

Modelling User Devices in Security Ceremonies

Taciane Martimiano and Jean Everson Martina
Departamento de Informática e de Estatística
Universidade Federal de Santa Catarina
Campus Universitário - Florianópolis - Brazil
taciane.m@inf.ufsc.br, everson@inf.ufsc.br

M. Maina Olembo
Technische Universität Darmstadt
maina.olembo@cased.de

Marcelo Carlomagno Carlos
Information Security Group
Royal Holloway, University of London
Egham, TW20 0EX, UK
marcelo.carlos.2009@rhul.ac.uk

Abstract—User constrained devices such as smart cards are commonly used in human-protocol interaction. Modelling these devices as part of human-protocol interaction is still an open problem. Examining the interaction of these devices as part of security ceremonies offers greater insight. This paper highlights two such cases: modelling extra channels between humans and devices in the ceremony, and modelling possession when the device also acts as an agent in the ceremony. Case studies where such devices are used during authentication ceremonies are presented to demonstrate these use cases.

Keywords—Threat models, Security Ceremonies, Security Protocols

I. INTRODUCTION

Security protocols are widely used nowadays. From web browsing to cash machines, we are all, directly or indirectly, using protocols in order to assure that certain properties, such as integrity or confidentiality, hold. User devices are also commonly used in security protocols, mainly to introduce an additional factor in authentication protocols. For an attacker to use such devices to successfully impersonate a valid user, the attacker needs to learn a secret from the user, and also possess this physical device.

When modelling such devices, the device is commonly treated as another agent in the protocol, but an attacker cannot listen to the communication channel between agents and their respective devices (as we see in [1]). In other words, other than using the powerful and widely used attacker model proposed by Dolev and Yao [2], a different threat model is used (no attacker, in this specific case). This will leave a gap of unspecified assumptions.

However, it is not comprehensive enough to model devices under a protocol point of view. A smart card, for example, does not belong to a computer agent. Instead, it is in the possession of a user. Additionally, user interaction in this type of protocol is very relevant, and to simply leave it as an assumption (or a set of assumptions) may lead to security problems due to usability issues or due to the fact that the assumption is unrealistic. Instead, modelling user devices in ceremonies seems more appealing because the level of details we can add to the description is potentially higher. The idea of ceremony design and analysis was introduced by Ellison [3]. In his research paper, he describes ceremonies as an extension of the concept of protocols by adding human beings, user interfaces, key provisioning and all instances of the workflow

to the protocol description. More importantly, we consider that one of the most important characteristics of ceremonies is that it includes a more precise description of assumptions.

Modelling user constrained devices, such as smart cards and one-time password generators for ceremonies has not been thoroughly discussed yet. These devices are widely used in several real-world scenarios, ranging from ATM systems to web authentication. They cannot be modelled as computers or human agents. They are different types of agents in a protocol and therefore under a different threat model as well.

In this paper, we present different user constrained devices which are frequently used in human-protocol interaction and discuss how they should be specified and analysed in a security ceremony setting. We show how to model such devices and how to approach the threat modelling for those devices based on the threat model proposed by Carlos et al [4].

Our paper first clarifies the differences and characteristics of ceremonies when compared to protocols in Section II. Then, we list different types of user devices and propose ways to model them in security ceremonies in Section III. The threat model that we should consider for each device is discussed in Section IV. An example scenario is presented in Section V. Finally, we present a discussion of our findings and future work in Section VI.

II. SECURITY CEREMONIES

Both security ceremonies and security protocols can be defined as a sequence of interactions among entities (or agents) designed to achieve a certain security goal, such as entity authentication, key distribution, secrecy, anonymity, etc. The difference between ceremonies and protocols is that, as described by Ellison [3], ceremonies can be seen as a superset of security protocols. Figure 1, from Carlos et al [4], represent the relationship between these two approaches.

While in security protocols we include several assumptions for operations that are considered out-of-scope, such as human-device interaction, human-human interaction, key provisioning, human knowledge, etc; in security ceremonies we explicitly include such interactions. Nevertheless, security ceremonies still need the use of certain assumptions, as for example some initial knowledge of the human agent, but the assumptions tend to be more precisely described, fine grained and more realistic.

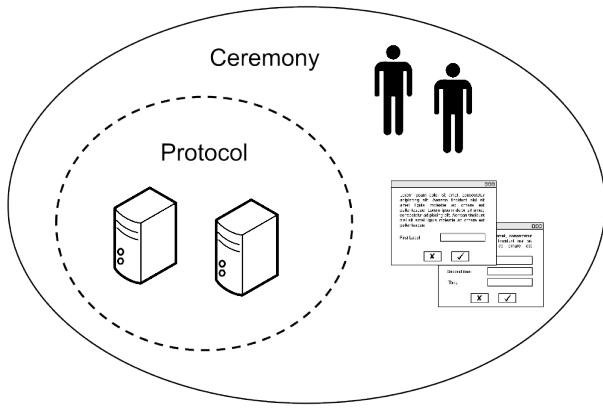


Fig. 1. Protocol and Ceremony association [4]

However, the gains from designing and analysing security ceremonies are directly linked to the quality and accuracy with which we describe the additional components a ceremony has when compared to protocols. In a protocol setting, it is relatively simple to predict the agents' behaviour and then expect a deterministic list of possible outcomes. The communicating media is usually a network medium and it follows some rules regarding the threat model it is subject to. In ceremonies, we also have to deal with the challenge of adding human nodes into the specification. This brings additional communication channels, such as the human-device channel and the device-device channel [4], [5] and the human agent type in addition to the already existing device (or computer) type. All of these new additions have their own nuances and complexity, making ceremony design and analysis a daunting challenge.

The threshold between a realistic and secure ceremony and an overly protected ceremony is also challenging to find. If we overestimate the attacker capabilities in a ceremony, we will probably end up designing extremely complex and unusable ceremonies. However, if we underestimate the attacker, we might have a flawed ceremony. To approach such a challenging task, Carlos et al. [4] proposed a dynamic threat model for security ceremonies. According to their idea, the ceremony designer would start with a very powerful attacker model known as Dolev-Yao, to which he would remove from the set of attacker capabilities in order to make the attacker more consistent with real-world scenarios. The opposite could also be done, where the attacker would start with no capabilities and then a set of capabilities would be added accordingly. Finally, Carlos et al. also present an example of a ceremony where unrealistic attacks would be found in case of using an inaccurate threat model for ceremonies.

One point that has not yet been covered in the ceremony design and analysis field is how to model user constrained devices, such as smart cards, one time password generators, tokens, etc. Such devices are commonly used in various scenarios, especially in banking and authentication systems. Modelling such devices and describing the threat model that they are subject to is something that has not been discussed yet and it is the focus of this paper.

III. USER DEVICES FOR CEREMONIES

When we enter the realm of user devices that can be used to engage in security related actions, we find a large set of gadgets. These gadgets are usually given to the user so that user and transaction authentication can be achieved by the proof of possession of such a device. The device itself is usually tamper proof and holds some secret information shared with the device issuer so that it is very unlikely to be released without the destruction of the gadget itself.

One important characteristic of such devices is that they have a constrained interface (or API). With a tamper-proof or tamper-evident nature to ensure resistance to the threats present in the physical environment, most of these devices are constrained logically so that attacks are very unlikely. As noted by Clulow [6], the size of an API is directly proportional to its attack surface, so this is a wise choice. But our view extends the use of such constrained API for devices to the adaption to a different realistic threat model, as already noted by Carlos et al [4]. The main purpose of this section is to list a series of such constrained devices, trying to identify their main characteristics. With this characterisation we can then move on to group them for the specific dynamic threat model using Carlos et al's [4] ideas.

A. Smart Cards

We start our listing with the most widely used type of security device that is the smart card used by financial institutions. These cards can have very different purposes, such as ATM cash withdrawal, and debit card or credit card payments. Despite that range of uses they share a great deal of similarities, being for our purposes classified as a single constrained device. A usual smart card has the features shown on Figure 2 and Figure 3. The following information is present at the front: (1) Issuing bank logo, (2) EMV chip, (3) Hologram, (4) Card number, (5) Card brand logo, (6) Expiration date and (7) Cardholder's name. The back contains: (1) Magnetic stripe, (2) Signature strip and (3) Card Security Code.

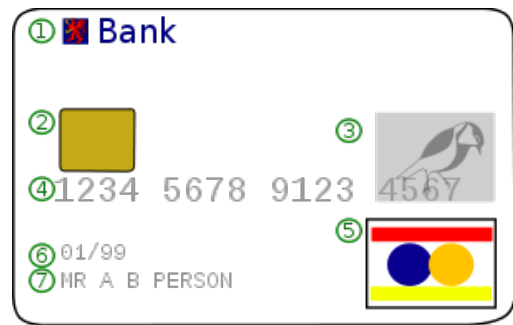


Fig. 2. Smart Card Token - Wikipedia Photo

An important characteristic of the plastic cards is their Unicity. Whenever we use such cards in transactions we have to prove its possession, which in turn is based on the assumption of the card being a unique and an atomic physical entity. This property turns out to be one of the basic building

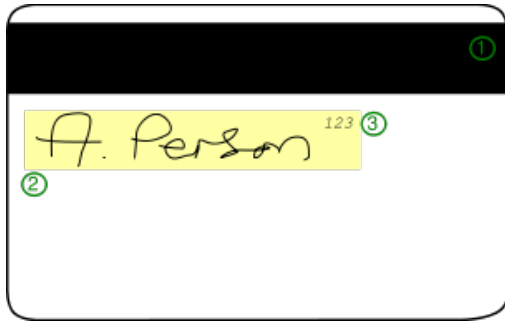


Fig. 3. Smart Card Token - Wikipedia Photo

blocks for the construction of authentication based on two factors.

Plastic cards evolved to have micro-processors and started to be called smart cards [7]. These smart cards offer cryptographic functions to the user, but they are still based on the same unicity assumption. Smart cards are often protected by the usage of security PINs, which in turn are part of the knowledge possessed by the card-holder. When protected by a PIN, a smart card would not respond to requests unless the PIN informed is correct. An important characteristic of plastic cards is their necessity of contact with other devices so that the user can prove its possession. By touching another device, this interface is out-of-bound to the user inspection and is usually a vector for attacks.

B. One-time Password Generator



Fig. 4. One Time Password Generator

A different type of device is known as a one-time password generator (OTP). These OTP devices, as shown on Figure 4, are generators for a disposable type of password that can be used by their holder to prove the possession of the device. The holder only has to copy the number displayed by the device to be authenticated. Such devices are produced and synchronised with a central server so that they use some sort of cryptographic function to generate the disposable passwords so that it can be verified by the other side. These devices are usually synchronised with the authentication server by the use of a common time-stamp or a counter [8]. These devices do not authenticate their holders to output an OTP value.

One of the main characteristics of such devices is the usage of a shared secret to generate the disposable passwords and a

very constrained user interface. The interface only consists of the display window and (sometimes) an on/off button. No user input is accepted (which could allow for transaction authentication, for example). The OTP device also has the Unicity property but lacks any sort of user authentication, meaning it only yields a very basic form of possession instead of authenticated possession as in the smart card device. Another important characteristic of OTP generators is that the user can prove its possession without contact of other out-of-bounds interfaces.

C. Keypad Token



Fig. 5. Keypad Token

A third type of constrained device used for authentication is the Keypad Token (Figure 5). These devices are an evolution of the OTP devices in the sense they will also generate a random password based on a shared secret and a cryptographic setup [8]. The main advantage of such devices is their extended input capabilities when compared to OTP devices.

The operation of a keypad token consists of the user turning on the device and typing a PIN to unlock it. Once unlocked it will be able to generate OTPs on demand. Thus, the first advantage of such a device is that it requires the authentication of its holder (with the obvious drawback of requiring an additional effort by the user). Another relevant advantage is its capability of data input. In this case, data input can be used to authenticate transactions. The input data can be concatenated with the time source or the counter so that it is used to calculate the OTP used for authentication [9]. Any change on this semantic data will inherently mean an authentication failure on the authentication server's side.

From the point of view of the ceremony design and execution, a keypad token provides two important properties: Unicity and Authentication. Similar to the previous devices, unicity guarantees the possession of the device, while the authentication property indicates the shared knowledge between the device and its holder. A difference with the previous

devices is that now the ability to operate it is controlled by the knowledge shared between the master and its device. Also by using two different Human-Device channels where the same data needs to be input by the Human user, we can assure that the attacker can not gain any advantage if he does not control both channels with the same powers.

D. Card Reader Calculator



Fig. 6. Handheld Readers

An evolution of the keypad token is the addition of a card reader calculator to it. Shown in Figure 6, a handheld reader allows the user to use a smart card with a keypad token proving the possession of both devices to authenticate himself and authorise transactions.

In this device, the user inserts his smart card into the handheld's reader. His card will hold the cryptographic secret needed to operate the calculator. The handheld will allow the user to prove the possession of the card without inserting it into any other (unknown) device. This is very important for the ceremony using a keypad token or smart card. With the card only being inserted into a device that is constrained and subject to a specific threat model under the control of the user, the Unicity and Authentication assumptions are stronger. In this sense the card reader provides a controlled and constrained contact interface to the smart card. In this setting we can weaken the attacker because we can assure our card reader calculator was never in the wild or in the attacker's possession, so the contact interface is as it should be. This solution yields the strong cryptographic assumption of a smart card with the non-touching capability of a password generator.

Later on in the paper (Section V) we will show how the usage of such a tool can prevent the invalidation of the Unicity assumption. By not allowing the attacker to have physical access to the device holding cryptographic material, the chances of copying are nullified.

E. Smart-phone

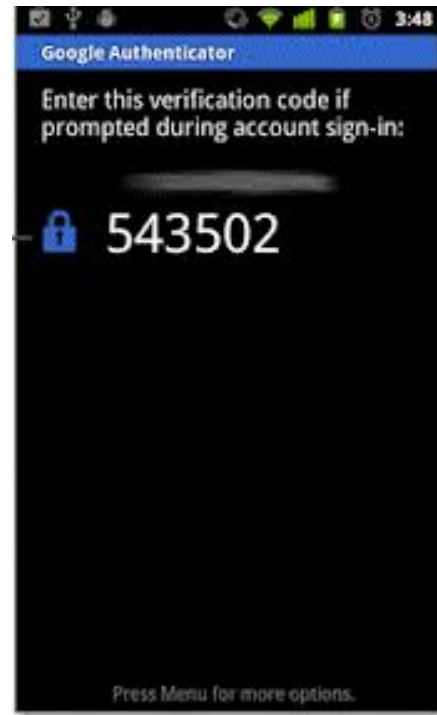


Fig. 7. Smartphone One Time Password Generator

Finally we have a “not-so-constrained” device, that is a smart-phone. Most of the techniques present in the other constrained authentication devices can be easily deployed on smart-phones these days. As shown on Figure 7, a smartphone can be easily setup to work as an OTP device, or even to authenticate transactions as a keypad token [10]. A smart-phone can also be extended with the inclusion of some peripheral devices to the equivalent of a handheld reader. One of the capabilities being explored by the banking industry is the Near-Field Communication present in such devices [10].

Even with the wide availability of smart-phones, we must stress that they are generic purpose devices that do not operate in a typical constrained fashion. Smart-phones are as susceptible to attacks as a standard desktop computer. The use of vetted software does not imply the guarantees needed for assumptions such as unicity and strong authentication as provided by the single purpose contained devices. Nevertheless, the high availability of such devices makes them very good candidates to embed security applications, especially because the cost of deployment is already taken by the user.

In this section we provided a brief overview of some constrained devices used for user and transaction authentication. These devices will be used in our examples later on the paper and do not constitute the complete universe of solutions available. We have already seen other variations of the above, such as smart cards with integrated keypads, such as Mastercard's Display Card or the usage of different authentication channels, such as the usage of SMS for authentication. In the next

section, we will provide a through review of the threat models for such constrained authentication devices.

IV. THREAT MODELLING

The use of constrained devices in security ceremonies is an interesting addition due to the security properties they provide. However, as mentioned earlier, such devices are under different threat models than the usual devices used in a ceremony (e.g. computers). The use of a powerful attacker in the threat model (such as Dolev-Yao [2]) would be excellent to find a wide range of possible attacks and threats. On the other hand, several attacks would be completely unrealistic and an attempt to prevent such attacks would imply additional complexity to the ceremony and usability. Our approach follows the threat model proposed by Carlos et al [4]. Constrained devices are designed assuming a different threat model (using a not so powerful attacker) since such devices are in most cases simpler than computer interfaces and often do not send information via a classic network channel.

Carlos et al's adaptive threat model introduces the idea of adding and/or removing capabilities of the attacker. They use Dolev-Yao's threat model and attacker capabilities set as a starting point and adjust the threat model according to the real threats that the ceremony being analysed is subject to. Such a dynamic threat model can also be used when modelling the threats for constrained devices. In their paper, Carlos et al make use of three different channels namely human-human (HH), human-device (HD) and device-device (DD). The HH channel involves humans communicating (e.g. via speech); the HD channel represents the interaction between users and devices (e.g. via user interface); and the DD channel involves the interaction between two or more devices (e.g. a network channel). For each channel, a different threat model is used and it is tailored to each ceremony's requirements and real threats. The threat models for the DD and HH are less likely to have big differences among ceremonies since the channels are usually similar (e.g. a exchange of speech packets between humans or network messages between devices).

However, the HD channel is more susceptible to changes. This channel can be used to model a complete interface between a computer agent and human user, but can also be used to model the interactions with constrained devices as shown in the previous section. There are clear distinctions on the channel's threats in this case. On HD channels involving a computer it is plausible to assume that the attacker will be able to control the device and induce users into false beliefs about the ceremony being executed. The same can not said for certain for constrained devices. Moreover, some of these HD channels have as an anchoring point the human user, which is not subject to the same attacks we usually see in security protocols' settings.

The constrained nature of some devices will inherently thwart certain types of attacks. For example on a read-only OTP device, an attacker can not inject any software that will automatically copy the current password to his knowledge. Nevertheless we know that this password can be caught by the

HD channel between the user and his computer. Some other threats are not existent due the notion of strict possession of a device. In contrast to a generic computer, a constrained device is not shared and can be directly traced to one single user. Due to their constrained interface, if they are kept under the possession of the user at all times, we can assume a weaker threat model.

Thus, to model constrained devices and define accurate threat models for them, we need to create additional channels to those already described by Carlos et al. These additional channels will be used to represent the interaction between agents and the constrained devices.

A. Modelling multiple HD Channels

As we have seen in section III, there are several types of devices that although all of them are mainly used for authentication, they provide some additional (and different among them) security properties. Therefore, the communication channels between such devices and the human agents are susceptible to different attacks and require different threat models. Additionally, constrained devices are (in most cases) used as auxiliary devices in ceremonies that includes other non-constrained devices (e.g. computers). Due to the reasons we explained earlier, the threats that a constrained device is subject to are not the same as a non-constrained device is. Thus, since our focus is on defining realistic threat models, we cannot model the HD channel for both device types in the same way.

Differentiating the threat models even when the agent types are, in principle, similar is relevant since it allows us to design a ceremony describing the interaction with all the devices involved and make use of a specific and realistic threat model for each channel involved. For example, a two factor-authentication ceremony frequently involves a human interacting with a computer (e.g. his/her laptop) and the same human interacting with a constrained device (e.g. OTP generator). Although from a very generic point of view the human agent is interacting with two devices via a HD channel in this scenario, from a threat model point of view, the threats are very different based on the channel used to interact with each device. In the first HD channel (HD), an attacker is more likely to possess some kind of control over the channel, such as spoofing messages or eavesdropping the communication. In the second HD channel (HD_C), between the human and the OTP generator, an attacker would have no capabilities, or in the worst case, the eavesdrop capability. Additionally, in contrast to the normal HD channel, the HD_C channel includes the concept of possession of the device, that represents the fact that only the user has access to it.

B. Modelling possession

When using constrained devices in security ceremonies, a very important factor to consider is the possession of the device. The fact of a user possessing a device involves the assumption that only the user has access to the device and in the case this assumption does not hold, an attack may happen.

For example, just the act of a user inserting a card into a card reader (e.g. in a ATM) may imply that, in practice, the user no longer possesses the device or that the device loses its unicity property, meaning that it is no longer unique.

Once again, a realistic modelling of the threats involved is the key to model the possession of a device. In OTP generator devices, the generated values (OTPs) are sent to the user via a constrained HD channel (HD_C). Such values are then forwarded by the user to another agent in the ceremony (e.g. a computer) via another channel (typically a HD channel), which then sends the OTP value to the verifying agent (e.g. bank server). The verifying part must be able to check whether the value inserted by the user has been produced by a trusted constrained device linked to the user. Note that at any moment the device itself has not been forwarded to another agent, just the value generated by the device.

When smart cards are used, on the other hand, the possession of the device changes at least momentarily. To perform the requested operations, such as encrypting or signing data, a smart card has to be inserted into a reader. That means that the possession of the card changes from a human (owner of the card) to a reader (another device). In this case, there is a risk of an attacker being able to control the DD channel where the smart card and the reader are communicating and obtain relevant information. Therefore, there is a change in the threat model that the card is subject to, which must be accurately and realistically described. For example, a card that does not output sensitive information (e.g. keys) and requires the user to insert a PIN to authenticate its owner's approval to the operation being performed should be used instead of a classic card that uses magnetic stripes (which is easy to copy). However, even when we use modern smart cards, there is a risk while the card is inside the reader. At that moment, the card is not in possession of the user, and the user has to provide the PIN to another device (e.g. reader) via HD channel that forwards the PIN to the card via DD channel. Since the user does not directly provide the PIN to the card, there is a period when the card is in the reader and the PIN has been given by the user to the reader where all the information required to require the card to perform an inappropriate operation is not under the control of the owner of the card.

In both cases (when the constrained device is always in possession of the user and when the device is forwarded to another device), modelling the possession of the constrained device involved the distribution of knowledge among the agents. When the constrained device is always in possession of the user, the knowledge of the required values to create and verify OTPs are only in the knowledge of the constrained device itself and the verifying party. The possession of the device is given by the fact that the user was the only one to interact with the device via a restricted channel. In the second case, when the constrained device is forwarded to another device, the constrained device communicates with more than one agent (for example, a user and a reader). In this case, the possession is given by the assumption that the device does not reveal sensitive (or private) information to anyone and only

acts (e.g. signs data) when receiving a pre-established approval value (e.g. PIN).

In the next section, we will present case studies that will provide a more practical description of modelling the additional channels and the possession of the devices. In these case studies only use OTP generators and smart cards due to space limitations. Showing usage ceremonies for the all the devices is part of our future work. Furthermore, we explicitly leave out smartphones because we do not believe they are actually constrained as they are considered in several scenarios. Smartphones are fully fledged computing devices what makes their attacker model closer to the full Dolev-Yao model but not actually it. We mention them here because we believe they are indeed a good example to the limitations for a model based on Dolev-Yao such as the one proposed by Carlos et. al.

V. CASE STUDIES

In this section we will present two case studies where we model different uses of constrained devices and present our analysis. The first case is a classic two-factor authentication. The second one is the case where the device (smart card) never leaves the user's possession compared to the classical use where the user inserts the card into the reader.

The notation used in this work uses lower-case words to represent user input (strings) and messages to be displayed to users (e.g. "email" represents the email value that the user has entered). Human readable messages are represented between quotes (e.g. "Enter your PIN"). Upper-case components (characters or abbreviations) represent "classic" protocol messages sent between devices. Channels are represented by the letters below the arrow, where HD and DD represent the human-device and device-device channels. There is a secondary HD channel, named HD_C , to represent a constrained channel (which involves human-device channel) but under a different threat model. Similarly, we also have a secondary DD channel, named DD_C , to represent a constrained channel involving a device-device channel under a different threat model. For agents, we use single upper-case characters (e.g., G is used to represent the OTP Generator, C to Card and T to Terminal). Finally, brackets "{" and "}" are used to indicate the encryption operation. For example, $\{m\}_K$, means a ciphertext made by encrypting m with a key K .

For the ceremonies presented in this section, we consider a threat model where we have a Dolev-Yao attacker for DD channels, no attacker (N) for constrained channels DD_C and HD_C . This is a plausible assumption because we assume that the constrained devices are only under user's possession and then there is no threat on their use in the constrained channels. Finally, as proposed by Carlos et. al. we consider an attacker with just the Eavesdrop capability for HD channels (N+E).

A. Two-factor Authentication Ceremony

The ceremony we present below is derived from the authentication scheme used by various websites to authenticate users using two factors. We must stress that we have in mind here

the use of a dedicated device instead of an smart-phone, since a smart-phone may be under a less constrained threat model due to its nature.

We assume that the user U possesses a one time password generator, a server B can verify the one time password generated and that there is an established session key K_{AB} between the agent A (U 's computer) and B . The assumption of an in-place session key is reasonable since most websites using dual-factor authentication do it over SSL/TLS channels.

1. $U \xrightarrow{HD} A : email, password$
2. $A \xrightarrow{DD} B : \{email, password, N_a\}_{K_{AB}}$
3. $B \xrightarrow{DD} A : \{authresult_1, N_a, N_b\}_{K_{AB}}$
4. $G \xrightarrow{HD_C} U : OTP_A$
5. $U \xrightarrow{HD} A : OTP_A$
6. $A \xrightarrow{DD} B : \{OTP_A, N_b\}_{K_{AB}}$
7. $B \xrightarrow{DD} A : \{authresult_2, N_a\}_{K_{AB}}$
8. $U \xrightarrow{HD} A : "Welcome"$

Objective: If the messages M1, M5 and M8 are run against a N+E attacker, message M4 is considered as having no attacker and messages M2 to M3 and M6 to M7 are run against a DY attacker, the attacker (Att) can learn the email, password and OTP_A information by eavesdropping the HD channel.

$$\frac{(M_{1,5,8} \cup N + E) \wedge (M_4 \cup N) \wedge (M_{2,3,5,6} \cup DY)}{email \wedge password \wedge OTP_A \in knows(Att)}$$

Proof: Using the capability of Eavesdrop, the attacker Att is able to learn *email*, *password* and OTP_A . Therefore, he can only perform an attack if he sends the message $\{OTP_A, N_b\}_{K_{AB}}$ to B before A does it, otherwise the OTP will be invalid if used for a second time.

As described in related work by Bruce Schneier [11], the use of OTP prevents the case where the attacker obtains email and password offline and wants to use it to deploy an attack on himself without user collaboration. However, it does not stop an attacker from deploying an online attack, such as a man in the middle attack (MITM).

Our ceremony analysis shows that, even assuming a constrained channel, the use of OTP does not really help much in terms of improving security against an online attacker. The attacker is able to learn the OTP and use it on-the-fly to perform an online attack. Nevertheless, it does protect against an attacker who wants to deploy the attack when the user is offline. This is mostly true because of the strict possession we implied the device has towards its user. By not being able to physically manipulate the device the attacker cannot deploy an attack without user participation.

B. Simple Smart Card Authentication Ceremony

The next ceremony we show a classic case where we have to assume that there is a constrained DD channel in

order for the ceremony to work. This ceremony represents the authentication against an ATM machine. The ceremony was derived by the protocol shown in Anderson's book on security engineering [12]. We refer to this ceremony as classic authentication based on smart card operations.

1. $U \xrightarrow{HD} T : C$
2. $C \xrightarrow{DD_C} T : C_{info}$
3. $T \xrightarrow{HD} U : "Enter your pin"$
4. $U \xrightarrow{HD} T : PIN$
5. $T \xrightarrow{DD_c} C : N_T, PIN$
6. $C \xrightarrow{DD_c} T : N_T s k C$
7. $T \xrightarrow{DD} B : \{C_{info}, N_T s k C, N_T\}_{TMK}$
8. $B \xrightarrow{DD} T : \{AuthResponse\}_{TMK}$
9. $T \xrightarrow{HD} U : "Welcome"$

Objective: If the messages M1, M3, M4 and M9 are run against a N+E attacker, messages M2, M5 and M6 are considered as having no attacker and messages M7 to M8 are run against a DY attacker, the attacker (Att) can learn C_{info} and the card's PIN.

$$\frac{(M_{1,3,4,9} \cup N + E) \wedge (M_{2,5,6} \cup N) \wedge (M_{7,8} \cup DY)}{C_{info} \wedge PIN \in knows(Att)}$$

Proof: The first thing we notice is that the card is at the same time an agent and a message in this protocol. When the card is sent to the terminal (inserted into the terminal's reader), the card's information is under a different threat model. Information such as C_{info} can now be learned by the attacker in case it is sent via the DD channel that exists between the card C and the terminal T . A security concern in this case would be that any message sent from and to C could be accessible and modified by the attacker. While C is in the card reader (that is, C is an agent in the protocol that can directly exchange messages with T), C is susceptible to an attacker that know the PIN value, which could be obtained in 4 or 5. In the case of an ATM, an attacker has no access to TMK since it is pre-established during the initial ATM setup and it is assumed to be secure. It limits its capabilities on the DD channel. However he could learn the PIN , which could be used in other scenarios.

Clearly our analysis for the security ceremony shown above is also expected. As already noticed in practice by parallel research [13], a card can be copied or tampered with while inserted in the card reader terminal. This only happens because there is the assumption we made above that the DD channel between the card and the reader is indeed constrained. But this is not always true.

One proposition we can draw from both our analysis shows that a new ceremony could be devised to use a smart card in authentication scenarios where the channels could actually be

constrained. In fact if we use a card reader terminal such as the one shown in Figure 6 we may be able to achieve such supposition. By having strict control of the card reader the user can be sure that the attacker will not be able to act over it. Also the user is assured by the “air gap” he introduces between the point of attack off the ATM and the card. In this sense we may be able to devise a ceremony where we could guarantee that a smart card could not be cloned by the simple fact that it never leaves the user’s possession.

VI. CONCLUDING REMARKS

This work has presented user constrained devices, such as smart cards and one time password generators, and considered how they can be better modelled with minimum assumptions. To this end, we have applied the adaptive threat model presented by Carlos et al. [4]. We have considered the example of authentication ceremonies, and looked at two case studies: two factor authentication and a smart card at an ATM terminal.

The contribution of this work is the listing of the different constrained devices and their properties, as well as the translation of these properties to the threat modelling proposed in Carlos et al. From the case studies, we have proposed an ATM ceremony that can be proven not to allow for card copying because the card never leaves the possession of the user. The user is indeed the hub that passes information between the constrained device and the not constrained device. The user acts as a proxy avoiding the attacker to access the secrets from the constrained device.

While in this work we have evaluated authentication ceremonies for OTP generators and smart cards, we deliberately left out some of the constrained devices we presented. Some were due to space limitations; describing the ceremonies for Keypad Token and Card Reader Calculator and evaluating them would not fit in due to space limitation. While the smartphone device was left out because we believe it already stresses the capabilities of Carlos et. al. threat model. The smartphone is much more like a fully capable computer than a constrained device, nevertheless we recognise it adds to the security of a ceremony.

As future work we plan to develop a tailored threat model based on the ideas of Carlos et. al and The Multi-Attacker from Arsac et al. [14] specially to sort out the limitations from Carlos et. al model towards smartphones. Our main idea is to consider a multi-attacker approach, once smartphones are susceptible to several attackers with different capabilities sets that may act independently from each other or not. In the multi-attacker model, each protocol participant may behave as a Dolev-Yao attacker, neither colluding nor sharing knowledge with any other participants[14]. We want to investigate the application of multiple Carlos et. al constrained attacker to the ceremony environment. We believe this new threat model will help us to encompass the peculiar way ceremony designers use smartphones in the ceremony context.

In a parallel line for future work will also believe investigating various application contexts, for example voting ceremonies, using this constrained approach may help users

to have better assurances that his vote was taken into account and the election was run as expected.

REFERENCES

- [1] G. Bella, *Formal Correctness of Security Protocols*, ser. Information Security and Cryptography. New York: Springer Berlin Heidelberg, 2007.
- [2] D. Dolev and A. C. Yao, “On the security of public key protocols,” *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.
- [3] C. Ellison, “Ceremony Design and Analysis,” Cryptology ePrint Archive, Report 2007/399, Oct. 2007.
- [4] M. C. Carlos, J. Martina, G. Price, and R. F. Custodio, “An updated threat model for security ceremonies,” in *Proceedings of the 28th Symposium On Applied Computing*, ser. SAC. ACM, Mar. 2013.
- [5] M. C. Carlos, J. E. Martina, G. Price, and R. F. Custodio, “A proposed framework for analysing security ceremonies,” in *Proceedings of the 7th International Conference on Security and Cryptography*, ser. SECRIPT 12, P. Samarati, W. Lou, and J. Zhou, Eds. SciTePress, Jul. 2012, pp. 440–445.
- [6] J. Clulow, “The design and analysis of cryptographic apis for security devices,” Master’s thesis, University of Natal, Durban, 2003.
- [7] W. Rankl and W. Effing, *Smart card handbook*. Wiley, 2010.
- [8] L. Lamport, “Password authentication with insecure communication,” *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981. [Online]. Available: <http://doi.acm.org/10.1145/358790.358797>
- [9] F. F. I. E. Council, “Authentication in an internet banking environment,” 2011.
- [10] C. Soare, “Internet banking two-factor authentication using smartphones,” *Journal of Mobile, Embedded and Distributed Systems*, vol. 4, no. 1, 2012. [Online]. Available: http://www.jmeds.eu/index.php/jmeds/article/view/Internet_Banking_Two_Factor_Authentication_using_Smartphones
- [11] B. Schneier, “Two-factor authentication: Too little, too late,” *Commun. ACM*, vol. 48, no. 4, pp. 136–, Apr. 2005. [Online]. Available: <http://doi.acm.org/10.1145/1053291.1053327>
- [12] R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 2001.
- [13] S. Drimer and S. J. Murdoch, “Keep your enemies close: Distance bounding against smartcard relay attacks,” in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, ser. SS’07. Berkeley, CA, USA: USENIX Association, 2007, pp. 7:1–7:16. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1362903.1362910>
- [14] W. Arsac, G. Bella, X. Chantry, and L. Compagna, “Multi-attacker protocol validation,” *Journal of Automated Reasoning*, vol. 46, no. 3-4, pp. 353–388, 2011. [Online]. Available: <http://dx.doi.org/10.1007/s10817-010-9185-y>