Daemones Non Operantur Nisi Per Artem

Daemons Do Not Operate Save Through Trickery: Human Tailored Threat Models for Formal Verification of Fail-Safe Security Ceremonies

Taciane Martimiano and Jean Everson Martina(✉)

Programa de Pós-Graduação em Ciência da Computação, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil
tacianeemartimiano@gmail.com, jean.martina@ufsc.br

Abstract. In this paper we argue that we must impoverish (or enrich in a different sense) threat models in order to be able to verify fail-safe security protocols that include human peers (a.k.a. security ceremonies). Some of the threat models we use nowadays for establishing the security of communication protocols are far too much concerned with failing deadly and do not encompass subtleties of the real world. Security is then maintained at all costs, especially in the presence of human constraints and expectations. Our position is that we must assume omnipresent and omnipotent evil beings (daemons) do not exist in order to be able to verify fail-safe security protocols that include human peers. We show how a popular security ceremony could be made fail-safe assuming a weaker threat model and compensating for that with usability. We also discuss the impact of our work for formal verification techniques and how they can be expanded for security ceremonies.

Keywords: Threat models · Security ceremonies
Formal verification · Fail-safe security ceremonies
Human-peer security protocols

1 Introduction

In the past we discussed how important it is to establish the goals of security protocols in the presence of human constraints and in the real environment where they will be used [9]. Security protocols fail too often, and mostly due to misunderstandings at design time. These misunderstandings have to do with misconceptions in preconditions and/or in environmental assumptions. Sometimes protocols assume strong assumptions or twist the environmental conditions where they execute so that a worst case scenario threat model can be addressed.

We also argued in the past [5] that Dolev-Yao's [6] threat model, the "de facto" standard used in protocol design and verification is powerful but not suited to be used as-is when we are trying to encompass human interactions

V. Matyáš et al. (Eds.): Security Protocols 2018, LNCS 11286, pp. 96–105, 2018.
https://doi.org/10.1007/978-3-030-03251-7_11

with protocols. In our paper [5] we break down Dolev-Yao's attacker capabilities into atomic actions and create an adaptive way of using them. In that paper we argued that some of Dolev-Yao's capabilities are not present when we have human-to-human interactions, such as the fabrication or blocking capabilities. Furthermore, allowing such things which do not respect the laws of physics in the presence of humans is equivalent to assume omnipresent, omnipotent evil beings (daemons) exist.

More recently we went deeper into the relation of threat models and the subtleties they entail on the formal verification process of security protocols in the presence of humans [8]. We started to symbolically evaluate security ceremonies in the presence of multiple attackers which are constrained in their capabilities depending on the layer of execution they are attacking [4]. We try to leave the daemons out of the equations and to construct more human friendly interactions with security protocols. This experimentation taught us that to represent the reality where humans and protocols co-exist we must take into account potentially uncountable threat models. This is a requirement if we want to make protocols fail-safe to the human peer behind the screens.

We have also seen that this topic of human interaction with security protocols has been fully discussed in the last couple years. We can cite Roscoe's detection of failed attacks on human interactive protocols [11]. He proposes a method to test and design security protocols that have human interactions where attacks can not be misdiagnosed as communication failure. In his work he introduced the concept of protocols being auditable for humans and the fact that transformations are time-dependent. Our position is that Roscoe's work can be reinterpreted taking a different view on threat models. We believe that Roscoe's work can be extended in the sense of transforming communication failures into fail-safe scenarios for users.

Bella [3] argue that security should encompass human beings while remaining invisible. They introduce various security protocols that can be constructed involving human peers and advocates that making security less present can help the user to participate more willingly. He also brings to attention that integration of security protocols that involve human peers need to be done with other socio-technical facets of the task that the user is trying to accomplish. Our position regarding Bella's work is that to properly integrate human interaction with security protocols, we need to be able to assess its security in more formal manner. To do so we need threat models to establish whether security is available or not. We can, by establishing plausible threat models for security ceremonies, enter the realm of verifying fail-safe characteristics of such human-protocol interactions and deliver the invisibility proposed by Bella et al.

Finally our position is that to properly design security ceremonies in a systematic and secure way to fail-safe, we need to understand and adapt threat models to presence of human peers. We also defend that this can ultimately be done using theoretical tools such as First-Order Theorem provers with standard protocol verification techniques.

This paper is organised with this introduction, followed by a background Sect. 2 to present some concepts we used to develop our verification framework. Then, we describe our mechanisation strategy in Sect. 3. Section 5 has our Bluetooth case study ceremony, alongside with our justification for threat modelling choices. Finally, Sect. 6 brings our final thoughts and conclusions.

2 Background

To be able to demonstrate our strategy for the design and verification of security protocols in the presence of human peers (a.k.a. security ceremonies), we need to present some of the background work on that field (Sect. 2.1). We also present some threat models for symbolic evaluation (Sect. 2.2) by giving a historical context, and classifying threat models regarding capabilities.

2.1 Security Ceremony Layers

Bella et al. argue that a ceremony can be layered and analysed on specific sections of its description. For that, the Concertina approach establishes security and privacy in the presence of humans. By doing so, they present a ceremony model which links technology to society through five layers, ranging from computer processes to user personas [2], as follows:

- Layer 1 (L1) - Informational: stands for the insecure network, where we have the protocol running between processes on the communicating devices;
- Layer 2 (L2) - Operating System: intermediate level between the protocol (executing on behalf of the user) and the process that runs the interface for that user;
- Layer 3 (L3) - Human-Computer Interaction: socio-technical protocol whereby a user interacts with a user interface. This layer is crucial for the protocol to reach its end, and achieve its goals;
- Layer 4 (L4) - Personal: related to the user expressing a given persona while interacting to the interface;
- Layer 5 (L5) - Communal: shows the society influence over the user behaviour.

From all five layers, protocols just focus on the first layer once they only handle network traffic. Ceremonies enable us to study layers 2 and 3 as well. Thus, we cover layers L1 through L3 in our work (being the layers related to computer science research). As stated in the work of Bella et al. [2], layers L4 and L5 are strongly related to social science (which is out of our scope), as such layers deal with the non-deterministic nature of the human being itself. Layer 4 concerns the mindset of the user when interacting with the system, while layer 5 represents the influence of the society over the user's behaviour.

2.2 Threat Models

In Ellison's words: "A ceremony can be designed and analysed with variants of the mature methods already in use for a network protocol."[7]. As such we need threat models to enable such symbolic evaluation. We start with the Dolev-Yao (DY) which is the "de facto" standard attacker model for security protocol design and symbolic evaluation.

A Dolev-Yao attacker controls the network channel, being capable of altering, copying, replaying and creating messages [6]. However, the attacker is not allowed to perform cryptanalysis and guessing random numbers. A protocol considered secure against such attacker is also considered secure against less powerful ones.

In addition to the DY model, we have one of its variants: the Multi-Attacker (MA) model [1]. This model allows each participant protocol to behave maliciously, intercepting and forging messages. Nevertheless, each agent of this model neither colludes nor shares knowledge with any other agent, bringing insights such as retaliation and anticipation attacks [1].

We use in our verification framework MAs (besides the well-known Dolev-Yao) to exemplify attackers acting in more than only one channel. As standard for protocol messages, a DY only acts on messages being transmitted through the network (namely L1 only). It is convenient to emphasise that both models do not allow the attacker to share information with any other attack in the system.

Alongside attacker types DY and MA, we use Martimiano and Martina's Distributed Attacker (DA) [8]. They consider that each layer can have more than one attacker associated to it, so several attackers are allowed to be distributed throughout the layers of the ceremony. The distinction among the three attacker types we used is summarised in Table 1 [8].

The DY attacker has fixed capability set and appears only in layer 1 (network). It is possible for MAs to control several layers, however their set of capabilities must not change. As for DAs, they can also be in several layers, with the differential of possible distinct capability sets for each one of them.

Table 1. Comparison among attacker types

Attacker type	Share knowledge	Fixed capabilities	Different layers
DY	No	Yes	No
MA	No	Yes	Yes
DA	Yes	No	Yes

The DA approach is most unique for its sharing feature: an DA may share knowledge with other DA attackers, attempting to corrupt the system by himself and/or colluding with others. In contrast, a DY never shares his knowledge as it would contradict its own rules. Similarly, MAs cannot share either.

In real life, DAs who share information stand for different attackers being in distinct places (attacking a network from home and eavesdropping the user on

the street, for instance). Those DAs who do not share can be seen as waiting for the opportunity to share. Such behaviour is encouraged in order to demonstrate the power of sharing, mainly when human peers are considered.

Regarding the attacker's capabilities our verification framework uses Carlos et al. ideas [5]. If we overestimate the attacker's capabilities in a ceremony, we will probably end up designing complex ceremonies which tend to be more complicated and not followed by users. On the other hand, if we underestimate the attacker, we might have a flawed ceremony. They proposed an adaptive threat model approach for security ceremonies, which we based our framework upon. In their model, the designer can remove capabilities from the whole DY capability set in order to make the attacker more realistic. They define the DY set as containing *Eavesdrop* (E), *Initiate* (I), *Atomic Break Down* (A), *Block* (B), *Crypto* (C), *Fabricate* (F), *Spoof* (S), *Re-order* (O), *Modifying* (M) and *Replaying* (R). Over the network, they set the standard DY threat model as it is its default setting [5].

2.3 Notation Syntax

The notation chosen started with the early paper from Needham and Schroeder [10]. Showing each step of the protocol in numbered lines, it has the sender on the left of the arrow denoting the flow of information and the receiver on its right. The message contents are shown after the receiver.

Martimiano [8] added the Ceremony Concertina layers in order to have a more detailed view of the information pathway. The idea here is to pinpoint the layers being crossed by each message as it is sent from one end to the other of the communication parties.

Furthermore, Martimiano [8] related each layer of the Concertina methodology to one or more attackers, and their correspondent capability set, accordingly to the threat models mentioned in the previous subsection.

The steps now are numbered in a dot separated version to represent the crossing of subsequent Concertina layers by each message. For example, step "2.1" stands for the first layer of the second message of the ceremony. Sender and receiver remain each on either side of the arrow just as in the protocol syntax. Nevertheless it is below the arrows that layers L1, L2 and L3 were added for the ceremonies description. The threat model (capability set) and attacker type (DY, MA or DA) are beside the layer contents in each step. Lastly, the messages come out at the rightmost part of each step [8].

The capability set for each attacker is either the full Dolev-Yao set of capabilities or a subset of such set - accordingly to the adaptive threat model of Carlos et al. [5].

Martimiano defines four different threat models for a Concertina layer [8]:

- "N": represents safe model, meaning the absence of threat model (no attacker);
- In case of a subset (of the DY full set of capabilities) with just one capability, we write it in capital letter (e.g. "E"). In this case, the attacker has only such capability;

– In case of a subset of capabilities with size greater than one, we use the symbol "+" to split the notation of the capabilities. For instance, "E + B" stands for the attacker subset of capabilities containing Eavesdrop and Block.
– "DY": full set of the DY attacker capabilities; in other words, an active attacker. Mainly used for the network layer.

A DY attacker is always linked to the DY full capability set, and never to a subset of these capabilities. On the other hand, MAs and DAs may have varied capability sets. More importantly, they can appear in more than one layer throughout the ceremony. For this reason, we give each MA and DA a number to serve as their unique identification (e.g. MA_1).

3 Mechanisation Strategy

We developed our verification framework as a set of scripts in Python that parses ceremonies in .tex format following Martimiano's syntax to the input format .dfg of the theorem prover SPASS [12]. As we displayed the ceremonies in LaTeX throughout this paper we as well use it as input of our program.

TexReader is our class responsible for parsing the tex, via regular expressions, and creating a ceremony object properly initialised with its contents. Our Ceremony class stores all needed information about the scenario in case so that it is clear what a writer should receive. In our case, our writer is SpassWriter, which passes through all the agents, threat models and messages to create all the formulae specified in the SPASS syntax, such as predicates and conjectures.

Our code is easily adaptable to other formats, though. The reader and writer classes (or even methods, if one prefers) can be programmed to translate a ceremony, say, in JSON to some other theorem prover once provided all needed ceremony info. This information, mentioned as attributes of class Ceremony, regards basically ceremony agents (senders and receivers of messages, besides attackers), message steps themselves, layers and threat models in the order occurred.

Our main method arranges for the list of ceremonies to be parsed and run in SPASS in order. The produced files have the same name of the original one, only the extension changes. We follow Martimiano and Martina's ceremony basic model for SPASS [8], so each new ceremony .dfg file starts with as a copy of this model and is incremented through the iterations of the writer. All our scenarios in tex and its corresponding generated files, as well as our code is available at: https://github.com/tacianem/SpassModel.

In the list of conjectures we can test the possibilities of sharing among the DAs for each scenario, or to test different scenarios to find the best solution for a specific problem, such as failing-safe.

4 The Devil Lies in Details

Carlos et al. [5] believe an over-powerful attacker to be unrealistic in certain cases, especially those related to the human peers. Having an attacker that

may overcome the laws of physics, and interfere with human speech or direct human action is usually above reason for ceremonies. We go even further, in that assuming such threat models to be possible is equivalent to allow for daemons (omnipotent, omnipresent and evil being) to dictate how we design security protocols and ceremonies.

Thus, the adaptive threat model of Carlos et al. is also part of our solution, since we assume attackers only capable of attacks reasonable from a human perspective. This is in fact one of the main characteristics of our verification framework. It also allows us to design and verify security ceremonies with reasonable attacks from the user perspective, enabling us to easily encompass security features that enable fail-safe situation in the execution for security ceremonies.

5 Bluetooth Fail-Safe Security Ceremony

We now describe the design and verification process to add a fail-safe strategy to the Bluetooth pairing protocol, adapting it from the work of Carlos et al. [5] by adding the non-diabolic threat models. We also show how it can be proven to fail-safe securely in the presence of reasonable attacker using our security ceremony verification framework.

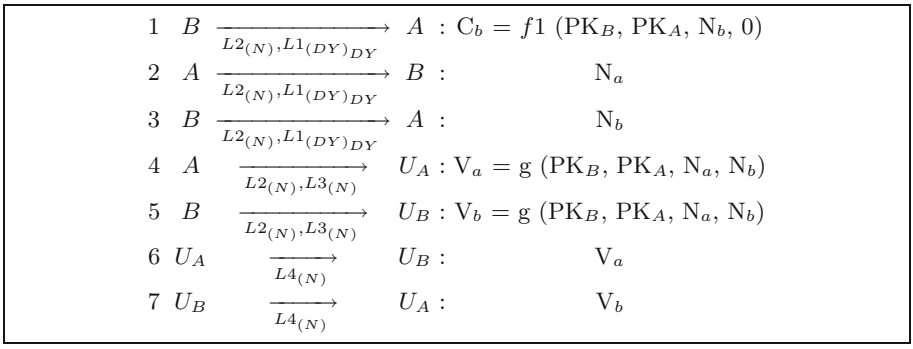

Fig. 1. Scenario 1 - default bluetooth ceremony

Figure 1 is our ceremony adaptation for phase two of the Bluetooth secure simple pairing (SSP) using the Numeric Comparison (NC) mode from Carlos et al. work, where we apply the Concertina layers instead of the DD (device-device), HD (human-device) and HH (human-human) channels as presented in their work. As we simply put it as a protocol, we have the standard DY attacker for the network and safe environment for the remaining layers. Besides, we have compressed layers as it is even suggested by Bella et al. [2] in their proposal of the Concertina methodology, once it is simpler to analyse what is happening at each step.

Essentially, the first three messages start on each device's own operating system and cross to the other device via Internet. Messages 4 and 5 also start on the operating system of devices A and B, and reach the user interface (layer L3). Finally, messages 6 and 7 are related to the users' interaction in order for the ceremony to achieve its goals, and be completed. For more details on the contents of each message payload, see [5].

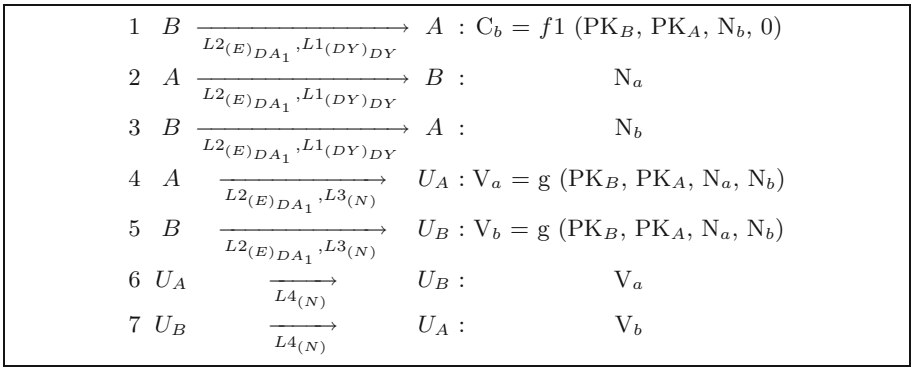

Fig. 2. Scenario 2 - Eavesdrop on both devices

For our second scenario (Fig. 2) we only vary the threat model for the first three messages, considering a DA attacker on the devices operating system. Such attacker has only the subtle yet powerful capability of Eavesdrop, as it is already sufficient enough for him to get all information computed and processed in each device. In here, the attacker is not able to actually modify any contents, however.

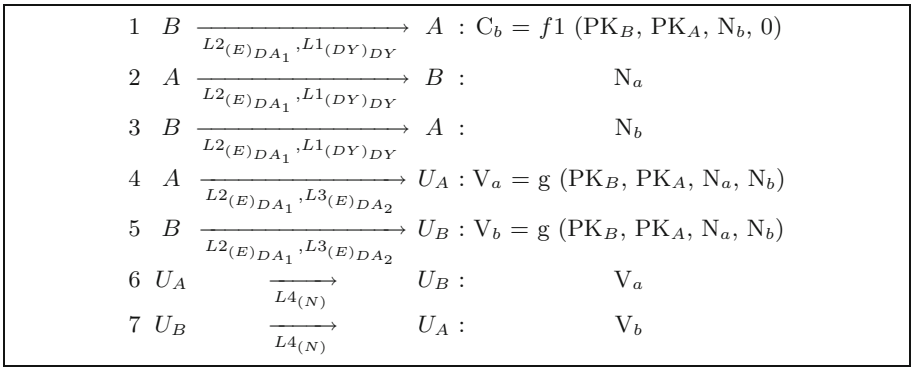

Fig. 3. Scenario 3 - Eavesdrop on both devices and on users' environment

Next, Fig. 3 brings our third scenario, we bring another DA attacker, now on the same environment as the users ("shoulder-surfing" them). Such physical

intrusion can compromise significantly the insurance of several ceremonies once most people do not watch their back while entering critical credentials in their devices, for instance.

Given that we expect both users to meet and exchange messages 6 and 7, we set our second DA to be around both users (which are not distant from each other themselves), hence controlling two different communication channels over the same layer (L3).

Similarly to DA_1, the second DA is only able to observe at best (in this case, the users' devices screens) and does not alter anything. The two of them are capable of sharing their knowledge (as explained in [8]), if they both agree.

It is worth noting that we are not varying the threat model of layer L4 as it is out of computer science scope and therefore we limit ourselves to assessment of layers L1 up to L3 solely. Our code for the ceremony scenarios above is in: https://github.com/tacianem/SPW18

6 Conclusions

During the formal verification and specification of security ceremonies for the Bluetooth pairing protocol we were able to detect that some actions could be included in the security ceremony to make it fail safe.

With a reasonable (non-diabolic) threat model we can design a ceremony where the user can be advised to execute the pairing in a specific setting. For example for devices that execute numeric comparison, but that do not have a good interface for showing the numbers (or use fixed numbers), we could change the protocol so that before pairing a precondition is to scan the Bluetooth spectrum trying to find other devices around. This can be used to either interrupt the pairing because it could lead to a possible man-in-the-middle attack from the other device, or at least to let the user know that he is in this somewhat insecure setting.

Having an automated verification framework where realistic, fine grained threat models can be easily tested for security ceremonies, allowed us to come with a series of ceremonies which aim to avoid that the human-being behind the device fall in an insecure setting which could jeopardise his secure communication effort. This is one of our main contributions for this work.

As future work we plan to embed more threat models to our security ceremony verification framework, as well as creating a more usable interface and to make the proof interpretation process easier for the ceremony designer.

References

1. Arsac, W., Bella, G., Chantry, X., Compagna, L.: Multi-attacker protocol validation. J. Autom. Reason. **46**(3–4), 353–388 (2011)
2. Bella, G., Curzon, P., Giustolisi, R., Lenzini, G.: A socio-technical methodology for the security and privacy analysis of services. In: COMPSACW (2014)

3. Bella, G., Christianson, B., Viganò, L.: Invisible security. In: Anderson, J., Matyáš, V., Christianson, B., Stajano, F. (eds.) Security Protocols 2016. LNCS, vol. 10368, pp. 1–9. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62033-6_1
4. Bella, G., Coles-Kemp, L.: Layered analysis of security ceremonies. In: Gritzalis, D., Furnell, S., Theoharidou, M. (eds.) SEC 2012. IAICT, vol. 376, pp. 273–286. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30436-1_23
5. Carlos, M.C., Martina, J., Price, G., Custodio, R.F.: An updated threat model for security ceremonies. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC 2013, pp. 1836–1843. ACM, New York (2013)
6. Dolev, D., Yao, A.C.: On the security of public key protocols. IEEE Trans. Inf. Theory **29**(2), 198–208 (1983)
7. Ellison, C.: Ceremony design and analysis. Cryptology ePrint Archive, Report 2007/399, October (2007)
8. Martimiano, T., Martina, J.E.: Threat modelling service security as a security ceremony. In: 2016 11th International Conference on Availability, Reliability and Security (ARES), pp. 195–204, August 2016
9. Martina, J.E., Carlos, M.C.: Why should we analyse security ceremonies? In: First CryptoForma workshop, May 2010
10. Needham, R.M., Schroeder, M.D.: Using encryption for authentication in large networks of computers. Commun. ACM **21**(12), 993–999 (1978)
11. Roscoe, A.W.: Detecting failed attacks on human-interactive security protocols (transcript of discussion). In: Anderson, J., Matyáš, V., Christianson, B., Stajano, F. (eds.) Security Protocols 2016. LNCS, vol. 10368, pp. 198–205. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62033-6_22
12. Weidenbach, C.: SPASS input syntax version 1.5. Max-Planck-Institut fur Informatik (2007)