OpenHSM: An Open Key Life Cycle Protocol for Public Key Infrastructure's Hardware Security Modules*

Jean Everson Martina[1,**], Tulio Cicero Salvaro de Souza[2], and Ricardo Felipe Custodio[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 (LabSEC)
Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88040-900 – Florianópolis – SC – Brasil
Jean.Martina@cl.cam.ac.uk, salvaro@inf.ufsc.br, custodio@inf.ufsc.br

Abstract. The private keys used in a PKI are its most important asset. Protect these keys from unauthorised use or disclosure is essential to secure a PKI. Relying parties need assurances that the private key used to sign their certificates is controlled and managed following pre-defined statement policy. Hardware Security Modules (HSM) offer physical and logical protection and should be considered for any PKI deployment. The software that manages keys inside an HSM should control all life cycle of a private key. Normally this kind of equipment implements a embedded key management protocol and this protocols are not available to public scrutiny due to industrial interests. Other important issue is that HSMs are targeted in their development to the Bank industry and not to PKI, making some important PKI issues, like, strict key usage control and a secure auditing trail, play a secondary role. This paper presents an open protocol to securely manage private keys inside HSMs. The protocol is described, analysed and discussed.

Keywords: Key management protocol, Hardware Security Modules.

1 Introduction

Key management includes key establishment, rules and protocols for generating keys, and the subsequent handling of those keys. Securely manage cryptographic keys is one of the most important and resource consuming efforts to guarantee the security on public key cryptosystems. It means we must have a rigid control on the life cycle of those keys and this is not a trivial task. Moreover, we can

* Work supported and founded by Rede Nacional de Pesquisa/Brazil.
** Supported by CAPES Foundation/Brazil on grant #4226-05-4.

J. Lopez, P. Samarati, and J.L. Ferrer (Eds.): EuroPKI 2007, LNCS 4582, pp. 220–235, 2007.
© Springer-Verlag Berlin Heidelberg 2007

assume that a public cryptosystem can be considered as secure as the keys are secured. Taken this as a premise we should guarantee that a key is strictly secure during all events on its life cycle. A way to achieve this is by designing systems to securely create, manage, copy, and destroy private keys maintaining an audit record of all uses during the key life.

Hardware security modules are specific hardware designed to protect key against any kind of logical and physical tampering or extraction of cryptographic material from its environment. The HSMs are normally hardware that passed by certification procedure. The most widely known are FIPS 140-2 [1], a certification developed by USA's Department of Commerce, and Common Criteria [2], developed by a consortium having in mind the creation of protection profiles for such equipment. Normally these equipments implement their own key management protocols, which due to industrial concerns are not made publicly available for scrutiny, making us reasoning about their true correctness. Another important issue to the actual HSMs is their targeted development to the Bank industry and not to PKI, making some important PKI issues, like, strict key usage control and auditing, play a secondary role in the security context, normally making the HSM just a digital safe where we trow our keys.

Key management life cycle has been studied by many researchers [3,4,5]. Menezes et al. [6] discuss the public key management in a general context, including from user registration and initialization to key revocation.

However, protecting a private key in a CA context was always one of the main concerns in any PKI deployment, and is discussed by Jeff Schiller [7]. He states that protection schemes can be broken into two basic classes: schemes where no human ever has access to the raw private key material and schemes where a human may have access to the raw private key material. In the first, the private key is stored in a hardware device which itself requires a hardware token to operate. He advises that when a key is generated by this kind of devices, special attention should take in account to deploy facilities to recover the key from a failed unit.

Having an open protocol is an important matter when concerning to cryptographic algorithms and to cryptographic protocols. This was stated by Auguste Kerckhoffs[8] in the 19th century and by Claude Shanon[9] in 1948, and our main concern when designing this protocol is the lack of access to the industry owned protocols due to their intent to protect their copyrighted material. This makes us always suspicious when using a so sensitive equipment like a HSM to control keys for a Certification Authority in a PKI environment.

This work presents a cryptographic protocol to manage private keys. Our focus is an open key life cycle protocol for public key infrastructure's Hardware Security Modules which will fit on Schiller's first category. The proposed protocol was embedded in a hardware designed to be a HSM holding all physical tampering countermeasures.

The paper is structured with this introduction section, followed by section 2, where we present all the protocol basic ideas and concepts, as well as the premises we assumed during the protocol development in subsection 2.1. Later on section 3