Dirk Fox

# Hardware Security Module (HSM)

## Hintergrund

Jede kryptographische Sicherheitslösung – sei es eine Verschlüsselung, eine elektronische Signatur oder eine Authentifikation – sorgt für eine „Verschiebung" des Schutzbedarfs von den zu schützenden Daten auf einen kryptographischen Schlüssel. Damit reduziert sich der Schutzaufwand in der Regel erheblich, da ein Schlüssel einfacher zu schützen ist als die Daten selbst – allerdings sind die Daten auch nur so gut geschützt wie der Schlüssel.

Kryptographische Schlüssel müssen jedoch unterschiedlichen Anforderungen genügen, die es praktisch unmöglich machen, sich einen Schlüssel auswendig zu merken – sie müssen eine bestimmte (Mindest-) Länge besitzen,[1] zufällig (oder so gut wie zufällig) gewählt und einmalig sein.

Aber allein die geeignete Wahl der Schlüssel ist noch kein ausreichender Schutz. Über den gesamten Lebenszyklus, d. h. von der Erzeugung über die Nutzung bis zur Vernichtung, müssen unberechtigte Zugriffe auf einen Schlüssel wirksam verhindert werden.

Da liegt es nahe, kryptographische Schlüssel nicht in PCs und anderen Mehrzweck-IT-Systemen, sondern ausschließlich in speziellen Geräten zu erzeugen, zu speichern und zu nutzen, die einen unberechtigten Zugriff verhindern – so genannten Hardware Security Modules (HSMs).

## Anforderungen

Damit ein HSM kryptographische Schlüssel wirksam vor unberechtigten Zugriffen schützt, muss es verschiedenen Anforderungen genügen:

- Es muss über einen Schlüsselgenerator verfügen, der zufällige Ausgaben erzeugt – also eine „echte" Zufallsquelle besitzen.
- Es muss alle benötigten kryptographischen Operationen (Signier-, Verschlüsselungs- und ggf. Hashalgorithmen) beherrschen, damit die Schlüssel das HSM nie „verlassen" müssen.
- Es muss Schutz vor Seitenkanalangriffen bieten, wie z. B. Timing Attacks oder Differential Power-Analysis, die aus Rechenzeit bzw. Stromverbrauch eines HSM den Schlüssel gewinnen.[2]
- Es muss „tamper resistant" sein, d. h. auch Angriffen auf die Gerätehardware widerstehen.

Die Tamper Resistance-Sensorik stellt dabei eine besondere Herausforderung dar: Da sich Angriffsversuche auf die Hardware nicht verhindern lassen, muss das HSM sie erkennen und ggf. das Schlüsselmaterial löschen, bevor der Angreifer darauf zugreifen kann. Das führt jedoch zu einer weiteren Anforderung, sofern eine Löschung des Schlüsselmaterials Schaden verursacht (z. B. die Daten einer verschlüsselten Festplatte unzugänglich macht) und einen Denial of Service-Angriff ermöglicht:

- Das HSM muss über eine Möglichkeit zum „Cloning" verfügen, d. h. die Erzeugung eines HSM-Duplikats mit demselben Schlüsselmaterial.[3]

## Realisierung

HSMs werden seit über 20 Jahren in Banken in Gestalt spezialisierter Systeme für die schnelle Verarbeitung einer hohen Zahl kryptographischer Transaktionen eingesetzt, entweder als externes Gerät oder als Erweiterungskarte.

Mit der wachsenden Leistungsfähigkeit von Chipkarten-Prozessoren wurden HSMs auch als SmartCards möglich, die alle gängigen kryptographischen Verfahren beherrschen, Speicher für mehrere Schlüssel und Zertifikate bieten und teilweise sogar über einen Zufallszahlengenerator verfügen. Zwar sind Rechen- und Kommunikationsgeschwindigkeit von SmartCards niedrig; als Zugangstoken oder privater Schlüsselspeicher eignen sie sich jedoch sehr gut.

In Public Key Infrastrukturen (PKIs) werden für den Schutz von Zertifizierungsschlüsseln auch PCMCIA-HSMs eingesetzt: sie sind klein, ausreichend schnell und bieten im Unterschied zu SmartCards mehr Möglichkeiten für Tamper Resistance-Mechanismen.

Für den Datenaustausch mit HSMs haben sich standardisierte, anwendungsspezifische Ein-Ausgabe-Schnittstellen durchgesetzt, wie z. B. PKCS#11 (auch ‚cryptoki' genannt) für SmartCards, OpenSSL für SSL-Server und die Crypto-API (CAPI) von Microsoft, beispielsweise für .NET-Anwendungen.

## Einsatzbereiche

HSMs werden zu sehr unterschiedlichen Zwecken eingesetzt, insbesondere

- als kryptographischer „Token" oder spezieller Hardware-Chip (z. B. zur Verschlüsselung von E-Mail, für VPN-Schlüssel und zur Authentifikation) wie bei Trusted Computing[4],
- als kryptographisches Transaktionssystem („Crypto-Coprozessor") z. B. in Banken oder SSL-Servern,
- für Zeitstempel und
- für die Erzeugung von Schlüsselzertifikaten und Rückruflisten in PKIs.

## Zertifizierung

Um die Eignung von HSMs vergleichbar zu machen, hat das US-amerikanische NIST den Standard FIPS 140 entwickelt (publiziert am 11.01.1994). Derzeit gültig ist die Fassung FIPS 140-2 vom 25.05.2001.[5] Dieser Standard, der auch Anforderungen an Software-Krypto-Module umfasst, spezifiziert vier Sicherheitslevel, nach denen Produkte zertifiziert werden können. Dazu müssen sie unterschiedliche Anforderungen in elf Kategorien erfüllen.

Seit 1995 wurden vom NIST knapp 1.200 Produkte zertifiziert,[6] darunter allerdings nur 14 Produkte nach dem höchsten (Level 4) und etwas über 200 Produkte nach dem zweithöchsten Niveau (Level 3). Ein Zertifikat für Level 1 bestätigt wenig mehr als Selbstverständlichkeiten.

1 Zu Mindestlängen kryptographischer Verfahren siehe Lenstra/Verheul, Selecting Cryptographic Key Sizes, DuD 3/2000, S. 166.

2 Für Chipkarten siehe Wohlmacher/Fox, Hardwaresicherheit von Smartcards, DuD 5/1997, S. 260 ff.

3 Schon mehrmals gingen in größeren PKIs (Identrus, gematik) die Root-Schlüssel aufgrund eines Hardware-Defekts oder eines Angriffs-Fehlalarms im HSM verloren, ohne dass Backup-HSMs (Clones) existierten.

4 Zu Trusted Computing siehe die Schwerpunkthefte DuD 9/2004 und 9/2005.

5 Eine erweiterte Neufassung des Standards, FIPS 140-3, ist derzeit in Entwicklung.

6 Die vollständige Liste findet sich unter http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm