of the supermarket chain where they return their containers. However, different supermarket chains use different technology, and hence the technical protocol may vary, although this is transparent to the customer. For example, RVMs deployed in Kvickly and Coop supermarkets are similar, but they produce different vouchers compared to the RVMs deployed in Netto supermarkets.

Since there is hardly any information about the technology behind DRSs available, besides a few patents, this work follows a reverse engineering approach and reconstructs the technical aspects and the ceremony of DRS. In particular, this work adopts the road map for reverse engineering proposed by Müller et al. [9] and focuses on field observation as a primarily investigative technique to gather information regarding the ceremony.

3.1. Reverse Vending Machines

Reverse vending machines (RVM) are the main technological element in the DRS, hence it is essential to gather as much information as possible regarding the functioning of RVMs to build a correct ceremony. Most of the RVMs in Denmark are built by Tomra, and their specifications available to the public in the form of patents. This work considers three Tomra machine models: T-710, T-820, and T9. Every machine is built into a wall, which has a room on the other side which can be accessed through a locked door. An RVM can either accept a single empty container at a time or a beverage crate. Each container is validated on the basis of its weight, barcode, and size. In general, an RVM accepts only glass containers

Fig. 1. An example of a voucher printed by a Tomra T-710 machine

that have a barcode. The sole exemption is the traditional shape of the Danish beer bottle, which does not need to be equipped with a barcode for being accepted. Cans, instead, are accepted with or without barcode. However, the latter case entails no reward for the customer.

From a security perspective, Tomra has filed several patents for detecting fraud attempts in reverse vending machines [10–12]. However, the effort is almost exclusively concentrated on making sure that the machine does not accept invalid containers. Thus, we can assume that no RVM would accept an invalid container. Such an assumption can be confirmed by our observations of the machines. In particular, we had access to look through one of the Tomra RVMs while being emptied from its containers.

In Denmark, RVMs are equipped with thermal printers that print paper vouchers. A voucher attests the number of containers filled by the customer and entails a reward to them. An example of a voucher is in Figure 1. A voucher includes the following information

- The redemption value in Danish kroner
- A machine-readable serial number (SN1)
- The number of containers
- The model of the RVM
- A non-machine-readable serial number (SN2)
- Time and date of the printing of the voucher

3.2. Reverse engineering the machine-readable serial number (SN1)

To the best of our knowledge, there is no document covering how the RVM generates the information included in the voucher, especially how the serial numbers are generated. According to the patents filed by Tomra [13, 14], the company has implemented some security measures against presentation of home-made vouchers. In particular, some RVMs implement voucher control by means of a communication from the RVM to a cloud-based service solution provided by Tomra [15]. Once the filling of the RVM is completed by the customer, the RVM generates the voucher and sends both redemption value and SN1 to the Tomra servers. When later the voucher is presented for rewarding, this is controlled against the

Tomra server, which authorises the payment to the customer. According to the patents, other solutions that do not require constant communication with the Tomra server may be implemented. For instance, the RVM can be set to communicate locally with a computer hosted at the store premises, which periodically updates the list of valid vouchers in point of sale stations.

Since no public specification of SN1 is available, we have derived it empirically by analysing the vouchers printed by the different Tomra machines this work has taken in consideration (i.e. T-710, T-820, T9) hosted in three different stores (i.e. Kvickly, Coop, and Netto). Kvickly and Coop belong to the same supermarket chain. In our case, the Kvickly store hosts three T-710 machines, the Coop store hosts two T-820 machines, and the Netto store hosts one T9 machine.

Kvickly and Coop supermarkets. Several vouchers with different values were collected at different times and dates. A sample of the batch of vouchers collected at Kvickly from a T-710 is in Figure 2a. It can be seen that, independently from date and time, SN1 is fixed when the RVM is filled with one container worth of 1.00 Kr. However, SN2 still slightly changes. This is because the three vouchers in Figure 2a were printed by three different machines. This is confirmed by the second batch of vouchers (see Figure 2b) obtained from the same store. The second batch also reveals that SN1 slightly changes accordingly the value of the containers filled in the RVM. The first nine digits are always fixed while the 10^{th} and the 13^{th} digits change. It can be seen that the 10^{th} digit represents the total value of the voucher. It is also confirmed that the same approach is used at the Coop supermarket as depicted in Figure 2c. Here the 2^{nd} digit of the SN1 digits changes because the voucher is printed in a different store. However, the rest of the SN1 reflects the value of the containers.

Finally, in order to fully predict all the digits of the SN1, it is necessary to understand how the last digit is generated. We found that the last digit $\mathrm{SN1}_{13}$ is the check digit from the EAN-13 standard, which can be computed as

$$\mathrm{SN1}_{13} = x - y \text{ where}$$
$$y = \mathrm{SN1}_{[1\ldots12]} \cdot [1\ 3\ 1\ 3\ 1\ 3\ 1\ 3\ 1\ 3\ 1\ 3] \ \wedge\ x = \lceil y \rceil \text{ s.t. } x \mathit{ mod } 10 = 0$$

For example, the SN1 in Figure 1 can be computed as

$$\begin{aligned} y &= [2\ 3\ 3\ 9\ 9\ 0\ 0\ 0\ 0\ 4\ 0\ 0] \cdot [1\ 3\ 1\ 3\ 1\ 3\ 1\ 3\ 1\ 3\ 1\ 3] \\ &= 2 + 9 + 3 + 27 + 9 + 0 + 0 + 0 + 0 + 12 + 0 + 0 = 62. \\ x &= \lceil 62 \rceil = 70 \end{aligned}$$
$$\mathrm{SN1}_{13} = x - y = 70 - 62 = 8.$$

Netto supermarkets. The T9 machine at Netto generates the SN1 in a different way. It can be seen that the value of the voucher is not anymore reflected on any of the SN1 digits. Instead, by analysing three vouchers printed in sequence by the machine, we found that the SN1 is implemented as a counter that increments by one unit every time a voucher is printed. Notably, the SN1 digits can be also fully predicted at Netto stores since the last digit of the SN1 is still a check digit from the EAN-13 standard.

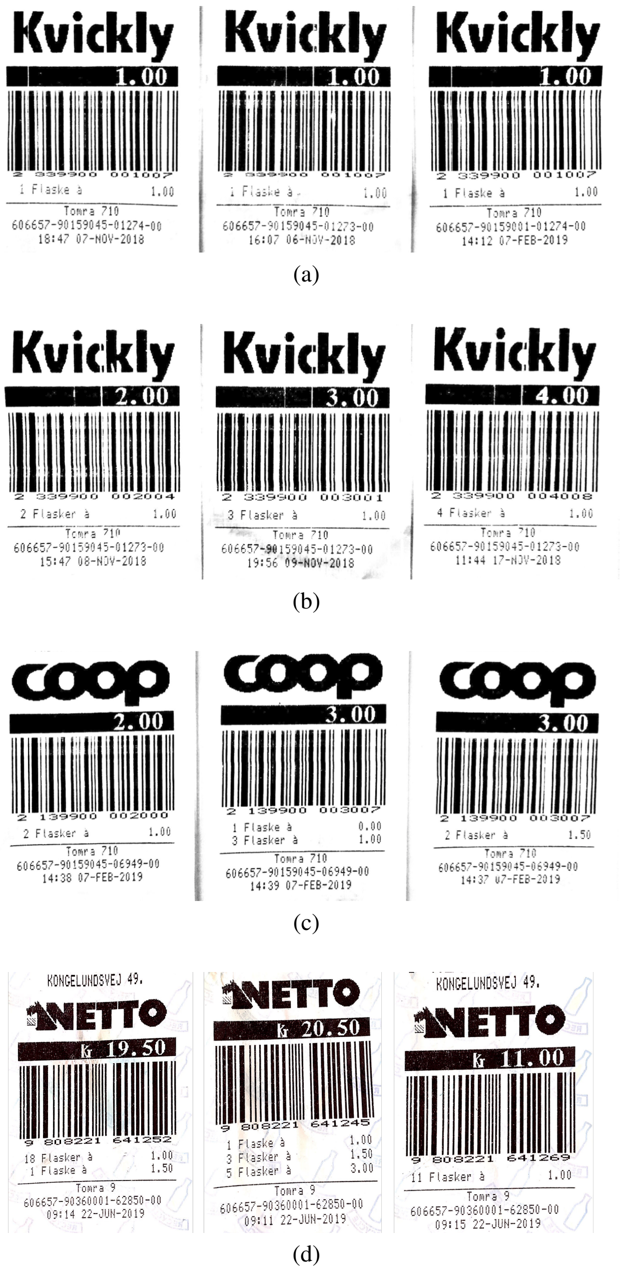

Fig. 2. The four different batches of vouchers obtained from different Tomra machines at Kvickly, Coop, and Netto. (a) the SN1 digits are fixed in each voucher; (b) some of the SN1 digits reflect the value of the voucher; (c) only the 2nd digit differs among Kvickly and Coop stores; (d) the SN1 digits increment by one unit at Netto

Discussion. The analysis of the vouchers printed at Coop confirms that the SN1 and all the other information printed in the vouchers can be fully predicted. Since a voucher can be redeemed at any of the stores of the same supermarket chain, we can rule out that barcodes are sent to the store's local computer. Also, since any two vouchers with the same value turn out to contain the same SN1, it is

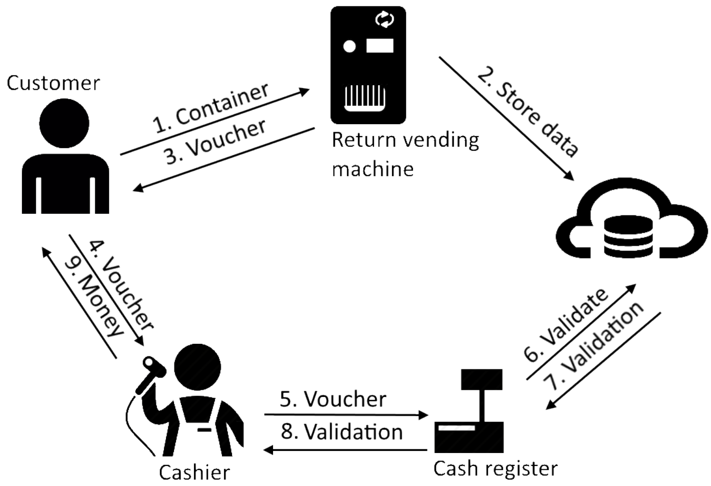

Fig. 3. The Danish deposit return system ceremony

unclear how the Tomra servers can prevent a fake voucher from being redeemed provided that other vouchers with the same value were printed. We believe that in this case there is no communication from the RVM and that the scanner reads the value of the voucher from the SN1 only. However, as we shall see later, we assume that such communication exists in the formal analysis of the Kvickly and Coop DRS ceremony.

Netto stores have a different way of generating the SN1, and the value is not stored in the SN1. Thus, we believe that the RVM should communicate to either Tomra servers or a store's local computer the details of the voucher. However, as for Kvickly and Coop stores, the SN1 is still fully predictable, but in this case one needs to know the value of the counter of the RVM.

3.3. Ceremony description

Having seen the descriptions of RVM and SN1, we can first present a full description of the ceremony, as depicted in Figure 3, and then appreciate the differences between the Netto and Kvickly & Coop ceremonies thanks to our notation in epistemic modal logic.

A ceremony begins with the customer approaching the RVM and inserting a number of containers (step 1). The RVM may either accept or reject each of the containers. It will stop accepting new containers when either the customer pushes the button to complete the filling phase or the RVM is full and cannot accept further items. Then, the RVM generates the data to be printed in the voucher, and, optionally, sends them to the Tomra servers (step 2). The RVM prints the voucher (step 3) that can be redeemed at the cash register at *any* of the stores belonging to the supermarket chain (step 4). There, the cashier scans the barcode that encodes the SN1 (step 5). As seen above, the cash register may check the validity of the voucher against the Tomra server or a local computer (step 6 and 7). Then, the cashier may either stamp the voucher with the supermarket mark or rip it and put it in the cash register (step 7). Finally, the cashier reads the import redeemable from the cash register (step 8) and hands to the customer the money matching the value read from the cash register (step 9).

Netto ceremony. We can now describe the ceremony for DRS at Nettos in epistemic modal logic. We begin with introducing the different principals: the RVM V, the customer C, the seller S, and the cashier

Ca. Containers are captured by the predicate *object* and vouchers are captured by the predicate [*receipt*. The ceremony can be defined by the following five rules.

$$(\mathit{Purchase}) : \Pi S.\, \Pi C.\, 1 \multimap \exists c.\, [\![S]\!]\mathit{container}(c) \otimes [C]\mathit{object}(c)$$

$$(\mathit{Return}) \quad : \Pi C.\, \Pi V.\, \forall c.\, [C]\mathit{object}(c) \multimap [V]\mathit{object}(c)$$

$$(\mathit{Output}) \quad : \Pi V.\, \Pi S.\, \Pi C.\, \forall c.[V]\mathit{object}(c) \otimes [\![S]\!]\mathit{container}(c) \multimap \exists id.[C]\mathit{receipt}(QR(c, id)) \otimes [\![S]\!]\mathit{info}(c, id)$$

$$(\mathit{Hand}) \quad : \Pi C.\, \Pi Ca.\, \forall r.[C]\mathit{receipt}(r) \multimap [Ca]\mathit{receipt}(r)$$

$$(\mathit{Cash}) \quad : \Pi Ca.\, \Pi S.\, \forall id.\, \forall c.\, [Ca]\mathit{receipt}(QR(c, id)) \otimes [\![S]\!]\mathit{info}(c, id) \multimap 1$$

We can associate each of the above rules for the Netto ceremony with one or more steps modelling the general description for a DRS. The rule *Purchase* models a customer buying a container. Rule *Return* captures step 1; rule *Output* captures steps 2 and 3; rule *Hand* captures steps 4 and 5; finally, rule *Cash* captures steps 6 to 9.

Kvickly & Coop ceremony. The model of the ceremony of the Kvickly & Coop supermarkets can also be defined by five rules. However, while in the previous ceremony vouchers were printed on the basis of a single container, namely a new *id* was generated each time containers were returned, here vouchers are printed on the basis of the type of container. This type *id* is captured by what we call a *qualified* container (c, id), which will be used to identify the value of the container, when it is being returned.

$$(\mathit{Purchase}) : \Pi S.\, \Pi C.\, 1 \multimap \exists c.\, \exists id.\, [C]\mathit{object}(c, id) \otimes [\![S]\!]\mathit{container}(c, id)$$

$$(\mathit{Return}) \quad : \Pi C.\, \Pi V.\, \forall q.\, [C]\mathit{object}(q) \multimap [V]\mathit{object}(q)$$

$$(\mathit{Output}) \quad : \Pi V.\, \Pi S.\, \Pi C.\, \forall c.\, \forall id.\, [V]\mathit{object}(c, id) \otimes [\![S]\!]\mathit{container}(c, id) \multimap [C]\mathit{receipt}(QR(id))$$

$$(\mathit{Hand}) \quad : \Pi C.\, \Pi Ca.\, \forall r.[C]\mathit{receipt}(r) \multimap [Ca]\mathit{receipt}(r)$$

$$(\mathit{Cash}) \quad : \Pi Ca.\, \Pi S.\, \forall id.\, \forall c.\, [Ca]\mathit{receipt}(QR(id)) \otimes [\![S]\!]\mathit{container}(c, id) \multimap 1$$

4. Formal Analysis

4.1. The Tamarin prover

Our mechanised analysis is carried out in Tamarin [4], an interactive protocol verifier that can prove reachability and equivalence-based properties in the symbolic model. It has an expressive language based on multiset rewriting rules, which are similar to the rules in epistemic modal logic that we have used earlier in this work to define the threat models and the different DRS ceremonies. The Tamarin code modelling threat models, ceremonies, and properties is publicy available. [1]

[1] https://www.dropbox.com/sh/zlzk8473e26nk0o/AACXcmt4CnL8iZLpQIhCBuJpa?dl=0

In Tamarin, terms are variables and functions ranging over terms; facts are predicates that store state information and are parametrised by terms; facts may be linear (i.e. can be consumed only once) or persistent (i.e. can be consumed arbitrarily often by rules). The Tamarin multiset rewriting rules define a labelled transition system. The labels are used to reason about the behaviour of a protocol. Thus, to analyse our ceremonies in Tamarin, we need to annotate our rules with appropriate labels that will serve to the specification of our security properties. Rules are essentially defined as transitions from one multiset of facts to another. For example, a rule can be described as

```
rule RuleName:
    [ F(t1,t2), G(u1)] --[ L(t1) ]-> [H(t1,t2,u1)]
```

`F` and `G` are a multiset of facts regarding the starting state of the rule; `L` labels the state transition with the instatiation `L(t1)`; and `H` would be the new fact generated by the rule. The behaviour of a protocol is the captured by a number of rules. Thus, the execution of the rules creates a labelled transition system of the protocol in which all facts in the premise of a rule are consumed by the facts in the conclusion. Tamarin comes with a built-in network attacker model (i.e. the Dolev-Yao attacker). The attacker controls the network *channel* by two built-in rules that use the facts `Out` and `In` to model respectively messages learned by the adversary and messages built by the attacker and received by other participants. Controlling the channel allows the attacker to eavesdrop, block, modify, and inject messages to other participants. One can model different private channels to limit the control of the attacker over participants' communication. Note that the analysis of security ceremonies does not exclude the Dolev-Yao attacker. A ceremony that contains encrypted messages sent by its technical nodes would use the network channel controlled by the Dolev-Yao attacker to exchange such messages.

Conventionally, cryptographic primitives can be modelled in Tamarin by means of equational theories. An equational theory E describes the equations that hold on terms built from the signature. Terms are related by an equivalence relation $=$ induced by E. For instance, the equation $dec(enc(m, pk(k)), k) = m$ models an asymmetric encryption scheme. The term m is the message, the term k is the secret key, the function $pk(k)$ models the public key, the term *enc* models the encryption function, and the term *dec* models the decryption function, namely a deconstructor for the function *enc*.

4.2. From epistemic modal logic to Tamarin

We exploit equational theories to model physical actions such as reading a piece of paper or getting meta-information about an object. More specifically, we include the equations *receipt* and *object* to capture the actions of printing a receipt and building an object respectively, and their deconstructors $look(receipt(m)) = m$ and $get(object(m)) = m$ to capture the actions of reading something that was printed on a receipt and getting information about an object respectively. In the Danish return system, *receipt* is used to print a voucher, while *object* is used to build a container.

We introduce two additional channel models in Tamarin to capture epistemic modeal logic's modalities for knowledge and possession. One channel allows participants to share knowledge of some information (through facts `Out_I` and `In_I`). The other channel allows the participants to share objects (through facts `Out_O` and `In_O`). We then manually encode our rules from epistemic modal logic to Tamarin. Whether a fully automated and sound translation is possible is left for future work. Nevertheless, we make an effort to establish a systematic approach for the encoding. Terms and formulas in epistemic modal logic have straightforward encoding in Tamarin: linear implication $\multimap$ can be encoded with arrow symbol `--[]->` into Tamarin; linear conjuction $\otimes$ with comma symbol `,`; predicates and facts $P(t_1, \ldots, t_n)$ and F as Tamarin's facts `F(t_1,...,t_n)`; existential quantified individuals $\exists x$ are represented in Tamarin as fact `Fr(~x)`; and finally, universally quantified principals ΠK can be represented in Tamarin as prefixed (public) variables `$K`. As a general rule, the left-hand side of epistemic modal logic formulas, that is, everything before the $\multimap$ can be mapped as an input channel in Tamarin, while the right-hand side can be mapped as an output channel. For example, the rule *Hand* in the Netto ceremony can be translated into the following Tamarin rule

```
rule Hand:
    [In_O($V,$C,receipt(r))]
    --[]->
    [Out_O($C,$Ca,receipt(r)]
```

The encoding from epistemic modal logic to Tamarin is almost straightforward for most of the rules apart that, in Tamarin, we need also to state explicitly the sender of a message or the giver of an object. This is needed in Tamarin to avoid loops in the proof strategy, hence to avoid non-termination.

Existential quantified individuals, which are normally located at the right of the $\multimap$ in epistemic modal logic, must be expressed on the left-hand side of a Tamarin rule. For example, the rule *Purchase* in the Netto ceremony can be translated into the following Tamarin rule

```
rule Purchase:
    [Fr(~c)]
    --[Purchase($S,$C,~c)]->
    [!Container(object(~c)), Out_O($S,$C,object(~c))]
```

The rule above also shows how, in epistemic modal logic, each rule application together with the instantiations of the universal and existential quantifiers can partially serve as a Tamarin label, which in this rule is `Purchase($S,$C,~c)`. For example, the label $Cash(Ca, S, id, c)$ in epistemic modal logic for the rule *Cash* in the Netto ceremony, is expanded in Tamarin as `Cash($Ca,$C,$S,id,c)`, which notably includes the customer `$C`. This make it possible to use Tamarin to verify properties in which agreement on the customer identity is needed.

4.3. Properties

Trace properties can be modelled via metric first-order logic. Predicates are labels and properties can be expressed using quantification over time. For example, the following property expresses that for all trace, the *Cash* label is always preceeded by a *Purchase* label. This models a non-injective agreement on the terms C, S, and c, and can be formally written as

$$\forall C\ S\ c\ \#i.\ Cash(Ca, C, S, id, c)@i \implies \exists\ \#j.\ Purchase(S, C, c)@j \wedge j < i$$

We analyse the Danish return system against the following three security properties

- *Cash for voucher*, which says that if a cashier cashes out a voucher, then a corresponding barcode has been printed earlier by a vending machine.
- *Cash for container*, which says that if a cashier cashes out a voucher, then a corresponding container has been returned earlier to a vending machine.

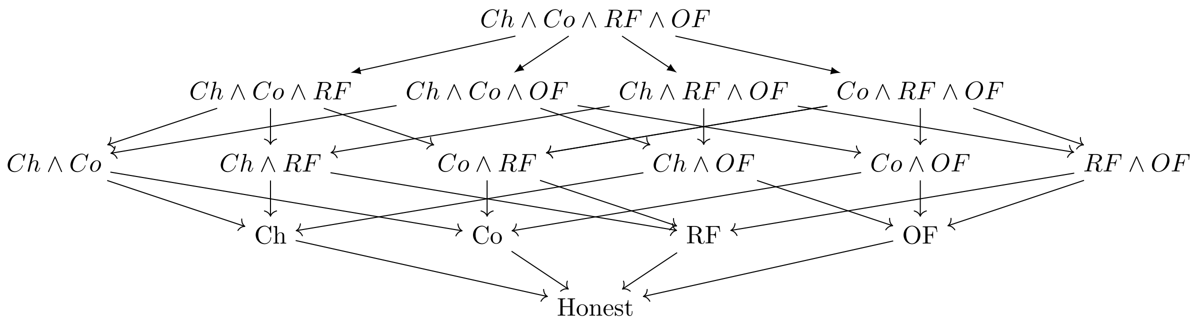

Fig. 4. The threat combinations

- *Cash for purchase*, which says that if a cashier cashes out a voucher, then a corresponding container has been bought earlier.

The properties above are all relevant for the security of a supermarket. If these properties hold, then the supermarket would not lose any cash. We can also formulate two additional properties that instead are more relevant for the security of customers. These properties can be expressed as follows

- *Cash for container customer*, which says that if a cashier cashes out a voucher to a customer, then a corresponding container has been returned earlier to a vending machine by the same customer.
- *Cash for purchase customer*, which says that if a cashier cashes out a voucher to a customer, then a corresponding container has been bought earlier by the same customer.

The last two properties are intuitively stronger than the previous three as they additionally require a strong injective agreement on the customer identity.

As an example, we present the Cash for purchase property

$$\begin{aligned}(\mathit{Cash\ for\ purchase}) :\ & \forall\ \mathit{Ca}\ C\ c\ \mathit{id}\ \#i\ .\ \mathit{Cash}(\mathit{Ca}, C, \mathit{id}, c)@\mathit{i1} \implies \\ & \exists\ S\ \ \mathit{C1}\ \#j.\ \mathit{Purchase}(S, \mathit{C1}, c)@j \wedge j < i \\ & \wedge \Big(\neg\big(\exists\ \mathit{Ca1}\ \mathit{C2}\ \mathit{id1}\ \#\mathit{i2}.\ \mathit{Cash}(\mathit{Ca1},\ \mathit{C2}, c, \mathit{id1})@\mathit{i2} \wedge \neg(i{=}\mathit{i2})\big) \\ & \vee \exists\ \mathit{C2}\ \mathit{S1}\ \#\mathit{j2}.\ \mathit{Purchase}(\mathit{S1},\ \mathit{C2}, c)@\mathit{j2} \wedge \neg(j = \mathit{j2})\Big)\end{aligned}$$

The agreement on the term c captures that the container for which a customer receives money has been sold previously by the supermarket. Here, we provide injectivity by ensuring that either no other similar container has been sold or the supermarket chain has previously sold another similar container. The remaining properties can be modelled similarly, possibly relaxing term agreements where necessary and replacing labels with the ones needed to capture the properties.

4.4. Lattice of threat models

Being the four models of human threat unrelated to each other, a ceremony may withstand several different scenarios that combine one or more threats assigned to one or more human participants. For example, a ceremony may withstand a scenario including the combination (Chatty ∧ Cocky) assigned to a specific human participant as well as the combination (Rforger ∧ Oforger) assigned to another participant. Obviously, a ceremony that withstands any threats assigned to any participants , namely a

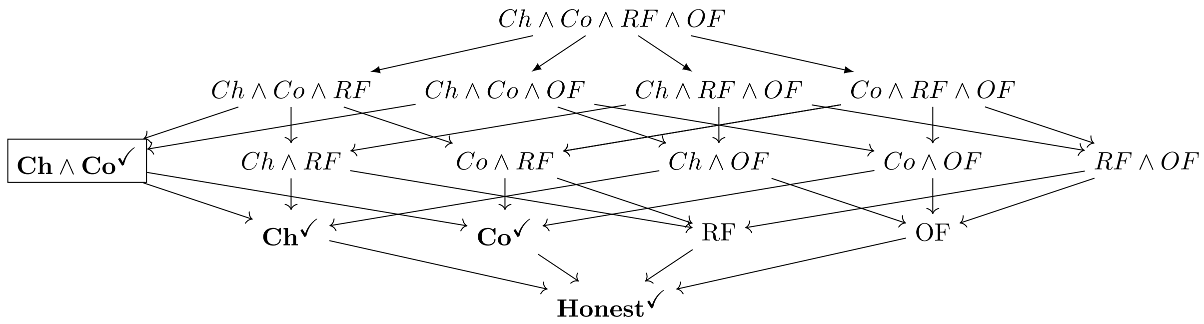

Fig. 5. If the property fails with the combination Ch ∧ Co, then the property fails with superset combinations, hence there is no need to explore the superset combinations.

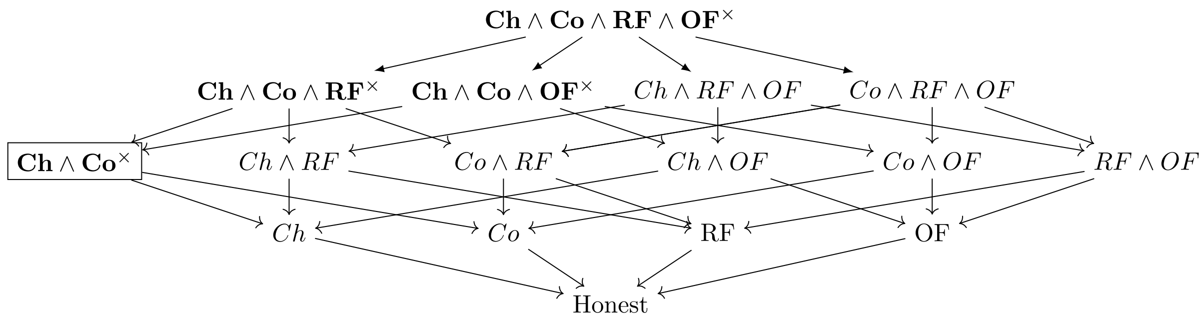

Fig. 6. If the property holds with the combination Ch ∧ Co, then the property holds with subset combinations, hence there is no need to explore the subset combinations.

ceremony that is secure against the combination (Chatty ∧ Cocky ∧ Rforger ∧ Oforger) assigned to any possible participant, withstands any other possible combinations, hence it is the strongest threat model combination for which a property may hold.

We observe that all the possible threats combinations form a partially ordered set, a Boolean lattice where any two elements have unique least upper and greatest lower bounds, which can be represented through a Hasse diagram as depicted in Figure 4. Here, each vertex of the diagram is one of the possible combinations of the four human threats. Vertices are connected by edges whose directions, e.g. $A \rightarrow B$, express strict superset, i.e. $A \supset B$. The strongest threat model combination is the maximum element, while the honest human is the minimum.

If a property fails against a threat model combination, we wonder what are the maximal threat model combinations (MTMC) for which the property holds. For simplicity, we do not focus on which participant executes which threats, but investigate this question considering threats being assigned to any human participants. However, threats can be assigned to an unbounded number of role instances, namely, participants, if the tool supports such a feature. This leads to stronger security guarantees when the property holds.

Finding the MTMC of a given ceremony and property requires us to explore a certain number of vertices of the diagram. A top-down approach, i.e. an algorithm that explores larger subsets first, would be optimal if the property holds against the strongest threat model, and pessimal if the property only holds against the honest human. Vice versa, a bottom-up approach in which smaller subsets are analysed first would be optimal if the property holds against few human threats and pessimal against the strongest threat model combination. To minimise the number of vertices of the diagram to explore for finding an

MTMC, we propose a greedy approach that priorities the analyses of vertices with a maximum degree and minimum difference between outdegree and indegree. By doing so, we certainly avoid exploring all vertices because the outcome of the analysis of such a vertex determines whether subset or superset combinations hold or not. For example, Figures 5 and 6 show that by analysing the vertex (Ch $\wedge$ Co), we avoid either exploring all the subset combinations or the superset ones.

We define Algorithm 1 to label all the vertices in the Hasse diagram when we analyse a property in a ceremony. Our algorithm takes in a graph $G = (C, E, L)$, where C is the set of vertices/combinations, $E \subseteq C \times C$ is the set of edges, and $L : C \rightarrow \{\checkmark, \times\}$ is a labelling function where $L(c) = \checkmark$ if the combination $c \in C$ holds in the ceremony. Note that our labelling function is the Tamarin analysis of the property against the combination c. We denote with $G[K]$, with $K \subset C$, the induced subgraph of C on vertices K. With C_l , we denote the set of vertices labelled either with a ✓or with a $\times$. The function `Max_edges` returns the vertex that has a maximum degree and minimum difference between outdegree and indegree. The function `DFS` takes in a graph, a vertex, a label, and a pointing direction, and labels the directed graph starting from the vertex using Depth-First Search. Note that the pointing direction is reversed if the labelling function returns ✓for a combination c, namely if the property fails with the given combination, we label with $\times$ the ancestors of c (e.g. Figure 6).

```
procedure Greedy_check(G):
  c ← Max_edges(G)
  if L(c) → ✓ then
    | DFS(G, c, ✓, ↓)
  else
    | DFS(G, c, ×, ↑)
  Greedy_check(G [C-C_l ])
```

Algorithm 1: A greedy algorithm to label all nodes in the Hasse diagram with ✓or $\times$

Algorithm 1 returns a fully labelled Hasse diagram. For each property, we select the maximal elements of the labelled Hasse diagram.

Lattices are well known constructions in cybersecurity and have been extensively used for modelling information flow polices [17], and then applied to several access control models [18]. However, to the best of our knowledge, they have not been considered for the threat modelling of security protocols or security ceremonies. In fact, their usefulness derives from the complexity of security ceremonies. Different from the standard Dolev-Yao network attacker, whose adversarial traits are clear and universally accepted, the socio-technical attacker can be malicious in different ways. The lattice-based approach to structuring the socio-technical attacker's capabilities is a mathematically sound tool to ensure that the attack space is fully and completely explored. The approach is independent from the logic that formalises the threat modelling (e.g. epistemic modal logic) as well as from the tool used for the analysis (e.g. Tamarin).

4.5. Findings

Table 1 shows the results of checking the two ceremonies of the Danish DRS in Tamarin against our five properties.

Table 1

The outcome of the analysis of the Danish DRS. A semicolon ; separates two or more maximal threat model combinations, if any. Legenda: Ch: Chatty; Co: Cocky; RF: Receipt forger; OF: Object forger.

	Maximal threat model combinations for which the property holds	
	Kvickly & Coop	Netto
Cash for voucher	(Ch ∧ Co ∧ OF) ; (RF ∧ OF)	(Ch ∧ Co ∧ RF ∧ OF)
Cash for container	(Ch ∧ Co ∧ OF) ; (RF ∧ OF)	(Ch ∧ Co ∧ RF ∧ OF)
Cash for container customer	(Ch ∧ OF) ; (RF ∧ OF)	(Ch ∧ OF) ; (RF ∧ OF)
Cash for purchase	(Ch ∧ Co) ; (RF ∧ OF)	(Ch ∧ RF ∧ Co) ; (RF ∧ OF)
Cash for purchase customer	Ch ; (RF ∧ OF)	Ch ; (RF ∧ OF)

Kvickly & Coop supermarkets. Table 1 shows that the Kvickly & Coop ceremony resists against attacks perpetrated by a combination of forgers (i.e., RF ∧ OF). Cash for container holds also against the combination Chatty, Cocky, and Oforger (i.e., Ch ∧ Co ∧ OF) while cash for purchase holds also against the combination Chatty and Cocky (i.e., Ch ∧ Co).

The common issue that leads to the falsification of the properties in the Kvickly & Coop ceremony is that Rforger humans may create fake vouchers based on what a Chatty customer reveals to them. Namely, Rforger can print vouchers that were never printed by any RVM. This is possible because the SN1 depends only on the number of containers. More precisely, for cash for voucher, cash for container, and cash for container customer Tamarin finds an attack trace in which a Chatty customer reveals the barcode printed on the voucher to another Rforger customer, who can now print as many vouchers they want since the barcode is fixed. The Rforger customer can successfully redeem one of the freshly printed vouchers.

For cash for purchase and cash for purchase customer, Tamarin exhibits an attack trace in which a Chatty customer reveals the information about the containers they bought to two Oforger humans, who in turn build two new identical containers. These are eventually returned to an RVM, and the corresponding vouchers printed by the machines are redeemed by the Oforgers.

Netto supermarket. Differently from the previous ceremony, there is no common issue leading to the systematic falsification of the properties in the Netto ceremony. Tamarin confirms that cash for voucher and cash for container hold against the largest threat model, which in Table 1 is reported as (Ch ∧ Co ∧ RF ∧ OF). Similarly to the previous ceremony, Table 1 shows that the Netto ceremony resists against attacks perpetrated by a combination of forgers (i.e., RF ∧ OF) for any property. Cash for purchase holds also against the combination Chatty, Rforger, and Cocky (i.e., Ch ∧ Rf ∧ Co). However, for cash for purchase, Tamarin shows an attack trace in which Chatty customer reveals the information about the containers they bought to another Oforger customer, who builds an identical container and returns this container to the RVM. The latter accepts the container and prints a valid voucher, which is eventually redeemed by the Oforger customer.

For cash for container customer and cash for purchase customer, Tamarin shows an attack trace in which a Chatty customer reveals the barcode printed in their voucher to another Rforger customer, who prints the voucher and redeems it with the cashier, eventually.

Discussion. Having analysed formally the ceremonies for the Danish DRS, we can conclude that, according to our models, the Kvickly and Coop ceremony is less secure than the one in Netto stores. We can also conclude that both ceremonies are strictly more secure towards protecting the supermarket's interests rather than the customers' ones. In fact, the MTMC for cash for container and cash for purchase withstand stronger threat model combinations than the respective customer versions of the properties.

Specifically, cash for container and cash for purchase hold despite Cocky humans, while cash for container customer does not hold in both ceremonies.

The main problem with the Kvickly and Coop ceremony is that one can generate a voucher by guessing the number of containers that other customers may have previously filled into any of the RVMs that belong to the supermarket chain. This is worsened by the fact that vouchers have the same SN1 independently from which RVM they have been generated, hence an attacker can likely be successful in redeeming a fake voucher holding the same SN1.

The Netto ceremony sees the RVM generating the SN1 based on an internal counter. Thus, each voucher is unique and is validated against a check with the Tomra servers. However, the Netto ceremony still fails to meet three properties against the strongest threat model combination. The main problem can be found by looking at the MTMC. We can observe that the combination of Cocky and OForger threats breaks all the tree properties. An OForger human can build new containers provided by a Cocky customer. While this can be an expensive attack procedure, we found a variant to the ceremony that would avoid such attacks.

4.6. Ceremony variant

For our variant, we get inspiration from the two-factor authentication DRS that has been piloted in Australia [19], named *myTOMRA app*. Here, container deposits are refunded digitally through an app. Each customer has an account and, when they return empty containers for recycling, they scan their myTOMRA app barcode at the RVM. The RVM will then refund the customer with a PayPal payout.

Differently from the Danish DRS, we had no chance to investigate the Australian DRS solution in depth. However, our variant exploits some services that are already available in Denmark. One of these services is *Storebox* [20], an app that allows customers to get their receipts digitally on their phone when making purchases. Storebox accounts are linked to the customer's credit card so that every time the customer makes a purchase with their credit card, the seller can send the corresponding receipt to the customer's Storebox account. In Danmark, Storebox is linked to another service, *E-boks* [21], which is the digital mail platform that Danish institutions and companies must use to communicate with Danish citizens. Access to E-boks is secured via another service, *NemID* [22], which is the common two-factor log-in solution for accessing services offered by Danish institutions and companies. Access to Storebox is secured by the two-factor authentication provided by NemID.

In our variant, when a customer purchases a container, the seller sends the digital receipt related to the purchase, which is identified with *rid*, to the customer's Storebox and to the RVMs. Upon return of the container *c*, the customer logins to their E-boks account through NemID and provides the digital receipt stored in their Storebox account to the RVM. The latter checks whether the receipt information matches with the information sent by the seller and, if so, refunds the customer digitally.

We can again use epistemic logic to express the variant. In the rules below, the predicate *appid* captures that a customer must have access to an identity provider, for example, through the NemID app installed on the customer's mobile phone. This provides a form of identity evidence referred to as *aid*.

$$(\mathit{Registration}) : \Pi C.\, 1 \multimap \exists aid.[\![C]\!]appid(aid)$$

$$\begin{aligned}(\mathit{Purchase}) \quad & : \Pi C.\, \Pi V.\, \forall b.\, \forall aid.\, [\![C]\!]appid(aid) \\ & \multimap \exists c.[C]object(c) \otimes \exists rid.[\![C]\!]receipt(rid, aid) \\ & \quad \otimes [\![V]\!]info(rid, aid, c)\end{aligned}$$

$$\begin{aligned}(\mathit{Return}) \quad & : \Pi C.\, \Pi V.\, \forall aid.\, \forall rid.\, \forall c.,\, [C]object(c) \otimes [\![C]\!]receipt(rid, aid) \\ & \quad \multimap [V]object(c) \otimes [\![V]\!]receipt(rid, aid)\end{aligned}$$

$$\begin{aligned}(\mathit{Cash}) \quad & : \Pi V.,\, \forall c.\, \forall rid.\, \forall aid.\, [V]object(c) \otimes [\![V]\!]receipt(rid, aid) \\ & \quad \otimes [\![V]\!]info(rid, aid, c) \multimap 1\end{aligned}$$

Discussion. Tamarin confirms that all the five properties hold in our variant against the strongest threat model combination. We observe that this is due to several reasons. First, the RVM now directly refunds customers, so we get rid of the cashier and of the associated threats. Second, we get rid of vouchers and use instead purchase receipts for refunds. In fact, the RVM refunds a customer upon the return of a container if and only if the customer has a valid receipt of purchase for that container. Third, we move critical security tasks, such as authenticating receipts from humans to the machines. A two-factor authentication mechanism ensures that our human threats are not sufficient to break the ceremony.

Our variant potentially introduces novel issues as well. Stronger customer authentication might allow RVMs to track container consumption habits of the customers, hence posing augmented privacy and usability risks than the ones in current solutions implemented in Denmark. Threats that concern machines, such as security bugs on RVMs or on customer mobile phones, would clearly have a stronger impact on the proposed variant rather than the current solutions. A Denial of Service attack on supporting services such as Storebox or NemID would prevent customers from getting refunded and RVMs to work at all. While privacy and availability properties are not in the scope of this work, the importance of verifying how ceremonies cope with them is increasingly important when critical security tasks are moved from humans to machines.

5. Related Work

A few works have already approached the formal analysis of security ceremonies to some extent, hence are related to the present contribution. Bella and Coles-Kemp model security ceremonies as a *concertina*, with various folds corresponding to the various layers interposing a technical system and its users [23]. The model sees each human's expression of various *personas*, arguably including personas that pose threats. Johansen and Jösang pick up the concertina and describe the expression of personas using probability theory but do not use computer assistance [24]. Probst et al. take an attack-tree approach and mechanise it via interactive theorem proving to study insider threats over a toy example that features a baker, his wife and a cake [25]. The approach exhibits vast potential to be generalised to more realistic applications and related security properties. Giustolisi et al. take a UML and model checking approach to analyse TLS certificate validation as carried out by modern browsers. They focus on security properties that also depend on user interaction, but users do not deviate from the possible choices that each browser supports [26].

Martina et al. [6] and more recently Martimiano and Martina [27] reinforce the need to shift away from the classical Dolev-Yao attacker model to capture human-centred threats. Stojkovski et al. respond to that need by modelling socio-technical misalignments between a technical system and its users and demonstrate their approach over an end-to-end email encryption system [28].

Sempreboni and Viganò [3] propose "mutations" of the human users of a ceremony with respect to the behaviour that the ceremony originally prescribed for such users. Such mutations are then paired with matching mutations of the underlying technical components. Thus, their work enables reasoning about situations in which the implementation of a ceremony does not conform to the ceremony's intended specification. By contrast, in our work, all technical components are assumed to behave as intended. This allows us to find out, through the lattice of human threat models, the maximal human threat model that makes the ceremony secure. Also, there are differences in the formal modelling of security ceremonies. The tool proposed by Sempreboni and Viganò extends the Tamarin prover to model and analyse security ceremonies, although their approach can potentially be applied to other provers. In our work, the modelling of security ceremonies is realised in epistemic modal logic and is fully independent from the prover. This choice has the advantage that ceremonies, threats, and properties can be formalised in a more intuitive and clearer way than using Tamarin's constructs, at the cost of an extra effort for their encoding into the prover of choice. Moreover, their human threat model is distributed in the sense that a number of humans may deviate from the standard steps, but humans who pose threats may only implicitly interact with each other and each may explicitly interact with the technical system. By contrast, the present work explicitly formalises *distributed and interacting* human threats, so that every human may misbehave for his personal sake, without any fixed prescription to collude with others, yet may directly favour someone else.

Basin et al. formalise human error in security protocols and analyse a few two-factor authentication case studies in Tamarin [29]. Notably, their work introduces the *untrained human*, who "may blindly follow any adversarial instruction he is given", hence "can perform any action permitted by the execution model". A few axiomatic restrictions transform the untrained human in the *rule-based human*, thus limiting his erroneous behaviour. The rule-based human behaviour can therefore be interpreted as a limitation of the arbitrary behaviour of the Dolev-Yao attacker. Our approach is conceptually and technically different. At the conceptual level, we deem threats to be broader than errors: while some errors, such as revealing information to someone else, may become security threats, it is clear that not all threats are errors because many threats are inherently intentional, such as forging a receipt. At the technical level, the differences are evident in the modelling approach. Basin et al. work resort to Tamarin for the modelling of security ceremonies. In contrast, we express security ceremonies in epistemic modal logic so that our approach is completely independent from the prover. In so doing, our approach can model ceremonies and threats in a more natural way, which can be eventually encoded in the prover of choice (e.g. Tamarin). Also, Basin et al. work leverages Tamarin's built-in Dolev-Yao attacker as a source of erroneous actions for humans. This is appropriate to capture human error in security protocols because it sources, for example, information that a human would not have or an arbitrary step that the human would not take otherwise. However, human-level threats for security ceremonies, which form the focus of our work, obviously also include tampering with objects, beside threats being distributed over interacting humans, as mentioned above. In consequence, we take the *augmentative approach* of modelling human threats as Tamarin rules that enrich those whereby honest humans interact with the given technical system. This approach is well known, at least dating back to Paulson's notion of *Oops* to enable humans to leak secrets [30], and easily supports extensions with explicitly modelled error rules as well as with additional threat rules. The augmentative approach underlines several approaches to capture threats to cyber-physical systems and their physics-based properties [31–34]. One may then even conjecture a stretch-out of the augmentative approach to combine the rules capturing threats for cyber-physical systems with our additional rules for interacting human threats. It can be expected that the challenge would then be offset to the efficiency of the analysis tools.

6. Conclusions

This article advanced a distributed and interacting threat model for humans participating in a security ceremony. It was formalised in epistemic modal logic, encoded in Tamarin and then tried out against two deployed deposit-return systems. A hierarchy of security properties for such systems was laid out, but many were found to fail over the two examples against the threat model incorporating all formalised offensive capabilities. This sparked off two separate developments. One was to design a variant for the deposit-return systems, which was later formally proved as effective. The other one was innovative and sparked off when a property was found to fail. It led to the search for threat models, arguably weaker than the strongest possible, that would not break the given property. A lattice of threat models originated as a general contribution to the field of formal analysis of security ceremonies, confirming that our approach to human threat modelling indeed is modular and scalable.

The general technological progress immerses us in a world of socio-technical systems. More and more everyday tasks will be carried out by leveraging the latest electronic inventions, and human beings will be increasingly engaged with socio-technical interactions. In consequence, room for error and explicit human threats may only expand, hence the challenge exacerbates for modular and scalable approaches to analyse the security properties that may hold. This article took that challenge and provided a general approach that can be conveniently reused over other security ceremonies. However, it is clear that the definition of an ultimate threat model demands a full assessment of the weakened versions that could preserve security and still be realistic. This in turn expands the future work towards optimising both the construction of, and the automated search through, the lattice of human threat models by means of computer-assisted formal analysis tools.

7. Acknowledgements

We are grateful to Ivan Garbacz and Kasper Møller Nielsen for helping out with the reverse engineering of the RVMs hosted in several Danish supermarkets. Rosario Giustolisi and Carsten Schürmann are supported by the Villum Foundation, within the project “Enabling User Accountable Mechanisms in Decision Systems”.

References

[1] R. Giustolisi, Free Rides in Denmark: Lessons from Improperly Generated Mobile Transport Tickets, in: *The 22nd Nordic Conference on Secure IT Systems*, Springer International Publishing, Cham, 2017, pp. 159–174. ISBN 978-3-319-70290-2.
[2] C. Ellison, Ceremony design and analysis, *IACR eprint* (2007).
[3] D. Sempreboni and L. Viganò, X-Men: A Mutation-Based Approach for the Formal Analysis of Security Ceremonies, in: *2020 IEEE European Symposium on Security and Privacy (EuroS P)*, 2020, pp. 87–104. doi:10.1109/EuroSP48549.2020.00014.

[4] S. Meier, B. Schmidt, C. Cremers and D. Basin, The TAMARIN Prover for the Symbolic Analysis of Security Protocols, in: *Computer Aided Verification*, N. Sharygina and H. Veith, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 696–701. ISBN 978-3-642-39799-8.

[5] I. Garbacz, R. Giustolisi, K.M. Nielsen and C. Schuermann, A Security Analysis of the Danish Deposit Return System, in: *Socio-Technical Aspects in Security and Trust - 9th International Workshop, STAST 2019, Luxembourg City, Luxembourg, September 26, 2019, Revised Selected Papers*, T. Gross and T. Theo, eds, Lecture Notes in Computer Science, Vol. 11739, Springer, 2019. https://www.springer.com/gp/book/9783030559571.

[6] J.E. Martina, E. dos Santos, M.C. Carlos, G. Price and R.F. Custódio, An adaptive threat model for security ceremonies, *International Journal of Information Security* **14**(2) (2015), 103–121.

[7] D. Dolev and A.C. Yao, On the security of public key protocols, *Information Theory, IEEE Transactions on* **29**(2) (1983), 198–208.

[8] D. Garg and F. Pfenning, A Proof-Carrying File System, in: *31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berleley/Oakland, California, USA*, IEEE Computer Society, 2010, pp. 349–364. doi:10.1109/SP.2010.28.

[9] H.A. Müller, J.H. Jahnke, D.B. Smith, M.-A. Storey, S.R. Tilley and K. Wong, Reverse Engineering: A Roadmap, in: *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, ACM, New York, NY, USA, 2000, pp. 47–60. ISBN 1-58113-253-0.

[10] A. Nordbryhn and A.H.H. Hansen, Fraud Detection, 2019, EP3440641A1. https://patents.google.com/patent/EP3440641A1/en.

[11] G. Saether, R. Sivertsen, T. Lunde and J. Njastad, Fraud detection system and method, 2018, US20180232745A1. https://patents.google.com/patent/US20180232745A1/en.

[12] T.O. Kavli, J. Njastad and G. Saether, Method and apparatus for detecting fraud attempts in reverse vending machines, 2012, US9189911B2. https://patents.google.com/patent/US9189911.

[13] A. Jorgensen, Method, system, reverse vending machine and use thereof for handling empty packaging, 2005, US20050246225A1. https://patents.google.com/patent/US20050246225A1/en.

[14] G. Saether, Means in a reverse vending machine (RVM) for receiving, handling, sorting and storing returnable items or objects, 2010, US7754990B2. https://patents.google.com/patent/US7754990B2/en.

[15] Tomra Systems Asa, Voucher control, Accessed 05-July-2019.

[16] Tamarin Code, 2021. https://www.dropbox.com/sh/zlzk8473e26nk0o/AACXcmt4CnL8iZLpQIhCBuJpa?dl=0.

[17] D.E. Denning, A Lattice Model of Secure Information Flow, *Commun. ACM* **19**(5) (1976), 236–243–. doi:10.1145/360051.360056.

[18] R.S. Sandhu, Lattice-Based Access Control Models, *Computer* **26**(11) (1993), 9–19–. doi:10.1109/2.241422.

[19] Tomra Systems Asa, myTOMRA app, Accessed 05-July-2019.

[20] Storebox ApS, Storebox, Accessed 05-April-2020.

[21] e-Boks A/S, e-Boks, Accessed 05-July-2020.

[22] Nets DanID A/S, NemID, Accessed 05-July-2020.

[23] G. Bella and L. Coles-Kemp, Layered Analysis of Security Ceremonies, in: *27th IFIP SEC 2012*, Springer, Berlin, Heidelberg, 2012, pp. 273–286.

[24] C. Johansen and A. Jøsang, Probabilistic Modelling of Humans in Security Ceremonies, in: *3rd International Workshop on Quantitative Aspects in Security Assurance (QASA 2014)*, Springer, 2015, pp. 277–292.

[25] C.W. Probst, F. Kammüller and R.R. Hansen, Formal Modelling and Analysis of Socio-Technical Systems, in: *Semantics, Logics, and Calculi: Essays Dedicated to Hanne Riis Nielson and Flemming Nielson on the Occasion of Their 60th Birthdays*, Springer, 2016, pp. 54–73.

[26] R.Giustolisi, G. Bella and G.Lenzini, Invalid Certificates in Modern Browsers: A Socio-Technical Analysis, *IOS Journal of Computer Security* **26**(4) (2018), 509–541.

[27] T. Martimiano and J.E. Martina, Daemones Non Operantur Nisi Per Artem, in: *Security Protocols XXVI*, Springer International Publishing, Cham, 2018, pp. 96–105. ISBN 978-3-030-03251-7.

[28] B. Stojkovski, I. Vazquez Sandoval and G. Lenzini, Detecting misalignments between system security and user perceptions: a preliminary socio-technical analysis of an E2E email encryption system, in: *4th European Workshop on Usable Security*, 2019.

[29] D. Basin, S. Radomirovic and L. Schmid, Modeling Human Errors in Security Protocols, in: *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, 2016, pp. 325–340. ISSN 2374-8303.

[30] L.C. Paulson, The Inductive Approach to Verifying Cryptographic Protocols **6** (1998), 85–128.

[31] M. Rocchetto and N.O. Tippenhauer, CPDY: Extending the Dolev-Yao Attacker with Physical-Layer Interactions, in: *Formal Methods and Software Engineering*, K. Ogata, M. Lawford and S. Liu, eds, Springer International Publishing, Cham, 2016, pp. 175–192. ISBN 978-3-319-47846-3.

[32] M. Rocchetto and N.O. Tippenhauer, On Attacker Models and Profiles for Cyber-Physical Systems, in: *Computer Security – ESORICS 2016*, I. Askoxylakis, S. Ioannidis, S. Katsikas and C. Meadows, eds, Springer International Publishing, Cham, 2016, pp. 427–449. ISBN 978-3-319-45741-3.

[33] R. Lanotte, M. Merro, A. Munteanu and L. Viganò, A Formal Approach to Physics-Based Attacks in Cyber-Physical Systems, *ACM Trans. Priv. Secur.* **23**(1) (2020). doi:10.1145/3373270.
[34] D. Sempreboni, G. Bella, R. Giustolisi and L. Viganò, What Are the Threats? (Charting the Threat Models of Security Ceremonies), in: *Proceedings of the 16th International Joint Conference on e-Business and Telecommunications, ICETE 2019 - Volume 2: SECRYPT, Prague, Czech Republic, July 26-28, 2019*, M.S. Obaidat and P. Samarati, eds, SciTePress, 2019, pp. 161–172. doi:10.5220/0007924901610172.
[35] L.C. Paulson, The Inductive Approach to Verifying Cryptographic Protocols, *J. Comput. Secur.* **6**(1–2) (1998), 85–128.