# Ceremonies Design for PKI's Hardware Security Modules

**Jean Everson Martina**[1]*, **Túlio Cícero Salvaro de Souza**[2] **and Ricardo Felipe Custódio**[2]

[1] University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge – CB3 0FD – United Kingdom

[2]Laboratório de Segurança em Computação
Universidade Federal de Santa Catarina
Caixa Postal 476 – 88040-900
Florianópolis – SC – Brasil

Jean.Martina@cl.cam.ac.uk,{salvaro,custodio}@inf.ufsc.br

***Abstract.*** *Ceremonies are a useful tool to HSMs in PKI environments. They state operational procedures and usage scenarios. Their correct construction can lead to a safer operation. This paper presents basic ceremony procedures to manage the life cycle of cryptographic keys and ideas of requirements needed to assure security throughout the usage of ceremonies in the context of an HSM implementing the OpenHSM protocols. It presents ceremonies to make the OpenHSM protocol operational establishing basic building blocks that can be used by any PKI application based in an HSM. Our main contributions are the re-usage of ceremony phases and a survey on formal methods to verify them.*
***Keywords:*** *Key management protocol, Hardware Security Modules, Ceremony Design, Ceremony Analysis*

***Resumo.*** *Cerimônias são ferramentas muito úteis para Módulos de Segurança Criptográficos em ambientes de ICP. Elas descrevem os procedimentos e cenários de uso. Sua correta construção nos permite uma operação mais segura. O presente artigo apresentas as cerimônias básicas para o gerenciamento do ciclo de vida de chaves criptográficas e idéias de requerimentos necessários para assegurar a segurança de um MSC executando os protocolos OpenHSM. São apresentadas cerimônias para tornar o protocolo OpenHSM operacional estabelecendo os blocos básicos que podem ser usados por quaisquer aplicações baseadas em um MSC. Nossas contribuições são o re-uso de fases de cerimônias e a proposição de métodos formais para verificá-las.*

## 1. Introduction

Key life cycle management is one of the most important and challenging issues in public-key cryptosystems. The strict controlling of key life cycle is a difficult but crucial point in Public Key Infrastructures and applications. Although not a trivial task, research and

*Supported by the CAPES Foundation/Brazil on grant #4226-05-4

practice have established good techniques to achieve a reasonable level of protection for PKI sensible data.

The use of hardware devices is a common way to secure key storage and control key life cycle management. There are among them tokens, smart-cards and more elaborate devices called Hardware Security Modules (HSMs). An HSM is a specialised cryptographic hardware built to create a protected environment where keys' unwrapping and use of sensitive data are safe. HSMs allows the control of when and who can use a key. These equipment are normally subject to scrutiny by trusted third parties, who will issue a certification of compliance [FIPS 2002, Estrutura de Chaves Públicas Brasileira 2007]. The certification states that the device is capable to keep security material protected against attacks up to a specific level, and under certain circumstances. It can also take into account the scrutiny level the process was subject to.

The evaluation process takes into account environmental circumstances in which the equipment is operating. This environment must not differ from that where the equipment will operate during its actual usage. Together with environmental constraints, the equipment is also subject to operational procedures that must be followed. This happens in order to assure security not achievable by cryptographic means, like the possibility of the device being stolen or misused. It also tries to address the non-determinism of human peers involved in the process, and how to access the security of their participation. The equipment manufacturer describes these operational procedures in order to try to minimise the environmental threats.

Operational procedures in this sense can be considered *security ceremonies*. Security Ceremonies, as described by Ellison [Ellison 2007], are extensions of protocols, where it is considered environmental consequences and importantly human actions as part of them. Moreover, the protocols are reviewed to reduce the stronger assumptions of the protocol security in favour of weaker ones, or at least show how the former assumptions can be constructed in an expected way. With this idea it can be extended the security threats and its properties to the environment. Ceremonies should be part of the security certification process of any HSM. As such, they should be subject to scrutiny. A better evaluation of threats can give assurance in terms of effectiveness of countermeasures taken by the equipment producer against problems present in the HSM operational environment.

By including the environment, ceremonies end up by including the human peers that will execute the protocols, or at least will expect the results of its execution. In the specific case of PKI-related ceremonies, answering these expectations is a key issue that will help to establish trust. When it is included such type of peers it is also taken care of peculiarities of their operational semantics and weaknesses. In the human peer case, ceremonies should be designed to cope with the usual faults present in human beings. It is known that memory is not reliable, operations are not accurate and the capacity of storing strong keys is not present in humans as they are in computers. As they are part of the protection mechanism in any security system, their weaknesses, if not correctly addressed in a ceremony, can lead to a whole insecure process or protocol.

Martina et al. and de Souza et al. [Martina et al. 2007, de Souza et al. 2008] proposed the creation of an open HSM software architecture called OpenHSM, where key

life cycle can be controlled under a generic protected environment but no operational procedures were discussed. A next challenge is to establish basic ceremonies to the OpenHSM. This should be done in order to address environmental and procedural threats in the previously proposed protocols. OpenHSM was chosen due to its openness characteristics and because of its availability, due to wide deployment in Brazilian academic institutions[Rede Nacional de Ensino e Pesquisa 2009].

The present work will focus heavily on the OpenHSM protocols, so they are a recommended reading for those who wish to understand their details[Martina et al. 2007, de Souza et al. 2008]. It was tried to cover the minimum for understanding the ceremony process. Although focused on this specific HSM architecture, the basic ideas in ceremonies and ceremony analysis can easily be adapted to any HSM, in any environment. Furthermore, as our contribution, it is proposed the re-usage of identifiable phases to ceremony design. This can be applied to simplify any future ceremony design and analysis

The OpenHSM is an environment where PKI keys can be created and managed. It includes protocols to create keys based on threshold cryptography and heavy auditing schemes. Thus, keys' consequent usage in a PKI scenario can be controlled. Although already researched, will not cover some supported operations present in the OpenHSM protocol, such as, changes in administrative groups participants or changes to the ownership of a key.

The focus will be the narrative regarding the ceremony description in a system that will enable to create evidence for support witnessed ceremonies. All ceremony processes will finish with a written act signed by everybody present in the ceremony, which will help to describe the key life cycle. The creation of such records is seen as part of the required environmental procedures needed to establish trust in the HSM and for consequence the PKI, and to satisfy human peer expectations on controlled and audited execution regarding the protected keys.

Corroborating this idea, in PKI ceremonies the presence of auditors is mandatory and often mentioned in standards [Chokhani et al. 2003]. Works in the Management and Governance field also show the importance of such auditing and tracking. Spira [Spira 1999] states that the importance of auditing and ceremonies in corporate governance standards is due to the fact that they validate the legitimacy of operations and enable access to the history of procedures when needed. It also can detect the roots of problems if they occur, thus making the governance standards higher.

It should also be paid attention that such PKI related ceremonies must be already documented somewhere by the big companies that operate in the Digital Certification market, but due to industrial and security concerns - such as the lack of correctness analysis tools - they remain secret. This work should not be compared with those ceremonies, since their stage of evolution could be beyond of what is described in this paper, but seen as an open proposal to drawn attention to its importance. Other work that corroborates in a parallel way the usage of witnesses was recently published by Brainard [Brainard et al. 2006], where he emphasises the importance of people's relations to give security to authentication processes and the threats that can be avoided when other external parties are involved in a collaborative security process.

Although important to a safe protocol operation, ceremonies are not often seen

in academic research. Recently, ceremony design and analysis were introduced by Ellison [Ellison 2007, Ellison 2002]. He states: ”ceremonies extends the concept of protocols by including human beings as nodes in network”, but certainly this can be extended even further to the environment and the relations between subjects, environment and security targets. This can give a broader coverage of *out of bounds* operations and problems to security protocols

This paper shows the concepts of ceremonies and the related work stressing the importance of this area of research. Section 2, introduces HSMs' ceremonies in a PKI environment, describing the basic building blocks and their usage in this HSM context. The focus was a description of the ceremony process. It also introduces the main contribution that is the re-usage of ceremony phases. Section 3 describes an approach to verify in a systematic way humans peers cognition, a small part of the ceremonies problem. Finally in Section 4 discussed the outcomes of this ceremonies and ceremony analysis proposals. Finally, it was proposed some future research to be done in this field.

## 2. HSMs' Ceremonies

The ceremonies presented in this section were designed to show the importance of the following points:

- Ceremonies can be divided into phases, where later, during the design of other ceremonies, can be reused and making them the basic blocks that can keep the proprieties needed for tackling out-of-bounds operations;
- Auditing phases are a good strategy to detect off-line attacks and tamper attempts, without putting in danger any operation in the PKI environment;
- Strong assumptions should always be reduced to weaker ones, trying to relate and understand them and propose less questionable statements instead;
- Common human peers cognitive slips should be avoided in order to create a safer environment. It is done by introducing some non by-passable operations to human peers that are checked by the ceremony process.

The ceremonies presented below are those required to deploy a simple PKI application using the OpenHSM's protocols. These ceremonies can be easily adapted and used with any HSM. They are a basic block of a whole PKI ceremony set. The HSM-related ceremonies described here are: Initialisation, Key Generation and Key Usage. In their representations, there are six involved entities - 4 humans groups and 2 devices. These entities are:

- Auditors Group (human): represents the auditing body of the use of the HSM. It is responsible for checking the correctness of the HSM's activities;
- Operators Group (human): this is the group responsible for authorising the use of HSM's managed keys;
- Administrators Group (human): initialises and creates groups and keys in the HSM. Is responsible for maintaining the HSM running;
- Host Machine (device): a trusted platform that was prepared by the administrators group and contains a trusted operational system. This machine allows the groups to interact the HSM;
- HSM (device): the cryptographic hardware that manages the key's life cycle in a secure way;
- HSM Producer (human): represents the manufacture of HSMs.

## 2.1. Initialisation Ceremony

The initialisation ceremony is an important step towards a secure HSM. Although not always taken into account in protocols, a ceremony starts even before an HSM is bought, when it is still under production. These issues should be considered as Trust Management ones, since to include an HSM in any process it should come from a trusted source and in a trusted, or at least verifiable, path.

Ceremonies are used to establish trust anchors, building the trust from scratch and allowing actions to be traced to something that is trusted. The difference between protocols and ceremonies is that the trusted assumptions are generally smaller in ceremonies if compared to protocols. Thus, without evidence that the process was strictly followed, there is no basis for trusting in a ceremony. Also, a design that is not secure against human inherent problems can be subject to vulnerabilities. The initialisation ceremony starts with a Pre-initial Phase as illustrated in Figure 1. In this phase, it is proposed that the HSM's producer issue its own certificate (Step 1) that will be used as its identity and to sign HSM's certificate and software.

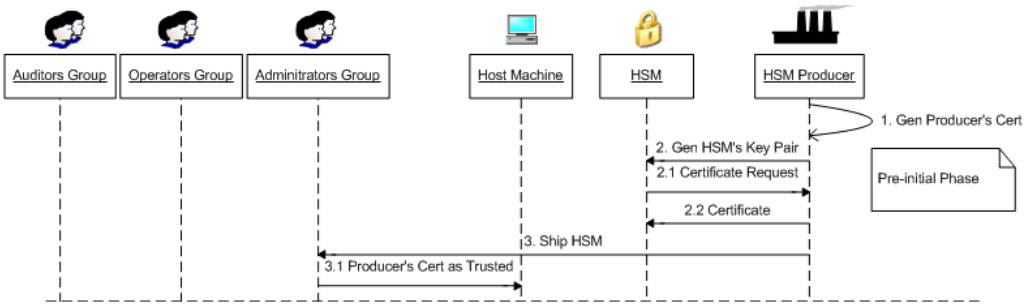


**Figure 1. Ceremony to initialise the OpenHSM - Pre-initial Phase**

As part of the fabrication process, the producer requests the generation of the HSM's key pair during its test run at factory (Step 2). Thus, the HSM issues a certificate request and returns it to the HSM's producer (Step 2.1). Then, using the HSM's serial number that is unique and is marked on the HSM's chassis, the producer issues the HSM's certificate. The producer and HSM's certificates are uploaded to the HSM (Step 2.2). The software is also signed and included in the HSM shipment (Step 3). Therefore, the HSM certificate and the software signature will guarantee the origin of the HSM shipment and will be anchored to the HSM's producer certificate and in the human-to-human relationship that was previously established. The last step of this phase consists in the recipient of the HSM to trust the producer's certificate. To do this, the recipient copies the producer's certificate, verifies and installs it into the host machine. After this, the host machine can verify the integrity of the HSM's software and communication's channels.

The producer certificate will be distributed to its clients, assuring the virtual anchor already existent in the real world: The customer must trust the HSM producer, since that without this trust can be assumed that the HSM is compromised by default. The transference of this certificate to the HSM user will happen in a human-to-human data channel, allowing the interaction between parties in a form that gives enough confidence to the other human counterpart.

It is difficult to establish what is a good human-to-human channel, but it can be

considered good as it is not possible for an attacker monitor or interfere with the channel giving confidence to the humans involved. Some humans require face to face contact with a company representative, receiving the digital version of the certificate together with a signed letter containing the certificate digest, while for others, just receiving the letter by post in a signed-for delivery could be enough. This analysis is far beyond from what is achievable today with the current method for verification, and is up to peer's discretion.

In this sense, it can also be seen that the producer is able to create a Cryptographic Identity for the HSM, binding it physically, and by doing so he ties the logical contents of the HSM with its physical instance, using the protections in the security perimeter of the HSM as a shield to avoid tampering in this process. An important objective achieved by this Pre-initial Phase is to reduce the strong assumption that the HSM is not compromised in transit before it arrives in the destined domain. It is reduced to the security of the human-to-human interaction that exist with the HSM producer. The steps that are in between can be easily reproduced and checked to give confidence to the user. Additionally, an auditing process can easily detect erros and misconduct.

The second phase starts with the HSM and software installation and is called Initial Phase, shown in Figure 2. In this phase, the management software is uploaded to the host machine (Step 4). Before installing the software, the installation procedure in the host machine verifies the signature of the management software validating its signature and certificate (Step 4.1). If the signature is verified then the management software is installed (Step 4.2). We should point that the preparation of the Host Machine in itself is another ceremony, and is not covered here.

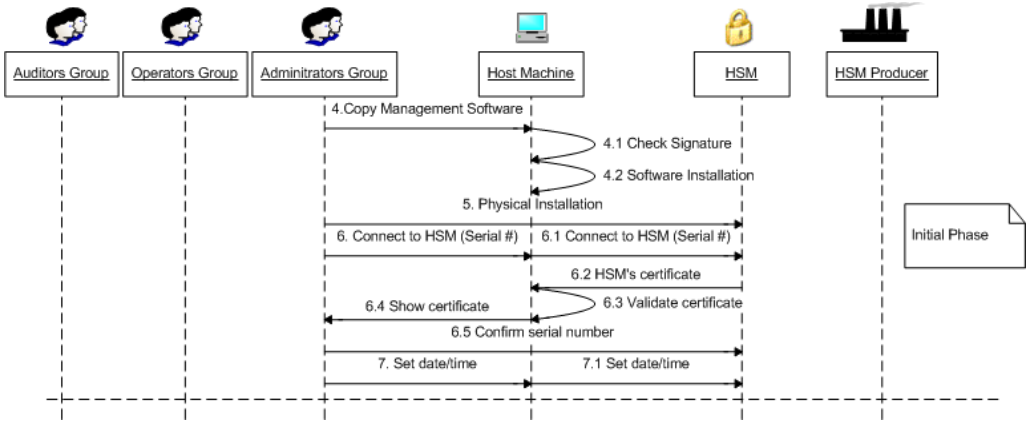


**Figure 2. Ceremony to initialise the OpenHSM - Initial Phase**

With the software installed, the HSM's physical installation and identification can be proceed (Step 5). From this point on, it can be established the first connection to the HSM. To connect, the administrators group request the Host Machine to connect to the HSM (Step 6) and in turn, the Host Machine forwards this request to the HSM (Step 6.1). As the HSM is being connected by the first time - even if the HSM has already been used it can be reset and reach this state again[Martina et al. 2007] - the connection will be established using an encrypted channel, where, the HSM will send its certificate to the Host Machine (Step 6.2). The Host Machine will then check the certificate's validity, using the previously trusted producer's certificate (Step 6.3), and upon validation, the certificate will not be shown to the user (Step 6.4). Then, the user enters the HSM serial number

marked on the HSM chassis (Step 6.5). If the number entered by the user - remember that the user does not have access to the content of the HSM certificate before he has entered the serial number - is equal to the value in the certificate extension, the HSM sends a message to the computer, that it will forward the user, stating that the firmware in the HSM is original, was created by the expected producer and that it is the expected HSM (Confirms the HSM identity).

With this phase ran, the user could be prevented to use a tampered with or unexpected HSM. By blindly asking the user for information and letting the system deal with the comparison we solve a possible human peer weakness point. This proved to give a better degree of assurance that the user was not fouled by an attacker just using some software or tampering trick. After being assured that they are connected to the right and trusted equipment, the Administrators Group must set the date and time in the Host Machine (Step 7), which will then synchronise its clock with the HSM's (Step 7.1). This step is important to guarantee further auditing steps, since clock de-synchronisation can lead to severe problems in auditing processes.

After having the HSM properly connected, validated and having the correct software installed, the next phase will execute the protocol initially proposed by Martina et al. [Martina et al. 2007]. This phase is called Initialisation Phase as shown in Figure 3 and incorporates the Auditors group and backup procedures proposed by de Souza et al. [de Souza et al. 2008] which is essential to the goals of having witnessed ceremonies. To deeply understand this phase, it is recommend the reading of the above referenced work [Martina et al. 2007, de Souza et al. 2008], since the abstract level of this paper could obscure key aspects covered by these previous works.

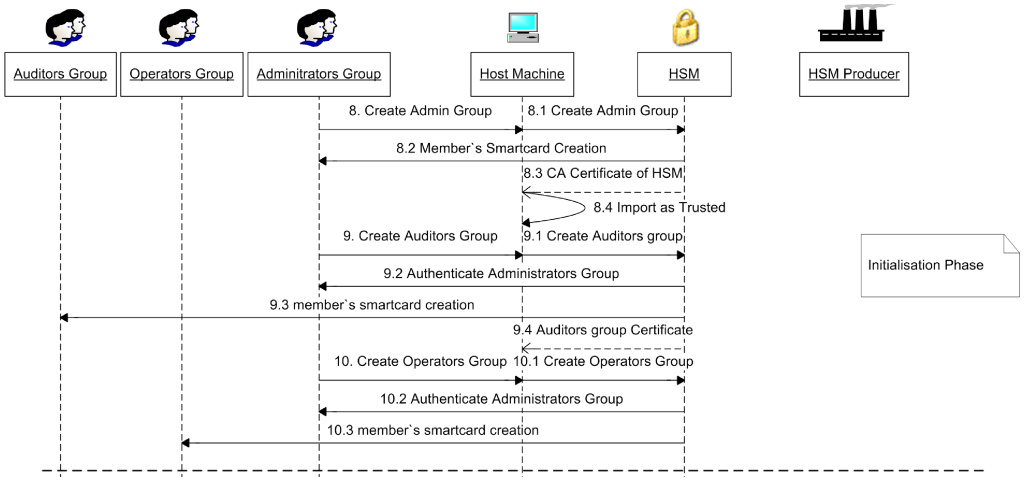


**Figure 3. Ceremony to initialise the OpenHSM - Initialisation Phase**

The Administrators Group will issue a command to the Host Machine software interface demanding the creation of the Administrators group representation within the HSM (Step 8). The Host Machine will forward that request to the HSM (Step 8.1), which will proceed with the creation of each Administrators physical token (Step 8.2) using the OpenHSM protocol. These physical tokens can range from simple smart-cards to a collection of biometric authentication devices. Their purpose is to bond the identity of an Administrator with its representation in the HSM instance. With all Administrators in

the group having their token created, the HSM will output its self signed certificate to the Host Machine (Step 8.3), which will import it into the trusted certificate list (Step 8.4).

Following the Initialisation Phase, the Administrators will request the Host Machine to create at least one Auditors group (Step 9), and the Host Machine will forward this request to the HSM (Step 9.1). To do so, the HSM will authenticate the previously created administrators using the standard procedures and parameters required by the protocols (Step 9.2). Then, will create for each member of the new Auditors Group a physical token (Step 9.3) and will export a certificate that represents this group to the Host Machine (Step 9.4). This step is an extension to the protocols proposed by Martina et al., and is a requirement in the auditing process established by de Souza et al. It is meant to start the auditing life of the newly initialised HSM, and to establish confidence in all procedures in the ceremonies from here on. In the HSM ceremony process this is a very important point, since that up to here it can be reduced to the assumptions of the initial human-to-human communication channel.

To reach full functionality the OpenHSM needs at least one Operators Group. Administrators will request the Host Machine to create one Operators Group (Step 10), and the Host Machine will forward this request to the HSM (Step 10.1). To do so, the HSM will authenticate the previously created administrators (Step 10.2) and will create for each member of the new Operators Group a physical token (Step 10.3). Once this procedure finished, the HSM is initialised.

When authenticating any group belonging to the HSM, it is introduced new steps where human cognition can be subject to attacks from malicious parts. In this case it is supposed that the messages will be clear enough to the user to bond it with its physical actions towards the authentication. In our analysis process, it has been checked that the order on the messages issues by the protocol in the ceremony would not help an attack to succeed in the presence of either an active or a passive attacker. Much of this comes from the intrinsic characteristics of the threshold cryptography used to spill the responsibility among the group participants .

After being correctly initialised, it is started the documentation of the HSM's life and the keys protected by it. The Auditing Phase is shown in Figure 4. In this phase, the auditors group will request the management software in the Host Machine to export the HSM logs (Step 11). The Host Machine will pass this request to the HSM (Step 11.1). The HSM will then authenticate the Auditors (Step 11.2) and export the requested logs (Step 11.3), signed by the private key related to the auditor group that requested them to be exported. This signature can be verified using the certificate produced in Step 9.4.

With the logs already in the Host machine, the Auditors group will create the Ceremony Act (Step 12), that consists of a textual description of all events that happened during the ceremony, in accordance with the PKI requirements and policies. This act can contain the name and roles of all principals present in the location where the ceremony happened and also everything that was noted by the attendants. The Act will then be printed (Step 12.1), and signed (manually and/or digitally) by all present auditors (Step 12.3), being followed by the operators (Step 12.4) and finally the administrators (Step 12.4).

A new trust-verification point can be attested by the signatures of all attendants.

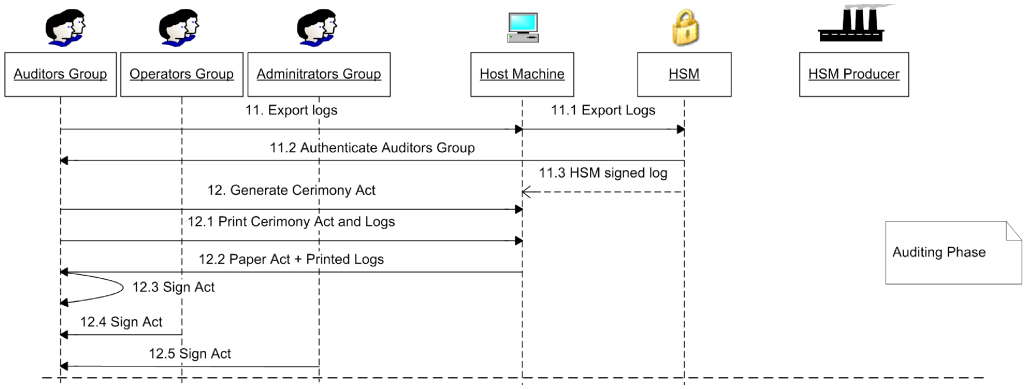


**Figure 4. Ceremony to initialise the OpenHSM - Auditing Phase**

They will declare that up to this moment the ceremony followed the stated procedures, and the results are those expected.

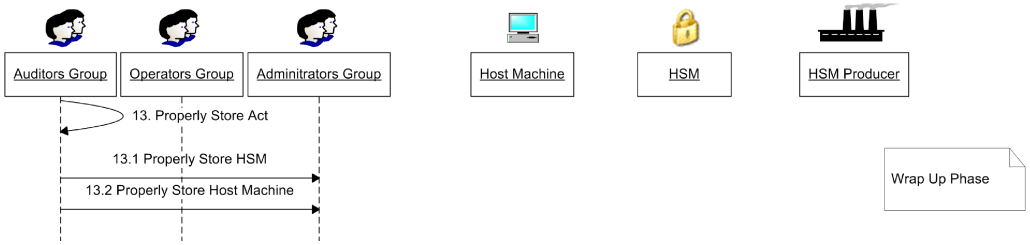


**Figure 5. Ceremony to initialise the OpenHSM - Wrap Up Phase**

Now, the Wrap Up phase is shown in Figure 5. It must be assured that all the cautions are taken to safely store the Act (Step 13), e.g., be published in a reference newspaper. The HSM (Step 13.1) and the Host Machine (Step 13.2) also need to be properly stored, thus preventing them being subject to any tamper attempt while waiting to be used again. The storage facilities should be a Safe Room, locked and physically sealed by the Administrators Group until the next usage.

During this first ceremony process in the OpenHSM life, it was introduced the ideas of ceremony phases, which will permit them to be reused in later ceremonies, and will help in their verification processes. It also used techniques of blind input by the user, thus avoiding him being fooled by a single attack technique, letting the protocol to deal with the error prone comparisons. And, finally, it is introduced the idea of human trust-point verification which can be used to reconstruct later in the HSM's life all the operations. This reconstruction can help the tracking of potential problems by interviewing the attendants in the disputed ceremony, if necessary.

## 2.2. Key Generation Ceremony

The HSM, once initialised, is ready to manage keys in its security environment, and this ceremony is covered by Figure 6. It is the next step after the initialisation.

Initially, the administrators group, in order to cover the Setup Phase, initialise the host machine (Step 1). This step consists in switching the host machine on and getting it ready to start the HSM management tool. Following, the administrators verify the tool's

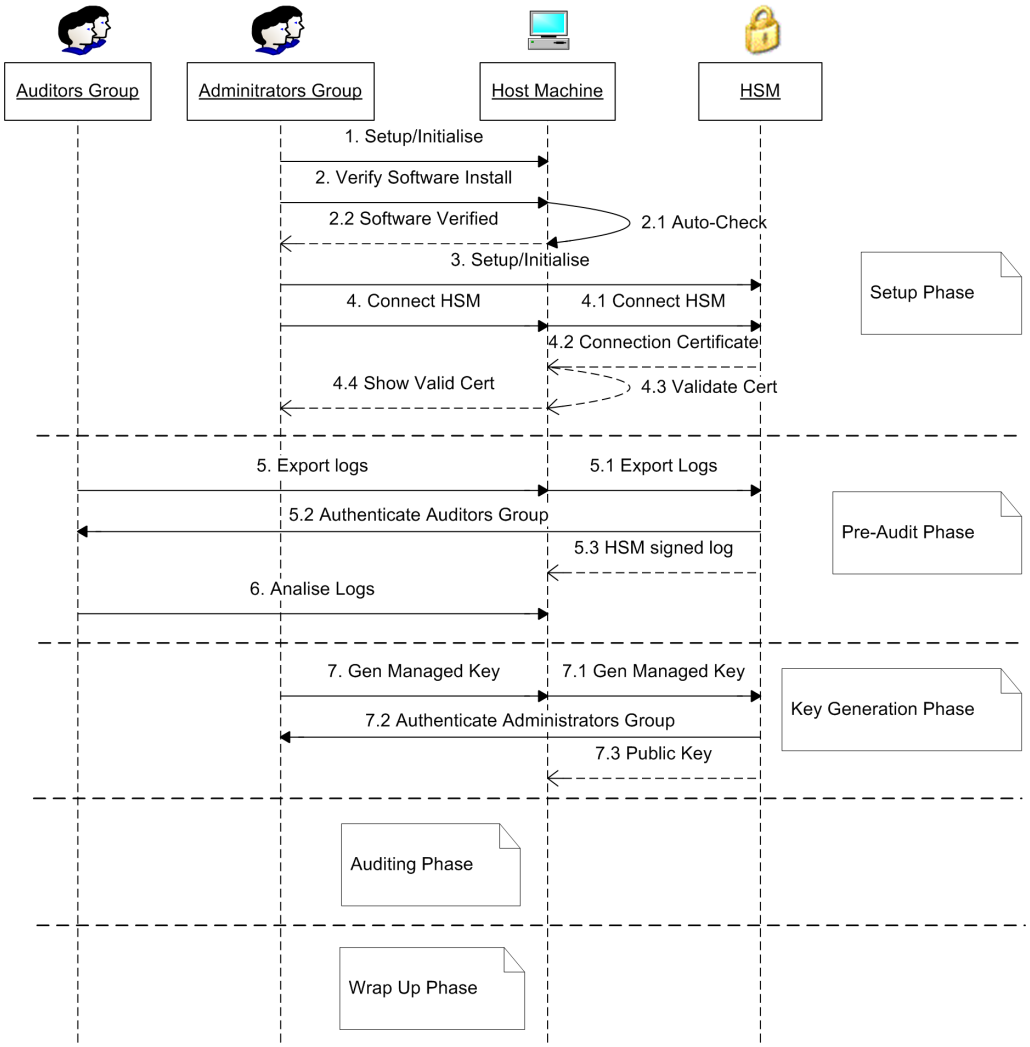


**Figure 6. Ceremony to Generate Key in the OpenHSM Environment**

signature (Step 2) and run the auto-check (Step 2.1) to confirm its authenticity (Step 2.2). This checking should be extended to the host machine regarding to its integrity from the previous ceremony, a ceremony not covered here. The administrators group must also start the HSM (Step 3). This process includes energising it, connecting the token reader and confirm its non tampered state. Finally, in the setup phase, the administrators connect to the HSM through the management tool (software) (Step 4) and call a self check routine. This routine should verify the crypto algorithms against its standard procedural tests and assure no tampering attempt or unauthorised run happened while the HSM was stored. The connection certificate is verified again by the Host Machine (Step 4.3), and it once again requests the Administrators to input the HSM serial number written on the chassis to confirm the software-hardware bonding, using a blind test(Step 4.4).

The Pre-Audit phase takes place in order to guarantee that no unauthorised activity has been made since the last HSM usage. The main goal is to connect the executions in the next phase with the previous runs, whatever they were, giving subside to claim full life cycle documentation by the written acts. Starting this phase, the auditors request the logs of the HSM through the Host Machine (Steps 5 and 5.1). The operations must be

authorised, then the auditors must be authenticated (Step 5.2). On success, the signed HSM's logs are exported (Step 5.3) and can be analysed by the auditors group (Step 6). This analysis should give enough confidence to the auditors to assure the three requirements stated above: no unauthorised activity, no tampering and self-correctness of the HSM and Host Machine.

Next phase is Key Generation. This is the centre of the ceremony and will enable the administrators to generate a new managed key in the HSM protections environment. The process, as stated by Martina et Al.[Martina et al. 2007], starts when the administrators requests to generate a key to the HSM through the Host Machine (Steps 7 and 7.1). In the step 7.2, the Administrators are authenticated and, on success, the HSM will internally generate the key-pair and will export its public part (Step 7.3).

The next phases in this ceremony, Auditing and Wrap Up Phase, have already been stated in section 2.1 and will not be covered here. Their purposes are to document the key's life-cycle and safe storage oh the equipment.

Actually, this is considered one of the contributions to the ceremony analysis and design process, since by reusing ceremony parts, they can be written more generally, and test them just once, instead of re-including it in every single sub-ceremony of the whole design. In brief, this ceremony has a preparation and finishing stages (Setup and Wrap Up Phases), the chronological records exported before and after the key creation (Pre-Audit and Auditing Phases) and the key creation itself. This means that it should be self contained and self verifiable, making the external auditing easier and less error prone.

### 2.3. Key Usage Ceremony

The usage of a managed key is the main purpose of an HSM. This makes this routine the most called and present during the life-cycle management, since the key must be released in order to be used in any application. Its ceremony, shown by Figure 7, is split in 5 phases:

In this ceremony, it is achieved a very good level of already designed phases, which makes it really easy to express. As a goal in the design strategy, it is now shown this ceremony just using the intrinsic protocol-related instructions, reusing then all the previous phases.

The Key Usage Phase involves only operators, who are responsible for releasing a key. In order to load the key and release it for usage, the operators send a request to the Management Tool, installed in the Host Machine, and it forwards it to the HSM (Steps 1 and 1.1). As specified in the Protocol [Martina et al. 2007], the operators are responsible for the key usage, and must be authenticated as shown in Step 1.2. Once the key is loaded on the HSM's memory, it is possible to use it, for instance, in a PKI environment, for issuing a certificate or sign a certificate revocation list (Step 2). The loading of a key can be used based on a policy. The usage policy is set by the operators group when loading the key (Step 1). Possible policies can include a maximum number of usage or period of time, for instance, 30 minutes. Considering this nuances, the *unload key request* (Step 3) must only be executed if the policy was not follow correctly.

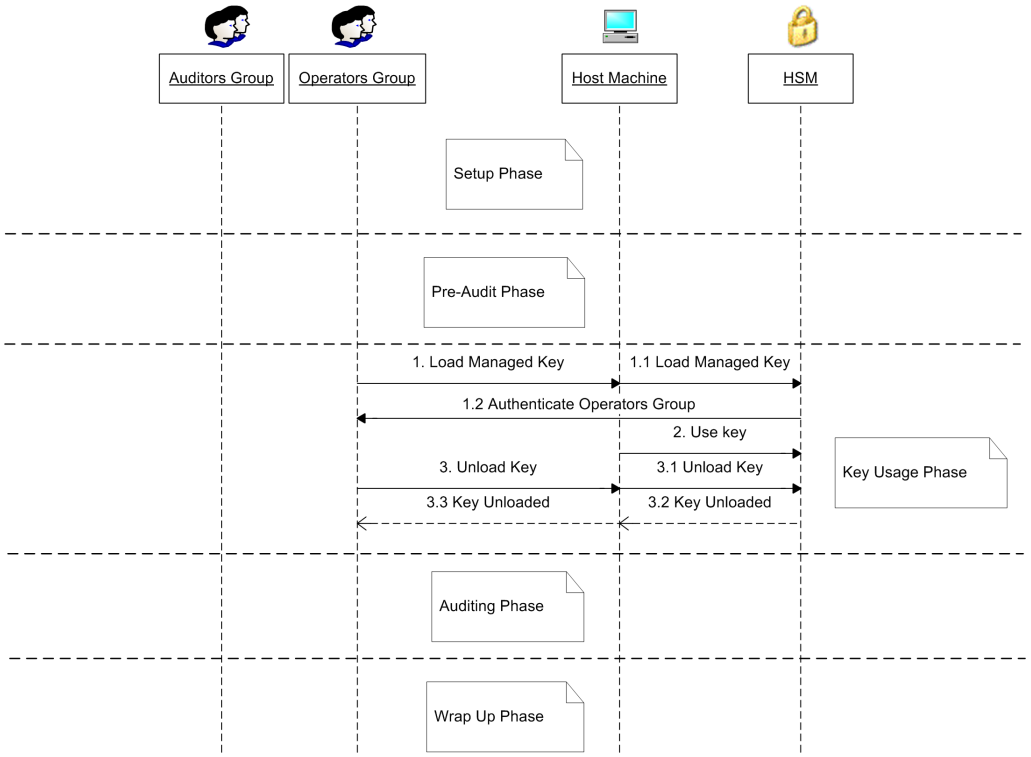


**Figure 7. Ceremony to use a key managed by the OpenHSM**

## 3. Initial Formal Verification Method

Taking Ellison's ideas on security ceremonies can be seen a vast amount of things that must be covered to declare a ceremony secure and trusted. Due to this fact, one reasonable approach is breaking the problem into reasonably small parts and try to verify one at a time. Dividing his approach we can find two different classes of problems to deal with when analysing ceremonies: The first one regards humans peers interaction/expectation and the second regards environmental conditions that the protocol is subject to. The problems regarding humans peers - choice of the current analysis work - can be presented as how adapted the protocol is to cope with limitations of humans behind the computer screens. Especially how this affects the security and trust in a systematic manner. The second class is broader, and can include almost anything that is not included in the ceremony as a protocol or a human peer. Furthermore, the representation and verification of environmental conditions in ceremonies, and by extension protocols, can be a key to understand better problems on protocols composability, making them trustworthy.

The work of Rukšėnas et al. [Ruksenas et al. 2008, Ruksenas et al. 2007] can be an answer to part of the first class of problems that should be verified in ceremonies. They developed a human error cognitive model, which initially applied to interaction on interfaces. This model can be directly applied to a human-protocol interface. Furthermore, taking recent Rukšėnas et al. [Ruksenas et al. 2008] extensions, cognitive slips can be easily verified in protocols in the presence of different types of attacker. Their model supports a description of any human peer in protocols, taking its point of view and describing his interpretation and the effects he can cause in security and the peers impressions on the protocol run (trust). Another important feature of Rukšėnas et al. is the support of envi-

ronmental descriptions. Although not yet very well developed, it shows great potential to model the second class of interactions. Rukšėnas et al. doesn't formalise the environment due to its complexity, but leaves an input to add it later to the model.

One reasonable approach to embrace Rukšėnas et al. method and use it in ceremonies is by applying it using Bella's [Bella 2007] goal availability ideas. Following this principle, it must be taken into account each peers' point of view and work towards formal guarantees available to each human peers in the ceremony. Furthermore, the approach requires that these guarantees must be checked only by what is visible to the human peer during his participation in the ceremony. This correlates with the human idea of trust, where we build up trust over facts that can be easily verified and understood.

Rukšėnas' verification method, describes a dynamic interaction. It is fairly easy to understand their choice on modelling a device, and then try to map user interpretation and effect inside an environment, since their main goal was verifying the usability of interfaces. Within this structure they can easily distinguish among user perception (input signal to the human peer), consequences of user's actions (output signals) and the user's internal state, in this case his memory. The cognitive architecture built by them is a higher-order formalisation of plausible cognitive behaviour. It is implemented in the HOL Theorem Prover[Gordon 1993] and in the SAL Model Checker[Bensalem et al. 2000]. This tries to map standard user behaviour, but of course, it is unable to detect those who act outside the cognitive standard. As it is targeted to find general pitfalls in ceremonies the approach seem reasonable to use.

During the design phases of the ceremonies shown in Section 2 we used the method proposed by Rukšėnas et al. adapting it to the specific scenarios of our ceremonies. The method enabled us to identify and measure the application of concepts, such as the blind copy and paste (Section 2.1), in a controlled way. The method seems promising, but more studies are need in order to develop a generic system, which can make the design and verification of ceremonies less error prone, at least in the optics of human peer cognition capabilities.

## 4. Final Considerations

The initial target was the description of basic operational procedures to an HSM in a PKI environment. It was taken into account recent work on the area of ceremonies and applied to the OpenHSM protocols. The ceremony set was the natural extension of the security descriptions and were stated to give assurance and guarantees in the usage of the module. It was tried to describe the ceremonies by sketching real scenarios and environment where the OpenHSM would be introduced.

It was shown that ceremony design to an HSM could be divided in reusable phases, where can be established trust and threat mitigation requirements and then reuse such phases in other ceremonies that share the same principles. This contribution is important to ceremony design since by doing that it is possible to reduce the explosion in details the ceremony design can lead to. We also surveyed the until now untouched formal ceremony verification area, trying to sketch a reasonable approach to validate the claims in the ceremony construction process.

As future work, in the ceremony design area, it is being proposed the expansion of such ceremonies to cover all HSM operations in a PKI environment, and the extension

to a complete Certification Authority operation, reusing basic blocks already described in this paper.

## References

Bella, G. (2007). *Formal Correctness of Security Protocols*, volume XX of *Information Security and Cryptography*. Springer Verlag.

Bensalem, S., Ganesh, V., Lakhnech, Y., Munoz, C., Owre, S., Rue, H., Rushby, J., Rusu, V., Sadi, H., Shankar, N., Singerman, E., and Tiwari, A. (2000). An overview of SAL. Technical report.

Brainard, J., Juels, A., Rivest, R. L., Szydlo, M., and Yung, M. (2006). Fourth-factor authentication: somebody you know. In *Proceedings of the 13th Conference on Computer and Communications security*, pages 168–178, New York, NY. ACM.

Chokhani, S., Ford, W., Sabett, R., Merrill, C., and Wu, S. (2003). Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework. RFC 3647 (Informational).

de Souza, T. C. S., Martina, J. E., and Custódio, R. F. (2008). Audit and backup procedures for hardware security modules. In *Proceedings of the 7th Symposium on Identity and Trust on the Internet*, New York, NY. ACM.

Ellison, C. (2002). Improvements on conventional pki wisdom. In *Proceedings of the First Annual PKI Research Workshop*, Gaithersburg, MD.

Ellison, C. (2007). Ceremony design and analysis. Cryptology ePrint Archive, Report 2007/399. http://eprint.iacr.org/.

Estrutura de Chaves Públicas Brasileira (2007). Manual de Condutas Técnicas 7 - Vol I (MCT 7 Vol. I) - versão 1.0. Technical report, Instituto Nacional de Tecnologia da Informa cão - ITI.

FIPS (2002). Security requirements for cryptographic modules, FIPS PUB 140-2.

Gordon, M. J. C. (1993). *Introduction to HOL: A Theorem Proving Environment*. Cambridge University Press.

Martina, J. E., de Souza, T. C. S., and Custódio, R. F. (2007). Openhsm: An open key life cycle protocol for public key infrastructures hardware security modules. In *Fourth European PKI Workshop: Theory and Practice*, volume 4582 of *LNCS*, pages 220–235. Springer-Verlag.

Rede Nacional de Ensino e Pesquisa (2009). ICPEDU - Infraestrutura de Chaves Pblicas para Pesquisa e Ensino . `https://www.icp.edu.br/`.

Ruksenas, R., Curzon, P., and Blandford, A. (2007). Detecting cognitive causes of confidentiality leaks. In *First International Workshop on Formal Methods for Interactive Systems*, volume 183 of *ENTCS*, pages 21–38.

Ruksenas, R., Curzon, P., and Blandford, A. (2008). Modelling and analysing cognitive causes of security breaches. *Innovations in Systems and Software Engineering*, 4(2):143–160.

Spira, L. F. (1999). Ceremonies of governance: Perspectives on the role of the audit committee. *Journal of Management and Governance*, 3:231–260(30).