Quick Intro to HSM Attest

Author: Chris Cracknell, 26.5.2026

Overview

This Document is a short introduction to the basic concept of HSM Attest (the name is WIP) and the scenario that is intended to address. It is intended to be shown and discussed, rather than read in isolation.

It contains:

- Problem Scenario
  - Threats from within
  - Required Protection
  - Required Solution
- Proposed Solution
  - Confidential Computing and HSMs
  - Protection Strategy
  - Design Model
  - Components
- Delivery Channels
  - Software elements plus Consultancy
  - Managed Service
  - Partnerships

Implementation Details will be addressed in a separate document, but they must specifically cover Governance, High Availability and Operations support (tooling).

The current state of HSM Attest development is outside the scope of this document.

Problem Scenario

Threats from Within

The commonly used **boundary security** model where a hardened boundary to the system host is assumed to protect all the contents with an assumption of trust for other elements inside the boundary is **no longer seen as adequate**.

It is now considered necessary to deal with threats coming from within the hosting boundary (“zero Trust”, “Sovereignty” …).

This includes:

- Threats from the Hosting organisation (Cloud Provider)
- Attacks coming through the Cluster Hardware.
- Devops and software supply chain attacks.
- Internal actors, both malicious and inadvertent.

Copyright © Chris Cracknell 2026 2026

Required Protection

Classically seen as:

- Data at Rest
- Data in Motion
- Data in Use

It also requires:

- Software and Hardware in use

The first three are often referred to as ‘”3d encryption”. The third one is the domain of confidential computing and the software supply chain.

Required Solution:

The solution to the threat scenario above is **strictly defined control** of:

- **Who** can access **What** Data?
- **Where** do they access it?
- **Who** said they could?

Secondary to these main aims, a solution must:

- **Enforce** the above.
- Have well defined paths for **changing** any of the above.
- Support Monitoring of the **History** and C**urrent State** of the above.
- Provide appropriate **tooling** to support these functions with a sustainable amount of effort. (If security involves too much effort, it will be subverted.)

Proposed Solution

Confidential Computing and HSMs

Confidential Computing (CC) is based on trusting the CPU Hardware Manufacturers. It can do 2 things:

- Process Data inside protected CPU Modes with CPU Memory encrpyted everywhere outside the CPU Process (**Enclaves**).
- Identify Running software and hardware by **measurement** by the CPU. (**Attestation**)

Hardware Security Modules (HSMs) can do 1 thing:

- Prevent side channel attacks on secret values (usually keys). The only access to these values is through **known and limited APIs**.

Protection Strategy

So…

Copyright © Chris Cracknell 2026 2026

- Important secrets (“Roots of Trust”) never leave the HSM.
- Working secrets derived from the roots of Trust are released by the HSM to Applications running in known and trusted instances of CC Enclaves.
- Working secrets are, where possible, time-limited and use-restricted.
- Some new HSM Module is needed to gateway access to the HSM Functionality based on Attestation.

Design Model

The role of this new module is sometimes referred to as “TDCAS”, “CAS”, “Coordinator”, “ccHSM”, “Configuration and Attestation Service”, … The role can be fulfilled both with or without HSMs resulting in different levels of security.

The new module is implemented as a custom HSM firmware module sitting on top of the standard HSM firmware and toolsets.

It supports:

- **Gatewaying access** to working secrets based on **attestation** against a **white list** of known images.
- Manage access to and use of Roots of Trust to **derive working secrets**.
- Support and enforce **Governance** of the above.
- Support Secure Operations, e.g. Auditing, Scalability, SIEM, Revocation and Tooling.

Components

The deliverables of this system include:

- HSM Custom Firmware Module
- Documentation, examples and best practices
- Tooling for Governance management and Operations
- Support, Security Updates, Certifications

Delivery Channels

This section is WIP.

Software Components and Consultancy

Under this model, the developer of this solution provides software components under license together with T&E support and Documentation/Examples for a customer to introduce and operate this model themselves.

Managed Service

Under this model the developer of this solution cooperates with hosting providers to offer a preinstalled and preconfigured solution allowing a move to this model of security with minimum effort in return for an ongoing fee (e.g. Monthly).

Copyright © Chris Cracknell 2026 2026

This could also be packaged as part of a larger hosting agreement including such things as SIEM, load management, Secret provisioning, delegated Governance.

Partnerships

HSM Attest will work best with established partnerships aimed to provide near-turnkey solutions with:

- Hosting Providers
- HSM Technology providers
- Confidential Computing framework providers.
- Confidential Computing Consortium?

Appendix - Confidential Computing Consortium and HSMs

The Confidential Computing Consortium(CCC) is an alliance of interested parties dedicated to furthering the uptake of confidential computing. This is a summary of what I have been able to find about relevant efforts by the consortium in this area. I can envisage bringing elements of this approach under the CCC umbrella, but not to the point of making it easy for competitors to out-fund and displace me. I could support a role for the CCC here in defining architectures and interfaces to support interoperability in such a project.

In "CCC_outreach_whitepaper_updated_November_2022", the CCC says:

> *Security is only as strong as the layers below it, since security in any layer of the compute stack could potentially be circumvented by a breach at an underlying layer. This drives the need for security solutions at the lowest layers possible, down to the silicon components of the hardware. By providing security though the lowest layers of hardware, with a minimum of dependencies, it is possible to remove the operating system and device driver vendors, platform and peripheral vendors, and service providers and their admins, from the list of required trusted parties, thereby reducing exposure to potential compromise.*
>
> *With the goal of decreasing the reliance on proprietary software for confidential computing environments, the Confidential Computing Consortium has excluded from its scope TEEs that have only software roots of trust, and focused on hardware-based security guarantees for confidential computing environments.*

Applying the ideas from this quote, we see a properly used HSM as the lowest layer of hardware on which security can be built.

Going through the list of current CCC Projects, none of them address the issue of **how** to integrate the lowest level hardware root(s) of trust **AND ITS' USE** into the frameworks of Enclaves and protection of working cryptographic artefacts that is the focus of this solution. There are a number of commercial offerings (Edgeless, Scontain, Enclaive, Fortanix) that aim to provide a coordinator role similar to that described here but either

Copyright © Chris Cracknell 2026 2026

using no Hardware Security module or only using it for single encryption of a working coordinator database and backup/recovery scenarios. (Access to the HSM, if present, in these systems is protected by changing the static HSM logon during the Coordinator bootstrap process, leaving the backup/restore path via the HSM only protected by organisational measures.)

In the context of the above I see this proposed solution as the missing link between Confidential Computing attestation based authorisation and access to HSM-protected secret (e.g. keys) and authenticated (e.g. Identities) artefacts.

Under the Trusted Computing Group (TCG) ietf RATS architecture, the HSM in this model would be the relying party with governance tied to the HSM combining the roles of the reference value provider, verifier ownership and relying party owner. The reliant operation would be the providing of working cryptography by the relying party to the attested party.

Additionally, there are commercial offerings aiming to implement HSM functionality inside an enclave and then sell it as a "virtual HSM" or "Software HSM". These are not equivalent to real HSMs since they do not provide the same level of certified and proven resistance to attacks based on physical access to the device, nor do the frameworks used (e.g.Kata) fulfil the primary rule of keeping the trusted base small and stable. They are however very useful for development-time test deployments… Note that the "virtual HSM"s offered by e.g. Utimaco (cHSM) or Thales (virtual slots) ARE real HSMs since they exist inside a real, physical HSM device that is internally (and certifiably) partitioned into virtual devices that can be provided to customers separately – often as a managed service.

This development seeks to cooperate with - not replace – other mechanisms for workload build and deploy (e.g. Kata/CoCo, Gramine, etc.), nor is it linked to any specific CPU hardware manufacturers CC technology such as AMD SEV/SNOP, SGX, TDX, or nVidia.

IBM also seem to be trying to do something in this direction (Access to the HSMs from known TEEs only) with Secure Encrypted Linux and CryptoExpress HSMs on PCI cards in the same host, but it does not seem very mature, nor flexible enough for the real world of diverse Cloud Applications.

Appendix – Application Protection by Provisioning

The concept being applied to protect an application here assumes that an application is able to run and prove its' identity only after having been provided with the full set of security artefacts that enable this. Any software running without access to these artefacts should be easily identifiable as non-functional as well as being unable to provide any visibility into the data managed by the application. Artefacts will commonly include:

Copyright © Chris Cracknell 2026 2026

- Identity protection, e.g. use of TLS private key
- Data at Rest protection – e.g. entry-specific data encryption keys
- Data in motion protection – e.g. short-lifetime Secure Mesh mTLS keys, e.g. secure ENVOY
- Authenticated information on third parties, e.g. someone else's root CA certificate.
- Miscellaneous, e.g. Access Credentials for other entities in cluster, e.g. an SIEM report upload destination.

Copyright © Chris Cracknell 2026 2026