Input to the Commission on Enhancing National Cybersecurity

Submission date: September 6, 2016

Joint submission made by:

Benjamin Gittins
Chief Technical Officer
b.gittins@synaptic-labs.com
+356 9944 9390

Ronald Kelson
Chief Executive Officer
r.kelson@synaptic-labs.com
+356 9944 9390

Synaptic Laboratories Ltd.
www.synaptic-labs.com
13 Nadur Heights,
Nadur NDR-1390,
MALTA, Europe

Designers of safe and secure computing and communication architectures. Developers of general-purpose soft IP for FPGA devices, to increase security and performance, and to reduce circuit area.

Topic of this submission:

Significant Progress In The Design Of Universally Trustworthy and Dependable Identity and Access Management: Synaptic Labs' post quantum secure Identity Management and Cryptographic Key Management Solution

RFI topic areas this submission relates to:

- **Identity and Access Management**
- International Markets
- Cybersecurity Research and Development
- Critical Infrastructure Cybersecurity
- Internet of Things

Input submission contents:

(1) A 1 page executive summary for this comment, in the format requested by the RFI, which "identifies the topic addressed, the challenges, and the proposed solution, recommendation, and/or finding." Citations in the Executive Summary map back to the references listed at the end of the 15 page article attached to this submission. We have inserted headings that match these points in the executive summary.

(2) A 15 page article: B. Gittins. "Outline of a proposal responding to E.U. and U.S. calls for trustworthy global-scale IdM and CKM designs." Report 2011/029, Cryptology ePrint Archive, 2011.

That 15 page article is based on an earlier peer-reviewed article: "Overview of SLL's proposal in response to NIST's call for new global IdM-CKM designs without Public Keys", (dl.acm.org/citation.cfm?id=1852733)in Proceedings of the 6th Annual Workshop on Cyber Security and Information Intelligence Research (April 21-23, 2010), ACM, 2010. Also based on**:** B. Gittins and R. Kelson. Overview of SLL's proposal in response to NIST's call for new global IdM/CKM designs without PKC. Video. In IEEE Key Management Summit 2010 website, Lake Tahoe, Nevada on May 4-5, 2010., May 2010. IEEE. (https://www.youtube.com/watch?v=8Z3Prkc2eng)

(3) A 158 page document: B. Gittins, R. Kelson. "Feedback to NIST DRAFT Special Publication 800-130 (June 15, 2010). Submitted and received as feedback to the draft NIST "A Framework for Designing Cryptographic Key Management Systems" (SP 800-130) document which scope originally targeted global scalability. Our feedback was by far the largest received by NIST on this project. (The URL's contained in that 158 pages of feedback have been updated in 2016.)

(4) Brian Snow. We need assurance! In ACSAC '05: Proceedings of the 21st Annual Computer Security Applications Conference, pages 3–10, Washington, DC, USA, Dec. 2005. IEEE Computer Society. Full text published online on the ACASC website (https://www.acsac.org/2005/papers/Snow.pdf).

Significant Advances In The Design Of Universally Trustworthy and Dependable Identity and Access Management: Synaptic Labs' post quantum secure Identity Management (IdM) and Cryptographic Key Management (CKM) Solution

Executive Summary

RFI Topics: Identity and Access Management, International Markets, Cybersecurity Research and Development, Critical Infrastructure Cybersecurity, Internet of Things

Problem: In 2007, the E.U. FP6 SecurIST called [31] for trustworthy international identity management (IdM) that was user-centric. In 2009, the U.S. Department of Homeland Security (DHS) called [28] for trustworthy [70] global-scale IdM and the U.S. National Institute of Standards and Technology (NIST) called [13] for new cryptographic key management (CKM) designs.

Progress being made: In 2010, Synaptic Labs outlined the core architecture for (apparently) the first globally scalable, post quantum secure, symmetric key based platform for provisioning IdM, key distribution/agreement and inter-enterprise CKM services. Our proposal has received positive peer reviews by cryptographic experts. **As of 2016, we are NOT aware of any comparable proposal.**

Our proposal employs a **decentralised trust model that employs cryptographic "all-or-nothing transformations"**, that also employs compartmentalisation, redundancy and diversification simultaneously across service provider, software developer, hardware vendor, class of cryptographic primitive, and protocol axis. It **supports the collaborative management of international name spaces, support for store-and-forward messaging services (instant secure messaging, secure email, secure databases distributed across enterprises, …) using public identifiers** and supports **user-centric cross-cutting control mechanisms**. Our proposal is suitable for use with commercial off the shelf hardware and **is designed to wrap-around and protect the output of existing security deployments**. **It is designed to provide services that can be used to protect both low-assurance and high-assurance systems.** The platform addresses the U.S. Networking and Information Technology Research and Development Program (NITRD) call [56] to create a digital immune system (multi-layered protection, decentralised control, diversity, pattern recognition), the DHS call [28] for combating insider attacks and malware, achieving survivability and availability, and **NIST managers' call for a CKM design supporting billions of users without the use of public key technologies** [13]. Our system can credibly scale to billions of clients. This proposal has been designed as part of our Trustworthy Resilient Universal Secure Infrastructure Platform project [38].

Note: In support of our IdM+CKM project, S/Labs' has proposed an independently studied (Brian Snow - U.S. NSA, Miles Smid - U.S. NIST, …), Trustworthy resilient universal Secure Infrastructure Platform (TruSIP). TruSIP is designed to provide high-assurance security controls that prevent the public cloud provider and their hardware and software suppliers from maliciously or unintentionally learning or exposing the value of the cloud client's data, even though the data is being processed in the cloud. The TruSIP architecture provides an exceptionally high-assurance computing platform for storing and processing sensitive identity management and cryptographic key management operations. In short, TruSIP protects the commercial viability of IdM+CKM service providers from security breaches that may undermine end-user trust.

The recommendation: We respectfully propose that the Commission's detailed recommendations to strengthen cybersecurity should include the following points:

1. Perform an in-depth survey to identify, catalogue and evaluate the viability of all candidate next-generation globally scalable identity and access management solutions that are credibly trustworthy and dependable in multi-jurisdiction, multi-stakeholder Internet-scale environments that can be incrementally deployed to protect existing security systems while permitting the transition to higher levels of security assurance and improved capabilities. *(Note: we are unaware of any other proposal with this stated scope, so this recommendation will not require expensive surveying or the study of a lots of proposals!)*
2. Perform a high-level security aware Failure Mode and Effects Analysis of today's dominant identity and access management solutions (e.g. X.509, OpenID, …) that considers the impact of identified design and implementation security flaws wrt. the stakeholders in multi-jurisdiction, multi-stakeholder Internet-scale environments. *(Note: there is already an extensive body of publications on the known issues - so this recommendation will build on known work and does **not** require expensive, start from scratch, funding.)* Quantify the costs to the global community of those security flaws. Quantify the returns of developing a "fit for purpose" globally scalable high-assurance identity management and access platform. Fund the top 5 candidate next-generation identity and access management solutions that are credibly trustworthy and dependable, ensuring sufficient diversity between the research agendas / techniques. Ensure equal access and adequate support for (and team building around) innovative small-to-medium sized enterprises.

Sincerely, Benjamin Gittins and Ronald Kelson.

Outline of a proposal responding to E.U. and U.S. calls for trustworthy global-scale IdM and CKM designs

Benjamin Gittins
Synaptic Laboratories Limited
PO Box 5, Nadur, Gozo, NDR-1000, Malta, Europe
cto@pqs.io

ABSTRACT

In 2007, the E.U. FP6 SecurIST called [31] for trustworthy international identity management (**IdM**) that was user-centric. In 2009, the U.S. Department of Homeland Security (**DHS**) called [28] for trustworthy [70] global-scale IdM and the U.S. National Institute of Standards and Technology (**NIST**) called [13] for new cryptographic key management (**CKM**) designs. In this paper we outline the core architecture for (apparently) the first globally scalable, post quantum secure, symmetric key based *platform* for provisioning IdM, key distribution/agreement and inter-enterprise CKM services. Our proposal employs a decentralised trust model that exploits compartmentalisation, redundancy and diversification simultaneously across service provider, software developer, hardware vendor, class of cryptographic primitive, and protocol axis. It employs behavioural analysis techniques and supports the collaborative management of international name spaces, management of client transactions using public identifiers and supports user-centric cross-cutting control mechanisms. Our proposal is suitable for use with commercial off the shelf hardware and is designed to wrap-around and protect the output of existing security deployments. The platform addresses the U.S. Networking and Information Technology Research and Development Program (**NITRD**) call [56] to create a digital immune system (multi-layered protection, decentralised control, diversity, pattern recognition), the DHS call [28] for combating insider attacks and malware, achieving survivability and availability, and NIST managers' call for a CKM design supporting billions of users without the use of public key technologies [13]. This proposal has been designed as part of our Trustworthy Resilient Universal Secure Infrastructure Platform project [38].

Categories and Subject Descriptors

E.3 [**Data encryption**]; C.2.1 [**Computer-communications networks**]: Network architecture and design—*distributed networks, store and forward networks, network topology.*

1. INTRODUCTION

In 1976, two fundamentally different techniques were published that enabled authenticated private conversations between any two parties over a public network. The first unnamed technique, proposed by W. Diffie, M. Hellman and L. Lamport, employed a symmetric key distribution protocol [30] exploiting m key distribution nodes (aka key distribution centers) [10] that was secure against a collusion of up to m-1 participating key distribution nodes. We name this proposal **DHL-SKD**. The second technique, proposed by W. Diffie, M. Hellman and R. Merkle, employed public key encryption and required digital signatures [29]. Unfortunately, derivatives [21] of Shor's 1994 quantum algorithm [65] threaten the confidentiality and integrity of classical public key algorithms [55] based on the problem of factoring large numbers, the discrete logarithm problem, or elliptic curve schemes. Many identity based encryption schemes [18] are based on the same problems and so are also at risk. Identification of a trustworthy post quantum secure asymmetric key exchange remains an open hard problem [16], [60]. Independent of the quantum computing threat there are many other serious flaws [40], [46], [47] that have plagued the civilian global-scale PKI and fundamentally undermine its utility [48].

In 2009 the U.S. President's cyberspace policy review [70] near term action plan called for game-changing technologies that have the potential to enhance the security, reliability and trustworthiness of digital infrastructure and to "*build a cybersecurity-based identity management vision that addresses privacy and civil liberties interests*". The DHS responded to this call with their "Roadmap for Cybersecurity Research" [28] which outlines 11 current hard problems in information security, including global-scale IdM. NIST formally responded to the policy review by declaring that the development of new CKM capable of billions of users must be part of the U.S. national cybersecurity initiatives [13]. In both cases, current technologies are not considered adequate.

In this paper we show how to extend the 1976 symmetric key distribution scheme [30] to create a platform for a semi-online global-scale IdM, key distribution/agreement and inter-enterprise CKM that responds to the above calls. The fundamental principles of our design were well received by J. Patarin and L. Goubin in their 2008 review. The precursor to this paper was peer-reviewed and published by the 2010 CSIIRW-6 [35]. The applicability of our model in

network behavioural analysis (and remote malware detection) was published [50] by O. McCusker and others at the NATO IA&CDS [6]. Network behavioural extensions to our model were also published at ORNL CSIIRW-6 [49]. Our design was published at the 2010 IEEE Key Management Summit [36], [37].

2. STRUCTURE OF THE PAPER

This paper has 2 parts: the context around our proposal and the proposal itself.

Part 1: Context. In §3 we re-evaluate the original watershed decision that promoted public key distribution over symmetric techniques [30]. In §4 we survey the drivers motivating our work: In §4.1 we outline design requirements found in the 'Spirit of Laws' political theory treatise [27]. In §4.2 we summarise E.U. FP6 SecurIST's published position on user centricity. In §4.3 we recite the 11 current hard challenges to achieving trustworthiness as identified by the U.S. DHS. Finally in §4.4 we outline NIST's 2009 CKM drivers [13]. In §5 we observe that IdMS and CKMS are interdependent §5.1 and discuss trustworthiness framed in the context of global-scale IdM-CKM §5.2. The cryptographic foundations of our platform rely on symmetric techniques §6: In §6.1 we perform a short survey of early symmetric key distribution results. In §6.2 we quote W. Diffie, M. Hellman, and L. Lamport's description of their symmetric key distribution proposal and make observations on it in §6.3.

Part 2: Proposal. In §7 we describe the network topology of our IdM-CKM proposal: In §7.1 we rewrite the 1976 DHL-SKD [§6.2] to scale wrt. service providers and server nodes. In §7.2 we describe the network topology between a client and a store and forward node (**SFN**) and in §7.3 the network topology between a pair of SFN. In §7.4 we compare the security properties of homogenous and diversified realisations of this topology. In §7.5 we illustrate the network connectivity between two clients on the network. Finally in §7.6 we indicate various deployment strategies and walk through a pedagogical global-scale deployment scenario involving communication between regional and international clients. In §8 we make explicit our a priori vulnerability assumptions §8.1 and we survey the design's conformance with our drivers in §8.2. In §9 we sketch how to provision a variety of services. In §9.1 we describe how high-availiability communications is achieved between nodes of the cryptographic overlay network (IdM-CKM platform). In §9.2 we sketch how to assigning public identifiers to clients. In §9.3 we outline the context of inter-enterprise key management, describe how to scale secret sharing schemes wrt. to the number of shares, and then apply this within the context of our global-scale cryptographic overlay network (IdM-CKM platform). In §9.4 we describe how clients recall keys on demand. In §9.5 we describe the push based distribution of keys. By using (or rewriting) §9.4 and §9.5 we show how to perform: key agreement in §9.6, key agreement with crypto diversity in §9.7, provision authenticated assertion records in §9.8, provision secure file sharing in §9.9 and provision secure messaging in §9.10. In §9.11 we sketch how Exoskeletons^{TM}(protocol aware point-to-point tunnels) employing services provisioned by our proposed IdM-CKM platform can protect deployed infrastructure without requiring changes to software or hardware implementations of standards based security standards. In §10 we discuss (dis)trust and accountability before ending with a conclusion in §11.

Part 1: Context

3. RE-EVALUATING PKI DRIVERS

In 1976, W. Diffie and M. Hellman (**D&H**) conjectured [29, 30] that offline public key infrastructure (**PKI**) was required to achieve scalability and availability. Today online techniques are routinely applied to scale offline X.509 based PKI. This negation prompts us to reconsider their drivers.

Driver 1: Avoid secure key distribution channels. The use of self-signed certificates relaxed the original requirement for a trusted courier to deliver pair-wise unique symmetric keys down to the authenticated delivery of a public root certificate. The mass availability of CPU based smart cards is relatively new phenomena that was unavailable to D&H in 1976. These programmable smart card modules, when mounted on reels, can be efficiently used as a secure distribution channel for pair-wise unique symmetric keys. An enrolling party can visually fingerprint the smart card modules (using high-resolution laser imaging) and install custom applets before supplying them to service providers for key-injection operations. The tokens can then be optically inspected (for similarity and tampering) and electronically queried on return. The enrolling parties then act as authenticated distribution channels by supplying smart cards to end-users. Public key techniques, using merkle tree digital signature based algorithms [51, 25], can *also* be used to validate the authenticity of the smart cards.

Driver 2: Enable private conversations between any two parties regardless of whether they have ever communicated before. In 1976, D&H held that offline public key distribution was more bandwidth/latency efficient at key distribution than their m-1 secure symmetric key distribution proposal (**DHL-SKD**). Today, public key distribution with Online Certificate Status Protocol (**OCSP**) involves a network transaction. In 1976, ARPANET [26] and X.25 [22] clients were not designed to support concurrent network sessions. Today, concurrency is uniformly available which reduces the network transaction latency by a factor of m. Today, the difference in network latencies between public key distribution with OCSP checking and DHL-SKD is much less than anticipated in 1976. With the advent of CPU based smart cards, DHL-SKD network costs can be amortised by securely managing symmetric keys over multiple network sessions and by performing key derivation.

Driver 3: Enable scalable authentication of communication parties. In 1976, D&H expressed concern with node scalability and network availability issues and sought offline methods. Offline authentication operations in X.509 [41] require certificates and digital signature technologies. The responsibility for certificate/public key life-cycle management (discovery, validation) was shifted away from online servers. Users were left to find their own ad hoc solutions. Today, this heavy burden shifted to users is considered a serious hindrance to ubiquitous encryption [56]. These problems do not exist in symmetric systems. In key distribu-

tion and key translation architectures [10] pair-wise unique symmetric keys are employed to perform mutual authentication and key exchanges with low CPU overhead, either directly or through tickets. Advantageously, all reachable identities are discoverable in one location and the freshest key material is always supplied to users.

Driver 4: Remove the need for online servers. Summarising 3 results from P. Gutmann's paper [40]: 1) It is not possible to explicitly *validate* certificates in the X.509, instead offline certificate *revocation* lists are used. 2) The Online Certificate Status Protocol (**OCSP**) is a proxy service designed to improve the scalability of the certificate *revocation* lists. 3) The OCSP requires computationally expensive digital signatures for authenticated operations. OCSP also has vulnerabilities [46].

4. DRIVERS MOTIVATING OUR WORK

We propose that cryptographic systems should seek to address relevant requirements and calls as found below.

4.1 L'esprit des lois design requirements

The "Spirit of Laws" is a treatise on political theory first published anonymously by Charles de Secondat, Baron de Montesquieu in 1748 [27]. Montesquieu was the most frequently quoted authority on government and politics in colonial pre-revolutionary British America, cited more by the American founders than any source except for the Bible [45]. Montesquieu advocated constitutionalism, the separation of powers, checks and balances, the preservation of civil liberties, and the rule of law with the objective to reduce citizens fear of the political system. The important role true anonymity (as opposed to Government revocable pseudo-anonymity) has played historically in democracies should be considered in the design of, and laws concerning, IdM and CKM systems.

4.2 E.U. FP6 SecurIST on user centricity

Based on text and quotes from SecurIST publications [62, 31]: "*In the E.U., privacy is generally defined as a right of self-determination, namely, the right of individuals to determine for themselves when, how and to what extent information about them is communicated to others.*" SecurIST calls for international user-centric IdM in which the end users are empowered to determine his or her own security and dependability requirements and preferences. "*User-centric mechanisms are required to allow controlled release of personal, preference-related and location-based information, and to deliver assurances to owners about how personal information will be used by third parties.*" This marks a shift "*from Security and Dependability by 20th century central command and control approaches*", towards architectures that could lead to an "*open and trustworthy Information Society through empowerment*" of the individual with the purpose of protecting the central systems, the citizen and society interests (i.e. protecting the legitimate interests of all stake holders). "*Responsibility, authority and control have to move more towards the end user.*"

4.3 U.S. DHS on trustworthiness

The Nov. 2009 DHS "Roadmap for Cybersecurity Research" [28] outlines 11 current hard problems, eight of which "*were selected as the hardest and most critical challenges that must be addressed by information security research if trustworthy systems envisioned by the U.S. Government are to be built.*" The 8 challenges being: global-scale IdM, insider threats, availability of time-critical systems, building scalable secure systems, situational understanding and attack attribution, information provenance, privacy aware security and enterprise-level security metrics. The remaining 3 hard challenges being: system-evaluation life cycle, usable security and combating malware and botnets. Information processing systems striving for trustworthiness should address as many of these challenges they can from the onset of their design. The call for global-scale IdM was stressed again in June 2010 [33].

4.4 U.S. NIST's CKM drivers

At the 2009 NIST Cryptographic Key Management (**CKM**) Workshop [13], NIST managers identified that new CKM designs should be highly available, fault tolerant, secure against destructive attacks, scalable to billions of users, enable the ubiquitous take up of encryption, be secure against quantum computer attacks and use means other than public key technologies. Additionally they must support accountability, auditing, policy management, and be interoperable. NIST subsequently published their draft "Framework for Designing Cryptographic Key Management Systems" (SP 800-130) [14] in June 2010 resulting in comments received [17]. Over 90% of the points raised in NIST's summary of public feedback comments [24] presented at the second NIST CKM Workshop [3] were submitted by Synaptic Laboratories. Among other things, our feedback identified the need for the CKMS framework to be reconciled with other standards, special publications, guidance and forms such as SP 800-57 [12], the DHS Cybersecurity Roadmap [28], IEC 61508 Safety Integrity Levels [4], the US National Strategy for Trusted Identities in Cyberspace [7] and so on. At the end of the 2-day workshop, the results of the 2 breakout study groups correlated with our recommendations.

5. GLOBAL-SCALE IDM AND CKM

5.1 IdMS and CKMS are interdependent

The New Oxford American Dictionary defines a secret as "*something that is kept or meant to be kept unknown or unseen by others*". Cryptographic systems employ a) CKMS to manage keys and establish authenticated private channels and b) IdMS to identify and authenticate identities. Electronic IdMS use cryptography to authenticate identities and physical IdMS to identify people. We can't define an electronic-IdMS without defining a CKMS and vice versa. IdMS and CKMS are as interdependent as Yin and Yang. Global-scale cryptographic systems require collaboration between CKM, electronic IdM and physical IdM specialists.

5.2 Trustworthy global-scale IdM-CKM

To paraphrase Montesquieu, a global-scale IdM-CKM should be set up so no stake-holder need be afraid of another. This requires a conceptual shift away from the 'us vs. them' adversarial model inherited from the military origins of cryptography and towards an inclusive regulative system between peers. We assert that principles and requirements outlined in §1 and §4 can be embodied and realised in a unified trustworthy and cost-effective IdM-CKM system. A system

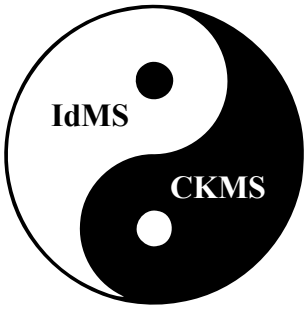


Figure 1: The Yin-Yang of IdMS and CKMS

that enhances democratic principles and protects the legitimate and diversified interests of all stake holders/users, even in a global context of competing nation-states. A global-scale IdM-CKM system provides the opportunity to realise user-centricity envisioned by the E.U. and others in a way not possible with today's uncoordinated silo'd (federated) based security solutions. In this paper we outline the core architecture of a global-scale platform that can be extended to comprehensively address international CKM, electronic-IDM and physical-IDM in a co-ordinated but distributed, decentralised and diversified manner. Our proposal exploits diversity in membership to improve security through a system of checks-and-balances and separation of powers in a way that ensures the system remains highly available and robust to all stake holders. Diversity used in this manner also encourages international competition in the open market place.

The IdM-CKM proposal as described in this paper protects clients from security compromises as a result of latent vulnerabilities or malware present in the software or hardware used by IdM-CKM service providers, or by the service provider's privileged technical or managerial staff. Our IdM-CKM proposal will achieve further improved confidentiality, integrity and availability properties for the IdM-CKM service providers when the IdM-CKM server software is hosted on our Trustworthy Resilient Universal Secure Infrastructure Platform proposal [38].

6. SYMMETRIC KEY DISTRIBUTION (SKD)

6.1 A short survey of early SKD results

In 1970 H. Feistel [32] described the use of symmetric keys to perform mutual authentication and this was applied to a network context by D. Branstad in 1973 [19] and 1975 [20]. In 1976 W. Diffie, M. Hellman and L. Lamport proposed the use of m key distribution nodes, where $m \geq 2$ [30]. We call this unnamed proposal **DHL-SKD**. S. Kent's 1976 thesis [42] gave the first description of a cryptographic system that employed two factor authentication, $m \geq 1$ symmetric key distribution networks, chaining of symmetric secrets between network sessions (stored on magnetic cards), and the authenticated encryption of data. Our proposal extends these results.

6.2 The DHL-SKD proposal

With reference to figure 6.2 we quote [30]: "*A small number m of the network's nodes will function as 'key distribution nodes'. Each user has m keys, one for communicating with each of these m nodes. These keys vary from user to user, so while each user must remember only m keys, each of the key distribution nodes remembers n, one for each user of the net. When users A and B wish to establish a secure connection they contact the m key distribution nodes and receive one randomly chosen key from each. These keys are sent in encrypted form using the keys which the users share with the respective nodes. Upon receiving these keys, the conversants each compute the exclusive or of the m keys received to obtain a single key which is then used to secure a private conversation. None of the nodes involved can violate this privacy individually. Only if all m nodes are compromised will the security of this connection fail.*" The paper goes on to say under the usual idealized security assumptions DHL-SKD is secure against a collusion of any combination of m-1 key distribution nodes. If one or more of the key distribution nodes is performing a denial of service attack the users select a subset n of the m key distribution nodes, in which case the protocol is secure against of any combination of (n-1) key distribution nodes.

6.3 Our observations on DHL-SKD

The 1976 DHL-SKD proposal did not specify if the m key distribution nodes are operated by 1 or m different service providers (that is, did we achieve m-Independence). It did not specify if the m key distribution nodes should run on identical platforms or exploit hardware and/or software diversity [23, 58]. The DHL-SKD proposal can be implemented using NIST FIPS 140 approved symmetric cryptographic primitives/modes of operation. NIST Advanced Encryption Standard [53] with 256-bit keys is widely considered post quantum secure for encryption [39]. NIST Secure Hash Algorithm [54] with 256-bit digest is widely considered PQS for message authentication. Key distribution nodes (**KDN**) and key translation centers (**KTC**) are both a type of secure store-and-forward node (**SFN**). For the purpose of two devices establishing a secure authenticated network connection, each of the m key distribution nodes in DHL-SKD can be trivially adapted to operate as key translation centre without invalidating the original security argument. An *idealised* key translation centre (a link-level secure key relay service) can be rewritten as: a network of unsecured processing elements enclosed and operating within the protection of a TEMPEST SDIP-27 [1] certified electromagnetic shielded enclosure performing key translation operations.

Part 2: Proposal

7. SLL'S IDM-CKM TOPOLOGY

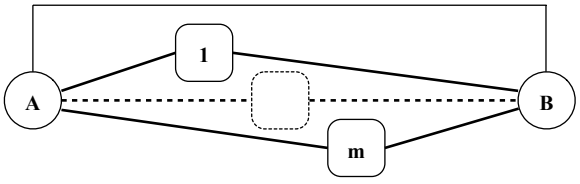


Figure 2: Topology of DHL-SKD with $m = 3$ key distribution nodes and 2 clients A and B

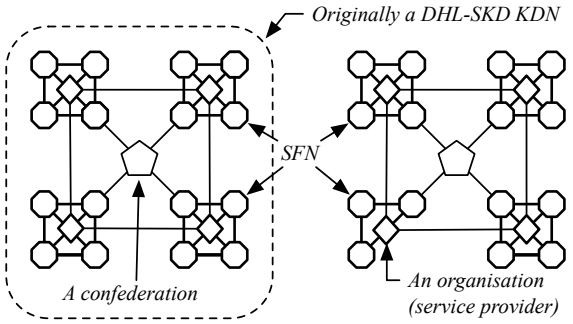


Figure 3: Topology of our scalable architecture

Our primary contribution in this paper is to outline a global-scale cryptographic overlay network, derivable from the DHL-SKD proposal. This overlay network is a platform suitable for delivering a wide range of inter-organisation, authenticated, policy driven, store-and-forward based cryptographic services such as secure messaging, key distribution, key agreement, key storage, and IdM operations. Our cryptographic overlay network has a semi-regular topology with certain well defined topological constraints that ensure consistent operational performance. Similar to the DHL-SKD model, most client transactions provisioned from a IdM-CKM deployment are distributed across a subset $x \geq 3$ service providers that the client is enrolled with, those service providers selected from x of the c confederations. In some cases a client may perform administrative operations, such as billing, with a single store and forward node.

7.1 Rewriting DHL-SKD to scale

With reference to figure 3 we consider each deployment of the DHL-SKD scheme to be an instance of a cryptographic overlay network (i.e. there may be multiple independent deployments of the DHL-SKD scheme). We substitute the m idealised key distribution nodes (**KDN**) of the DHL-SKD proposal with c confederations (illustrated as pentagons), where $c = m$. Each of the c confederations has at least 1 service provider (illustrated as a diamond). A service provider is assigned exclusively to one of the c confederations in this cryptographic overlay network instance. (A service provider may participate simultaneously in multiple cryptographic overlay network deployments.) Each service provider must have at least 1 store and forward node (**SFN**) (octagon). In practice, each confederation should have at least x SFN, where $x \geq 3$. Each of the x SFN shares at least one pairwise unique symmetric key ($\geq$ 256-bits in length) with the other x-1 SFN in its confederation. Each of the x SFN operates within the protection of an TEMPEST SDIP-27 electromagnetic shielded enclosure [1, 2]. Each of the x SFN communicate with the other x-1 SFN in a confederation using post quantum secure authenticated encrypted communications using the corresponding symmetric key. Efficient methods of m-1 post quantum secure bootstrapping of confederations and incrementally enrolling SFN are known.

In this way we have rewritten a SFN implemented as a network of unsecured processing elements enclosed within a single TEMPEST SDIP-27 certified electromagnetic shielded enclosure as a SFN implemented as a network of TEMPEST certified processing elements, where the TEMPEST certified processing elements communicate with each other using pair wise unique post quantum secure channels. Under idealised conditions, both versions of the SFN description are at least post quantum secure against outside adversaries.

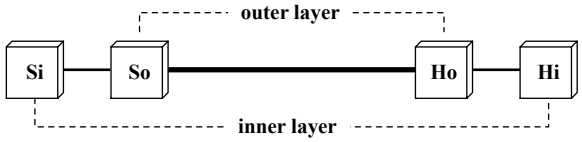


Figure 4: Topology between a client and a SFN

7.2 Topology between a client and a SFN

Each client is enrolled with c store and forward nodes (**SFN**), one from each of the c confederations. Figure 4 illustrates the topology between one client and one of the c SFN. In this higher assurance embodiment the client has two CPU based smart cards Si and So and the SFN has two network attached hardware security modules Ho and Hi. The smart card Si and Hi share a pairwise unique symmetric key ($\geq$ 256-bits in length). Likewise, the smart card So and Ho share a pairwise unique symmetric key ($\geq$ 256-bits in length). The hardware security module Hi executes the SFN server logic, and the smart card Si executes the SFN client logic. The hardware security module Ho and So execute point-to-point secure tunnel logic. The encrypted ciphertext generated in the inner communications layer between Si and Hi is re-encrypted by So and Ho resulting in an outer layer of security. Each enrolled client has a total of $2c$ pairwise unique symmetric keys. Key injection operations, performed by c service providers, can be executed while the smart cards processors are still on reels. Each SFN has 2 pairwise unique symmetric keys with every enrolled client. Ideally, Ho and Hi operate within the protection of a TEMPEST certified enclosure, and So and Si employ side-channel and fault injection protection mechanisms.

7.3 Topology between a pair of SFN

In preferred higher assurance embodiments, each of the x SFN shares at least two pairwise unique symmetric key ($\geq$ 256-bits in length) with the other x-1 SFN in its confederation. Communications between every pair of SFN involves an inner and outer layer of communications security, similar to the technique described in §7.2. Cross-cutting communication between SFN may also be required. These pair-wise unique keys would be exchanged online, on demand, as required.

7.4 Assigning agents to the abstract topology

The security properties of our proposal vary based on the agents participating.

Homogeneity: Let us consider a small **degenerate** homogenous deployment with $c = 4$ confederations, 1 service provider per confederation, and 1 store and forward node (**SFN**) per service provider. We assign all these resources to one division of one organisation. The $2c$ hardware security modules are provided by the same hardware security module vendor. The $2c$ modules are installed and run from the same

room. The operations of the inner and outer smart cards are all assigned to one smart card. All smart cards enrolled into the system are from the same smart card vendor. The protocol software for the hardware security modules and smart cards is implemented by one software developer. The deployment standardises entirely on NIST standards running in identical modes of operation (AES-CTR, SHA2-HMAC) for all cryptographic operations. In this way the hypothetical degenerate deployment strives to aggregate control and responsibility towards fewer agents, making the system more vulnerable to common mode of failures.

Diversity: Let us consider a similar sized deployment which preferentially exploits diversity and independence. It has $c = 4$ confederations, 1 service provider per confederation, and 1 store and forward node (**SFN**) per service provider. For simplicity of description, we select only two different smart card vendors, a first vendor for Si and a second vendor for So. For simplicity of description, all clients enrolled into the system will use a token from the same 2 vendors. We assign each of 4 confederations one of the following countries {Iceland, Russia, China, United States}. The 4 service providers are autonomous/independent organisations (wrt. other service providers) and each service provider is incorporated in the country assigned to their respective confederation. The 4 SFN are installed in the country of their respective confederation. Each of the 4 SFN are randomly assigned 2 different hardware security module vendors from the set of all available hardware security module vendors, where that random selection is refined to ensure each hardware security vendor is present within the deployment and also well represented (avoid heavy biases). Each service provider assigns one of their divisions to vetting/implementing their local copy of the software for the smart card So and hardware security module Ho for their instance of the outer layer. (In this way, the client smart card So receives c applets implementing the outer layer operations, a different applet for each SFN.) Each service provider assigns a different division of their organisation to vetting their local copy of the software used for their instance of the inner layer software for the smart card Si and hardware security module Hi. The inner layer employs NIST standards based cryptographic primitives and modes of operation. The outer layer employs alternate cryptographic primitives, such as non-US regional standards such as the GOST standards [5] and [71] for the Russian provider or other popular primitives. In this way the preferred hypothetical deployment strives for diversity, separation of powers (influence) in a redundant way with the aims of improving security (and at times improving availability).

7.5 Enrolled clients

Figure 5 illustrates 3 confederations of a IdM-CKM overlay network deployment. Label A illustrates a first client that is enrolled with three store and forward nodes (**SFN**) selected from the three confederations. Label B illustrates a second client that is enrolled with three store and forward nodes (**SFN**) selected from the same three confederations. Recall that every SFN shares a pair-wise unique symmetric key with every other SFN in a confederation, permitting a post quantum secure channel between every pair of SFN in that confederation. Client A and Client B can establish post quantum secure link-level encrypted paths across each confederation of the overlay network.

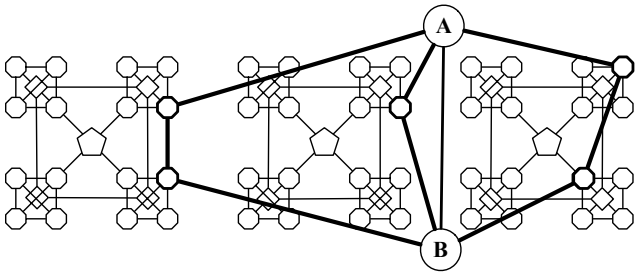


Figure 5: Paths between 2 clients over 3 confed.

7.6 Deployment strategies and example

A deployment of the IdM-CKM overlay network can organise it's confederations in a variety of ways including: as a global system with service providers grouped by aligned countries, as a regional system, as a national system with service providers grouped by different agencies/organisations, or even as an enterprise system. Deployments of the IdM-CKM overlay network can be layered, permitting a global IdM-CKM infrastructure for international communications, and several independent regional, national, industry controlled overlay network for localised traffic as described below.

Hypothetical global scale deployment. One possible configuration of a global scale IdM-CKM overlay network has $c = 6$ confederations with membership criteria as follows: { {UK, USA, CA, AU, NZ}, {EU member states (excluding the UK which is already assigned)}, {Arab States}, {Asian States}, {African States}, {all other remaining States} }. (Other configurations may be more desirable). Each confederation has 4 service providers, and each service provider has 4 SFN. The deployment employs diversification techniques as described in §7.4.

One of many possible international deployment layers participating in the global scale deployment. The five countries in the first confederation of the global scale deployment can reuse their existing investments and simultaneously participate in a second IdM-CKM overlay network with $c = 5$ confederations with membership criteria as follows: {UK, USA, CA, AU, NZ}. This configuration may be highly desirable for supporting their inter-government communications, and for commercially sensitive transactions between those countries.

Case use of an enrolled client. A client from Canada may be enrolled in their choice of service providers, one from each of the following countries: {UK, USA, CA, Iceland, Dubai, South Korea}. That Canadian client can use these 6 service providers to enrol (exchange keys) online with other service providers participating as clients in the global system. In this way the Canadian client can enrol with both a AU and a NZ service provider. The Canadian client exchanging key material with a New Zealand client may chose to set a default policy to use service providers from the 'Five-Eyes' countries {UK, USA, CA, AU, NZ} of which they are both a member, thereby minimising information leak-

age to other countries that may not normally share intelligence with this group of Nation-states. The same Canadian client, exchanging key materials with a Norwegian client would most likely use the service providers selected from the global-scale deployment as this guarantees availability of secure paths for exchanging key material. The ability for the client to chose their preferred service providers, and which service providers to use depending on the transaction, supports the U.S. NITRD's call for Tailored Trustworthy Spaces, and the E.U. call for user-centric empowerment.

Metadata. The regrettable leakage of connection information in our proposal is comparable with the information leakage already resulting from the international use of secure socket layer/transport layer security. SSL/TLS network communications over the Internet between clients in different countries leaks information to the countries that the Internet packet traverses (and any countries the respective certificate revocation query traverses), and through those countries to their respective allies they share intelligence with, and so on. e.g. SSL connections between two Japanese citizens located in their own country potentially leaks information to American intelligence organisations if they rely on the U.S. based Verisign as one of their root certificate authorities. Our proposal, as described above, can reduce this type of leakage.

8. ASSUMPTIONS AND CONFORMANCE

8.1 A priori vulnerability assumptions

In our proposal we work under the conservative assumption that latent unknown security vulnerabilities (malicious or otherwise) are *present* within the software and/or hardware of a cryptographic overlay network deployment. Our design objective is to limit one or more colluding agents to induce a service failure wrt. availability, confidentiality, integrity or maintainability of the system. Our goal is to ensure the reliability and safety of operations on behalf of all stake holders/users, even in the face of destructive attacks/natural disasters.

8.2 Conformance with our drivers

We will now comment on the conformance of our design with our drivers.

Achieve scalability of topology. Our IdM-CKM proposal permits scalability in the number of confederations, the number of different service providers within a confederation, and the number of store and forward nodes (**SFN**) within a service provider. Uniform performance characteristics across the system can be met by defining quality of service level requirements that must be met by service providers with regard to every client token they manage. This permits variation in the aggregate computing power of service providers and even confederations (on the provision that the number of confederations in a deployment is larger than the number of confederations each token is enrolled in).

Achieve scalability of provisioned services. This requires care in selecting what services to offer, and how to deliver them. Advantageously the constraints behind the semi-regular topology of the IdM-CKM permit certain assumptions and design optimisations to be made. For example, the number of SFN within a confederation required to forward messages in a client-to-client transaction is upper bound to 2. This property ensures certain security properties are present, and that wide area network latencies are upper bound even as the system scales. In our experience so far, essential cross-cutting services can be efficiently realised in an arbitrarily scalable manner. The mappings of Uniform Resource Identifiers [15] to the SFNs managing the clients' tokens associated with that identifier change infrequently and can be synchronised system-wide relatively easily. In contrast, a volatile database mapping of client tokens with their current Internet protocol address is comparatively burdensome and entirely unnecessary because online tokens can disclose their SFN directly between each other over unsecured network channels (zero overhead for the SFN), and validated at the end of the cryptographic handshake.

Redundancy. IdM-CKM platform employs client transaction redundancy across confederations, and client communication security redundancy through layering of independently keyed cryptographic protocols (preferably with different cryptographic primitives).

Diversity. Our IdM-CKM platform employs diversity [23, 58] at every point of redundancy in the design, including diversification across confederations, service providers, software and hardware vendors, class of cryptographic primitives, and in layers of communication security.

Separation of powers. Separation of powers is where the functions of a system are divided into separate and independent powers and areas of responsibility. Similar to the application of separation of powers within a country, this principle is applied within the context of a service provider in our design - typically only one organisation is assigned to each component. In the same way that we can observe redundancy with diversity when we look at two or more countries that both implement separation of powers, we also see redundancy and diversity at a system-wide deployment level of our platform. Where this property has limited benefit on a day-to-day basis for citizens in the context of the organisation of nation-states, in our case every client gains improved assurances on each transaction they perform. By applying diversity at every point of redundancy in our model, we limit the total amount of power/authority/control/influence a vendor or component has within a cryptographic overlay network deployment.

Checks and balances. As (almost) all client transactions are distributed redundantly across several autonomous service providers there is implicitly some form of checks and balances in place for those transactions. This property is made explicit through cross-cutting negotiation between participating service providers, and possibly one or more other representative authorities, to determine if the requested client transaction is authorised.

Multilayered protection: Our IdM-CKM design promotes layering of different secure communication protocols for both client-to-store and forward node, and client to client operations (see §9.11). In addition we propose services provisioned by the platform implement behavioural analysis techniques that employ human-in-the-loop techniques to mitigate misconduct by users and privileged administrators.

Decentralised control: The core of our IdM-CKM platform is decentralised organisation of (semi-)autonomous service providers that collaborate together to perform client transactions. In an international deployment, there is no system-wide single point of authority/control. Furthermore, the layering of communication security protocols ensures that the protocols employed within a deployed system are not under any one organisations control.

Useability: Our IdM-CKM platform employs smart cards to simplify client side key-management. The ability to globally co-ordinate the assignment of public Uniform Resource Identifiers with clients, in an online system that ensures freshness of key material and validation of identifiers permits vastly simplified key management over current X.509 type solutions.

Collaborative management of name spaces: A single global-scale deployment of our IdM-CKM platform can act as a clearing house for each nation's registers (assertion providers) for people (registry of births, deaths, and marriages), corporations (corporate registery) and top-level domain names (.com, .br, .fr, ...). Each client can consult with the service providers it has a relationship with from the c different confederations to form a consensus opinion on the validity of an assertion, without the client having to know (or have a relationship with) the internationally recognised authority for the different types of assertions. Additional assertion providers may be responsible for managing assertions made from a specific portion of a name space (IANA[1], au.IANA, com.au.IANA, compay.com.au.IANA), for assigning tokens to identity assertions, for creating and assigning roles and responsibilities within an organisation, and so on.

User centricity: Each person and organisation is a single logical entity, independent of the ability for a person to have multiple names and roles or an organisation to have multiple directors and authorised agents within it. In a global-scale IdM-CKM deployment with multiple assertion providers attesting various attributes regarding the existence and status (e.g. dead or alive) of an entity, and the mapping of a token to that entity (or authorised agent for that entity), it becomes possible to provide a cross-cutting user-centric view of the information managed by a global-scale IdM-CKM system. This can be done while simultaneously ensuring that every organisation managing a relationship with that entity has a 'per organisation unique identifier' (pseudonym).

Privacy enhancing technology: Services provisioned from our IdM-CKM platform can be privacy enhancing in the way that is envisioned by the EU STORK [67] and US NSTIC [69] initiatives. e.g. ensuring conditional release of information and the use of pseudonyms where desired/required. As we proceed to advance the design we will be looking for opportunities to minimise the amount of meta-data trivially leaked to service providers. We will be asking questions such as: can a service provider manage data in a user-centric cross-cutting way while masking these relationships from the service providers through the use of indirection/pseudonyms and further compartmentalisation of information.

[1]Internet Assigned Numbers Authority (IANA)

Achieve fault-tolerance: Redundancy can be employed within the compute and storage elements of each store and forward node to improve availability of services in the case of hardware faults. The use of distributed atomic transaction (begin, commit, rollback) based programming techniques by a service provider can be used to mask hardware failures of store and forward nodes without exposing the hardware failure to clients [11]. The presence of redundant service providers can be adapted to increase availability of the system in the case of the failure of a service provider. This may vary depending on the client transaction. With client-to-client key exchange, fault-tollerance is achieved through negotiation between the clients by allowing the number of participating service providers to be reduced in response to a unavailable/misbehaving service provider. With the remote storage of data across multiple service providers the encoding of client data using an all-or-nothing transformation [59] that is further encoded with parity and then distributed over multiple service providers permits the client to access their data even if one service provider is unavailable.

Achieve availability: The presence of fault-tolerance in a design leads to improved service availability. In our design, where a service provider has one or more store and forward nodes, it is possible to dynamically re-assign the store and forward nodes responsible for processing token requests in response to work-load within that service-provider. More uniform assignment of work load increases the responsiveness and availability of the service. The systematic application of quality-of-service techniques through the system can increase the availability of mission-critical services and permit price differentiation of services.

Combat insider attacks: Some high-availability systems achieve software [44] and/or hardware fault tolerance [11] using redundant implementations of the same function, running on independent circuits, potentially implemented by different teams where the output of the functions passes through a ballot monitor. The principle of redundancy and diversification has been adapted to create intrusion tolerant systems such as SITAR [72] where it is assumed an adversary can introduce a service failure in software executing within the system. Our IdM-CKM platform when implemented with diversity can mitigate a wide variety of insider attacks residing within the hardware or software of any component from compromising a client's security. In contrast to some intrusion tolerant systems which seek to detect and respond to intrusion events on an otherwise un-compromised deployment, our design explicitly assumes the intruder has a persistent presence inside the deployment and seeks to limit their ability to leverage that presence against a IdM-CKM client. This line of approach to combating insider attacks has been refined further in our Trustworthy Resilient Universal Secure Infrastructure Platform [38].

Survivability against destructive attacks: Physically destructive attacks resulting from natural disasters or deliberate malicious human acts can result in catastrophic service failure at a site. If an attack is experienced by a service provider at one site, continuity of services for that service provider is possible if redundant systems are available at one or more physically different sites. If one service provider experiences total catastrophic service failure, it is

possible for clients to negotiate relationships with other service providers and restore redundancy in any information stored in the IdM-CKM deployments by substitution operations performed by the failed service provider with a new service provider.

Situational awareness: Unlike X.509 PKI systems which are intentionally designed as predominantly offline systems, (semi-)online IdM-CKM systems are designed to actively participate in the delivery of many client transactions. Online systems can be trivially adapted to maintain state, and this state can be used to achieve situational awareness. For example, online IdM-CKM systems can *selectively* store information about the access patterns of a Client, or an IP address. In this way our IdM-CKM platform can support situational awareness and provide useful and appropriate services to clients.

For a *service provider-client* relationship to be trustworthy (e.g. doctor-patient, attorney-client, specialist-layman, computer-user, cloud-user, ...) the party entrusted with sensitive information must not exploit that information in a way that undermines the legitimate interests of that stake holder. Likewise, trustworthy information processing systems (human or automated) should be designed to minimise the amount of exploitable clear-text information they receive, while ensuring they leverage sensitive clear text information entrusted to them solely for the benefit of the client (virtue). Systems that (individually or systematically) violate this axiomatic principle undermine the community and cannot/will not be trusted by the same. e.g. A corporation of lawyers would irrevocably undermine their client's trust if they exposed sensitive personally information. Likewise, it follows that to realise a global-scale trustworthy IdM-CKM deployment, as is called for by the E.U. and U.S. Government, it must be virtuous and uphold this axiom.

Behavioural analysis and pattern recognition: Behavioural analysis techniques can be used to detect behaviours which may indicate possible security risks. To maintain user centricity, behavioural analysis should be performed for the benefit of each stake holder in the system. Each stake-holder may have their own unique behavioural analysis policies which the system should enforce. A range of default policies should also be made available to make these services immediately available. A human-in-the-loop process should be used to manage risk events detected by the system. A client should be able to delegate the human-in-the-loop to their outsourced managed security solution provider if they desire.

Combating malware and botnets: U.S. Sonalysts Inc is designing a distributed sensor system for the Internet (Occulex), which delivers policy-driven behavioral-based trust of hosts, derived from analysing aggregated network behaviors over multiple time scales for threat behaviors. Malware and botnets often exhibit distinctive behaviors that can be remotely detected by sensor networks. Behavioral analysis of sensor data, when done without identity, enables the sharing of actionable information without infringing upon the privacy of individuals or the community. On the remote detection of certain classes of malware, notification (via reverse look-up through the IdM-CKM platform) can then lead to remedial action to the relevant stake-holders. Separation of powers should be enforced, ensuring that identity information is not supplied from the IdM-CKM to the sensor network, and behavioural data exchanged between sensor nodes should not be supplied to the IdM-CKM deployment. See our co-authored paper for more information [49].

Post quantum secure: Our IdM-CKM platform relies entirely on symmetric cryptographic primitives which can select operational parameters (such as key length and digest length) that are widely considered to be both classically and post quantum secure. These primitives are available and widely trusted today.

9. SERVICE PROVISIONING

Our proposed IdM-CKM platform can be used to provision a wide range of cryptographic services. In this section we outline how communication is achieved with high availability within the IdM-CKM overlay network and then outline several cryptographic client services.

9.1 Overlay network communications

Most client transactions and all client-to-client transactions provisioned by a IdM-CKM deployment are distributed across a subset $x \geq 3$ service providers that the client is enrolled with, those service providers being selected from x of the c confederations. In some cases a client may perform administrative operations, such as billing, with a single service provider.

With reference to figure 6, in preferred high availability embodiments clients are enrolled with two store and forward nodes (**SFN**) owned by the same service provider paired in an {active, hot standby} buddy system. The hot standby node is illustrated as a light grey octagon with thick black border. The pairing is on a per-client basis. The SFN buddies may be physically located in two geographically separated sites (located in the east and west borders of a country or continent). The client has a pairwise unique symmetric key with each SFN (the enrolment with the hot standby SFN may be performed online with first use). If the active SFN becomes unavailable, the client continues the transaction on the hot standby SFN (which becomes the active SFN). The client may be directed by a service provider to exchange one of the SFN pairs with a different SFN managed by the same service provider in response to work-load balancing or hardware failure. The low-level details of how the buddy system should be implemented is outside the scope of this paper. In our model a client can establish an authenticated secure channel with every (active or hot standby) SFN it is enrolled with as described in §7.2. A client can request a first active SFN it is enrolled with to establish a secure connection with a second active SFN in the same confederation. The preferred secure connection between the first and second SFN is described in §7.3. In high-availability deployments the hot-standby SFN mirror the active SFN, as illustrated in figure 6. A reference to a SFN now implies the active SFN unless otherwise indicated.

The client can relay messages through the first SFN to the second SFN, and through the second SFN to any of the clients enrolled with the second SFN. In this case, the first SFN is responsible for identifying the client to the second

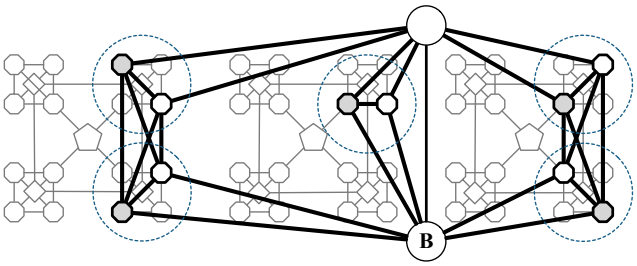


Figure 6: 2 clients, SFN buddy system, 3 confed.

SFN, and the second SFN is responsible for forwarding that identification information to other clients. Clients are responsible for correlating message parts from across the x confederations and checking that the identity assertions are the same from the x independent service providers.

In this paper, each SFN can reliably store data (with associated use policies), on behalf of an enrolled client in non-volatile memory. In high-availability systems, this data is replicated across the buddy system.

9.2 Assigning public identifiers to clients

After a client's token(s) are enrolled with a service provider it can be assigned one or more public identifiers (such as universal resource identifiers [15]) at low cost using an automated challenge-response process establishing the token's user has control of an e-mail account/website. This process can be reinforced through manual checking of physical credentials when higher levels of assurance are required. The process for the transfer and control of an identifier varies depending on the level of attestation previously provided and it's description is outside the scope of this paper. The redundancy in validating identifiers helps protect name spaces as assets of their respective owners/stake holders.

9.3 Inter-enterprise key management

Context. In the context of managing the private key of root certificate authorities, some commercial enterprise CKM products offer M of N split key controls where $(2 \leq M \leq\approx 5)$, $(2 \leq N \leq\approx 7)$ and $(M \leq N)$. The value of the secret is split into N shares, where any M combination of those N shares can reconstruct the original value of the secret. Split-key schemes are also known as secret sharing schemes [63]. The N person controls are often managed by people employed by one enterprise/organisation. All transactions performed with that split key requires the participation of M agents. Transactions include key exchanges, message signing, changing the membership of N and so on. The logistic effort to perform a transaction increases as the value of M increases. In practice, split-knowledge schemes do not arbitrarily scale wrt. the number of shares.

In the context of managing enterprise keys, sometimes M of N split authentication access controls are used. In this case the full value of the secret is entrusted to a hardware security module. The hardware security module is supplied a policy that requires M out of N parties to authorise a transaction on that secret. The stake-holders have to trust the hardware security module to consistently enforce that policy. Adding and removing authorised parties is easier in this case, as knowledge of the secret is not split across the N parties. To improve system availability and transaction workload capacity in conventional enterprise CKM deployments, two or more hardware security modules may mirror each others configuration. Unfortunately simple replication in this way increases the risk of a single hardware security module failure compromising that deployment. Furthermore, the attractiveness of attacking a hardware security module tends to increase along with the number of stake-holders that are dependent on it.

In an inter-enterprise key exchange environment, if the hardware security module is under sole control of one organisation, dependent organisations may have little to no assurances regarding their ability to control and audit transactions. If the full value of the key is known to that HSM, then a dependent organisation may have no assurances with regard their ability to control who can discover the value of the key. If one or more of those hardware security modules is attached to the Internet, the stake-holders require additional assurances that the split-authentication access controls cannot be subverted remotely. Unfortunately, it does not appear possible for any single vendor to demonstrate to their clients that this type of vulnerability is not present in their device[2]. One or more malicious software developers may covertly install vulnerabilities that could be exploited. Likewise, vendors of hardware security modules may be compelled to (covertly) install kill-switches or interception technologies in hardware security modules intended for local and/or foreign markets. Back-doors may be lurking in the components that hardware security vendors employ in their hardware security modules. Countries such as America are extremely concerned regarding the possibility of back-doors and kill switches under foreign control [8]. To quote the E.U. SecurIST [31]: "*The lack of trust is one of the main barriers for the establishment of a secure and dependable Information Society.*"

To summarise, it is not possible in practice to arbitrarily scale the number of parties that share partial knowledge of a secret, and schemes where knowledge of the value of the secret is not split across multiple parties are limited in the level of security assurances they can offer to stake-holders.

Scaling split key operations. We are proposing an inter-enterprise IdM-CKM scheme where the knowledge of a secret is split over a small manageable number of shares $3 \leq c \leq\approx 7$ and the authentication and access control is managed independently for each share (resulting in scalable split-authentication controls). Given it is unreasonable to require a vendor/organisation to demonstrate the complete absence of vulnerabilities in a product/process, we propose that each of the c shares is managed by a different service provider while ensuring those c shares are managed by several different hardware security module vendors, ideally c different vendors. See §7.4 and §7.6 for more information on our preferred deployment strategy.

[2]To quote B. Schneier: "*No one can guarantee 100% security*" ... "*There's no test possible that can prove the* ***absence*** *of flaws.*" ... "*A good cryptographic system strikes a balance between what is possible and what is acceptable.*" [61]

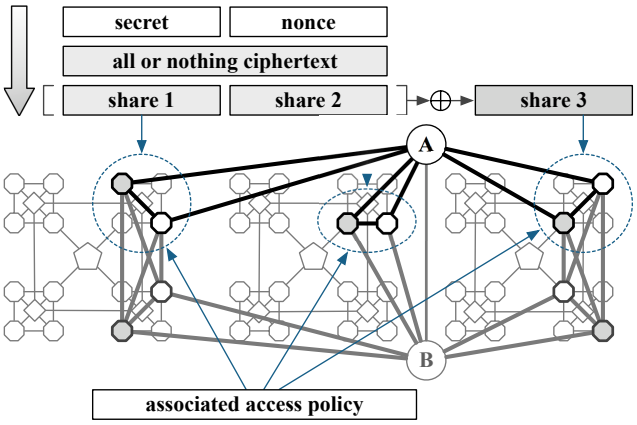


Figure 7: Inter-enterprise key storage

In this configuration clients rely on other service providers to independently manage some or all of the shares. If a secret is encoded in a N of M split key scheme where $M = c$ and $N = M - 1$ the client's secret remains *available* if one of the M service provider becomes unavailable and the clients secret remains *secure* against a collusion (or simultaneous compromise) of $N - 1$ service providers.

If a client wants to increase their security assurances they can participate as a service provider for their own transactions. This capability is available to each client, and between clients. i.e. two clients can be actively participating in the management of key shares between themselves (this requires their smart cards to be enrolled with each others hardware security modules).

Pedagogical example. Long-lived key material with associated access policies can be stored and enforced by a global-scale IdM-CKM deployment. In our model we require that client transactions provisioned from a IdM-CKM deployment are distributed across a subset $x \geq 3$ service providers (that the client is enrolled with) from x different confederations.

With reference to figure 7 client A receives the value of a 256-bit secret it wishes to store and an access use policy associated with that secret. In this pedagogical illustration client A is enrolled with $c = 3$ service providers from c confederations. (In preferred embodiments $4 \leq c \leq 7$). Client A wants to encode and distribute the value of the secret across the c confederations ensuring no service provider can discover the value of the secret, while ensuring the secret can be reconstructed if any 1 of the c service providers is unavailable (a $c - 1$ of c secret sharing scheme). In our illustration, one of the service providers will be assigned the parity of the encoded secret value distributed over the other $c - 1$ service providers. To ensure $\approx 2^{255}$ security against brute-force guessing attacks by one of the participating service providers, we require each share to be at least 256-bits in length. Client A allocates a contiguous array of 64 bytes (512 bits) in length in protected memory. Client A stores {the value of the 256-bit secret, the value of a 256-bit nonce} in the array. Client A encodes the 64-byte array using an unkeyed all-or-nothing transformation [59]. The 64-byte encoded message is partitioned into 2 shares of 256-bits in length ($share_1$, $share_2$). A 256-bit parity is created by calculating $share_3 = share_1 \oplus share_2$, where $\oplus$ is a binary exclusive OR operation operating on a 256-bit word. It is possible to reconstruct the original secret from any combination of 2 of the 3 shares in the usual way. Client A creates a meta-data record associate with each of the shares. The meta-data for each share stores information about the key, including the key type, key length, how many shares the key has been split into, which share of the split key is managed by this meta-data, and who is permitted to access the key. See [14] for a detailed description on the recommended fields required for key meta-data. Client A securely sends the first share and it's associated meta-data to it's enrolled service provider in the first confederation and receives back a public identifier. Client A repeats this process for the second and third shares to the second and third service providers in the second and third confederations respectively. Client A assembles the three public identifiers into a composite public identifier for that key material. The service providers are entrusted to enforce that policy with regard to their share of the key.

9.4 On demand recall of keys

Continuing from the previous paragraph and with reference to figure 7, client A has now encoded and stored a 256-bit secret across the c confederations. The secret is stored with associated meta-data instructing the store and forward nodes (**SFN**) how to manage that key material.

In a traditional 'enterprise CKM solution' key material is requested on-demand by one or more clients listed in the associated access policy. To achieve this operation in our design client A makes the composite public identifier for the key material known to client B. To access the key material, client B establishes secure authenticated connections with the c SFN it is enrolled with. Client B sends a request to it's c SFN to access the key material associated with the composite public identifier. Each of client B's SFN independently evaluate the request, establish a secure connection with the SFN assigned to managing the key material in their confederation, and forward the request. At this point the SFN responsible for storing the key material for Client A have all received an authenticated request from a SFN within their confederation. A cross-cutting query is performed by Client A's c SFN to establish that they have all received the same request (checks and balances). Having established a consensus to perform the request, each of client A's c SFN securely forward their respective share of the key material, along with associated policies to client B's c SFN. This material is then securely relayed to client B. Client B now has sufficient information to authenticate that the key material is from client A and to reconstruct the original value of the key. This process can be readily adapted to support a range of key management operations as authorised by the meta-data associated with that key material.

9.5 Push based distribution of keys

Continuing from the previous paragraph, the meta-data associated with client A's key material could instruct client A's store and forward nodes (**SFN**) to notify one or more target clients that key material is available. Client A's SFN are then responsible for identifying the SFN associate with

each of the target clients and informing them of the key's availability. Target clients are notified of the availability of key material immediately, or when they next log-in with their SFN. The policy for that key material may optionally instruct client A's SFN to delete the key material after a) all targets have successfully received the key material, b) an expiration time, c) or both.

9.6 Key agreement

Both client A and client B can use the push based distribution of keys to securely exchange nonce. Client A and client B would each receive the other's nonce via the IdM-CKM deployment, concatenate the 2 nonce in the same order and supply the output of the concatenation operation as input to a cryptographic hash function, using the resulting digest as shared key material.

9.7 Key agreement with crypto diversity

In a two-pass online key agreement protocol that exploits symmetric and asymmetric technologies, client A and client B use the push based key distribution function to securely exchange their respective public keys in the first pass. Client A and client B receive the authenticated public key of the other client. In the second pass client A and client B use the push based key distribution function to securely exchange the ciphertext resulting from their respective public key encryption of a nonce. Client A and B receive the ciphertext, decrypt the nonces, concatenate the 2 nonce in the same order and supply the output of the concatenation operation as input to a cryptographic hash function, using the resulting digest as shared key material. Advantageously, this method protects the ciphertext of public key operations using post quantum secure symmetric techniques and depending on the strength of the asymmetric algorithm chosen it may provide additional protection from a collusion of all participating service providers. Client A and Client B can choose to use different asymmetric algorithms to further increase crypto diversity or to satisfy their respective regional security standards.

9.8 Assertion records

Instead of key material, the client-to-client key distribution techniques described in §9.4 can be used to store public (or private) authenticated assertions such as SAML assertions [57], domain name server resource records [52], resource permissions, and so on.

9.9 Secure file systems

Instead of key material, the client-to-client key distribution techniques described in §9.4 can be used to store long-lived objects, the objects being either directories or files.

9.10 Secure messaging

Instead of key material, the client-to-client key distribution techniques described in §9.5 can be used to transmit shorlived objects, such as instant messages or e-mails.

9.11 Protecting deployed infrastructure

Secure tunnels are designed to wrap around and protect the (potentially insecure) output of programs without changing them. Protocol aware secure tunnels, which we call Exoskeletons^{TM}, would provide improved post quantum secure protection for the output of each network session generated by implementations of at-risk public key dependent security standards such as SSL/TLS, IPsec, RADIUS, and SSH. This capability could protect today's massive classically secure PKI deployments in a non-disruptive manner. Exoskeletons can be developed in a controlled environment without requiring existing standards to be adjusted. The technology can then be incrementally or rapidly deployed on a moments notice as desired/required.

10. (DIS)TRUST AND ACCOUNTABILITY

It is not appropriate to design global systems where insiders must be trusted. Today, approximately 86% of fraud happens by management level staff against their own organisation, in part **because they can** circumvent security mechanisms intended to prevent fraud [43]. Global systems that centralize trust in one 'trusted third party' (**TTP**) fuel the risk of cyber fraud and cyber war because they require users to absolutely trust the integrity of that trusted third party (or in the case of PKI, some 20+ Root certificate authorities [66], [40]).

Security systems should be designed so that no stake-holder is in fear of another. This can be done by redundantly distributing the execution of each provisioned service across m autonomously owned/managed service providers to mitigate insider fraud/attacks. Users do not need to buy into the altruism of any service provider. Instead users may choose to place their confidence in the mutual distrust and/or competitiveness between service providers. Such systems already employ "separation of powers" and can be adapted to employ cross-cutting "checks and balances" [27], provide redundant transaction audit logs to all users of the system, prevent liability shifting [9], and provide balanced security, accountability and privacy [68] for all stakeholders/users [31], [64].

11. CONCLUSION

Federal agencies and co-ordinating bodies in the U.S. and E.U. are calling in unison for trustworthy, resilient and dependable information and communications infrastructure that protects civil liberties and is user-centric. Calls for new trustworthy international/global-scale identity management and cryptographic key management designs have been made. This paper is a response to those calls. We have introduced (apparently) the first globally scalable, symmetric, IdM-CKM platform that is robust against a wide range of insider attacks. We have listed the ways our proposal addresses several drivers identified by U.S. and E.U. authorities. Our architecture can be derived from an existing proposal [30] which is already considered post quantum secure. Our proposal is practical, cost effective and can be implemented using commercial off-the-shelf hardware and implemented using NIST (or regional standards based) symmetric ciphers and hash functions which are already accepted to be post quantum secure. Our proposal can be used to provision a diverse range of client services by mapping traditionally specialised services (key distribution, key agreement, key management, name server, assertion server, file server, secure email, secure instant messaging) in a uniform way onto a authenticated store-and-forward network that exploits compartmentalisation, redundancy and diversification throughout the design. Our proposal can be used to protect exist-

ing at-risk public key cryptosystems. Our feedback [34] to the NIST draft framework for designing CKMS [14] appears to have also been positively received. Our (internationally distributed) decentralised trust model employs the democracy supporting Principles Of Laws and can be deployed in a manner that empowers all stake-holders and promotes goodwill and engenders trust between nations.

12. REFERENCES

[1] Compromising emanations laboratory test standard. SECAN Doctrine and Information Publication SDIP-27 Level A, NATO.

[2] Laboratory test standard for protected facility equipment. SECAN Doctrine and Information Publication SDIP-27 Level B, NATO.

[3] Cryptographic Key Management Workshop 2010. Project, National Institute of Standards and Technology, Sep. 2010.

[4] Functional safety of electrical/electronic/programmable electronic safety-related systems. IEC 61508, International Electrotechnical Commission, 2010.

[5] GOST 28147-89: Encryption, Decryption, and Message Authentication Code (MAC) Algorithms. In V. Dolmatov, editor, *RFC5830*, United States, Mar. 2010. RFC Editor.

[6] NATO Information Assurance and Cyber Defence Symposium in Turkey. Website, Apr. 2010.

[7] National Strategy for Trusted Identities in Cyberspace. Project, National Institute of Standards and Technology, January 2011.

[8] S. Adee. The Hunt for the Kill Switch. In *IEEE Spectrum*. IEEE, May 2008.

[9] R. J. Anderson. Liability and computer security: Nine principles. In *ESORICS '94: Proceedings of the Third European Symposium on Research in Computer Security*, volume 875 of *LNCS*, pages 231–245, London, UK, Nov. 1994. Springer-Verlag.

[10] ANSI. Financial institution key management (wholesale). Technical report, American National Standards Institute, 1985.

[11] A. Avizzienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. In *IEEE Transactions on dependable and secure computing*, volume 1, Jan. 2004.

[12] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendations for key management – part 1: General (revised). Special Publication 800-57 Part 1, National Institute of Standards and Technology, Mar. 2007.

[13] E. Barker, D. Branstad, S. Chokhani, and M. Smid. Cryptographic key management workshop summary (final). Interagency Report 7609, National Institute of Standards and Technology, June 2009.

[14] E. Barker, D. Branstad, S. Chokhani, and M. Smid. A Framework for Designing Cryptographic Key Management Systems. (Draft) Special Publication 800-130, National Institute of Standards and Technology, June 2010.

[15] T. Berners-Lee, R. Feilding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. In *RFC3986*, United States, Jan. 2005. RFC Editor.

[16] D. J. Bernstein, T. Lange, and P.-L. Cayrel. Post-quantum cryptography. Website, July 2009. Available at http://www.pqcrypto.org

[17] V. Bharadwaj, I. Clover, S. Eddy, B. Gittins, B. Nixon, S. Saha, and C.-R. Tsai. Comments Received on SP 800-130. Comments, National Institute of Standards and Technology, Aug. 2010.

[18] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, volume 2139 of *LNCS*, pages 213–229, London, UK, Aug. 2001. Springer-Verlag.

[19] D. Branstad. Security aspects of computer networks. In *AIAA Computer Network Systems Conference, Paper 73-427*, Huntsville, Alabama, Apr. 1973. AIAA.

[20] D. K. Branstad. Encryption protection in computer data communications. In *Proceedings Fourth Data Communications Symposium*, pages 7–9, Quebec City, Oct 1975. IEEE Computer Society.

[21] M. Brown. Classical cryptosystems in a quantum setting. Master of mathematics in combinatorics and optimisation, Waterloo, Ontario, Canada, Apr. 2004.

[22] CCITT. Recommendation X.25. Standard, International Telegraph and Telephone Consultative Committee, 1976.

[23] L. Chen and A. Avizienis. N-version programming : A fault-tollerance approach to reliability of software operation. In *FTCS-8*, pages 3–9. IEEE, 1978.

[24] S. Chokhani and M. Smid. A Summary of Public Comments on Draft Cryptographic Key Management Framework. In *NIST Key Management Workshop*. National Institute of Standards and Technology, Sep. 2010.

[25] C. Coronado. *Provably secure and practical signature schemes*. Doctoral thesis (elib.tu-darmstadt.de/diss/000642), Technische Universität Darmstadt, Nov. 2005.

[26] S. Crocker. Host software. In *RFC1*, United States, Sep. 1969. RFC Editor.

[27] B. d. M. de Secondat, Charles. *The Spirit of the Laws (Originally published anonymously in 1748)*. Crowder, Wark, and Payne, 1777.

[28] Department of Homeland Security. A Roadmap for Cybersecurity Research. Roadmap, DHS Science and Technology Directorate, Nov. 2009.

[29] W. Diffie and M. Hellman. New directions in cryptography. In *IEEE Transactions on Information Theory*, volume 22, issue: 6, pages 644– 654, Nov. 1976.

[30] W. Diffie and M. E. Hellman. Multiuser cryptographic techniques. In *AFIPS '76: Proceedings of the June 7-10, 1976, national computer conference and exposition*, pages 109–112, New York, NY, USA, June 1976. ACM.

[31] Z. Dooly, J. Clarke, W. Fitzgerald, W. Donnelly, M. Riguidel, and K. Howker. ICT Security and Dependability Research beyond 2010 - Final strategy. Deliverable 3.3, SecurIST EU-FP6-004547, Jan. 2007.

[32] H. Feistel. Cryptographic coding for data bank

privacy. Research Report RC2827, IBM T.J. Watson Res. Ctr, Yorktown Heights, N.Y., Mar. 1970.

[33] GAO. CYBERSECURITY: Key Challenges Need to Be Addressed to Improve Research and Development. Report to Congressional Requesters GAO-10-466, United States Government Accountability Office, June 2010.

[34] B. Gittins. Feedback to NIST DRAFT Special Publication 800-130. Comment, Synaptic Laboratories Limited, Jun 2010. www.tinyurl.com/jox78ju

[35] B. Gittins. Overview of SLL's proposal in response to NIST's call for new global IdM/CKM designs without public keys. In *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, CSIIRW '10, pages 60:1–60:4, New York, NY, USA, 2010. ACM.

[36] B. Gittins and R. Kelson. A survey and low-level comparison of network based symmetic key distribution architectures. Video. In *IEEE Key Management Summit 2010 website*, Lake Tahoe, Nevada on May 4-5, 2010., May 2010. IEEE.

[37] B. Gittins and R. Kelson. Overview of SLL's proposal in response to NIST's call for new global IdM/CKM designs without PKC. Video. In *IEEE Key Management Summit 2010 website*, Lake Tahoe, Nevada on May 4-5, 2010., May 2010. IEEE.

[38] B. Gittins and R. Kelson. Trustworthy Resilient Universal Secure Infrastructure Platform (TruSIP) Project. Website page., ICT Gozo Malta, Jan. 2011. http://www.ictgozomalta.eu/vision-and-projeets/project-trusip-ict-ics.html- -

[39] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th annual ACM symposium on Theory of Computing*, Annual ACM Symposium on Theory of Computing, pages 212–219. ACM, 1996.

[40] P. Gutmann. *Engineering Security*. (draft book), Dec. 2009. Available at http://www.cs.auckland.ac.nz/~pgut001/pubs/book.pdf

[41] ISO/IEC. "The Directory" series of standards. In *ISO/IEC 9594-x*.

[42] S. T. Kent. Encryption-based protection protocols for interactive user-computer communication. Master's thesis, Massachusetts Institute of Technology Cambridge Lab for Computer Science, May 1976.

[43] KPMG. Profile of a fraudster survey 2007. Forensic advisory, KPMG International, Apr. 2007.

[44] P. Kumar and A. P. Singh. Analyzing Software Fault-Tolerance in Real-time Systems using voting technique. Research report, University school of information technology, Sep. 2010.

[45] D. S. Lutz. The Relative Influence of European Writers on Late Eighteenth-Century American Political Thought. volume 78, No. 1 of *The American Political Science Review*, pages 189–197, March 1984.

[46] M. Marlinspike. Defeating OCSP With The Character '3'. Technical report, July 2009. Available at https://moxie.org/papers/ocsp-attack.pdf

[47] M. Marlinspike. Null Prefix Attacks Against SSL/TLS Certificates. Technical report, July 2009. Available at https://moxie.org/papers/null-prefix-attacks.pdf

[48] L. Martin, P. Gutmann, and B. Gittins. Is PKI really that bad? A series of postings, June 2010.

[49] O. McCusker, B. Gittins, J. Glanfield, S. Brunza, and S. Brooks. The Need to Consider Both Object Identity and Behavior in Establishing the Trustworthiness of Network Devices within a Smart Grid. In *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, CSIIRW '10, pages 53:1–53:4, New York, NY, USA, 2010. ACM.

[50] O. McCusker, J. Glanfield, S. Brunza, D. C. Gates, D. J. Hugh, and D. Paterson. Combining Trust and Behavioral Analysis to Detect Security Threats in Open Environments. In *NATO IACDS 2010, RTO-MP-IST-091*, April 2010.

[51] R. C. Merkle. *Secrecy, authentication, and public key systems*. PhD thesis, Stanford University, CA, USA, June 1979.

[52] P. Mockapetris. Domain names - implementation and specification. In *RFC1510*, United States, Nov. 1987. RFC Editor.

[53] NIST. The advanced encryption standard. Federal Information Processing Standard 197, National Institute of Standards and Technology, Nov. 2001.

[54] NIST. Secure hash standard. Federal Information Processing Standard 180-2, National Institute of Standards and Technology, Aug. 2002.

[55] R. A. Perlner and D. A. Cooper. Quantum resistant public key cryptography: a survey. In *IDtrust '09: Proceedings of the 8th Symposium on Identity and Trust on the Internet*, volume 373 of *IDtrust*, pages 85–93, New York, NY, USA, Apr. 2009. ACM.

[56] QinetiQ. National Cyber Leap Year Summit 2009 – Co-Chairs' Report. On behalf of the US NITRD Program, Sep. 2009.

[57] N. Ragouzis, J. Hughes, R. Philpott, E. Maler, P. Madsen, and T. Scavo. Security Assertion Markup Language (SAML) V2.0 Technical Overview. OASIS Committee Draft sstc-saml-tech-overview-2.0-cd-02, Mar. 2008.

[58] B. Randell and J. Xu. *Software Fault Tolerance*, volume 3 of *Trends in Software*. John Wiley & Sons Ltd, 1995.

[59] R. Rivest. All-or-nothing encryption and the package transform. In *Fast Software Encryption*, volume 1267 of *LNCS*, pages 210–218, Jan. 1997.

[60] A. Rostovtsev and A. Stolbunov. Public-key cryptosystem based on isogenies. Report 2006/145, Cryptology ePrint Archive, May 2006.

[61] B. Schneier. Why Cryptography Is Harder Than It Looks. Essay 037, Counterpane Systems, Mar. 1997.

[62] SecurIST Advisory Board. Recommendations for a Security and Dependability Research Framework: from Security and Dependability by Central Command and Control to Security and Dependability by Empowerment. Deliverable 3.0, SecurIST EU-FP6-004547, Jan. 2007.

[63] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, Nov. 1979.

[64] F. T. Sheldon, R. K. Abercrombie, and A. Mili. Methodology for evaluating security controls based on

key performance indicators and stakeholder mission. In *HICSS '09: Proceedings of the 42nd Hawaii International Conference on System Sciences*, pages 1–10, Washington, DC, USA, Jan. 2009. IEEE Computer Society.

[65] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, Oct. 1997.

[66] M. Stevens, A. Sotirov, J. Appelbaum, A. Lenstra, D. Molnar, D. A. Osvik, and B. M. M. de Weger. Short Chosen-Prefix Collisions for MD5 and the Creation of a Rogue CA Certificate. In *CRYPTO '09: Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*, volume 5677 of *LNCS*, pages 55–69, Berlin, Heidelberg, Aug. 2009. Springer-Verlag.

[67] STORK. Secure identity across borders linked. E.U. co-funded project INFSO-ICT-PSP-224993. Available at www.eid-stork.eu

[68] K. Sullivan. On the anonymity "versus" accountability debate. Whitepaper, Think-Trust EU-FP7-216890, June 2010.

[69] US DHS and others. DRAFT National Strategy for Trusted Identities in Cyberspace. Technical report, United States Department of Homeland Security, June 2010.

[70] USOWH. Cyberspace policy review: Assuring a trusted and resilient information and communications infrastructure (may 26, 2009). United States, Office of the White House, May 2009.

[71] E. V. Dolmatov. GOST R 34.11-94: Hash Function Algorithm. In *RFC5831*, United States, Mar. 2010. RFC Editor.

[72] F. Wang, F. Gong, R. Sargor, K. Goseva-popstojanova, K. Trivedi, and F. Jou. SITAR: A Scalable Intrusion-Tolerant Architecture for Distributed Services. In *DARPA Information Survivability Conference and Exposition*, volume 2, pages 153 – 155, April 2003.

The hyperlinks in this Sep 2016 revision have been updated.

Ronald Kelson
Chairperson and CEO
Tel: +356 7956 2164
Fax: +356 2156 2164
ceo@pqs.io

Benjamin Gittins
Chief Technical Officer
Tel: +356 9944 9390
Fax: +356 2156 2164
cto@pqs.io

Synaptic Laboratories Ltd.
All Correspondence to:
PO BOX 5, Nadur NDR-1000
MALTA, Europe
www.synaptic-labs.com

Monday, 16 August 2010

Feedback to NIST DRAFT Special Publication 800-130 (June 15, 2010)

Preamble

Synaptic Laboratories Limited (Synaptic Labs) has been researching the requirements for trustworthy **global-scale** cryptographic key management systems for several years. New global scale CKM and identity management systems are now identified as important objectives by NIST, DHS and others. Synaptic Labs has actively contributed input to NIST, DHS-NSTIC and NITRD projects particularly during 2009/10. To the best of our knowledge we are the only company publicly proposing a new global scale CKM and IdM model in the US Federal cyber security initiatives (such as the NITRD NCLY Summit) and at important events such as Oak Ridge National Laboratory CSIIRW and the IEEE Key Management Summit. More than that, we are working on overcoming the trust barriers that will enable our global CKM/IdM model to be delivered as a secure service, from the cloud, to SME's and SOHO's.

Since Synaptic Labs has a strong interest and several years experience in global scale CKM and IdM research and design, therefore we have allocated a great deal of time to an analysis of this NIST draft. It has been a genuine pleasure to read the draft text of "A Framework for Designing Cryptographic Key Management Systems". Congratulations to the drafting team and to all those who have contributed. This standard will materially advance the objectives of improved cyber security. It will become an essential aid making the work of cryptographic key management system designers simpler both in the USA and internationally.

However, we argue that the scope of the Framework must be expanded to go beyond today's enterprise CKM solutions to achieve the global-scale objectives as described by NIST. Furthermore, to ensure long term utility of this framework and to support other US Federal cyber security initiatives, the Framework must include guidance and requirements that are relevant to **global** CKM and IdM solutions that are now being sought as a a top priority by DHS and other agencies. The expanded scope should encompass cloud based CKM (and IdM) models. We argue for greater collaboration between this NIST project and cybersecurity initiatives in other federal agencies is essential.

Synaptic Labs now offers its next round of input in the following 158 page analysis of the draft SP 800-130. The length of our analysis is partly due to steps taken to make the information it contains as accessible as possible. We have provided a comprehensive index. Obvious 'white spaces' are left at the end of many sections to facilitate locating and reading of individual items of feedback. To remove the need to continually revert to the Framework draft we typically cut and pasted relevant sections of the draft into this document. To remove the need to continually revert to referenced sources we have inserted relevant quotes. There is necessarily a degree of repetition throughout the analysis to limit the need to revert to extensive cross referencing within the analysis itself when reading any one section.

As far as we are aware we have avoided suggesting proprietary techniques.

We begin our analysis with feedback and recommendations on issues that relate to multiple sections of the document. We then sequentially address specific portions of the draft. We offer alternate text in some places and propose additional text in others. We also make observations and ask questions that may lead to NIST revising or refining some other text.

All references to Draft SP 800-130 refer to the June 15, 2010 revision.

We appreciate this opportunity to provide feedback on the NIST "A Framework for Designing Cryptographic Key Management Systems" document and are available to provide further clarification on any part of our analysis.

Table of Contents

Part 1:
CKMS, IdMS and the US Federal Cybersecurity Research Agenda

1. Praise for the NIST CKM Project

CRYPTOGRAPHIC KEY MANAGEMENT PROJECT

Cryptographic Key Management (CKM) is a fundamental part of cryptographic technology and is considered one of the most difficult aspects associated with its use. Of particular concern are the scalability of the methods used to distribute keys and the usability of these methods. NIST has undertaken an effort to improve the ***overall key management strategies*** *used by the public and private sectors in order to enhance the usability of cryptographic technology, provide scalability across cryptographic technologies,* ***and support a global cryptographic key management infrastructure.***

http://csrc.nist.gov/groups/ST/key_mgmt/

This Cryptographic Key Management Workshop is the kickoff activity in a "leap-ahead" effort that we are undertaking as a part of the National Cybersecurity Initiative. The President recently announced the results of a cybersecurity policy review. Cybersecurity is a critical element in our national security posture. ***Our reliance on the internet is becoming nearly total.***

...

The role of key management in cybersecurity is critical.

...

We're going to accept very high risks in our research because we're going for very high payoffs. ***We're not going to accept high risks in the future Internet****, because we don't want the adversaries to have high payoffs.*

...

One requirement is to have ***scalable solutions in very large applications****. While we know how to handle key management reasonably effectively* ***for up to a million people, we need to go a couple of orders of magnitude beyond that in the relatively near future.***

William C. "Curt" Barker
NIST Computer Security Division Chief and NIST Cybersecurity Advisor
- NIST IR-7609

The NIST CKM Project goals are extremely important to the nation and the global community. The NIST target of a global cryptographic key management infrastructure could radically improve security beyond what is currently available when using the existing civilian public key infrastructure or commercial Enterprise-grade CKMS. A new global CKMS service that exploits the latest technological advances, such as cloud computing, can make simplified more effective and potentially even ubiquitous key management operations available, even to (and between) small and medium sized enterprises (SME) as well as Small Office / Home Office (SOHO) environments. As SME/SOHO's make up the majority of all business globally, the new CKM Framework and a simplified global CKM/IdM solution can deliver the broadest impacting benefits across a wide range of identified priority areas. We can block cyber crime while improving the ability of Internet users to identify themselves.

The deployment of a global CKMS infrastructure along the lines NIST is forging would help create a trustworthy and dependable interconnected and interdependent global cyber community for everybody, fuelling human development and technological r-evolution in an environment where access and responsibility can be balanced. The NIST CKM Project and its draft Framework are seen as positive and beneficial activities towards achieving these goals.

2. Praise for the Draft NIST SP 800-130

The objectives of the NIST SP 800-130 publication, as described in it's introduction, accurately reflect and address critical needs. As was identified at the IEEE Key Management Summit 2010, a gap exists in the ability of customers to evaluate the security of, or compare the functionality of, products that employ (or are predominantly focussed on) CKMS.

We acknowledge this draft document as a valuable resource to the security community. It has the potential to empower security engineers to build better security products.

> *"This Framework, does not mandate, requirements for the protection of U. S. government sensitive information. NIST Standards and Recommendations are referenced in this Framework as examples only. This Framework is intended to be general enough to encompass any reasonable, complete, and well designed CKMS."*
>
> *Section 1, page 9, Draft SP 800-130*

> *This document is not oriented to a particular CKMS or class of CKMS for an Enterprise or Enterprise Class (such as US Federal Government, Aerospace, Health Care, etc.). This Framework is intended to meet the needs of a wide variety of CKMS and Enterprises.*
>
> *Section 2.2, page 13, Draft SP 800-130*

Again, these objectives are commendable! We see it as particularly useful in the design of global-scale CKM where that one CKM system may need to satisfy different security standards in different countries, and even different security standards within the one country (NSA Suite A and NSA Suite B cryptography in one CKMS).

> *The CKMS design shall specify any human error prevention or failsafe features designed into the system.*
>
> *Section 3.2, page 16, Draft SP 800-130*

Design requirements such as the one above found throughout the NISP SP 800-130 help towards achieving trustworthy and dependable security systems as called for by the US Cyberspace Policy Review[1].

[1] USOWH. Cyberspace policy review: Assuring a trusted and resilient information and communications infrastructure (may 26, 2009). United States, Office of the White House. Available at http://www.whitehouse.gov/assets/documents/Cyberspace_Policy_Review_final.pdf

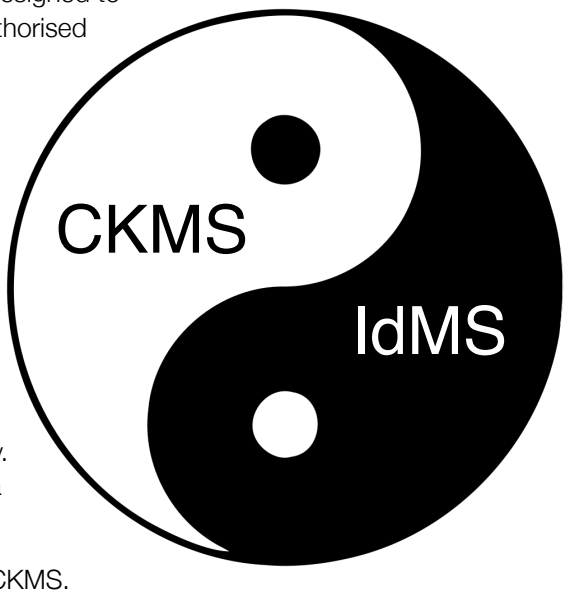
3. Relationship between CKMS and IdMS

3.1. CKMS-and-IdMS must be seen as Yin-and-Yang

> *… information systems should protect themselves and the information that they contain from **unauthorized** disclosure, modification and use.*
> *...*
> *Cryptography is often used to protect information from **unauthorized** disclosure, to detect modification, and to **authenticate** the **identities** of system users.*
> *…*
> ***Cryptographic techniques** use cryptographic keys that **are managed** and protected throughout their life cycles **by** the **CKMS**.*
>
> *Section 2.1, page 9, Draft SP 800-130*

> ***The role of key management in cybersecurity is critical.** We have cryptographic functions that are used for identification and authentication, both from the standpoint of protecting privacy, but **more importantly**, **for** integrity and **authentication mechanisms**.*
>
> *William C. "Curt" Barker (NIST), NIST IR-7609*

A secret is "*something that is kept or meant to be kept unknown or unseen by others*". (New Oxford American Dictionary). Any system or process that is designed to protect secrets must be able to correctly identify those who are authorised to know the secret and ensure an authenticated private channel between the secret holder and the authorised parties.



Cryptographic systems rely on a CKMS to manage keys. A cryptographic key management system must be able to identify and authenticate entities that are authorised to know the value of the key material. CKMS require identity management systems.

Electronic identity management systems employ cryptography to authenticate electronic identities. Electronic IdMS are cryptographic systems that require a CKMS.

CKMS and IdMS are the Yin-Yang of electronic information security. We can't define an Identity Management System without defining a Cryptographic Key Management System and vic-a-versa.

Any weakness in the IdMS system will result in a weakness in the CKMS. Any weakness in the CKMS system will result in a weakness in the IdMS.

(Yin-Yang image courtesy of Wikimedia commons)

IDMS have a CKMS. CKMS have an IDMS. It follows logically that any effort to define either a CKMS or IdMS without simultaneously defining the other component at an equivalent level of detail will be inferior to a cryptographic project that address both CKMS and IdMS in a holistic manner.

The discussion on identity management in the DRAFT NIST SP 800-130 can be found in it's entirety in two references, the first in the following sub-clause:

> *6.4.2 Owner Registration*
>
> *The initial registration of a security entity (i.e., individual (person), organization, device or process) and cryptographic key with bound metadata is a fundamental requirement of every CKMS. This requirement is difficult to fully automate while preserving security (i.e., protecting from the impersonation threat) and thus, it usually requires human interactions. There typically exists a registration process in a CKMS that associates each entity with an initial set of secret keys or public-private key pairs.*
>
> *The CKMS design **shall** specify the process for owner registration including the process for associating keys with owners.*
>
> *Section 6.4.2, page 35, Draft SP 800-130*

… and the second in one sentence on managing identity in the 5 page long section 6.2 on key metadata:

> *Owner Identifier: This field specifies the identifier (or identifiers) of the entity (or entities) that owns (or own) the key.*
>
> *Section 6.2, page 25, Draft SP 800-130*

A total of ~9 lines in the 88 page draft, surely not enough to provide a CKM designer with any guidance on the critical elements in CKM design that could be completely undermined if the IdMS introduces a weakness!

3.2. The complexity of identity management within a CKMS product

Managing electronic identities is complex and warrants much greater attention and detail in the draft standard. Let us consider the "owner identifier" field in the key metadata records:

Are we talking about owner as in:
John Smith, Birth XXYYZZZZ, Nationality XX, …

are we talking about owner as in a role:
A level X manager in an organisation with clearance level Y.

or are we talking about owner as in:
public key certificate or symmetric key secret managed on a smart card token?

or a combination and variation of all the above?
All managers and John Smith

How do we manage delegation to trusted agents and other complex legal relationships around roles and identities? How do we manage veto and *n*-out-of-*m* authorisation flow-processes?

Within a single CKMS, the CKMS' IdMS may have to rely on one or more different identity management protocols. The human key owner may be simultaneously enrolled in multiple IdMS e.g. the key owner may be enrolled in 3 OpenID compliant electronic credentials, have a certificate signed by a civilian X.509 compliant root certificate authority, an RFID ICAO MRTD (e-Passport), and an electronic National ID. Additionally the user may have a PKI based smart card token that is enrolled directly with the system. If the key-owner is identified as a "human identity",

how do we associate new electronic identities (due to expiration of old electronic credentials) associated with that same person/identity?

The draft Framework should set out the policies the CKM designer must apply to identification and authentication of a key owner, and how to manage authentication at different assurance levels throughout the CKMS.

Examples of different authentication levels include the IDABC (Interoperable Delivery of European e-government services to public Administrations, Businesses and Citizens) AAL (Authentication Assurance Levels), and the US OMB Memorandum M-04-04, E-Authentication Guidance for Federal agencies, December 16, 2003, available at: http://www.whitehouse.gov/omb/memoranda/fy04/m04-04.pdf .

3.3. Known attacks against civilian IdMS could undermine the security of NIST's next generation global-scale CKM project

As previously mentioned, a weakness in the IdMS component of a CKMS will undermine the security of the entire system. Let us illustrate this in the context of the OASIS Key Management Interoperability Protocol.

> *The OASIS KMIP TC works to define a single, comprehensive protocol for communication between encryption systems and a broad range of new and legacy enterprise applications, including email, databases, and storage devices. By removing redundant, incompatible key management processes, KMIP will provide better data security while at the same time reducing expenditures on multiple products.*
>
> *http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=kmip*

Subhash Sankuratripati (Chair of the KMIP TC) advised at IEEE KMS 2010 Summit in his presentation that KMIP relies on the Internet Standard Transport Layer Security (TLS) for identity management. TLS in turn relies on X.509 standards. There are many known problems with the civilian public key infrastructure. See the 3 publications in the footer [2] [3] for more information on the known problems. The key management experts assembled at the 2010 IEEE Key Management Summit were specifically asked if the status of PKI (x.509) was as bad as these experts were now pointing out. The consensus was 'yes' as summarised by Luther Martin in his blog on this very question[4] . As an example, let us consider the use of KMIP and TLS with the Civilian X.509 Public Key Infrastructure.

> *ABSTRACT: This paper introduces the compelled certificate creation attack, in which government agencies may compel a certificate authority to issue false SSL certificates that can be used by intelligence agencies to covertly intercept and hijack individuals' secure Web-based communications. Although we do not have direct evidence that this form of active surveillance is taking place in the wild,* ***we show how products already on the market are geared and marketed towards this kind of use—suggesting such attacks may occur in the future, if they are not already occurring.*** *Finally, we introduce a lightweight browser add-on that detects and thwarts such attacks.*
>
> *Soghoian, C., and Stamm, S. Certified Lies:*
> *Detecting and Defeating Government Interception Attacks Against SSL*
> ***April, 2010*** *http://files.cloudprivacy.net/ssl-mitm.pdf*

The above document talks about Government agency driven interception attacks against the SSL/TLS protocol. The (fair-use) illustrations on the next page shows one product already on the market explicitly designed to exploit vulnerabilities in the Civilian PKI X.509 infrastructure.

---

[2] Brooks, R. R., and Deng, J. "Lies and the Lying Liers that Tell Them - A fair and balanced look at TLS." CSIIRW-6 (April 2010)

[3] These videos are no longer online. Express your interested to receive these videos at info@synaptic-labs.com.

[4] https://www.voltage.com/technology/is-pki-really-that-bad/

Technical Details

Man-in-the-Middle Capabilities

Intercept any communication within Secure Socket Layer (SSL) or Transport Layer Security (TLS) sessions

All Packet Forensics targeting and policy capabilities can operate within the encrypted tunnel

Operational Configurations

In-line with hardware bypass / failsafe

Import any certificate / public key or generate your own for presentation

Availability

Available in firmware releases after August 31st, 2009 for all Packet Forensics platforms

Available under customization program

Contacts

Offices in Virginia and Arizona, USA

Headquarters

420 S Smith Rd

Tempe, AZ 85281

United States of America

Telephone & E-mail

Domestic US +1 (800) 807 6140

International +1 (757) 320 2002

salesteam@packetforensics.com

PACKET FORENSICS

HOW DOES IT WORK?

Deployment and Capabilities

Just as it sounds, engaging in a man-in-the-middle attack requires the interception device to be placed in-line between the parties to be intercepted at some point in the network. This could be at the subscribers' telecom operator or even on-premises, close to the subject. Packet Forensics' devices are designed to be inserted into and removed from busy networks without causing any noticeable interruption. Even the failure of a device due to power loss or other factors is mitigated by our hardware bypass fail-safe system. Once in place, devices have the capability to become a go-between for any TLS or SSL connections in addition to having access to all unprotected traffic. This allows you to conditionally intercept web, e-mail, VoIP and other traffic at-will, even while it remains protected inside an encrypted tunnel on the wire. All the same capabilities as other Packet Forensics products are still available, including the ability to extract pen/trap details only.

Technical Considerations: PKI

Using "man-in-the-middle" to intercept TLS or SSL is essentially an attack against the underlying Diffie-Hellman cryptographic key agreement protocol. To protect against such attacks, public key infrastructure ("PKI") is often used to authenticate one or more sides of the tunnel by exchanging certain keys in advance, usually out-of-band. This is meant to provide assurance that no one is acting as an intermediary. Secure web access (HTTP-S) is the best example of this, because when an unexpected key is encountered, a web browser can warn the subject and give them an opportunity to *accept* the key or *decline* the connection.

To use our product in this scenario, users have the ability to import a copy of any legitimate key they obtain (potentially by court order) or they can generate "look-alike" keys designed to give the subject a false sense of confidence in its authenticity.

Of course, this is only a concern for communications incorporating PKI. For most other protocols riding inside TLS or SSL tunnels—where no PKI is employed—interception happens seamlessly without any subscriber knowledge or involvement.

HOW CAN YOU USE IT?

Government Security

IP communications adoption dictates the need to examine encrypted traffic at-will, especially transiting government networks.

Investigations

Your investigative staff will likely collect its best evidence while users are lulled into a false sense of security afforded by web, e-mail or VoIP encryption.

Product Testing and Evaluation

All network products should be tested diligently for phone-home capabilities with encryption.

This is not a hypothetical risk limited to adversarial Governments around the world.

Another example of weakness in identity management relates to the arguments mounted by Gutmann and Brooks that it is impossible to differentiate certificates from placebo. **There are commercial Certificate Authorities that issue zero-verification certificates.** "*In late 2008 the founder of a low-cost commercial CA bought a certificate for MOZILLA.COM from another commercial CA with no questions asked in order to demonstrate just how easy it was to do*"[5].

SUMMARY: A new global CKMS must be accompanied by a new global-scale IDMS technology. In the meantime, the objectives of the NIST Framework (and any new Leap-Ahead CKMS that complies with it) will be continually undermined unless the Framework guides CKM designers on the critical IdM issues.

RECOMMENDATION. That, in recognition of the critical interdependency between CKM and IdM, the next revision of this draft CKMS standard to include drafting and insertion of a new comprehensive section on IdMS requirements for a CKMS. As a minimum it should set out the policies the CKM designer must apply to identification and authentication of a key owner, and how to manage authentication at different assurance levels throughout the CKMS.

RECOMMENDATION. That NIST expand the scope of this standard to encompass a comprehensive holistic global scale CKMS-IdMS security standard.

[5] Gutmann, P. "Engineering Security." (draft book), Dec. 2009. Available at http://www.cs.auckland.ac.nz/~pgut001/pubs/book.pdf

4. Relationships between NIST CKM project and other US Federal cybersecurity initiatives

In this section we outline possible synergies within concurrent US Cybersecurity initiatives that could be integrated with the NIST CKM project. This section is written in response to the NITRD 2010 Webcast's[6] call for feedback regarding US Federal cybersecurity project co-ordination as quoted below:

> *It's not a bad idea to talk [to government] as a partner in this co-ordination activity, pointing out areas where the Government might actually bring itself together and come out with a more joint, or at least something, a document, that co-ordinates those activities in the public view so you can understand … [this is a] completely legitimate business activity for you to do. It works both ways, we are trying really hard, and we need people to let us know where it doesn't appear from your end like any co-ordination has happened.*
>
> *See 1 hour 9 minutes into presentation, Patricia Muoio, ODNI*

4.1. Relationship between NIST CKM Project and the NIST Identity Management Systems Research & Development Project

> ***About the Identity Management Systems Program***
>
> *Electronic identities are routinely used to access logical and physical resources, and have become a ubiquitous part of our national infrastructure. …*
>
> *In conjunction with other federal agencies, academia, and industry partners, the NIST Identity Management Systems Program is pursuing the development of common models and metrics for identity management, critical standards, and interoperability of electronic identities. These efforts will improve the quality, usability, and consistency of identity management systems, protect privacy, and assure that U.S. interests are represented in the international arena.*
>
> *https://web.archive.org/web/20100528064008/http://www.itl.nist.gov/ITLPrograms/IDMS/external/index.html*

NIST has an IdMS project that could be brought into collaborate with the NIST CKM project. The NIST IdMS project[7] advertises existing established relationships in the identity management space which could play an important support role to the NIST CKM Project.

Of particular interest to the NIST CKM Project, the NIST IdMS project has conducted research into hybrid PKI and Symmetric **key management** solutions which overlaps nicely with the needs of the NIST CKM project and the requirements for compatibility, longevity and survivability of systems.

> ***Hybrid SKI/PKI Research***
>
> *Symmetric algorithms have advantages over asymmetric algorithms such as RSA: ... and it is believed that symmetric algorithms will be resistant to quantum cryptanalysis. …. PKI has the contrasting constraints implied by the comparison of symmetric and asymmetric algorithms.*
>
> ***…***
>
> ***Research Goals and Method***
>
> *The immediate goal of this research will be an exploration and analysis of alternative hybrid SKI/PKI **key management** architectures. ... Hybrid approaches will be described and analyzed against the criteria. Conclusions of the work will include recommendations for evolution of the FIPS 201 standard suite and other identity management programs.*
>
> *https://web.archive.org/web/20110311030732/http://www.itl.nist.gov/ITLPrograms/IDMS/external/IdMSRandD.html*

[6] NITRD Cybersecurity R&D Themes, 2010, https://web.archive.org/web/20140318071319/http://cybersecurity.nitrd.gov/events/nitrd-cybersecurity-rd-themes

[7] http://www.itl.nist.gov/ITLPrograms/IDMS/external/IdMSRandD.html

4.2. Relationship between NIST CKM Project and the DHS call for global-scale IdM

Global-scale Identity Management is a US Department of Homeland Security (DHS) cyber-security initiative. This call appears to have originated in the 2005 report[8] by the INFOSEC Research Council Hard Problem List. The associate director for NITRD in 2006 recognised the call for global-scale identity management[9]. It has since carried through to the DHS Roadmap for Cybersecurity Research in Nov 2009[10]. The U.S. Government Accountability Office (GAO) recently produced a document[11] titled: "*CYBERSECURITY: Key Challenges Need to Be Addressed to Improve Research and Development*". In that document they mention: "***global-scale identity management**, which **was identified by DHS as a top problem that needs to be addressed***".

> *Global-scale identity management concerns identifying and authenticating entities such as people, hardware devices, distributed sensors and actuators, and software applications when accessing critical information technology (IT) systems from anywhere. The term global-scale is intended to emphasize the pervasive nature of identities and implies the existence of identities in federated systems that may be beyond the control of any single organization. ... In this context, global-scale identity management encompasses **the establishment of identities, management of credentials, oversight and accountability, scalable revocation, establishment and enforcement of relevant policies, and resolution of potential conflicts.** ... It also necessarily involves the trustworthy binding of identities and credentials. It is much broader than just identifying known individuals. **It must scale to enormous numbers of users, computer systems, hardware platforms and components, computer programs and processes, and other entities.***
>
> *Global-scale identity management is aimed specifically at government and commercial organizations with diverse interorganizational relationships that today are hampered by the lack of trustworthy credentials for accessing shared resources.*
>
> *…*
>
> *Understanding the implications of quantum computing and quantum cryptography, and exploring the possibilities of global identity management without public-key cryptography or with quantum-resistant publickey cryptography.*
>
> *DHS, "A Roadmap for Cybersecurity Research"*

The latest call for global-scale identity management (2009) is framed within a wider context of eleven "Current Hard Problems", many of which hard problems need to be concurrently addressed to achieve a global-scale IdMS. Unfortunately, the global scale IdMS call did not directly address cryptographic key management as a related hard problem that had to be addressed. However, DHS did identify the need for useable key management as part of their current hard problem "Useable Security" which suggests a recognition of the need for new CKM solutions within the DHS.

The DHS 2009 roadmap and NIST IR-7609 report appear to share many common values with regard to trustworthy security and the necessary requirements to achieve security. This common overlap in value perception suggests a collaborative project between DHS and NIST to build a global scale hybrid IdM-CKM solution is possible.

---

[8] INFOSEC Research Council, "Hard Problem List", Nov 2005, https://www.nitrd.gov/cybersecurity/documents/IRC_Hard_Problem_List.pdf

[9] Sally E. Howe, "Remarks to the HCSS-Sponsored National Workshop on Beyond SCADA: Networked Embedded Control National Workshop on Beyond SCADA: Networked Embedded Control for Cyber Physical Systems for Cyber Physical Systems: Workshop Deliverables: Roadmap, Hard Problems, and Report", NITRD

[10] DHS, "A Roadmap for Cybersecurity Research", Nov. 2009. https://www.dhs.gov/sites/default/files/publications/CSD-DHS-Cybersecurity-Roadmap.pdf

[11] GAO. "CYBERSECURITY: Key Challenges Need to Be Addressed to Improve Research and Development", GAO-10-466, United States Government Accountability Office, June 2010. Available at http://www.gao.gov/products/GAO-10-466

RECOMMENDATION: Synaptic Labs feels there would be great value in harmonising the NIST CKM Project with both the NIST IdMS Project and the DHS Global-Scale Identity Management Project. The efforts already undertaken by the NIST Projects would complement and accelerate the DHS project.

4.3. Relationship between NIST CKM Project and the draft US National Strategy for Trusted Identities in Cyberspace

Ideally the US NSTIC project and NIST CKM project could potentially benefit greatly from each other if they were co-ordinated together. An example of an issue that could be explored is the apparently weaker cybersecurity values being suggested for the US NSTIC project, that may hinder and ultimately undermine the effectiveness of the NIST CKM Project and the CKM Framework.

Based on our reading of the US NSTIC draft strategy[12] the document appears to be more concerned with the interoperability of existing identity management systems, rather than addressing known architectural security weaknesses in these protocols. Unfortunately, a better co-ordinated identity management ecosystem built on insecure components remains inherently insecure and cannot achieve trustworthiness. However, the NSTIC interoperability goals supports “interoperability between first responders” using existing technologies as called for by NSA and others at the NIST CKM workshop[13].

The DHS November 2009 "A roadmap for Cybersecurity Research" outlined 11 hard problems, eight of which "**were selected as the hardest and most critical challenges that must be addressed by the INFOSEC research if trustworthy systems envisioned[14] by the U.S. Government are to be built.**"

The DHS Roadmap also states that "**experiences with failed or ineffective attempts in the past must be reflected in new directions**" **with regard to new global-scale identity management systems**.

The National Strategy for Trusted Identities in Cyberspace project does not seem to attempt to encompass these hard problems and critical trustworthiness issues identified by the DHS roadmap.

The NSTIC project also did not appear to consider the related cryptographic key management requirements that were identified in the NIST IR-7609.

Synaptic Labs has submitted three postings into the NSTIC public comment process addressing the apparent limitations and short comings in the NSTIC draft strategy in more detail here[15], here[16] and here[17].

---

[12] US DHS and others. “DRAFT National Strategy for Trusted Identities in Cyberspace.”, United States Department of Homeland Security, June 2010. Available at http://www.dhs.gov/xlibrary/assets/ns_tic.pdf .

[13] NIST IR-7609

[14] USOWH. Cyberspace policy review: Assuring a trusted and resilient information and communications infrastructure (may 26, 2009). United States, Office of the White House. http://www.whitehouse.gov/assets/documents/Cyberspace_Policy_Review_final.pdf

[15] B. Gittins (Synaptic Laboratories Limited), “We need to explore new distributed decentralised trust models that remove the current system-wide single point of trust failure”, NSTIC on IdeaScale, (no longer available online).

[16] O. McCusker (Sonalysts) and B. Gittins (Synaptic Laboratories Limited), “The Need to Consider Both Object Identity and Behavior in Establishing Trustworthiness”, NSTIC on IdeaScale, (no longer available online)

[17] B. Gittins (Synaptic Laboratories Limited), “NSTIC relies on cryptographic primitives known to be at risk of catastrophically breaking”, NSTIC on IdeaScale, (no longer available online)

4.4. Relationship between NIST CKM Project and the DHS definition of trustworthy systems

The word “trustworthy” occurs more than 100 times in the DHS roadmap[18] for cybersecurity research!

We agree with the eight current hard problems which "were selected as the hardest and most critical challenges that must be addressed by the INFOSEC research if trustworthy systems envisioned by the U.S. Government are to be built."

These 8 are as follows:

1. Global Scale Identity Management
2. **Insider Threats**
3. **Availability of Time-Critical Systems**
4. **Building Scalable Secure Systems**
5. **Situational Understanding and Attack Attribution**
6. **Information Provenance**
7. **Security with Privacy (Privacy aware security)**
8. Enterprise-Level security metrics.

We see the bolded themes arising and being addressed within the NIST SP 800-130 draft document at some level.

RECOMMENDATION: Harmonising the NIST CKM document to explicitly point to and cross reference the DHS Roadmap Cybersecurity Research document and its contents could help refine the clarity of the NIST SP 800-130 document and greatly assist CKMS developers in developing truly trustworthy CKMS as called for by the US Government (and by the global community).

RECOMMENDATION: In the same spirit, the DHS roadmap could be revised and harmonised to include requirements and current hard open problems as identified by the NIST CKM Project.

[18] DHS, “A Roadmap for Cybersecurity Research”, Nov. 2009. https://www.dhs.gov/sites/default/files/publications/CSD-DHS-Cybersecurity-Roadmap.pdf

4.5. Relationship between NIST CKM and US Networking and Information Technology Research and Development Program (NITRD)

It's not about security, its "Trustworthiness" of digital infrastructure.
Security, Reliability, Resiliance, Privacy, Useability.

...

Say "NO!" to Business as Usual. We don't want it, we can't take it anymore.

...

How can we:

- ***Enable risk-aware safe operations in compromised environments***
- *Minimize critical system risk while increasing adversaries' costs and exposure*
- ***Support informed trust decisions, necessitating flexible security strategies and allow for effective risk/benefit analysis and implementations.***

Dr. Jeanette Wing,
Assistant director for computer & information science and engineering (CISE), NSF (2010)

NITRD, like DHS, appear to have a full and working comprehensive of the scope of requirements necessary for creating trustworthy systems.

> ***We need to understand the requirements of the Federal government. We have to be working hand-in-hand with industries. What are our resources and what are our practical constraints?***
>
> *William C. "Curt" Barker - NIST IR-7609*

NITRD can help co-ordinate all relevant Federal parties to work with NIST's many experts to comprehensively answer the question regarding Federal requirements. Furthermore NITRD could help bring together the industry consumers of CKMS technologies. However, NITRD could be capable of going further than that.

Synaptic Labs asserts that we must think holistically, that we must be able to show how the next generation of global-scale IdMS and global-scale CKMS supports and advances all the different types of international cybersecurity initiatives.

NITRD appear to know what the technology problems are across the full networking and IT landscape, and the type of outstanding requirements / objectives that need to be researched[19]. Bringing that knowledge, those already identified needs, to the table to be addressed within the NIST CKM project could be highly beneficial to the global community. NITRD can bring together the various cybersecurity initiatives, group by group, to help identify the complex inter-relationships between different types of cybersecurity initiatives and global-scale IdMS-CKMS.

This process should seek to answer questions such as: In what way does the CKMS satisfy the US trustworthiness agenda? Why can our group's research agenda rely on your CKMS proposal to accurately enforce policies? Does the CKMS rely on trustworthy identity management infrastructure? In what way can the specialised behavioural security communities rely on, and enhance, a global-scale IdMS-CKMS deployment? In what way does the CKMS meet the needs of emerging malware protection technologies? How can the operating system community integrate this next-generation CKMS in a way that results in improved operating system security? What are the relationships between the hardware root of trust communities and the IdMs-CKMS project? How can or does IdMS-CKMS support the combat against spam?

This process should be iterative, and ensure that the requirements are flowing in BOTH directions, in and out.

---

[19] NITRD. Federal plan for advanced networking research and development. Report by the interagency task force on advanced networking, US Networking and Information Technology Research and Development Program, (Arlington, VA, USA), Sep. 2008. Available at http://www.nitrd.gov/PUBS/ITFAN-FINAL.pdf.

4.5.1. NITRD is promoting three cybersecurity themes.

NITRD is currently promoting three cybersecurity themes: tailored trustworthy spaces, moving target, and cyber economic incentives.

These three themes are directly relevant to both the NIST (global-scale) CKM project and the DHS global-scale Identity Management project.

In the next 3 sections we will outline some of the ways in which Synaptic Labs perceives possible synergies.

4.5.2. Supporting NITRD tailored trustworthy spaces theme

According to Dr. Jeanette Wing in the NITRD 2010 Cybersecurity R&D Themes webcast[20], based on federal consensus the tailored trustworthy spaces theme is considered the most important of the three NITRD themes. Furthermore, according to Dr. Wing in the same webcast, "Tailored Trustworthy Spaces supports context specific trust decisions". Dr. Carl Landwehr expands on this by stating:

> *The vision is of a flexible, distributed trust environment that can support functional, policy and trustworthiness requirements arising from a wide spectrum of activities (banking, e-commerce, schooling, ...) in the face of an evolving range of threats…*
>
> *Users can negotiate with others to create new environments with mutually agreed characteristics and lifetimes. …*
>
> *Dr. Carl Landwehr, Program Director, Trustworthy computing program, NSF*

To our minds, these statement relate directly to "User-centric design". In user centric design the system presents the user with a tailored interface that allows that user cross-cutting visibility and control over information and operations of which they are a stake-holder in.

Dr. Landwehr goes on to say:

> *Tailored Trustworthy Spaces is a New Paradigm. Users can select different environments for different activities (online banking, commerce, healthcare, personal communications) providing operating capabilities across many dimensions, including confidentiality, **anonymity**, data and system integrity, provenance, availability and performance.*

Clearly a global-scale CKMS **must** support different environments for different communities of interest that simultaneously co-exist within the one logical ecosystem. The broad-scope of tailored trustworthy spaces helps clarify the full-range of capabilities that should be present within a global-scale CKMS, ranging from "complete identification of every actor and device" through pseudo-anonymity down to full anonymity.

A tighter harmonisation of the NIST CKM project with the NITRD vision of tailored trustworthy spaces would increase the attractiveness and utility of the NIST project with regard to all potential stake-holders.

---

[20] NITRD Cybersecurity R&D Themes, 2010, https://web.archive.org/web/20140318071319/http://cybersecurity.nitrd.gov/events/nitrd-cybersecurity-rd-themes

4.5.3. Supporting NITRD moving target theme

According to Dr. Jeanette Wing in the NITRD 2010 webcast, the moving target theme is about providing resilience through agility.

Dr Patricia Muoio, Science and technology lead for cyber, Office of the Director of National Intelligence, expands on the definition of agility. It means to have the ability to control change across multiple system dimensions to:

- increase uncertainty and apparent complexity for attackers,
- reduce their windows of opportunity, and
- increase their costs in time and effort.
- increase resiliency and fault tolerance within a system.

Examples include address space randomisation, instruction set randomisation, and network port randomisation to achieve diversity and difference to keep an adversary guessing.

IdMS and CKMS are core components in almost every cybersecurity initiative. If we are going to create agile information processing environments, the IdMS and CKMS should at the very least support agility, if not employ agility from the onset internally.

4.5.4. Supporting NITRD cyber economic incentives theme

According to Dr. Jeanette Wing in the NITRD 2010 webcast, the cyber economics theme is about providing incentives to good security.

Dr Douglas Maughan, Program Manager, Cyber Security R&D, Science & Technology Directorate, Department of Homeland Security (DHS S&T), goes on to say that the consensus agreement today is: "Crime pays on the internet". He asks, in the future can we create an environment where being a good guy pays, and a bad guy doesn't?

While a global-scale IdMS-CKMS might be able to provide a direct support role to this important NITRD theme in other contexts, it is probably more important in the short term that best-practices with regard to cyber economic practice are employed during the design of the IdMS-CKMS to mitigate insider attacks and improve the ground-level trustworthiness of the deployment.

The question is, can we ensure “crime doesn’t pay” within a global scale IdMS-CKMS ecosystem deployment? Notions such as user-centric design, holding all parties equally accountable, and design for protecting the legitimate interests of all stake-holders are all proactive design strategies to begin addressing the cyber economic incentives. Are these sufficient? Can we expand on these further?

To our mind, If it is in fact possible to ensure “crime doesn’t pay” within a global scale IdMS-CKMS deployment, then it may also be possible to find general principles that might work in other environments!

Part 2:
The need for better CKMS domain models and nomenclature

In this section we will highlight various difficulties the NIST SP 800-130 draft has in defining the requirements for a CKMS design. It appears the problem stems from the lack of a sufficiently expressive CKMS domain model / taxonomy / nomenclature. To the best of our knowledge, this is a problem facing the entire open community.

In this section we will begin by pulling together text describing the CKMS Primary, Secondary, and Backup Facilities and their security controls as found scattered throughout various locations in the NIST SP 800-130 document. We create diagrams illustrating the components within, and interrelationships between, the primary, secondary and backup facilities.

Having drawn together this information, we then explore how other portions of the NIST SP 800-130 define a CKMS design and it's requirements. We highlight how the text describing the CKMS Primary, Secondary and Backup facilities only addresses an important sub-set of the overall CKMS vision as found in the NIST SP 800-130 documentation.

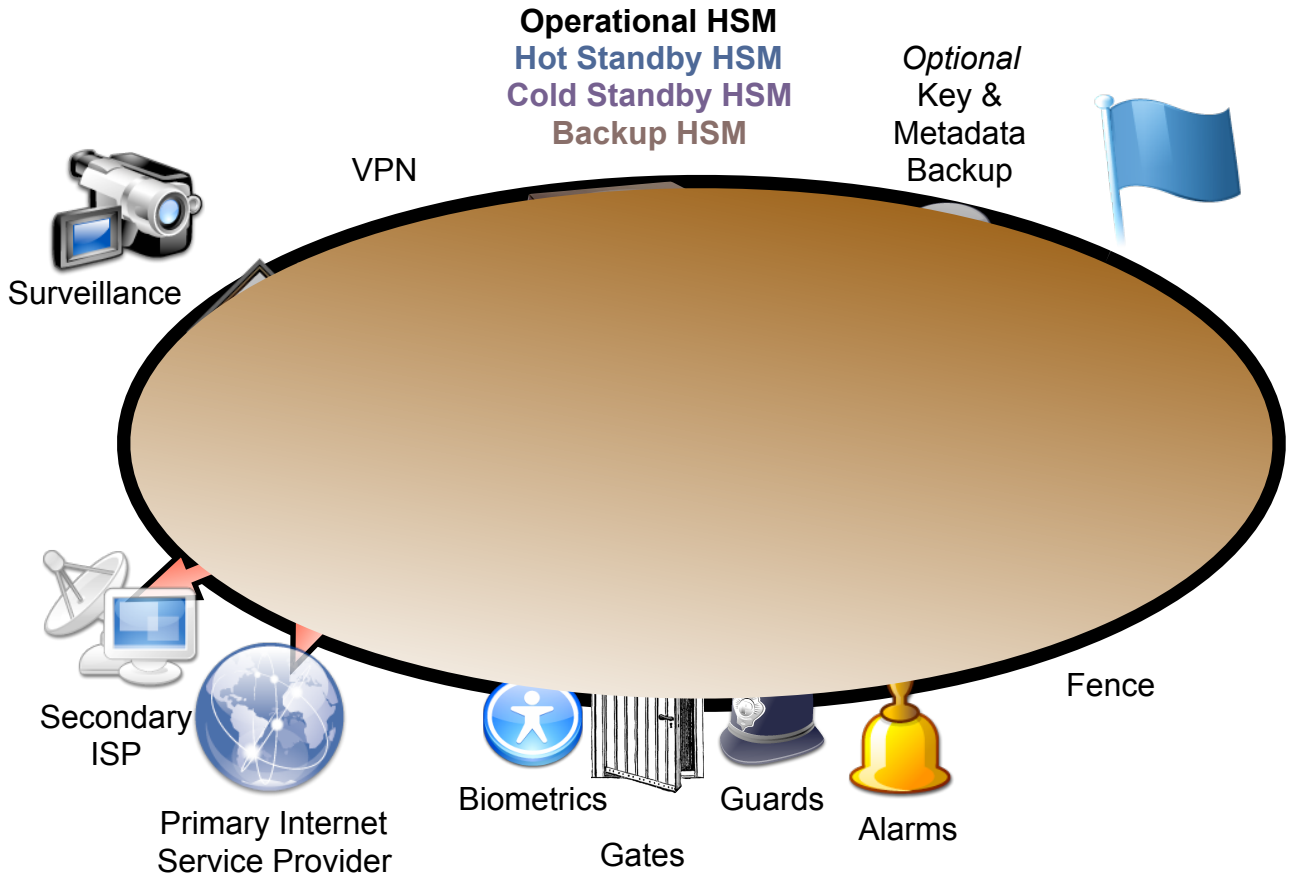
Continuing from this point, we then outline how the document itself has difficulty with the industry standard nomenclature for defining/describing CKMS. We highlight the recursive nature of the current CKMS definition (a global-scale CKMS uses cryptographic components that have embedded CKMS within them), and how the requirements for protecting "a CKMS" cannot apply to every device that employs a CKMS.

We will then move to illustrate one type of (public domain) CKMS system that should probably be describable with any new terminology.

5. A Visual representation of a CKMS as described in NIST SP 800-130

5.1. Visual illustration of a CKMS Primary Facility

The picture below illustrates most of the requirements identified for a primary facility as described in the quotations below from the draft NIST SP 800-130 document. [The iconic graphics were sourced from here[21]]



QUOTES from SP 800-130:

> **6.8.4 Network Security Controls and Compromise Recovery:** The scope of network security controls includes boundary devices, such as a firewall, a VPN, an intrusion detection system, and an intrusion protection system.
>
> **8.1 Physical Security Controls:** … One or more of the following mechanisms should be chosen to physically protect a CKMS, depending on the security criticality of the components. All components (regardless of type) listed above should require physical security. The following are examples of physical security mechanisms. Some of the mechanisms listed below are detection mechanisms, which should be augmented with appropriate prevention mechanisms.
>
> a) Fences ,
> b) Gates and doors,
> c) Guards,
> d) Locks (keyed or combination),
> e) Card readers,
> f) Biometric devices,
> g) Alarm systems
> h) Surveillance camera,
> i) Entry and exit log

[21] http://commons.wikimedia.org/wiki/Crystal_Clear (Lesser GNU License)
Computers licensed from iStock Photography. The nCipher HSM was sourced from the Wikimedia commons website.

8.1 Physical Security Controls: … CKMS components will likely be located at multiple facilities:

a) Primary Facility
i. Operational Components, ii. Hot Standby Components, iii. Warm Standby Components
iv. Cold Standby Components , v. Backup Components

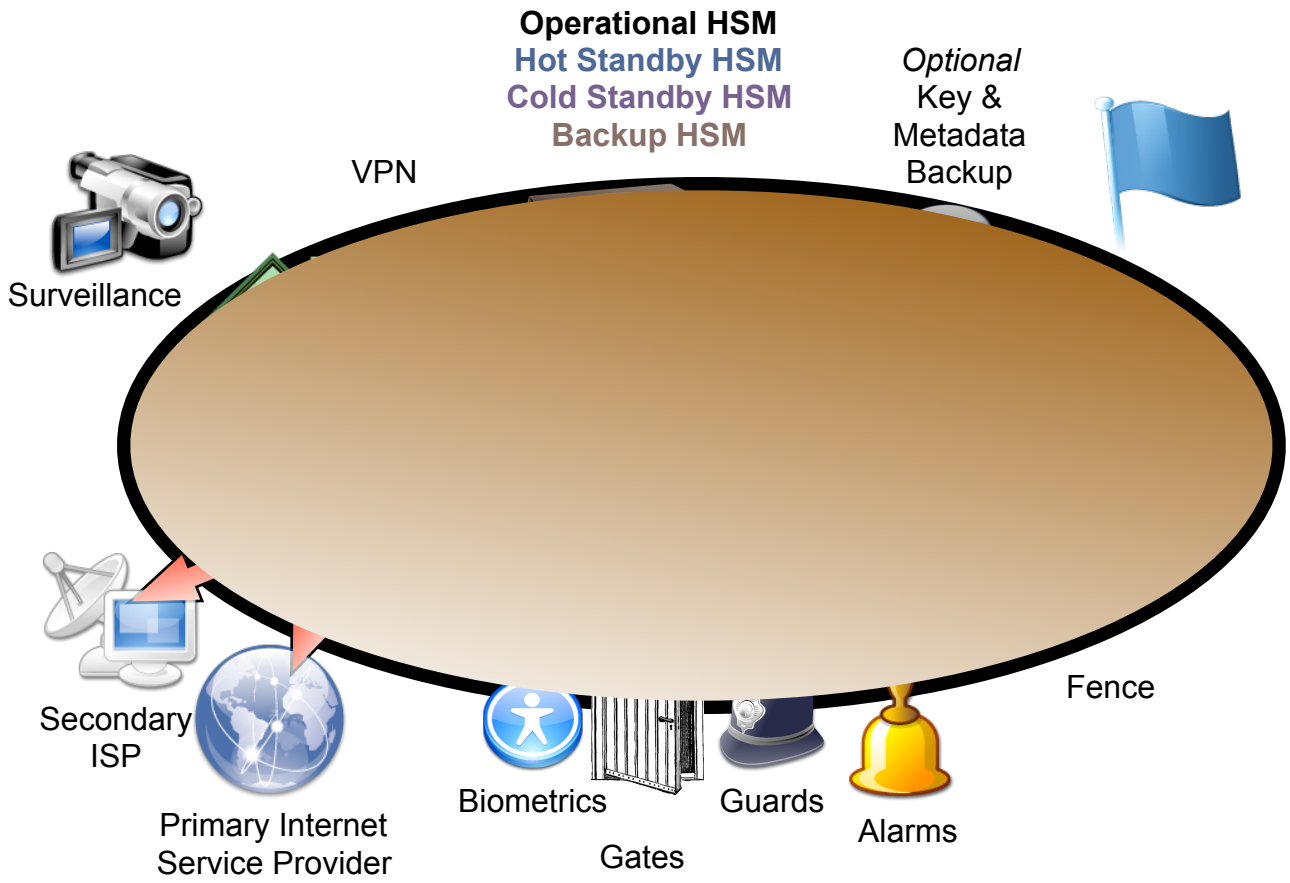
b) Secondary/Backup Facilities
i. Additional Operational Components, ii. Hot spare Components, iii. Warm Spare Components
iv. Cold Spare Components v. Additional Backup Components.

6.4.14 Operational Key Storage: Operational key storage involves placing a key in local storage for use during its cryptographic period without making a copy. Keys should be either physically or cryptographically protected when in storage (see [SP 800-57-part1]).

6.4.15 Backup Key Storage: Backup key storage involves placing a copy of a key in a safe facility so that it can be retrieved if the original is lost or modified. Backup copies of keys may be located in the same or a different facility than the operational keys to assure that the keys can be retrieved when needed even after a natural or man-made disaster.

5.2. Visual illustration of a CKMS Secondary Facility

The primary facility and secondary facility are implemented using equivalent components and security measures as illustrated below. Note the green building is used to indicate "secondary facility".

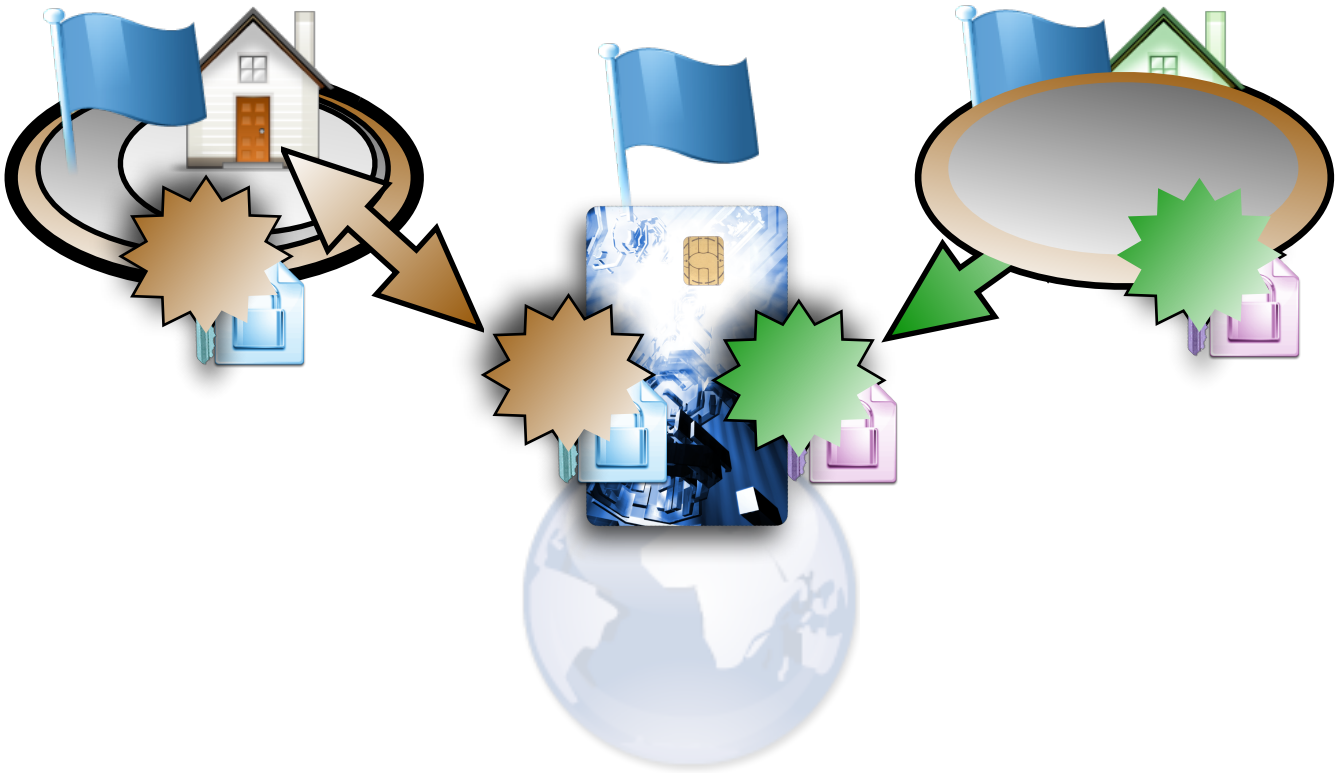


5.3. Visually illustrating the role of Primary and Secondary Facilities

According to Section 6.8.7, page 55, Draft SP 800-130, the Primary and Secondary systems, manage independent sets of keys, and need to be situated in physically different sites. This means that 'every logical key' managed by a CKMS must in reality be two keys with different values, one key per site.

To support this dual-key arrangement correctly in data-storage applications, each encrypted datum must have a unique key, that datum key encrypted two times, one under a key managed by the primary server, and one under a key managed by the secondary server. In this way the datum can be decrypted by accessing either of the primary or secondary facilities. During encryption, the data may need to be accessed twice - once when the primary facility is online, and then later when the second facility is online. **The requirements on dual-key management for data-storage applications has not been discussed in the Draft SP 800-130 document.**

In the illustration below, the blue flags show that the primary and secondary site are managed by the same organisation. Either the primary or secondary system is operational, but not both at the same time. This means that the system must cycle between systems so as to ensure all CKMS policy requirements (delete, key rolling, etc.) are maintained for all data. **This is not clearly articulated in the draft SP 800-130 documentation.**



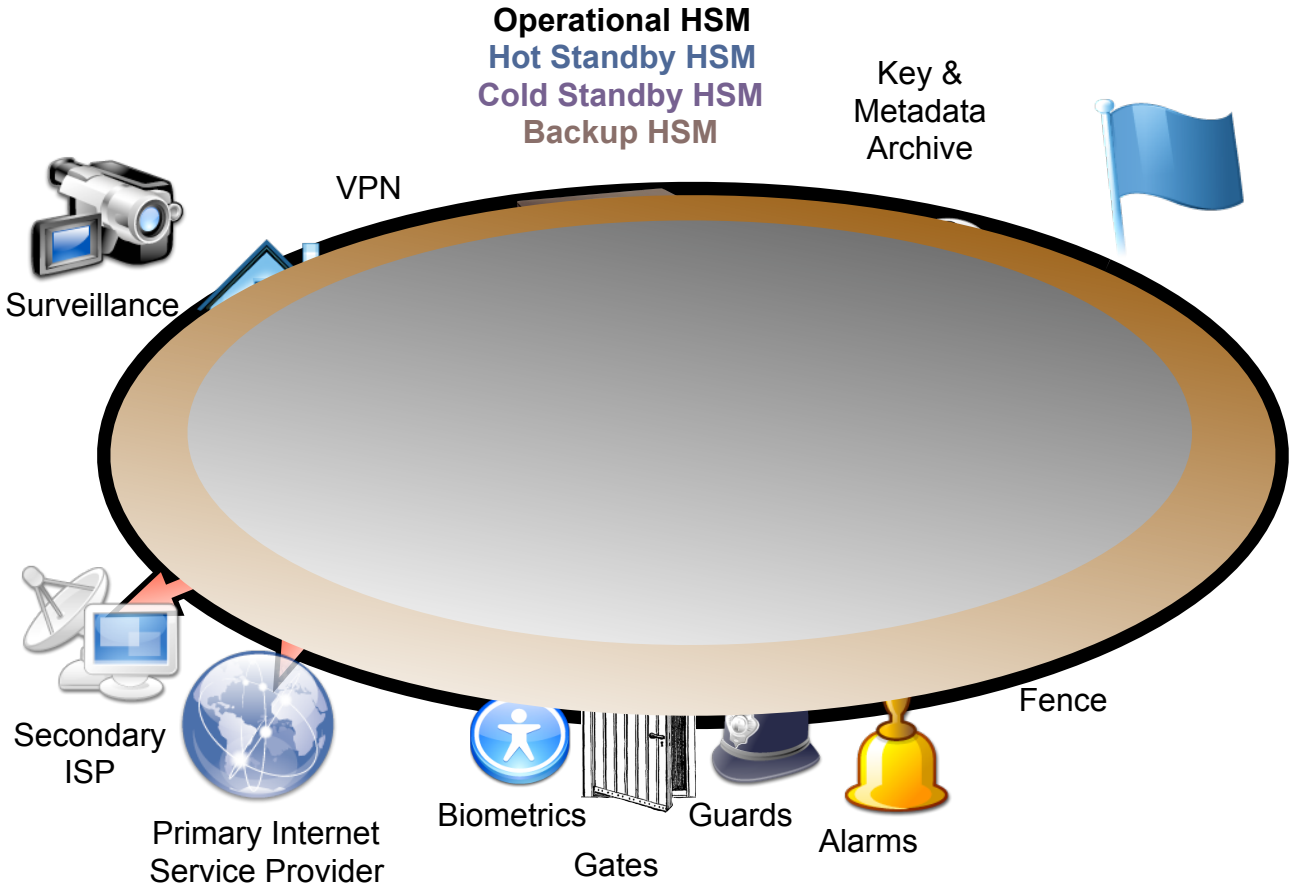
In the above illustration, the user accessing the key material (smart card) is also under the authority of the same organisation. The paper with a lock on it visually illustrates the associated meta-data bound to each key. The smart card CKMS must enforce the policies as provided to it by the primary or secondary CKMS facility.

The uniform 'enforcement of policies' throughout the entire CKM ecosystem, including edge hardware security modules, smart card HSM, and desktop computers using that key material needs to be articulated in the NIST Draft SP 800-130 documentation.

We observe in the illustration above that the smart card is a portable device that is normally used to authenticate humans. It is not possible for the CKMS in the smart card to be protected at all times in the way that the HSM in the primary and secondary facilities are. **This is not clearly articulated in the draft SP 800-130 documentation.**

5.4. Visual illustration of a CKMS Backup Facility

The picture below illustrates most of the requirements identified for a backup facility as described in the quotations below sourced from the draft NIST SP 800-130 document. Apparently, the most significant difference between a CKMS Primary and Secondary Facility and a CKMS Backup Facility, is that the CKMS Backup facility replaces the Optional Key & Metadata backup database with long-term data storage (such as possibly tape storage).

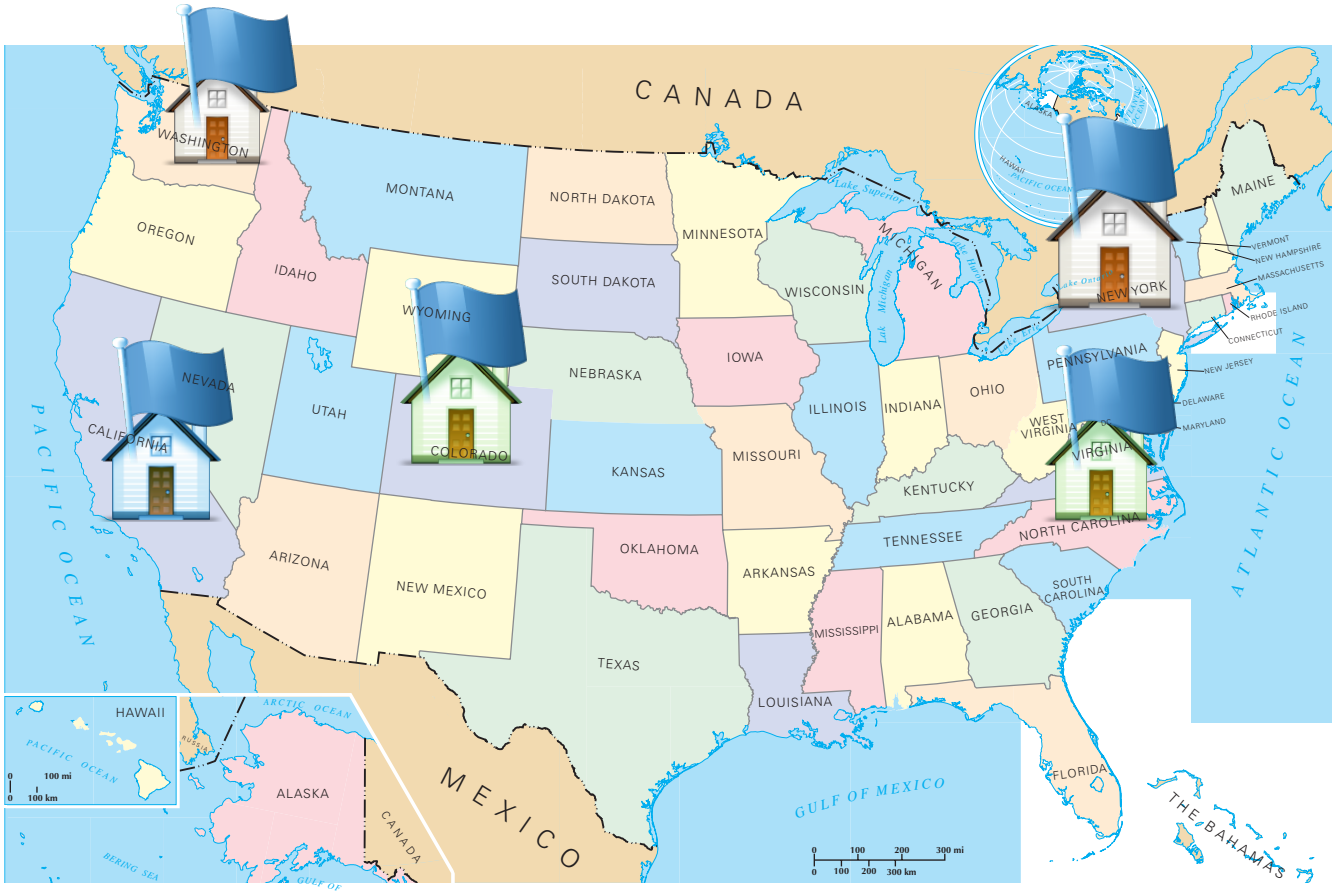


QUOTES from SP 800-130

> **Section 6.4.16 Key Archive:** **Key archive involves placing a key in a safe long-term storage facility** so that it can be retrieved when needed. Key archiving usually requires provisions for moving the key to new storage media when the old media are no longer readable because of aging of, or technical changes to, the media readers. Archived keys should be automatically retrieved from the old storage medium and restored on the new storage medium when a storage medium replacement is made.
>
> **6.4.17 Key Retrieval:** Obtaining a cryptographic key from **storage**, **a backup facility**, **or an archive** is considered retrieval if done during normal CKMS operation. If there has been an environmental or man-made disaster and the key cannot be normally retrieved and used, the key may have to be recovered by special means or with special permission (see Section 6.4.19). The CKMS security policy should state the conditions under which a key may be retrieved normally.

We have included HSM in our illustration as HSM are presumably required to perform all cryptographic operations on the key & metadata stored in the long term data storage media.

5.5. A Topology of Primary, Secondary and Backup Facilities



In this illustration above we have two primary facilities (Washington, New York), two secondary facilities (Colorado, North Carolina), and a backup facility (California) as suggested by the quote from the draft NIST SP 800-130 specifications below. It appears that all facilities are owned and managed by the one organisation.

> **8.1 Physical Security controls:** … A CKMS can consist of one or more primary facilities and one or more backup facilities. Each of these facilities should be protected. At each facility, CKMS components can consist of active, standby or backup components, each of which should be protected.

6. Definition of a CKMS in NIST SP 800-130

In this section we will draw out text (**bold** typeface) indicating the scope of the CKMS definition in SP 800-130. It is important to consider the following text in the light of the diagrams in the previous sections which describe the physical and logical security requirements for a CKMS as some type of "enterprise key management system". We comment on these quotes in the next section.

> *Cryptographic Key Management (CKM) is a fundamental part of cryptographic technology and is considered one of the most difficult aspects associated with its use. Of particular concern are the scalability of the methods used to distribute keys and the usability of these methods. NIST has undertaken an effort to improve the* ***overall key management strategies*** *used by the public and private sectors in order to enhance the usability of cryptographic technology, provide scalability across cryptographic technologies,* ***and support a global cryptographic key management infrastructure.***
>
> *http://csrc.nist.gov/groups/ST/key_mgmt/*

This Framework for Designing Cryptographic Key Management Systems (CKMS) contains descriptions of CKMS components that should be considered by a CKMS designer and specifies requirements for the documentation of those CKMS components in the design. This Framework places documentation requirements on the CKMS design document. ***Thus, any CKMS, that is properly documented, could have a design document that is compliant with this Framework.***

Abstract, page 2, Draft SP 800-130

The ultimate workshop goal was to define and develop technologies and standards that provide cost-effective security to cryptographic keys that themselves are used to protect computing and information processing applications.

Introduction, page 9, Draft SP 800-130

This document is intended for designers, implementers, security analysts, managers, system procurers, and users of CKMS to manage and protect keys.

Audience, page 10, Draft SP 800-130

Today's information systems *and the information that they contain are considered to be major assets that require protection. The information used by government and business is contained in computer systems consisting of groups of interconnected computers that make use of shared networks, often referred to as the Internet.* ***Since the Internet is shared*** *by diverse and often competing organizations and individuals,* ***information systems should protect themselves*** *and the information that they contain from unauthorized disclosure, modification and use. Even the denial of service to legitimate users is considered a significant threat. The information used by these systems requires protection when it is at rest within a protected facility, and also when it is transported from one location to another.*

2.1 Rationale for Cryptographic Key Management Page 11, Draft SP 800-130

Cryptographic techniques use cryptographic keys that are managed and protected throughout their life cycles by the CKMS. … The CKMS binds a key to its critical metadata in order to control the proper use of the key. … The CKMS is designed to provide the necessary protection for keys and bound metadata.

2.1 Rationale for Cryptographic Key Management Page 11, Draft SP 800-130

This document is not oriented to a particular CKMS or class of CKMS for an Enterprise or Enterprise Class (such as US Federal Government, Aerospace, Health Care, etc.). This Framework is intended to meet the needs of a wide variety of CKMS and Enterprises.

2.2 Framework Components and Requirements, Page 12, Draft SP 800-130

There is extensive use of cryptography in several security protocol standards (e.g., TLS, IKE, SSH, CMS, etc.) where ephemeral keys (i.e., cryptographic keys with short lifetimes that are changed often) are used by the protocols themselves. These protocols may also employ and distribute static keys (i.e., long-term keys) that are securely distributed using some other means. *While, the focus of a CKMS is on the generation, distribution and storage of the static keys,* ***a CKMS design covers the generation and storage of the ephemeral keys as well.***

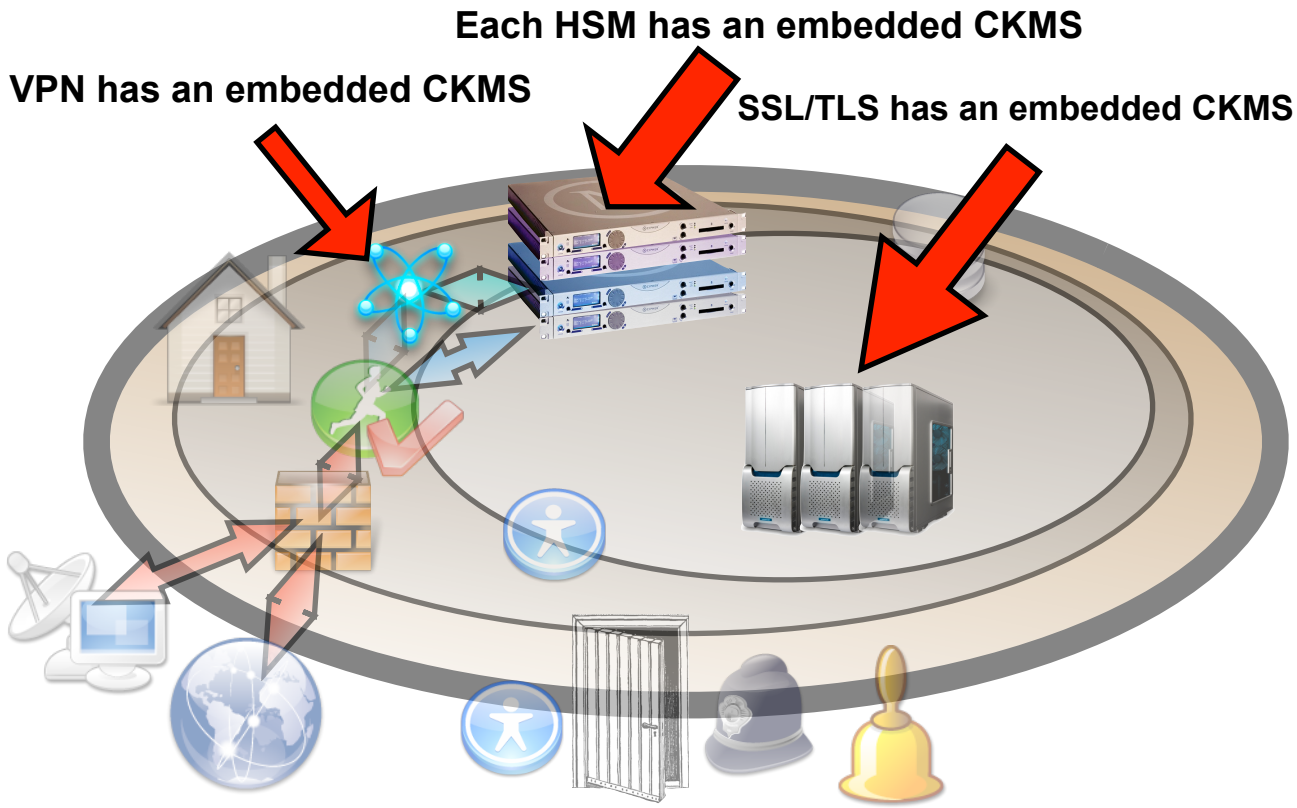
3.1 Providing Key Management to Networks, Applications, and Users, Page 14, Draft SP 800-130

7. CKMS nomenclature

The draft NIST SP 800-130 does not explicitly differentiate between different types of CKM products and case-uses. For example, when discussing network security controls of a CKMS, it becomes clear that the devices protecting that primary CKMS may themselves employ their own embedded CKMS.

> *The scope of network security controls includes boundary devices, such as a firewall, a VPN, an intrusion detection system, and an intrusion protection system. The scope of network security controls excludes cryptographic functions, cryptographic protocols, and cryptographic services,* ***except when used for the operation of the aforementioned network security control devices.***
>
> *6.8.4 Network Security Controls and Compromise Recovery, Page 52, Draft SP 800-130*

In addition, devices that are "implementing" the CKMS functionality themselves may have embedded CKM modules as illustrated below:



It is tempting to consider drawing on NSA terminology with regard to their Electronic Key Management System (EKMS) [22], however this appears purpose built for their operating environment and probably not flexible enough to express the comprehensive scope of NIST SP 800-130.

A CKMS framework COULD be applied to describe the essential functions almost every device that performs cryptographic services. Clearly different CKM products would be focussed on supporting different case uses. However, the same core CKMS library (or modules within that library) could be used by most of those devices. e.g, all CKM devices require key management and policy compliance, and that module could be implemented according to a unified standard across all devices. **i.e. dependent devices should enforce policies as described and managed by the CKM that supplied it the key material + meta-data.** It is generally not sufficient to simply issue the value of key material after certain gate keeper policies have been met.

[22] http://en.wikipedia.org/wiki/EKMS

Some CKMS products are designed predominantly to manage long-lived key material and manage it's re-distribution to dependent cryptographic devices. Other CKMS products might use key material received from a product designed to manage long-lived key material and perform data processing operations for a short period of time and then delete it's knowledge of the secret value of the long-lived key. A SSL/TLS device has a CKM module that relies on the digital signatures associated with one or more other CKM service providers (root certificate authorities)…

The design of a CKMS system must extend to all cryptographic processing devices, irrespective of their case-use. We need an improved taxonomy and nomenclature so that we can reference each different case-use without confusion.

Examples of different case uses include, but are not limited to:

- Security Application CKM profile:
  - a Secure Socket Layer, Secure Shell, type of application
    - will require ephemeral key management capabilities
    - may or may not require long lived key management capabilities, ...
- Security Appliance CKM profile:
  - A hardware dedicated device that implements a security application
- Business CKM profile:
  - single ownership, one domain
- Enterprise CKM profile:
  - single ownership
  - mutually suspicious semi-autonomous domains (accounts, sales, r&d, ...) within single enterprise
- Inter-enterprise CKM profile:
  - Interoperability between mutually suspicious Enterprise CKM servers owned/managed by different organisations
- Global-scale inter-enterprise CKM:
  - A single unified international system
  - Arbitrary number of CKM server owners (Federated)
  - Distribution of key material storage over different CKM service providers (prevent single point of trust failure)
  - Global co-ordination of name spaces
  - Support key management by public identifiers (Universal Resource Indicators, e-mail address, …)

The CKMS nomenclature must be able to support cases where one CKM device is "storing key material" on behalf of it's dependent CKM devices, and also where one or more CKM devices are facilitating key material exchanged between two dependent devices, such that the facilitators don't know the value of the final key negotiated by the two dependent devices.

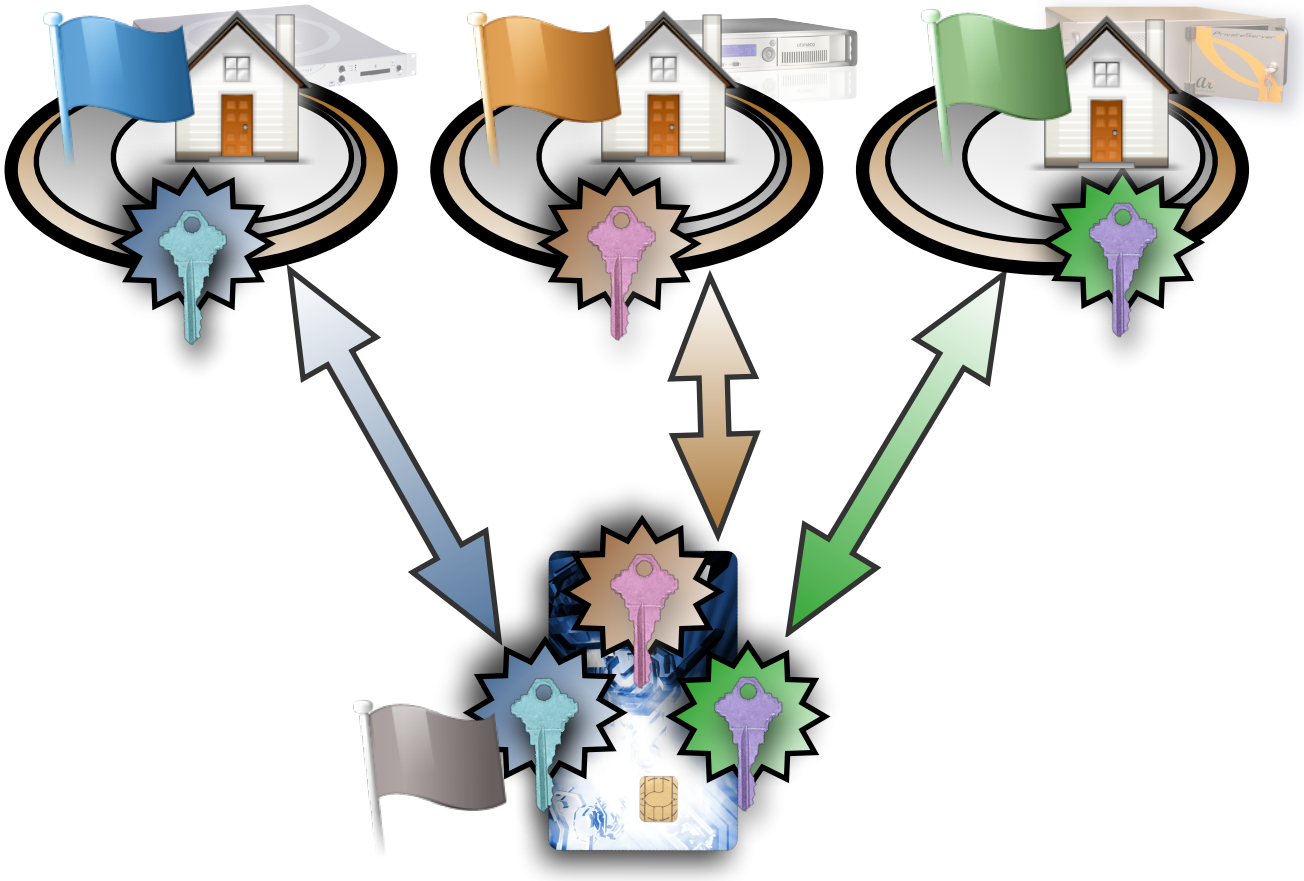
Unfortunately we have not had time to attempt to define a clear and comprehensive nomenclature. We would welcome the opportunity to explore collaborating with NIST on the creation of one.

8. A candidate (public domain) multi-organisation CKMS architecture published in 1976

By necessity, the NIST SP 800-130 must make certain assumptions about what the CKMS domain model looks like. Based on the public quotes around the NIST CKM project, the overall aim of the NIST CKM project is to design a scalable inter-organisation CKM. The current CKMS model as envisioned by the current draft standard may not be sufficiently expressive to capture a CKMS where facilities and computer equipment are owned and operated by different organisations. For instance, the current model may not readily facilitate the description of the civilian PKI or federal PKI X.509 architectures where there are several autonomous root-certificate authorities, and clients of those authorities belonging to yet further other organisations.

In 1976, Whitfield Diffie, Martin Hellman, and Leslie Lamport proposed a cryptographic key management system[23] involving m key distribution centres. Client operations are performed over *n* out of *m* KDC, where *n* is ≤ *m*, preferably *n* = *m*. The system offered security against a simultaneous compromise of *n*-1 of the *n* participating KDC.

If we enhance the original 1976 design implementation details by requiring that each KDC is run by a different organisation, we achieve protection against insider attacks from any one of the KDC owners. If we enhance the design implementation details further by requiring that each KDC runs different HSM, we protect against insider attacks from WITHIN the vendor of any one HSM provider. **These properties improve the survivability of a CKMS from many of the difficult problems posed by "system wide" CKMS compromise when using only one HSM vendor, or one CKMS provider. These are properties required of large trustworthy CKMS.**



[23] Diffie, W., and Hellman, M. E. Multiuser cryptographic techniques. In AFIPS '76, ACM, pp. 109–112. Available at http://doi.acm.org/10.1145/1499799.1499815.

Part 3:
Proposed adjustments to the structure of the CKMS document

9. The CKMS document could be enhanced to co-ordinate communication between client, vendors and integrators

> *This Framework for Designing Cryptographic Key Management Systems (CKMS) contains descriptions of CKMS components that should be considered by a CKMS designer and specifies requirements for the documentation of those CKMS components in the design. This Framework places documentation requirements on the CKMS design document.* ***Thus, any CKMS, that is properly documented, could have a design document that is compliant with this Framework.***
>
> *Abstract, page 2, Draft SP 800-130*

> *"This document is intended for designers, implementers, security analysts, managers, system procurers, and users of CKMS to manage and protect keys."*
>
> *Section 1.1, page 10, Draft SP 800-130*

The current CKMS design requirements document (SP 800-130) does not clearly articulate **how** the different types of organisations should use this document, or **how** SP 800-130 could be applied by one type of organisation communicates with another organisation of a different types. The quote above talks about different "job roles of humans" which could use this CKMS design framework. However, it does not talk about different "organisation" types that may need to use the CKMS design framework. For instance:

- how does a **vendor** organisation uses SP 800-130 to describes the properties of existing CKMS product they are offering for sale to one or more different client organisations
- how should a **client** organisation use this framework to describe their CKMS specific requirements to satisfy that organisations mission objectives
- how should a **lead contractor/integrator** organisation take the CKMS documents of one client and one or more vendors, , and proposes a new CKMS document that satisfies the client's requirements based on the performance details of the vendor CKMS product design documents.

For example: A client sets the scalability requirements of a CKMS implementation along with a justification for the necessary performance requirements. A vendor offers documentation outlining the actual capabilities of various CKMS deployment(s) of their product as measured. The integrator creates CKMS documentation that projects the capabilities of a specific tailored CKMS deployment configuration along with supporting justifications. The integrator then must show that the final deployment achieves the requirements of the client before turning the system on.

Section 4 of SP 800-130 describes a process that is exclusively client oriented.

It may be that different versions of section 4 could be written for the vendor and integrator perspectives.

10. Scope of the CKMS document

> *"This Framework for Designing Cryptographic Key Management Systems (CKMS) was initiated as a part of the NIST Cryptographic Key Management (CKM) Workshop. The ultimate workshop goal was to define and develop technologies and standards that provide cost-effective security to cryptographic keys that themselves are used to protect computing and information processing applications"*
>
> *Section 1, page 9, Draft SP 800-130*

> *"This document is intended for designers, implementers, security analysts, managers, system procurers, and users of CKMS to manage and protect keys."*
>
> *Section 1.1, page 10, Draft SP 800-130*

> *"There is extensive use of cryptography in several security protocol standards (e.g., TLS, IKE, SSH, CMS, etc.) where ephemeral keys (i.e., cryptographic keys with short lifetimes that are changed often) are used by the protocols themselves. These protocols may also employ and distribute static keys (i.e., long-term keys) that are securely distributed using some other means. While, the focus of a CKMS is on the **generation**, **distribution** and storage of the static keys, **a CKMS design** covers the **generation and storage of the ephemeral keys as well.**"*
>
> *Section 3.1, page 14, Draft SP 800-130*

As previously discussed briefly in the section on Nomenclature, the document is currently geared towards describing the design of a large-scale Enterprise Cryptographic Key Management System (ECKMS) e.g., OASIS KMIP, Voltage Key Management Server, SafeNet DataSecure Appliance, and so on. However, very large portions of this document could equally apply to an implementation of Secure Socket Layer, Secure Shell, the software within a Hardware Security Module or any other security product that uses cryptography. All such devices generate, distribute, store and use key material, making them cryptographic key management systems. Public key material and meta-data is managed for long periods of time in practically all those devices. Some of the document as it stands applies directly to ALL these cryptographic applications, where as some aspects of the document are geared at addressing specific cryptographic applications (mass aggregation of key material in online centres, ...).

The following question is an ontological one. Where does the boundary of a CKMS start and stop? If a HSM deployed in the field is processing cryptographic key material that is centrally managed in a data centre, does that field deployed HSM constitute part of the overall CKMS? Likewise if a user authenticates themselves with a smart card token, clearly the security controls to manage the keys on the token must be managed at a level of security commiserate to the level of risk that token poses to the entire system.

The current organisation of the document is generally quite good with regard to ECKMS. However, a re-factoring of the document to reflect the overlap and differences between applications would increase the utility of this standard.

> *The Framework contains many possible components, but the selection of which components are to be used is left to the CKMS designer who produces the CKMS design. Not all components have to be selected for a particular CKMS.*
>
> *Section 2.1, page 12, Draft SP 800-130*

Restructuring would help clarify "which components" may or may not be present, while still making it clear what **shall** be required in the specifications of those components when present.

11. The possibility of a range of NIST SP 800-130 compliant CKMS design profiles

Extending from the immediately preceding section, does it make sense to describe profiles for 800-130 compliant CKMS design documents, similar to the way common criteria supports product/case-use specific variations.

For example certain profiles might focus on functional capabilities:

- CKMS with long lived key management capabilities
- CKMS with ephemeral key management capabilities
- CKMS with identity assertion capabilities
- CKMS with public key agreement protocols
  - classical security with NIST compliance
  - experimental second generation public key technologies
- CKMS with symmetric key agreement protocols
  - using relays
    - using key distribution centers
      - symmetric key kerberos
    - using key translation centers (optionally on mesh networks)
      - secure ad-hoc wireless mesh network topologies
      - diffie-hellman-lamport simultanously employing several KTC
      - phase shift keying and frequency shift keying (wavelength agility) to secure optical layer
    - using key agreement centers (optionally on mesh networks)
      - quantum key distribution networks
- CKMS with hot backup capability

Other Profiles might target specific application domains. Compliant profiles below may incorporate one or many of the above functional profiles.

- Security Application CKM profile
  - a Secure Socket Layer, Secure Shell, type of application
    - will require ephemeral key management capabilities
    - may or may not require certain long lived key management capabilities, ...
- Security Appliance CKM profile
  - A hardware dedicated device that implements a security application
- Business CKM profile
  - single ownership, one domain
- Enterprise CKM profile
  - single ownership
  - mutually suspicious semi-autonomous domains (accounts, sales, r&d, ...) within single enterprise

- Inter-enterprise CKM profile
  - **Interoperability between mutually suspicious** Enterprise CKM servers owned/managed by different organisations
- Global-scale inter-enterprise CKM
  - A single unified international system
  - Arbitrary number of CKM server owners (Federated)
  - Distribution of key material storage over different CKM service providers (prevent single point of trust failure)
  - Global co-ordination of name spaces
  - Support key management by public identifiers (Universal Resource Indicators, e-mail address, …)

Part 4:
Robust interoperability is required to ensure cryptographic security and policy enforcement is uniformly maintained

12. Binary and Semantic Interoperability

Binary interoperability testing is independent to semantic interoperability testing of the individual fields in the protocol. e.g., if one systems receives keys and meta data from another system, does it then enforce the same semantic rules? Do both systems behave the same in response to the same binary message. (See 47 minutes into Subhash Sankuratripati, "Interoperable Key Management using the OASIS KMIP Standard"[24], IEEE KMS 2010)

Semantic interoperability is extremely important requirement that needs to be clearly addressed comprehensively by SP800-130 consistently through out the entire document. It may require a section dedicated to addressing the definition of semantic interoperability and how semantic interoperability can be validated.

We are glad to see semantic interoperability addressed at least once in the current draft:

> *The CKMS design shall specify all syntax, semantics, and formats of all keys types and their bound metadata that will be created, stored, transmitted, processed, and otherwise managed by the CKMS.*
>
> *Section 6.2, page 29, Draft SP 800-130*

However the word “semantic” was not found in section 9.3 “Interoperability Testing”.

Maybe the quoted text above could be refined to require that the specifications should be adequate to allow another party to accurately import key material stored within the CKMS being described.

With regard to binary testing, one possible requirement is that “exporting key material from vendor A, to vendor B, and then exporting that material from vendor B back to vendor A should result in the updated key material in vendor A being functionally equivalent to what that key material was before it was exported”. (Avoid the “Telephone”[25] problem where the meaning of a message gets corrupted as the information is relayed sequentially over a chain of different entities.)

[24] http://2010.keymanagementsummit.org

[25] Also known as the game: http://en.wikipedia.org/wiki/Chinese_whispers

Part 5:
Expanding the communities of interest by applying Safety Systems Standards

13. Possibility of adopting the Functional Safety Integrity levels within NIST SP 800-130?

*(Elaine Barker) We also need key inventory control, accountability/auditing of the keys, policies for managing the keys and metadata, **and safety requirements** for certain applications.*

Cryptographic Key Management Workshop Summary, NIST Interagency Report 7609

"Functional security addresses the ability of systems to perform their functions in the face of intentional or unintentional cyber threats while assuring fail-safe operation."

https://web.archive.org/web/20130523093337/http://community.controlglobal.com/content/deepwater-horizon-bp-oil-spill-appears-be-control-system-cyber-incident

Using IEC 61509 Safety Integrity Levels (SIL) in the NIST CKMS process could be seen to support US National objectives to improve critical infrastructure protection (CIP) by ensuring CKMS designs are engineered at a level suitable for use in CIP.

*IEC 61508 is concerned with achieving **functional safety,** where safety is defined as freedom from unacceptable risk of physical injury or damage to the health of people, either directly or indirectly as a result of damage to property or to the environment. So damage to long term health, including damage to property or the environment that leads to damage to long term health, is explicitly within the scope of the standard and is encompassed by the term safety.*

*It is recognised that **the consequences of failure could also have serious economic implications** and in such cases the IEC 61508 standard could be used to specify any electrical/ electronic/programmable electronic system used for the protection of equipment or product.*

http://www.iec.ch/functionalsafety/faq-ed2/ (See A5)

One benefit of the IEC 61508 standard is that it requires the CKMS designer to explicitly consider security/safety over the entire operational life cycle of a CKMS, including upgrades and decommissioning.

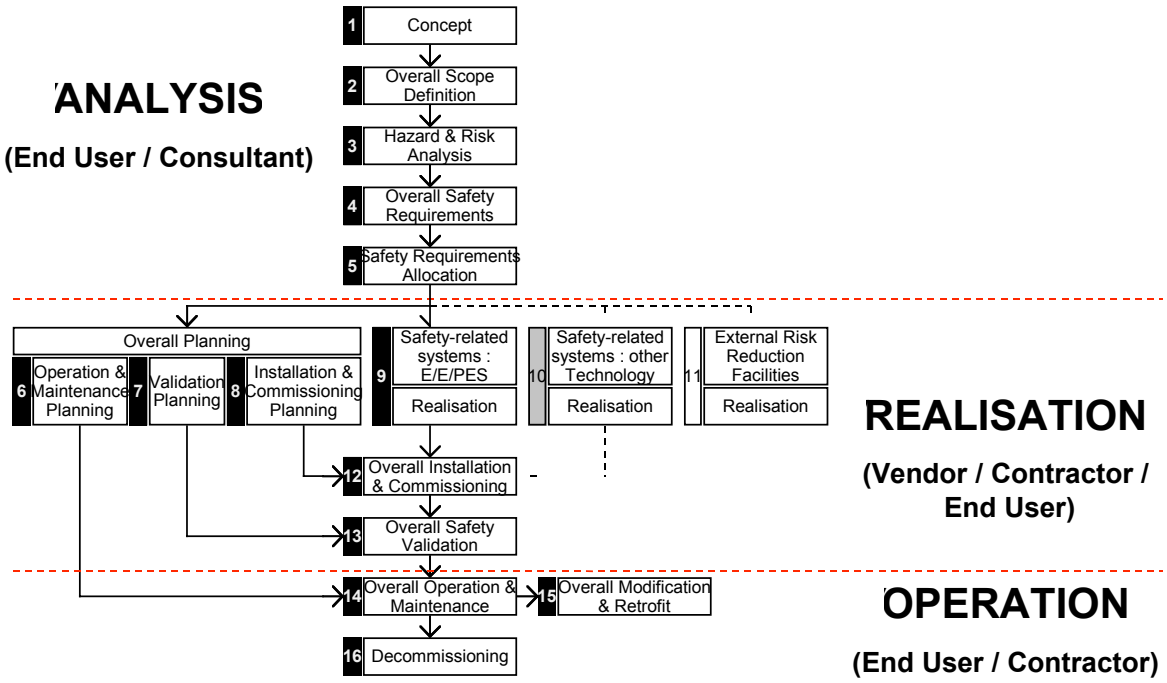


Figure Safety life cycle from IEC 61508.

By way of reference, nuclear power sites are typically designed to achieve SIL 3 and there are very few SIL 4 projects.

Safety Integrity Level	Probability of failure on demand, average (low demand mode of operation)	Risk reduction factor
SIL 4	>= 10^{-5} to < 10^{-4}	100000 to 10000
SIL 3	>= 10^{-4} to < 10^{-3}	10000 to 1000
SIL 2	**>=10^{-3} to <10^{-2}**	**1000 to 100**
SIL 1	**>=10^{-2} to <10^{-1}**	**100 to 10**

Alternatively, other safety standards such as DO-178B[26] for software systems might also be considered for suitability by a CKMS design.

See "*Basic concepts and taxonomy of dependable and secure computing*"[27] which gives the main definitions relating to dependability, a generic concept including as special case such attributes as reliability, availability, safety, integrity, maintainability, etc. Security brings in concerns for confidentiality, in addition to availability and integrity. The aim of this paper is to explicate a set of general concepts, of relevance across wide range of situations and help communications and co-operation among a number of scientific and technical communities.

[26] http://en.wikipedia.org/wiki/DO-178B

[27] Avizzienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. "Basic concepts and taxonomy of dependable and secure computing." In IEEE Transactions on dependable and secure computing (Jan. 2004), vol. 1.
http://www.loria.fr/~simonot/SlidesCSSEA/IEEETransonDependableComputing2004.pdf

Part 6:
Expanding the communities of Interest by encouraging adoption of Aerospace and Defence documentation standard

14. Possibility of adopting the S1000D Aerospace Documentation standard within NIST SP 800-130?

> *"S1000D is an international specification for the **production of technical publications**. Although the title emphasizes its use for technical publications, application of the specification to non-technical publications is also possible and can be very beneficial to **businesses requiring processes and controls**. … This specification was initially developed by the AeroSpace and Defence Industries Association of Europe (ASD)."*
>
> *http://public.s1000d.org/Pages/Home.aspx*

There may be benefit to requiring certain aspects of NIST compliant CKMS design specifications to be S1000D compliant. S100D defines a restricted use of the English language[28] (which improves the clarity of description, particularly for non-english readers) and can be used to describe information required to conduct a task[29] or describe the system itself[30].

Importantly, the use of S1000D increases the range of agencies in the US Government that can use the NIST CKMS compliant documentation.

S1000D is adopted by:

1. Members of Aerospace Industries Association of America (AIA).
2. Members of International Coordinating Council of Aerospace Industries Associations
3. (ICCAIA) not included in Categories 1 thru 2 inclusive.
4. Airlines and Armed Forces that are customers of Companies included in Categories 1 thru 3 inclusive.
5. Ministries of Defence of the member countries of ASD.
6. The Department of Defense of the USA.

Examples of adoption[31] include: NATO, Boeing, General Dynamics Canada, BAE Systems, Saab Military Aircraft, The Royal Navy, and Lockheed Martin.

---

[28] https://web.archive.org/web/20140513123932/http://cpf.s1000d.org/events/user_forum/munich/f09h30_ste_s1000duf_munich_berry.pdf

[29] See requirements found in section 6.8.5 "Violation of Procedures and Recovery from Violations" and the need to specify security procedures to be followed by personal.

[30] See requirements found in section 7. "Interoperability and Transition Requirements for CKMS"

[31] ~~http://www.allbusiness.com/technology/software-services-applications/5617087-1.html~~
(This hyperlink no longer works, no copy of this page could be found online)

Part 7:
CKMS, Compliance, and Enforcement

15. CKMS documentation shall specify what explicit support it has for compliance with different legislation when deployed internationally

How are legal requirements managed by a CKMS design?

Does NIST have advice that could be inserted into NIST SP 800-130 on how the requirements as dictated by International Law (UN), and various national Laws (US, European, UK, AU) are managed.. e.g. if the organisational objective is to support a CKMS system that supports a corporation operating in 100+ countries, how are the legal requirements that are then put on to the CKMS security policy managed?

How are changes with international law managed by a CKMS design over its operational life cycle?

Is there a mechanism to test each policy within a CKMS against legal requirements for the areas where that policy/ operation will be employed? (e.g. Let us say that a CKMS is managing a set of key material that will be accessed by devices located in the US and EU. Does the CKMS have a mechanism to perform run-time tests to evaluate if that set of policies are compliant with current US and EU laws regarding data processing between those two regions?)

Does each organisation have to "rebuild this legal requirement framework" or is this going to be done once with a safe-harbour arrangement? See for our paper[32] discussing this issue.

In some countries, key material/information generated in one country may have limitations in being accessed by users of other countries without appropriate data protection legal contracts in place. (such as with EU Data privacy Laws)

In some countries, there are requirements on requiring certain periods of data-retention, and in other, the need to delete information as soon as possible. These policies change over time. How does the CKMS manage these transient requirements throughout the key management life cycle.

In some countries there are specific protocols for legalised interception. They may have certain policy requirements regarding performance characteristics, assuring the target is not notified, and so on.

In some countries, the passing of key material or data through them, can result in special legal privileges that may need to be enforced by the CKMS [33]. For instance, the Swedish Press Freedom Act, on which part of the IMMI source protection requirements were modelled, requires that journalists & media organisations who promise confidentiality to sources must keep their promise. If they do not, a source has the right to initiate a criminal prosecution against them in Sweden. The Swedish constitution has protection for source anonymity and Belgium law has protection for journalist-source communications confidentiality. A CKMS should enforce those laws as appropriate.

Is there any advice on the mechanisms that a CKMS design should employ to manage these issues?

---

[32] Synaptic Labs, "The need for the EC to fund the development of an electronic requirements management process to support the conversion of existing standards, existing policy guidelines and existing laws of several nations simultaneously in a unified requirements model that also supports national and regional variations.", http://media.synaptic-labs.com/pub/papers/TT/20100127-TT-D3-1b-P4.pdf

[33] Icelandic Modern Media Initiative, https://web.archive.org/web/20100218092454/http://immi.is/?l=en&p=intro

Part 8:
Additional observations regarding cryptographic security

16. Concerning Security Ratings

16.1. Operational Use Period, Algorithm Security Lifetime (quotes)

Table 4: Recommended algorithms and minimum key sizes

Algorithm security lifetimes	Symmetric key algorithms (Encryption & MAC)	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
Through 2010 (min. of 80 bits of strength)	2TDEA[23] 3TDEA AES-128 AES-192 AES-256	Min.: $L = 1024$; $N = 160$	Min.: k=1024	Min.: f=160
Through 2030 (min. of 112 bits of strength)	3TDEA AES-128 AES-192 AES-256	Min.: $L = 2048$ $N = 224$	Min.: k=2048	Min.: f=224
Beyond 2030 (min. of 128 bits of strength)	AES-128 AES-192 AES-256	Min.: $L = 3072$ $N = 256$	Min.: k=3072	Min.: f=256

5.6.2 Defining Appropriate Algorithm Suites, NIST SP 800-57 part 1 (Mar 8, 2007)

Table 1: Recommended Cryptoperiods for key types[13]

Key Type	Cryptoperiod	
	Originator Usage Period (OUP)	Recipient Usage Period
1. Private Signature Key	1-3 years	
2. Public Signature Key	Several years (depends on key size)	
3. Symmetric Authentication Key	$\leq$ 2 years	$\leq$ OUP + 3 years
4. Private Authentication Key	1-2 years	
5. Public Authentication Key	1-2 years	
6. Symmetric Data Encryption Keys	$\leq$ 2 years	$\leq$ OUP + 3 years
7. Symmetric Key Wrapping Key	$\leq$ 2 years	$\leq$ OUP + 3 years
8. Symmetric and asymmetric RNG Keys	Upon reseeding	
9. Symmetric Master Key	About 1 year	
10. Private Key Transport Key	$\leq$ 2 years[14]	
11. Public Key Transport Key	1-2 years	

5.3.6 Cryptoperiod Recommendations for Specific Key Types, NIST SP 800-57 part 1

The recommended comparable key size classes discussed in this section are based on assessments made as of the publication of this recommendation using currently known methods. Advances in factoring algorithms, advances in general discrete logarithm attacks, elliptic curve discrete logarithm attacks and quantum computing may affect these equivalencies in the future. New or improved attacks or technologies may be developed that leave some of the current algorithms completely insecure. ***If quantum attacks become practical, the asymmetric techniques may no longer be secure.*** *Periodic reviews will be performed to determine whether the stated equivalencies need to be revised (e.g., the key sizes need to be increased) or the algorithms are no longer secure.*

5.6.1 Comparable Algorithm Strengths, NIST SP 800-57 part 1 (Mar 8, 2007)

Current cryptographic algorithms should be implemented so that they can be augmented or replaced when needed. See [SP 800-57-part1] for the NIST-recommended lifetimes of government-approved cryptographic algorithms. A CKMS should only use algorithms whose security lifetime will cover the anticipated lifetime of the CKMS and the information that it protects. If the CKMS is intended to remain in service beyond the security lifetimes of its cryptographic algorithms, then there should be a transition strategy for migration to stronger algorithms in the future.

7. Interoperability and Transition Requirements for CKMS, page 56, Draft SP 800-130

"Both of the fundamental intractability assumptions on integer factoring and discrete logarithms break down if a (large) quantum computer could be built as demonstrated by Shor." - page 25, section 6.4

"Advances have often been done in steps (e.g. the improvement from QS to NFS), and beyond ***approximately 10 years into the future****, the general feeling among ECRYPT2 partners is that recommendations made today should be assigned a rather small confidence level,* ***perhaps in particular for asymmetric primitives****." - page 31, section 7.3*

ECRYPT2, "Yearly Report on Algorithms and Keysizes" Deliverable D.SPA.7, Revision 1.0, ECRYPT ICT-2007-216676, July 2009. Available at https://web.archive.org/web/20091222051937/http://www.ecrypt.eu.org/documents/D.SPA.7.pdf

According to NIST SP 800-57:

- **Algorithm Security Lifetime** as the estimated time period during which data protected by a specific cryptographic algorithm remains secure.

- The period of time during which cryptographic protection may be applied to data is called the **originator usage period**, and the period of time during which the protected information is processed is called the recipient usage period.

16.2. Concerning the use of bits and years for quantifying security ratings - Quantum Computation

*The CKMS design **shall** specify the security strength (measured in bits of security) of the cryptographic mechanisms that are employed to protect keys and any sensitive parts of their metadata.*

Section 2.1, page 12, Draft SP 800-130

*The CKMS design **shall** specify the expected security lifetime of each cryptographic algorithm used in the system.*

Section 12.5, page 73, Draft SP 800-130

Key Metadata

j) Security Strength of the Key: A number associated with the amount of work (that is, the base 2 logarithm of the number of operations) that is required to break a cryptographic algorithm. For example, for a TDES key of 168 bits (not including parity bits), the security strength is specified as 112 bits; for a 2048 bit RSA modulus, the security strength is specified as 112 bits.

Section 6.2, page 24, Draft SP 800-130

The CKMS design is required to specify the security strength of an algorithm measured in bits, and years. However, this does not address the important contextual information/underlying assumptions surrounding these measurements.

A CKMS product claiming 30+ years security using ECC (f = 256), AES-128 and SHA-256, has a significantly lower level of assurance of achieving this than a CKMS product that relies exclusively on the security of AES-256 and SHA-512.

While both products can claim Algorithm Security Lifetimes beyond 2030 according to the NIST SP 800-57, only the later configuration could achieve it in the presence of code-breaking quantum computers. Furthermore, it is assumed that the AES-128 is weaker than AES-256 in NIST approved modes of operation, as is SHA-256 compared with SHA-512 in NIST approved modes of operation.

There is a general consensus in the international community, as the security requirements push beyond 10 years for algorithm security lifetimes, the risk to asymmetric primitives increases faster than symmetric key primitives.

*In light of quantum computing, **CKM system designers must look at means other than public key-based key management systems;** they must look at quantum computing-resistant algorithms and schemes.*

"3.13 New Technologies" - NIST IR-7609
also see "2.4.6 Overall Summary of the CKM Workshop: Elaine Barker" - NIST IR-7609

To avoid misrepresentation of the security of a CKMS design, and to improve clarification of design requirements, in addition to measuring security in "bits" and "years" it is necessary to indicate these measures for BOTH classical and quantum computing contexts.

It is necessary that stake-holders understand that certain classes of attack, such as quantum computer attacks, may retro-actively break any 'currently secure' ciphertext if it has been intercepted and archived for later decryption in "wait-and-see" attacks and that it is not possible to go-back and secure that ciphertext after the event.

16.3. Concerning the use of bits and years for quantifying security ratings - Information Theoretic Cryptographic Primitives

> *New CKM technologies are needed to keep up with the increased demand for security, due to significant increases in computer capability, applications, and usage. New or greatly improved technologies are needed in:* ***quantum cryptographic algorithms****/computing, cloud computing, identity-based cryptography, security improvements, speed improvements, usability improvements, and cost reductions.*
>
> *"3.13 New Technologies" - NIST IR-7609*

CKMS designs that support quantum cryptographic algorithms and other information theoretic techniques have a more complex security rating that may-or-may not be easily represented in bits. Furthermore, modern information theoretic techniques may fail to offer information theoretic security against insider attacks, yet safely falling back to the security of post quantum cryptographic primitives against those same insiders.

For example, a Quantum key distribution network may employ 3 non-overlapping/distinct paths across a mesh network to negotiate an information theoretically secure symmetric key. If an insider is able to compromise those three keys, the system may offer no security. To address this, some QKDN additionally employ end-to-end public key exchange operations over the public network as the 4th key between devices to improve security. In this scenario the key exchange will have a classically secure component.

An **interoperable** CKMS meta-data scheme will require the ability to express layered security properties with regard to different adversaries.

16.4. Selection of algorithms within a CKMS

> *Current cryptographic algorithms should be implemented so that they can be augmented or replaced when needed. See [SP 800-57-part1] for the NIST-recommended lifetimes of government-approved cryptographic algorithms.* ***A CKMS should only use algorithms whose security lifetime will cover the anticipated lifetime of the CKMS and the information that it protects. If the CKMS is intended to remain in service beyond the security lifetimes of its cryptographic algorithms, then there should be a transition strategy for migration to stronger algorithms in the future.***
>
> *7. Interoperability and Transition Requirements for CKMS, page 56, Draft SP 800-130*

Proposed additional text in bold:

> *A CKMS should only use algorithms whose security lifetime will cover the anticipated lifetime of the CKMS and the information that it protects.* ***Establishing the security lifetime of the information that it protects should also involve representative consultation with all categories of stake holder (from the owner of the CKMS deployment, through to communities and individuals who entrust their private information to that CKMS deployment). This Algorithm Security Lifetime is calculated as the anticipated operational lifetime of the CKMS plus the largest security lifetime required by the most conservative stake holder of a datum processed by the CKMS. The choice of algorithm must also consider advances in computation (such a quantum computer attacks) that may occur during that period of time.*** *If the CKMS is intended to remain in service beyond the security lifetimes of its cryptographic algorithms, then there should be a transition strategy for migration to stronger algorithms in the future.* ***When comparing the suitability of different algorithms, if an algorithm is anticipated to have a security lifetime less than the lifetime of the CKMS, the cost to upgrade that algorithm in the CKMS system (and all dependent systems) must be considered in the comparison process.***

16.5. Proposed Revision to CKMS Security Policy

> *The CKMS Security Policy should also specify individual responsibilities and the security mechanisms to be implemented and used in order to accomplish its goals and achieve its objectives. It is essential that the CKMS Security Policy support the goals of the organization's Information Management and Information Security Policies.* ***For example, if the Information Security Policy states that the confidentiality of the information is to be protected for up to 30 years, then the CKMS encryption algorithms and key management procedures must be selected to meet that requirement.***
>
> *4.3 CKMS Security Policy, page 19, Draft SP 800-130*

Proposed revised text:

> "For example, if the Information Security Policy states that the confidentiality **of each datum** is to be protected for **a minimum of** 30 years, then **all** the CKMS encryption algorithms, key management, **identity management procedures and security controls in the processing environment** must be selected to meet **and exceed** that requirement".

16.6. Human readable security ratings

The CKMS design might want to define a "human readable" security requirement that does not require deep expert knowledge to comprehend the security advisory.

> e.g. "This CKMS deployment is designed to achieve a minimum of 10 year security ratings for all encrypted data elements against classical computers. To do this the system use 128-bit classically secure primitives throughout all components. However, the CKMS deployment will not achieve 10 years security for each data element if code-breaking quantum computers arrive within 10 years as warned by some prominent quantum physicists. For this reason, we must advise that the duration of security for each utterance of this deployment will reduce as we approach the arrival of large code-breaking quantum computers. For this reason there is a distinct possibility this CKMS deployment will need to be upgraded in 5 years to use fundamentally different cryptographic primitives. It will not be possible to retroactively extend the security duration of utterances of this CKMS after-they have been exposed to potential adversaries."

17. Concerning requirements that may be specific to Key Translation Centres / Secure Relays

The NIST Draft SP 800-130 document does not advise on how to specify the requirements of CKMS design that use symmetric key distribution architectures that employ secure relays arranged in semi-regular or irregular mesh topologies. Key distribution networks follow the same topologies of communication networks: they are either centralised, decentralised or distributed[34]. Key material may be sent over several non-overlapping paths between two nodes and then mixed together to create the session key. There are several US military projects that use secure ad-hoc (distributed) mesh based architectures. **NIST draft SP 800-130 should consider the requirements of these increasingly popular CKMS architectures.**

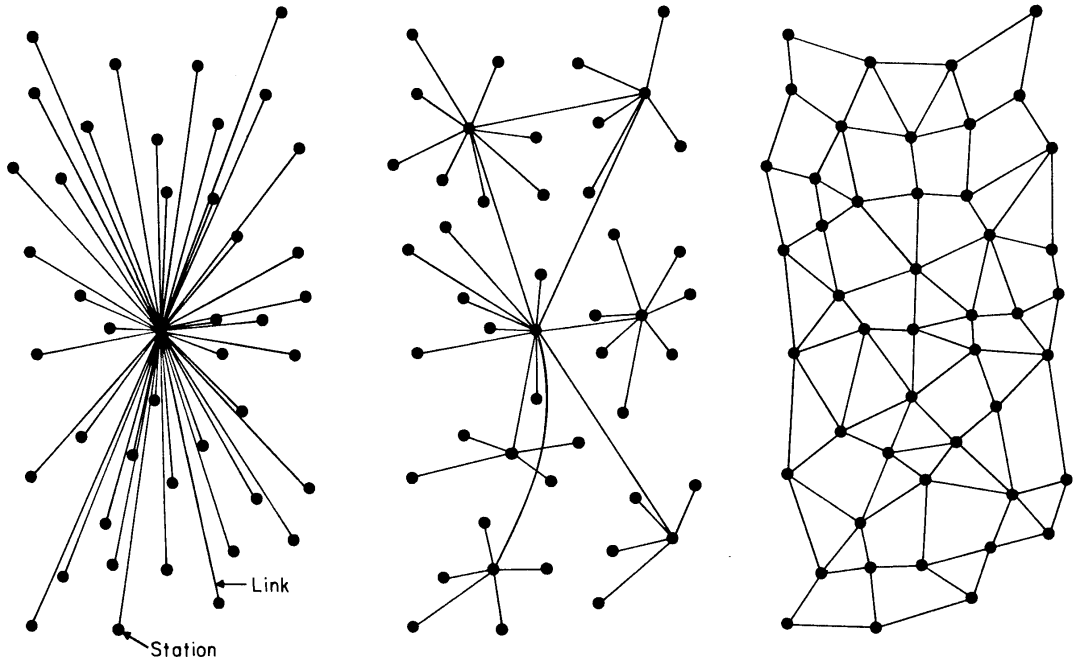


FIG. 1 — Centralized, Decentralized and Distributed Networks

Examples of various key-distribution networks include:

- http://en.wikipedia.org/wiki/Wireless_sensor_network
  - http://en.wikipedia.org/wiki/ZigBee
- http://en.wikipedia.org/wiki/Mobile_ad_hoc_network
  - Anderson, R., Chan, H., and Perrig, A. Key infection: Smart trust for smart dust. In ICNP '04: Proceedings of the 12th IEEE International Conference on Network Protocols (Washington, DC, USA, Oct. 2004), IEEE Computer Society, pp. 206–215. Available at http://www.cl.cam.ac.uk/~rja14/Papers/key-infection.pdf
- http://en.wikipedia.org/wiki/Delay-tolerant_networking
- http://en.wikipedia.org/wiki/Quantum_cryptography
- and so on

[34] Baran, P. On Distributed Communications: I. Introduction to Distributed Communications Networks. Memorandum RM-3420-PR, RAND, August 1964. http://www.rand.org/pubs/research_memoranda/RM3420/

18. CKMS clients should not be able to compromise unrelated CKMS clients

The NIST Draft SP 800-130 offers extensive detail on the 'back-end' of a CKMS system, however the 'client-end' of the CKMS system is almost non-existent. It would be highly desirable if the 'client-side' of CKMS systems was addressed at the same level of detail as the 'back-end'.

We observe that the clients of a CKMS extend the logical perimeter of the CKMS into potentially 'untrusted' spaces (such as users accessing key material from net-cafe's using their smart card token..) Furthermore, the key material on a smart card *might* be extracted using a physical attack.

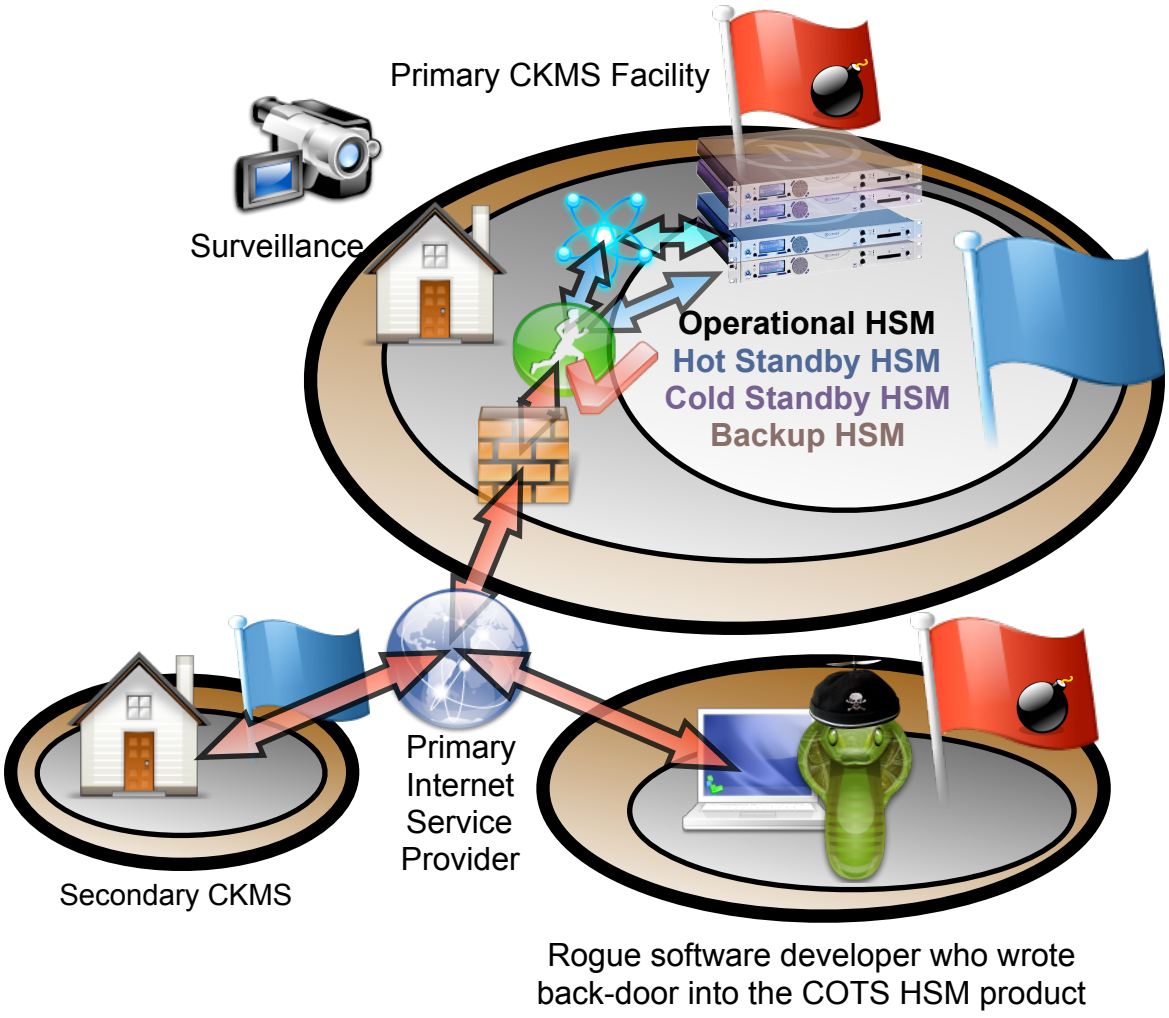
In a global-scale public CKMS, we have to assume that a researcher/adversary will purchase a large number of low-end tokens and extract the symmetric keys using potentially physically destructive attacks. With access to the keys and potentially applet software, an adversary may target the internal protocols of the CKMS system by emulating the smart card on a desktop computer. It may be desirable to:

- Ensure all of the CKMS protocols (external and internal) are robust (and have been tested using fuzzing)
- Ensure all CKMS protocols generate usage notifications that could be supplied to real-time behavioural analysis engines or diagnostic systems.
- Ensure that compromising the keys used to establish a secure session between a client and a CKMS back-end server does not lead to attacks unrelated to that token. (Compartmentalisation of domains).

19. Commercial Off The Shelf (COTS) insider attacks

Customers generally prefer Commercial Off-The-Shelf (COTS) products.

Section 3.5, page 17, Draft SP 800-130



The NIST draft SP 800-130 does not address insider threats in COTS. The most obvious problem with COTS equipment is that the 'client' has limited visibility into the vetting process and controls to protect against insider attacks. Depending on the configuration of the CKMS deployment, the vendor of a Hardware Security Module, or an Intrusion Detection System, or a Firewall, may have remote access to their products deployed in the CKMS. This is a particular issue for "global-scale" CKMS.

An insider attacker may inject malware into a COTS product that triggers when a certain unique 256 bit-pattern is detected in the ciphertext stream. This type of inline trigger injected by modifying the ciphertext of an authorised network session over the Internet. The attack could probably get through most Firewall, IDS, and so on. The malicious function may be as simple as "delete all keys". If the primary CKMS facility and secondary CKMS facility use the same malware infected HSM, both systems could be taken down in quick succession. Foreign Vendors may be forced by their Government to install such a trigger.

Other types of insider attacks may be as innocuous looking as removing bounds checking and creating the opportunity for a "buffer-overflow" attack. The malicious software is then installed remotely in products deployed in the field. **The NIST SP 800-130 should consistently require designs to consider all sources of insider threats, including from within COTS components.**

(Beanie: © Geek Culture. Used with permission. For real beanie caps, visit http://geekculture.com)

19.1. Generic Security Questions re COTS products and insider attacks

Q: Can the overall CKM system survive a compromised HSM product without compromising the confidentiality, integrity and availability of stake-holder information?

> Q: What assurances do we have that COTS does not have a remotely controlled kill switch? A kill switch could be triggered by the reception of a unique 256-bit string that is embedded in maliciously crafted ciphertext supplied to the CKM. (The probability of accidental trigger is negligible.)
>
> Q: Can a HSM receive a remote software upgrade in the field without physically pressing a button / presence of a key / ...

Q: With regard to smart card tokens, what assurance [35] do we have that a firmware upgrade of the smart card operating system code cannot be used to trivially expose secrets embedded with the card? That is, there maybe no overtly malicious source code present, just the presence of a weak security mechanism with regard to firmware upgrades.

Q: What software development practices where used?

> Q: Are their adequate controls in the software development process (is there a revision control system, are audit checks made on every line of code, are software development teams compartmentalised, and so on.

… and so on.

20. CKMS services provided by a cloud service on behalf of organisations that do not have the ability to ensure controls within the CKMS cloud service

The requirement to support billions of users implies that the ***vast*** majority of CKMS stake-holders will not be able to validate or oversee the operation of the CKMS service. They cannot ensure the physical controls nor ensure integrity of the software. Even if a CKMS system is "owned and managed" by the same organisation depending on it, there are always problems regarding insider attacks.

A large scale CKMS design should address the above issues. We observe that techniques such as Diffie-Hellman-Lamport symmetric key distribution design[36] can be used to manage some of the complex trust issues.

The NIST SP 800-130 should require designs to indicate if they are or are not suitable for use by CKMS cloud service providers to provision security services to clients. If a CKMS design says it is suitable, it should explicitly specify what attack vectors are present in the CKMS key-store, and all compute elements that are issued/exposed to key material or sensitive clear text information. The CKMS design should identify all insider attacks, including attacks by insiders managing the cloud compute platform.

The CKMS design should must be careful to avoid 'misrepresenting the trustworthiness' of a system. That is, a HSM at the 'client site' that manages key material may not be able to guarantee the security of the key material/ sensitive data if it is exposed in the clear to compute elements in the cloud. Applications such as "data-mining" and "key-rolling" could rapidly access vast quantities of sensitive data, potentially exposing the sensitive information to insider attacks conducted at the hypervisor/operating system level within the cloud compute environment.

---

[35] "Security Standards for Smartcards", CESG, http://webarchive.nationalarchives.gov.uk/20130128101412/http://www.cabinetoffice.gov.uk/media/311177/smartcard-security_1-1.pdf

[36] Diffie, W., and Hellman, M. E. Multiuser cryptographic techniques. In AFIPS '76: Proceedings of the June 7-10, 1976, national computer conference and exposition (New York, NY, USA, June 1976), ACM, pp. 109–112. Available at http://doi.acm.org/10.1145/1499799.1499815.

Part 9:
Suggestions Regarding Additional CKMS Functionality

21. On the need for user centricity and information self determination in CKMS

> *We rely on other Federal agencies to state their specific requirements so we can come up with CKM standards that satisfy their requirements and **are user-friendly**, cost effective, and secure.*
>
> *William C. "Curt" Barker, NIST IR-7609*

> *While cryptography can provide very effective protection for computer information, if it is not easy to use, then it likely will not be used. A strong case can be made that the largest impediment to the implementation of cryptography is that the burden of key management is often put on the user who is either not capable of, or not willing to, **perform all the security procedures required in a user-centric security system**.*
>
> *3.2 Ease of Use, page 15, Draft SP 800-130*

> *User-centric identity management approaches have received significant attention for managing private and critical identity attributes from the user's perspective. **User-centric identity management allows users to control their own digital identities**. Users are allowed to select their credentials when responding to an authentication or attribute requester and **it gives users more rights and responsibility over their identity information**.*[37]
>
> *Abstract, Privacy-Enhanced User-Centric Identity Management*

> *In the digital world, there are actual threats against the private sphere, be it that of natural persons or legal entities. … it is necessary to establish a **strict regulation of the utilization of specific files and of the traceability records of individuals and goods they carry.***[38]
>
> *ICT Security and Dependability Research beyond 2010, Deliverable 3.3, SecurIST*

User Centricity is **different** from "user-friendly". A program can be user-friendly, yet not orientate information and functionality towards every user/stake holder of the CKMS. User centricity can be thought of like a "customised portal" into the CKMS system for each user. Every stake-holder should be able to see all key material and audit transaction events relating to them from within a CKMS.

User centricity is central to empowering the individual with visibility (all necessary information) and control over their personal data. User centricity is an enabling technology for "informational self-determination".[39]

> *The European citizen's requirements, therefore, are mainly focused around an individual, personal perception of security and dependability and all its related implications. Individual, personal, democratic, self-determined control is much more important to citizens than the traditional, historic, government-controlled central approach to security and dependability.*
>
> ***In the European Information Society, security and dependability concepts must take into account not only central control requirements but also the individual need for security and dependability mechanisms that protect the citizens' privacy and identity.***

---

[37] http://dx.doi.org/10.1109/ICC.2009.5199363

[38] Dooly, Z., Clarke, J., Fitzgerald, W., Donnelly, W., Riguidel, M., and Howker, K. ICT Security and Dependability Research beyond 2010 - Final strategy. Deliverable 3.3, SecurIST EU-FP6-004547, Jan. 2007. Available at https://web.archive.org/web/20101105143643/http://www.securitytaskforce.eu/dmdocuments/d3_3_final_strategy_report_v1_0.pdf

[39] http://en.wikipedia.org/wiki/Informational_self-determination

*A research framework should pay special attention to areas of security and dependability that do not follow 20th century central command and control approaches, but that instead could lead to an open and trustworthy European Information Society in which the **end user is empowered to determine his or her own security and dependability requirements and preferences**. This need for self-determination is accompanied by a need for a reliable, dependable infrastructure that such self-determination can be applied to. Processes of the Information Society will be digitized more and more and there needs to be a reliable, failsafe communications environment and infrastructure in place to support these processes.* [40]

2.2 The Citizen's Perspectives on Security and Dependability, Deliverable 3.0, SecurIST EU-FP6-004547

*Privacy: in the European Union, privacy is generally defined as a right of self-determination, namely, **the right of individuals to determine for themselves when, how and to what extent information about them is communicated to others.***

Regulation addressing this is such as:

- *European Data Protection Directive that is rooted in the concept of consent, while*
- *California SB 1386 is putting a price tag on privacy*

Glossary, SecurIST EU-FP6-004547

*10. Build a cybersecurity-based identity management vision and strategy that addresses privacy and civil liberties interests, leveraging **privacy-enhancing technologies** for the Nation.*[41]

*Near-Term Action Plan, **US Cyberspace Policy Review**, USOW, 2009*

Information self-determination is a type of "Privacy Enhancing Technology". PET's are explicitly addressed by the US Cyberspace Policy Review with regard to IdMS. As IdMS are a type of CKMS, enhancing NIST SP800-130 to include privacy enhancing technologies would support this agenda.

PET is not currently addressed by the draft NIST SP800-130.

In a global-scale CKMS, **a user-centric focus would empower every entity (organisation, enterprise, business, individual, …) directly touched by the CKMS cross-cutting visibility** into data relating to them system-wide. A security breach of key material relating to them from one organisation using the CKMS should be visible to all dependent stake holders, down to the individual. If a global-scale CKMS is integrated with a global-scale IdMS, then each user has a 'single portal' into their 'portfolio of identity credentials', and complete visibility about which organisations have a relationship with data associated with that person.

User centricity (stakeholder centricity) helps empower and protect the legitimate interests of all stake holders. User centricity helps hold all parties equally accountable to each other.

The NIST SP800-130 should be expanded to require CKMS designs to specify what privacy enhancing technologies they use, in what ways they are user centric, and in what ways it empowers all dependent stake-holders to interact with the CKMS.

---

[40] SecurIST Advisory Board. Recommendations for a Security and Dependability Research Framework: from Security and Dependability by Central Command and Control to Security and Dependability by Empowerment. Deliverable 3.0, SecurIST EU-FP6-004547, Jan. 2007. https://web.archive.org/web/20130624085831/http://www.securitytaskforce.eu/dmdocuments/securist_ab_recommendations_issue_v3_0.pdf

[41] USOWH. Cyberspace policy review: Assuring a trusted and resilient information and communications infrastructure (may 26, 2009). United States, Office of the White House.

22. New proposed feature: “Runtime CKMS Risk Assessment and Management System”

New proposed feature: The CKMS should be able to dynamically generate an actionable “known risks report” based on the current/proposed configuration of the system and all known security rules. The report should include an ordered list of risks, for each risk[42]:

- component at risk
- name of risk (is there a well defined CERT advisory for the risk?)
- risk severity
- probability of risk occurring within 24 hours, 1 week, 1 month, 1 year, 2 years, 4 years, 8 years (graph)
- risk description, publications describing the risk/attack
  - keys and components that may be exposed if the risk materialises
  - number of stake holders / dependants that may be exposed if the risk materialises
  - ...
- risk mitigation automatically in place by the system, if any
- risk mitigation in human procedural systems in place?
- recommended actions to further mitigate the risk

The CKMS should provide an online knowledge base that can be explored to assist the operational management of the CKMS and also to assist with the evaluation of a proposed CKMS design. The known risk report system might be based on cert.org’s OCTAVE (Operationally critical threat, asset and vulnerability evaluation)[43] or another process more appropriate if one is known. The CKMS risk management system should probably tie into NIST Security Content Automation Protocol[44] initiatives.

The risk assessment and management system should be comprehensive and upgradable. Comprehensive coverage demonstrates that the CKMS designer is aware of the attack space against the CKMS design. It should also indicate what counter-measures have been taken to the known risks (with citations where available on who said that was a good countermeasure). Equally importantly it also provides the administrators of the system with “expert knowledge” and “situational awareness” to actively explore and respond to the most important risks in order.

If possible this risk assessment engine should be tied into the policy requirements engine. In this way advisories regarding security risks as a result of policies can be made visible to the operator.

Example advisories might include:

- “You have 103 root certificate authorities in your system, of which any one may provide a single point of trust failure for all identification and authentication operations in your CKMS deployment.”
- “25% of your users in the last 90 days have accessed the CKMS using SSL version 1.0 which has known security weaknesses .... Recommendation: systematically upgrade users to TLS 1.2”
- User centric: “Organisation X who you have authorised to manage your personal/organisation’s sensitive data employs the following weak security controls that could compromise your data privacy as follows: ...”

---

[42] For an existing requirement to address all known risks, see section 6.8.4, “Network Security Controls and Compromise Recovery”, page 53, “b) The CKMS design **shall** specify which of the mitigation techniques specified in this section were employed **for each envisioned compromise scenario.**”

[43] OCTAVE - www.cert.org/octave/

[44] http://dx.doi.org/10.6028/NIST.SP.800-126

23. Management of key material by public identifiers is absent from NIST SP800-130

We need to explore the advantages of alternative approaches like identity-based key management.

2.4.4 Leap-Ahead Technologies: Miles Smid, Orion Security Solutions, ***NIST IR-7609***

Identity based symmetric keys should be used to reduce the scale of the symmetric key distribution problem.

2.4.6 Overall Summary of the CKM Workshop: Elaine Barker, NIST, ***NIST IR-7609***

Management of key material based on public identifiers/identities (Kerberos, Identity Based Encryption[45]) is not addressed by the NIST draft SP 800-130. The specifications should be expanded to explicitly support the description of various identifier/identity based encryption schemes and their unique security properties and usability benefits.

Example risk: All data protected by some public key based identity based encryption systems [46] can be compromised if the singular master secret is exposed.

Example risk: If the private key of a corresponding public identifier is compromised, it is not possible to change the value of the key for that identifier. A new identifier is required.

[45] http://en.wikipedia.org/wiki/ID-based_encryption

[46] http://en.wikipedia.org/wiki/Boneh/Franklin_scheme

24. Additional work is required on Mirroring, Load-Sharing, Backup, Archiving and Disaster Recovery of a CKMS

The draft FIPS SP 800-130 has made strong strides to address Backup, Archiving and Disaster Recovery (section 10) of a CKMS. We feel this critically important area of the publication warrants further refinement.

Specifically we are concerned with the problems of keys and meta-data being lost, out of sync, or failing to meet policy compliance requirements.

We feel that pure online systems that are redundantly distributed over multiple sites can provide the necessary integrity, availability and durability without resorting to offline techniques. Conversely we feel the complexity of adding offline backup/archives could create more security problems (within the system design, and for stake-holders) then they solve.

We feel that the FIPS SP 800-130 document could be simplified by removing all references to offline/ semi-offline Backup Storage and Archiving of key materials. These should be replaced with a renewed focus on promoting online services that can load-share work-effort and provide overall improved availability.

See our detailed feedback on this subject in the following sections of this document:

- [page: 75] 34.6. QC: Section 3.3, page 16, Draft SP 800-130
- [page: 121] 35.26. VPQ: Section 6.2, page 24 (states)
- [page: 136] 35.59. VPQ: Section 6.4.9, page 37
- [page: 86] 34.26. QC: Section 6.4.12, page 38
- [page: 87] 34.28. QC: Section 6.4.14, page 38 (storage)
- [page: 87] 34.29. QC: Section 6.4.14, page 38 (upgrade)
- [page: 136] 35.61. VPQ: Section 6.4.15, page 38
- [page: 137] 35.62. VPQ: Section 6.4.16, page 38
- [page: 88] 34.30. QC: Section 6.4.17, page 38
- [page: 138] 35.66. VPQ: Section 6.5, page 43
- [page: 143] 35.80 VPQ: Section 6.8, page 49

25. A global-scale CKMS might want to deploy a CKMS site located in EVERY state of USA, in USA diplomatic buildings, and in other countries to ensure availability in crisis situations...

In a crisis, it is conceivable that a region such as a state, city or provence may loose wide-area Internet connectivity. This is a well recognised problem[47]. **A global-scale CKMS** needs to be designed from the onset to address the issue of intermittent or loss of WAN connectivity. Traditional SKI architectures, such as Kerberos, have known availability, scalability and security limitations[48]. However standards-based public key infrastructure (PKI) architectures also have known, but different, performance and long-term security limitations. According to the NIST website, the Identity Management Systems Research & Development Project is currently exploring hybrid SKI/PKI architectures to identify systems that exploit the strong-points of both technologies within the context of physical access control systems that may have unreliable network access.

> ***"Hybrid SKI/PKI Research***
>
> *… Symmetric algorithms have advantages over asymmetric algorithms such as RSA: cost/ performance is at least an order-of-magnitude better; keys may be much smaller; and it is believed that symmetric algorithms will be resistant to quantum cryptanalysis. …,*
>
> *well-known large-scale SKIs such as Kerberos and GSM telephony rely on the hub-and-spoke architecture.* ***Since messages between two spokes must pass through the hub, they are practical only when the hub and all pairs of communicating parties have a high degree of connectivity****."*
>
> ***"Research Goals and Method***
>
> *… If the user population is large (e.g., Federal employees and contractors) and geographically distributed,* ***a centralized hub may be infeasible****, and* ***continuous connectivity cannot be assured."***
>
> *http://www.itl.nist.gov/ITLPrograms/IDMS/external/IdMSRandD.html*

These observations hold as true for identity management systems as they do for cryptographic key management systems.

Let us consider an Integrated IdMS-CKMS solution

To increase availability we may desire a distributed decentralised IdM-CKM system, where data is redundantly stored in each of the 50 states. This means transactions operating in any state will need to be synchronised with all other states. With 50 full replication sites within the US, it is highly unlikely that offline backup/archive will be required.

Let us consider the CPU and Storage costs for this model.

With regard to CPU, the number of client CKMS transactions is **independent** of the number of sites the CKMS operates over. For each additional redundant site, there is a linear increase in the effort to maintain synchronisation. This effort can be "load-shared" across all available sites (1 site backs up to 2, those two sites backup to another 2 sites each, those 4 sites backup to another 8…). Synchronisation operations only need to be performed after a CKMS transaction has been committed, intermediate operations do not require synchronisation across sites.

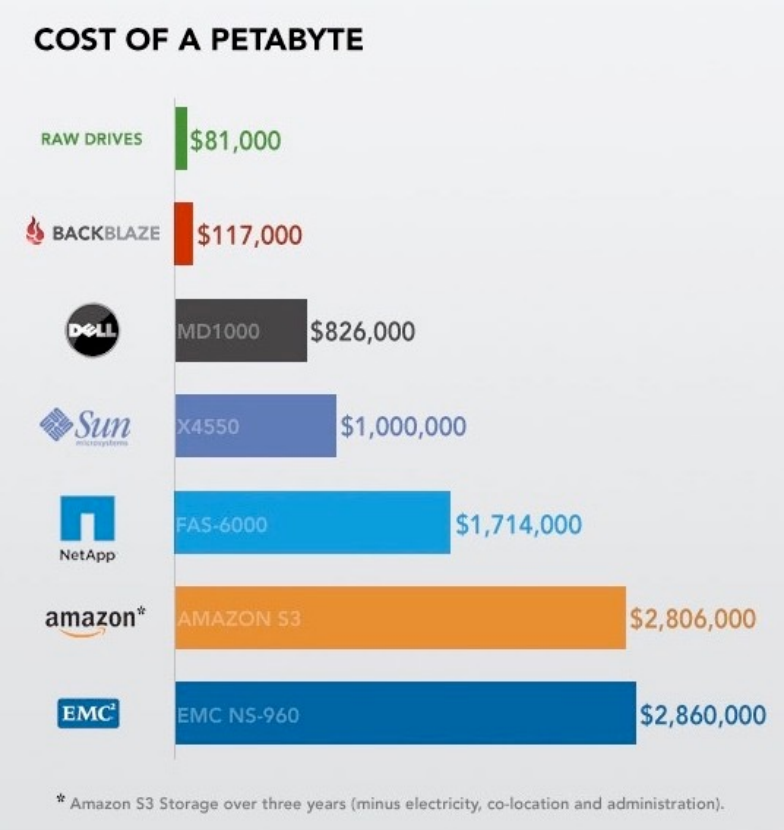
---

[47] http://www.computer.org/portal/web/csdl/doi/10.1109/WiMob.2008.103

[48] Formal Analysis Of Kerberos 5, http://dl.acm.org/citation.cfm?id=1226648

With regard to storage, storage is generally considered very cheap. We cite the various costs of a petabyte of storage from the following website[49]:

A Petabyte is 1000 terabytes of storage. The average key is 32 to 256 bytes in length. Lets say the average key and metadata in a system requires a generous 8192 bytes of storage to manage. It is possible to store 122 billion keys (with associated metadata) at a cost of USD 0.000013532c per key per site (assuming the DELL cost of two petabytes in a RAID1 configuration). With 50 sites with RAID 1 mirroring at each site, it is still only costs 0.000676599c per key to store the key. (Clearly there are other costs associated with running the system with regard to power, communications, personnel, physical site, …) 122 billion keys is probably just a little short of servicing the world's 6 billion people (Could support ~20 keys per person on average). Clearly increasing storage capacity is not a problem.



With regard to emergency services, as each state IdM-CKMS site will already need to support a very large number of keys, it is possible to establish pair-wise unique keys to be enrolled with all CRITICAL SERVICES before a crisis.

With regard to recovery after a catastrophe: the effort to re-synchronise a site after natural disaster could be load-shared across the other (*n*-1) full replications sites. The overall system performance due to the loss of a single site out of 50 due to a natural or man-made catastrophe would be negligible at around **2%**. Alternatively, if we take one of the remaining 49 sites offline to perform disk-to-disk replication, the overall system impact is still only **4%**.

In the case that the CKMS design requires "primary and secondary" sets of keys (see section 6.8.6 "Personnel Compromise Recovery"), and thus corresponding primary and secondary sites, it may be possible to partition the 50 CKMS sites distributed across the states of USA into 25 primary and 25 secondary CKMS sites. Emergency availability would not be diminished as it doesn't matter if the local state site is primary or secondary (all devices are enrolled into both).

Let us also consider a case-use of Physical Access Control Systems. A simplified IdMS-CKMS edge node could be run in each building on a LAN, enabling extremely rapid, low-latency identification and authentication of regular employees. When presented with an unfamiliar government issued card the edge system communicates with the back-end system over the Internet (or falling back to SMS over GSM or SMS over Satellite) to authenticate the smart card. In this way "availability", "scalability" and "survivability" are maintained. This edge CKMS could be implemented **using low-cost smart cards as HSM** (100s of $, not 10,000s of $ in HSM hardware)

The NIST Framework should address these case-use requirements and require a CKMS design to outline how it supports them. For example, how does a CKMS ensure 'locality of key materials and policy' by region, ensuring if international network communication fails emergency responders can communicate….

[49] http://www.smallnetbuilder.com/content/view/30922/79/

26. Temporary increased compatibility at the cost of lower security during times of crisis

> *NSA would like to have some interoperability among high-assurance government devices and commercial off-the-shelf devices, especially for emergency situations, such as 9/11 and hurricane Katrina.*
> *...*
> *NSA wants to support wider audiences of users, including FEMA, allies, charities, State governments, and emergency first-responders.*
> *...*
> *Steven Ranzini, University Bank ... discussed first-responder emergency access to patient data (i.e., the EMT doesn't know where the emergency victim's healthcare data are located when the responder first starts to look for it) in unplanned circumstances, which caused special security provisions to be needed.*
> *…*
> *(Elaine Barker) Prepare for emergency access to keys. … must be scalable and enhance interoperability in time of emergency.*
>
> *Cryptographic Key Management Workshop Summary, NIST Interagency Report 7609*

Participants of the CKM Workshop requested improved support for emergency situations to support emergency first responders. We feel the current NIST CKM design specifications need to address this in various ways.

In the same way a HSM may have "privileged" (access to key material not otherwise accessible) and "normal" modes of operation, a CKMS requires an "emergency responder" mode of operation that can be selectively enabled for certain domains. (For example in the case of a global scale CKMS two different states of America, and a third state in Europe may be in crisis mode, and all other states and regions in normal mode. The CKMS needs to support relaxed security and improved interoperability during these times, FOR THOSE REGIONS, for select authorised parties (emergency first responders), without compromising the security operations of other regions.

Furthermore, special audit mechanisms must be in place to ensure that:

- keys used during emergencies are updated after the emergency (to reduce risk exposure)
- the ability to perform post behavioural analysis on key access to identify and follow up abuses within the system

RECOMMENDATION: that NIST consider the possibility for the US NSTIC project, with it's focus on interoperability, to be the forum to address the requirement for increased compatibility and relaxed access controls for only certain parties/components within a CKMS during times of crisis and feed its results into the draft CKM Framework.

27. Improving internal security by checking the consistency of public key certificates assertions over resources

Various defensive strategies should be considered by a CKMS design. **RECOMMENDATION: that one of the CKMS goals be to ensure data/certificates/permissions are internally consistent**

For example, correlating which certificates exist for a given Universal Resource Identifier/Property, so that it is possible to detect if more than one root certificate authority has issued an equivalent certificate into the system. Occurrence of overlapping assertions should be investigated resulting in the creation of a policy rule for this specific instance. This may require sending a notification to the owner of the resource querying them as to the knowledge of the two certificates from different root certificate authorities. This can be used to detect and mitigate the system wide single-point of trust failure in the civilian public key infrastructure in some contexts.

28. Explicit support for different classes of devices

A large CKMS will probably need to support a variety of devices with a wide variation in capabilities.

Variations include but are not limited to:

- devices that are online 24h/7d (servers), periodically only 8h/5d (work computers).
- devices that can establish communications with the CKMS, but not the other way round
  - a device behind a NAT firewall
- devices that can only be communicated with via a proxy agent. (no internet protocol address)
  - See Marc Massar's IEEE KMS 2010 presentation, "Key Management In Hostile Environments"
  - See Petrina Gillman's IEEE KMS 2010 presentation, "National Security Agency Perspective on Key Management"
- devices that do not maintain trusted time sources (no battery, no reliable time source) such as:
  - certain types of RFID,
  - smart cards,
  - desktop computers (can i trust my time-source, do i update my clock remotely against authoritative time source?
- devices may be capable of different algorithms
  - a non-programmable PKI-token will require a device profile outlining the capabilities of that device. A CKMS should NOT issue key material of a type that the device cannot process.

The CKMS may need to know "what" type of device it is communicating with, so it can adjust it's behaviour to maintain policy compliance.

For example, certain classes of devices may be able to autonomously disable certificates because they have a trusted time-source. In other cases, the central CKMS may need to notify the device explicitly that it should disable the certificate it is subscribed with.

RECOMMENDATION: that the SP 800-130 should require CKMS designs to address the issues that arise from the range of devices supported.

29. Explicit support for event subscribers

A CKMS design may need to interact with other technical and human security systems. For example, behavioural analysis security engines may need notification on access to various key materials. See the immediately preceding section for other examples.

The NIST draft SP 800-130 talks about "revocation mechanisms" and the need for a design to specify them. However it is likely that a CKMS may employ one or more comprehensive notification mechanisms beyond Certificate Revocation.

RECOMMENDATION: that the Framework takes into account that notification mechanism may occur throughout the CKMS design, and that revocation mechanisms are but one potentially specialised type of notification mechanism.

30. Explicit comprehensive time-zone support, that can be revised over time

RECOMMENDATION: That the Framework should require a CKMS design to outline how it manages time-zones internally. Certain policies may require operations to be performed "8:00 am first Monday of the month" to coincide with human business processes at different company office locations around the world…

Time-zone information is not static, and countries occasionally change time-zone information. Respecting day light savings or not (+/-1 hour), etc.

31. Explicit support for anonymous connections to the CKMS

In RFID applications (such as e-passport), it may be necessary to contact a back-end CKM server, WITHOUT exposing the identity of the device to unauthorised parties monitoring the communications path. The requirements for privacy preserving connections must be considered.

RECOMMENDATION: that the Framework requires a CKMS design to express if and when privacy preserving or anonymising is required and applied within the design, and how it satisfies requirements.

32. Explicit support for authenticated devices facilitating a trusted path for accessing a CKMS for certain domains of information

A CKMS server may need to authorise 'terminals' that are forwarding requests on behalf of credit cards, e-Passports, National ID, or other tokens.

See US NSTIC for case-use scenario (A woman requests medical information from the hospital that her husband has recently visited) on page 15 of their draft report[50] for information on authenticating trusted platforms that then perform additional secure operations.

RECOMMENDATION: that the Framework requires a CKMS to specify how it manages authorised terminals as authorised access points for relaying communications for enrolled CKMS devices.

---

[50] US DHS and others. DRAFT National Strategy for Trusted Identities in Cyberspace. United States Department of Homeland Security, June 2010. Available at http://www.dhs.gov/xlibrary/assets/ns_tic.pdf

33. Explicit support for managing biometric key material

Even when we look at biometric methods for identification and authentication to control access to critical functions, we're still dependent on cryptographic functions for protecting the integrity of the biometrics.

William C. "Curt" Barker - NIST IR-7609

A CKMS should only allow authorized users to have access to stored symmetric and private keys. Thus, a CKMS should have some type of access control system (ACS). The ACS may be as simple as requiring a password or cryptographic key from the authorized user of the key, or it may make use of biometric authentication techniques.

6.5 Cryptographic Key and Metadata Security: In Storage, page 43, NIST draft SP 800-130

The NIST draft SP 800-130 talks about the use of biometrics [51] as part of physical access controls to protect key material, **but does not discuss the need for a CKMS design to specify how it manages biometric data.**

There are many security concerns with managing biometric secrets[52] and these may require special handling. If a CKMS design must handle biometric material it may also have to satisfy special legal requirements[53].

RECOMMENDATION: that the Framework requires a CKMS design to express if and when biometric data will be used in the CKMS, and how that data will be managed.

[51] http://en.wikipedia.org/wiki/Biometrics

[52] http://www.bromba.com/knowhow/temppriv.htm

[53] https://web.archive.org/web/20120913184351/http://www.sersc.org/journals/IJBSBT/vol2_no1/2.pdf

Part 10:
Questions and suggestions regarding compliant CKMS design requirements

34. Questions and suggestions regarding compliant CKMS designs

34.1. QC: Section 3.1, page 15, Draft SP 800-130

Original Requirement:

*The CKMS design **shall** specify its goals with respect to the communications networks*

Questions:

This requirement warrants greater detail.

RECOMMENDATION: that the Framework also requires that a conforming CKMS design document must spell out the networks it has been targeted for, t all relevant including specific network requirements, and why they are required in the CKMS. Examples include but are not limited to if and when a design MUST have UDP/IP support in it's clients/behind a firewall, if and when a CKMS design must have certain ports open, and if and when a CKMS design requires the client to open a server socket and receive "on-demand" connections from the CKMS server for push driven revocation notification.

34.2. QC: Section 3.1, page 15, Draft SP 800-130

RECOMMENDED Requirement:

*The CKMS design **shall** specify what expectations and requirements it can reasonably make on the Network Service Providers to achieve availability and survivability during times of crisis.*

Observation:

Internet pioneer Larry Roberts says *"The Internet's original design for 'best effort' service critically needs improvement. With Anagran's Flow Management installed, response time, throughput speed, quality and fairness can be greatly improved while virtually eliminating delay jitter, random packet loss, and substantially reducing delay".*

To support “mission-critical operations” during disasters, the CKMS may require the Internet / Network to have end-to-end Quality of Service functionality between certain links. Technologies such as Anagran’s Total Bandwidth Management solution[54] to upgrade Internet Protocol Services as a bump in the wire deployment may be essential to providing necessary communication assurance levels in times of crisis.

34.3. QC: Section 3.1, page 15, Draft SP 800-130

Requirement:

*The CKMS design **shall** specify the applications that it will support.*

Questions:

It "will" support or "does" support?

What is an "application"? Do you mean binary executable (Microsoft Office) or case use, or something else?

[54] https://web.archive.org/web/20111107080114/http://www.anagran.com/ , A Larry Roberts Company.

34.4. QC: Section 3.1, page 15, Draft SP 800-130

Requirement:

> *The CKMS design **shall** describe anticipated number of users and the responsibilities that the CKMS places on them.*

Questions:

A CKMS system may be capable of supporting various different "number of users" depending on the exact hardware configuration and how it is deployed and used at any given time. Can the Framework provide guidance on how the CKMS design document should address this flexibility with regard to describing various configurations from different perspectives or by different parties e.g. a vendor and an end user?

e.g.
Vendor Perspective: Depending on the number of hardware security modules, our system can easily scale to accommodate X to XXX users.

Client Perspective: our CKMS design requires X number of users initially, increasing at a rate of Y% year over year, to an anticipated maximum of Z users. We require certain performance characteristics to be met or exceeded at all times. We anticipate we will need to incrementally invest in additional hardware over the operational life time of the CKMS to accommodate the anticipated growth in our user base.

Should the Framework cross reference the above requirement back to section 3.3. using text such as "The description of anticipated number of users shall be framed in the context of WorkLoad Scalability, see section 3.3"

34.5. QC: Section 3.2, page 15, Draft SP 800-130

Requirement:

> *Ease of use provisions of a CKMS **should** assure that:*
> *a) ...*

Suggested additional text:

e) It is difficult/impossible to activate insecure configurations, thereby ensuring that if a CKMS is operational, it is also secure. This may require support for a 2 phase configuration process where the user begins the configuration process, makes various changes that may temporarily result in an insecure or non-operational configuration, submit configuration for security check, user selects abort or revise/commit based on feedback of the CKMS.

f) Complexity and choice is hidden from users that do not have the ability to make an educated decision regarding their choice. [users are not provided the opportunity to make decisions that should be made by trained/authorised system administrators][55].

g) The system pro-actively defends the users and owner of the CKMS by ensuring that it is not possible for users to "override" security checks. e.g., the error message: "the certificate has expired for this identity" should not provide the end-user the option to ignore / disregard the security violation.

h) the system remains secure even during user blunder/errors.

Should this requirement in the Framework also include a cross reference link to 12.2.2 (user interface design guidelines)?

---

[55] See Jay Jacobs, "Updating Shanon's Maxim", IEEE KMS 2010, "The Enemy knows the system, where the allies do not".

34.6. QC: Section 3.3, page 16, Draft SP 800-130

Requirement:

> *The CKMS design **shall** specify the performance characteristics of the CKMS, including average and peak workloads handled, and peak and average response times.*
>
> *The CKMS design **shall** specify the extent to which the CKMS can be scaled to meet workload demands beyond peak workload. This specification **shall** be in terms of additional workload, response times for the workload, and cost.*

Question:

By "performance characteristics" what functions are we measuring the performance of?
Which functions are performance critical and require documenting?

Does an enterprise / global-scale CKMS need to be designed as a real-time system[56] to satisfy client requirement specifications? (Particularly in the face of hardware failure which may require a CKMS operation to roll-back and execute that operation again on a different hardware module.)

Is there a notion of "Quality of Service" and "Service Level Guarantees" for certain users of the system? e.g. emergency responders granted higher priority over low-priority enterprise key rolling operations. Is there a notion that CKMS read requests are given higher priority to write requests (as is often the case with high performance file systems)?

We would argue that measuring the performance of recovery (see section 6.5 Cryptographic Key and Metadata Security: In Storage) is critical, even though this should only occur rarely. The performance of the system should be considered with regard to the ability to "recover requested operations on demand" faster than infrequently used key materials. This may require the backup system to group keys and metadata in different categories to facilitate responsive recovery (particularly in the context of sequential optimised tape access). Can the CKMS system support long-lived asynchronous pending requests, so that the CKMS 'pushes' the answer back to requesters as soon as it is available?

We should definitely measure best, average and worst case performance for simple individual CKMS operations under various "states" of the system. e.g., are we 100% operational, has a primary CKMS site gone down, are we simultaneously mirroring to another site, are nodes of the back-end CKMS (primary, secondary, tertiary site) disconnected from each other? ...And so on.

Question:

Is there a NIST measurements publication that would be useful for providing guidance on how the performance characteristics should be reported?

Question:

Can we include a "cost" metric? Does a CKMS product/proposal scale approximately at linear cost (after a certain ramp-up stage) or does it become exponentially expensive and then reach a max performance rating that can't be exceeded?

---

[56] See section 7. "Survivability of Time-Critical Systems", in the report "A Roadmap for Cybersecurity Research", DHS Science and Technology Directorate, Nov. 2009. Available at https://www.dhs.gov/sites/default/files/publications/CSD-DHS-Cybersecurity-Roadmap.pdf

For example, from the perspective of a vendor, I can envisage a CKMS system, with more-or-less same source code, mapped onto hardware in different ways depending on the scalability required:

Entry Level: a cluster of java card version 3 tokens[57] used as HSM

Mid Level: 4 network attached HSM providing database and mirroring services within the HSM

High Level: a massive cluster of network attached HSM with support infrastructure such as dedicated databases, replication services, and so on. The Entry/Mid level systems might be deployable as "proxy"/"edge" nodes for the globally scalable system to support performance improvements.

Clearly each level has it's basic up-front cost and then incremental costs for scaling to meet certain varying requirements (are we more concerned about raw key storage, access speeds, survivability, …)

34.7. QC: Section 3.3, page 16, Draft SP 800-130

Requirement:

*A CKMS **should** have the ability to rapidly replace compromised keys (both asymmetric and symmetric) and the ability to notify the relying parties (those who make use of the key) of compromise/revocation. Compromised Key Lists (CKL), Certificate Revocation Lists (CRL) (see [RFC 5280]), White Lists, Query White Lists, and the Online Certificate Status Protocol (OCSP) (see [RFC 2560]) are examples of mechanisms in use today. Each mechanism has its benefits and drawbacks.*

RECOMMENDED additional requirements:

A CKMS design **shall** specify what Revocation systems it uses.

A CKMS design **shall** specify the performance, timeliness characteristics of each revocation system.

A CKMS design **shall** specify if it has the capability to track which dependent devices have acknowledged a revocation operation. (Do we know if revocation has been applied, who is still vulnerable?)

A CKMS design **shall** report all known vulnerabilities against each revocation mechanism.

A CKMS design **shall** advise what counter measures, if any, are taken to mitigate known vulnerabilities, along with their effectiveness and for what communities they work. *(e.g. solution may be limited to working with Federal PKI Bridge, but not Civilian PKI infrastructure).*

A CKMS design **shall** advise on the assurance level of the revocation system and for what applications it is suitable for.

e.g. A known published weakness of OSCP is that it is a "Best Effort" system that can be trivially disabled by an adversary[58]. See here[59] for a description of other known security problems with OSCP.

---

[57] http://www.oracle.com/technetwork/java/embedded/javacard/overview/index.html Connected Edition 3.0.2 (Dec 2009).

[58] http://www.thoughtcrime.org/papers/ocsp-attack.pdf

[59] Gutmann, P. Engineering Security. (draft book), Dec. 2009.
Available at http://www.cs.auckland.ac.nz/~pgut001/pubs/book.pdf

34.8. QC: Section 3.4, page 16, Draft SP 800-130

Requirement:

A CKMS's key revocation notification mechanism(s) shall be designed based on the following considerations:

*a) Relying party requirements for **timeliness** of revocation information;*

*b) Relying party **computing** and **communication limitations**; and*

c) Infrastructure cost considerations.

Question:

What if a CKMS key revocation notification system **HAS NOT** been designed based on the criteria? Are you sure you want to prevent a vendor from producing an otherwise compliant document that describes the current state of their CKMS design?

34.9. QC: Section 3.5, page 17, Draft SP 800-130

Requirement:

*The CKMS design **shall** specify the specific COTS products used in the CKMS.*

*The CKMS design **shall** specify which security functions are performed by COTS products.*

*The CKMS design **shall** specify how COTS products are configured and augmented to meet the CKMS goals.*

RECOMMENDED additional requirements:

The CKMS design **shall** specify the known range of insider attacks originating from within EACH component, including each instance of a COTS product. The CKMS design **shall** specify the level of effectiveness of the anticipated attack and if the performance and availability of the system is reduced, and by how much.

The CKMS design **shall** outline the strategy for mitigating anticipated insider attacks on each component, including each instance of a COTS product.

For each insider attack, the CKMS design **shall** determine the ability of the system to detect the attack and where and how it can be detected. The CKMS design **shall** specify the level of effectiveness of each counter measure and if the performance and availability of the system is reduced, and by how much.

The CKMS design **shall** specify which anticipatable insider attacks cannot be addressed by the system.

34.10. QC: Proposed New Section: "How has the CKMS been designed to provide evidence against a hostile expert?"

Observation:

The CKMS design may be required to enforce access controls, enforce policies and employ audit controls.

RECOMMENDED additional requirement:

The CKMS design **shall** specify how the CKMS has been designed and certified to provide evidence on the assumption that the evidence will be examined in detail by a hostile expert[60].

This is probably a very complex issue that will need to be addressed by a team of security and legal experts at the onset of a design. For example, the CKMS might need to demonstrate it upholds certain non-repudiation properties for evidence to be admissible from the CKMS in court.

34.11. QC: Proposed New Section: "Runtime System Risk Assessment Management System"

See "22. New Proposed Feature: "Runtime System Risk Assessment Management System" [page 63].

34.12. QC: Proposed New Section: "Assessment and mandatory disclosure of all single point of (trust/security) failures"

Due to the significance of 'single points of failure' to the security, availability, and integrity of a CKMS and all it's dependents, the Framework should require a comprehensive report disclosing all known component wide, deployment wide and global system wide single points of failure and mitigation actions taken within the CKMS.

The risk assessment management system described above should also identify all single-point of failures, including paying particular attention to the existence of "CKM system-wide single point of failures". These system wide (global) SPOF may arise from the IDMS used to support the CKMS, reliance on a single HSM vendor throughout a design, etc..

[60] Anderson, R. J. "Liability and computer security: Nine principles." In ESORICS 94: Proceedings of the Third European Symposium on Research in Computer Security (London, UK, Nov. 1994), vol. 875 of LNCS, Springer-Verlag, pp. 231–245.

34.13. QC: Proposed New Section: "Multilateral Security and the protection of the legitimate interests of all stake holders within the CKMS design"

Multilateral security considers different and possibly conflicting security requirements of different parties and strives to balance these requirements [61]. To quote section 1 of that paper, with our emphasis in bold:

> *A lot of early security approaches are based on the assumption that it is quite clear who has to be protected against whom. E.g. the Trusted Computer Security Evaluation Criteria (TCSEC, [USA_DoD 1985]) focus very much on the protection of system owners and operators against external attackers and misbehaving internal users.* ***Protecting users against operators is not considered to be a major issue.***
>
> *Later criteria like the Information Technology Security Evaluation Criteria (ITSEC, [CEC 1991]) have expanded the scope of the TCSEC, but the following example illustrates that user protection still was not much in the focus. In an ITSEC evaluation a function for the selective logging of activities of individual users was classified as a non-critical mechanism that did not need evaluation. In the opinion of the evaluators, failure of this mechanism would not create weaknesses because if the function was not active, the activities of all users were logged [Corbett 1992]. From the operator point of view no real security risk existed, because no audit data would be lost – only perhaps more data than planned would be collected.* ***However, from the users' point of view this is a considerable risk, because excessive logging and the resulting data can lead to substantial dangers for users.***
>
> *Early security approaches, especially in the TCSEC, assume that a security policy can definitively describe which actions are authorized. Consequently to maintain a secure state the policy only has to be enforced by a secure and trusted entity.*
>
> *Clean cuts like these do not really apply when several parties with different and maybe conflicting interests are involved, as it happens in networks like telephone systems or the Internet.*

These observations concerning the need to protect the legitimate interests of all stake-holders, including the relative weaker stake holders, become increasingly critical in international global-scale CKMS design. The Framework should require a CKMS to comply with a multilateral security model and identify potentially competing and/or conflicting interests within the scope of the CKMS and how those interests will be protected and, if a the CKMS prioritises one parties interests at any time over another, a justification of when this might be permitted, how and why, and for how long; audit logging and behavioural analysis during such a period may be mandatory to protect all legitimate stake holder interests.

---

[61] Kai Rannenberg, "Multilateral Security", Microsoft Research, Cambridge
http://csrc.nist.gov/nissc/2000/proceedings/papers/202ra.pdf

34.14. QC: Proposed New Section: "All stake holders must be held equally accountable in a CKMS design"

In addition to protecting the legitimate interests of all stake holders, a CKMS design **shall** hold all parties equally accountable. This must include "software developers", "COTS vendors" and "management".

We propose that a CKMS design **shall** specify all mechanisms to hold all stake holders accountable to each other, either directly, or indirectly (such as by enabling users to make intelligent/informed requests about information managed by the CKMS under Freedom Of Information Act, or supplying evidence to legal proceedings).

According to Ross Anderson "Many (security) designers fail to realise that most security failures occur as a result of application and management blunders".

Let us consider the EMV protocol. EMV is the dominant protocol used for smart card payments worldwide, with over 730 million cards in circulation. The EMV protocols have been criticised due to architectural decisions that shift liability away from the banks and towards the merchant[62].

For example, several vulnerabilities have been found in the support for EMV secure messaging[63]. These attacks are significant because they show that the EMV protocol has not mitigated the risks of abuse by bank programmers at operations centres, and by exploiting this weakness insider attack there can rapidly undermine the system.

This is a serious concern. Celent, a research and advisory firm for financial institutions, estimates that approximately 60 percent of bank fraud cases where a data breach or theft of funds has occurred are the work of an insider[64]. Unfortunately, employees and contractors who access financial institution systems during the course of work know the system better than anyone else and they are better positioned to exploit the systems' vulnerabilities.

The problem of insider attacks is exasperated when the security systems has been designed to shift liability. Drawing results from [65]: Frequently, banks deny fraud victims a refund, asserting that a card cannot be used without the correct PIN, and concluding that the customer must be grossly negligent or lying. The consequence of this type of liability shifting is that the negative impact of insider attacks is born by outsiders. The self-correcting feedback mechanisms have been undermined [Think Cyber-Economics].

Just recently in 2010 it has been comprehensively demonstrated that, "Chip and PIN is broken"[66]. To quote Ross Anderson: "Merchants will be none too pleased either; the system no longer protects their interests but only those of the issuing bank."

RECOMMENDATION: If we are to achieve a significant step forward in CKMS design, then the Framework should require a CKMS design to address the issues of accountability and management of insider attacks.

---

[62] Ross J Anderson, "Liability and Computer Security: Nine Principles", http://www.formation.jussieu.fr/ars/2000-2001/UNIX/cours/5/COMPLEMENTS/DOC/why-cryptosystems-fail/liability.pdf

[63] B. Adida, et al. "On the security of EMV Secure Messaging", http://www.cl.cam.ac.uk/~mkb23/research/EMV-Secure-Messaging.pdf

[64] Celent, "Internal Fraud: Big Brother Needs New Glasses," October 2008, http://celent.com/reports/internal-fraud-big-brother-needs-new-glasses

[65] http://www.lightbluetouchpaper.org/2010/02/11/chip-and-pin-is-broken/

[66] S. Murdoch, S. Drimer, R Anderson, M. Bond, "Chip and PIN is Broken", Uni of Cambridge, http://www.cl.cam.ac.uk/research/security/projects/banking/nopin/oakland10chipbroken.pdf

34.15. QC: Proposed New Requirement: "Outline of all defense-in-depth strategies that have been employed in a CKMS design"

RECOMMENDED NEW REQUIREMENT: A CKMS design shall provide a brief description for all defense-in-depth strategies present in the design, with sub sections on "People", "Technology", and "Operations".

-- See http://www.nsa.gov/ia/_files/support/defenseindepth.pdf

34.16. QC: Section 4.3, page 20, Draft SP 800-130

Requirement:

*The CKMS design **shall** specify all types of CKMS Security Policy that it is designed to support and enforce.*

*The CKMS design **shall** specify the conditions under which keys and their related metadata may be shared by two or more entities and the security mechanisms that will be used to provide the protection required by the CKMS Security Policy.*

*The CKMS design **shall** specify how the CKMS Security Policy is to be implemented and enforced by the CKMS (e.g., the mechanisms used to provide the protection required by the policy).*

*The CKMS design **shall** specify the methods (e.g., tables, relational data structures, formal specification languages) to be used to express the CKMS Security Policy requirements.*

RECOMMENDATION: Additional requirements:

The CKMS design **shall** specify if it can, and when and how it can, support the management of key material that must be co-operatively managed by two mutually suspicious autonomous organisation (e.g. communication between Government agencies of 2 different countries; 2 competing companies that must collaborate on a contract etc) when both require (perhaps by law) some degree of command and control. How does the CKMS design manage the boundaries and conditions of distributed ownership/control.

Requirement:

*The CKMS design **shall** specify how the automated portions of the CKMS Security Policy are expressed in an unambiguous tabular form or a formal language (e.g., XML or ASN.1), such that an automated security system (e.g., table driven or syntax-directed software mechanisms) in the CKMS can enforce them*

Observation:

It is important that the semantic meaning of security policies is clear.
See section "12. Binary and Semantic Interoperability" in this feedback document. [page 42]

34.17. QC: Section 5.11, page 22, Draft SP 800-130

Requirement:

> *f) The CKMS design **shall** specify how individual accountability is enforced.*

RECOMMENDED additional requirement:

> The CKMS design **shall** specify how organisational accountability is enforced with regard to stake-holders dependent on the correct and secure operation of the system.

34.18. QC: Section 5.11, page 23, Draft SP 800-130

Requirement:

> *g) The CKMS design shall specify the collection and storage of "audit-able" events in order to ascribe security-relevant actions to individuals or roles.*

RECOMMENDED additional requirements:

> The CKMS design **shall** specify how all stake holders dependent on auditable events (such as security breach) are notified throughout the systems. In an Enterprise system this may require automatic notification to an oversight body (security breach notification laws) including notification down to individuals who do not have direct access to the CKMS (person on the street).
>
> The CKMS design **shall** specify what Security Breach Notification Laws it is compliant with, and support the ability to 'mass update' policies on CKM to be compliant with new laws as they emerge or change. (eg. security breach laws being instituted in states that did not previously have such laws).
>
> In the context of a global-scale CKMS, ideally the system should support opt-in "auto-enforcement" for all registered commercial entities employing the use of the system who are subject to various laws. (A Delaware company would then benefit from auto-notification of security breaches by the CKMS as administrators report the event, where as administrators in a company in another state may not have an appropriate body to inform, …). All organisations could benefit from "advisories" generated by the CKMS risk management system informing them of possible legal obligations as they change and evolve ("heads up, you may want to seek legal guidance on this law x with regard to your policies on key Y").

34.19. QC: Section 6, page 23, Draft SP 800-130

Original text:

> *Cryptographic Keys and Metadata*

Comments:

> Following are a few requirements as requested by Anthony Stieber of Wells Fargo from his IEEE KMS 2010 presentation.
> I have rewritten the requests in the affirmative.
>
> * We need to be able to create own keys
> * We need to be able to renew/replace keys
> * We need to be able to renew/replace keys without major downtime
> * Need to be able to store enough keys
> * Need to be able to manage enough keys
> * Need to scale without high administrative effort
> * Need to be able to recover from compromise
> * Need the ability to export all key material out of CKMS and import into another different CKMS, potentially from a different vendor.

Comments:

> Synaptic Labs adds to Anthony Stieber's list by noting the need to manage bulk changes of meta-data on key material based on legal and policy changes within an international system. A significant amount of effort in the CKMS design needs to be allocated to finding efficient methods to maintain legal and policy requirements on potentially millions to several billions of keys. [For example a view on the meta-data may need to be a combination of meta data specific to the key, and further meta data dynamically determined as relevant to that key by analysing the first meta data with regard to current laws known by the system.]

Comments:

> Does NIST have a standard for implementing Proxy Re-encryption[67] or similar constructions that support the conversion of cipher text from one key to ciphertext of another key?
>
> This could be particularly important for rolling symmetric keys of encrypted data without exposing clear text.
>
> See this paper[68] and this paper[69] for the use of double encryption with interleaved keys as used in ad-hoc mesh networks which could be trivially adapted for secure re-encryption of archived data without exposing cleartext. Such a scheme would require the co-operation of at least two independently managed HSM.
> Read old ciphertext -> HSM 1 (rekey 1) -> HSM 2 (rekey 2) -> store rekeyed ciphertext

---

[67] G. Ateniese, "Improved Proxy re-encryption schemes with applications to secure distribute storage", http://eprint.iacr.org/2005/028.pdf

[68] C.Castelluccia et a., Authenticated Interleaved Encryption and its application to WSN, submitted to IEEE Infocom 2007 (July 2006)

[69] M. Goodrich, "Leap-frog packet linking and diverse key distributions for improved integrity in network broadcasts," in IEEE Security and Privacy, May 2005

34.20. QC: Section 6.3.2, page 33, Draft SP 800-130

Original text:

> *The CKMS **shall** specify any exceptions to the key states and transitions that apply to asymmetric keys.*

Suggested amended text:

> The CKMS **shall** specify the full behaviour of key states and transitions as they apply to both symmetric and asymmetric keys. This may be achieved with a first "generic state transition diagram" with high level description, along with detailed refinements of the description for symmetric and asymmetric key exchanges. Additional detailed refinements may need to exist if there are key types that differ in key states and transition behaviour from the previous mentioned two.

34.21. QC: Section 6.4, page 34, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify the key and metadata management functions to be implemented and supported.*

Suggestions:

> What about requiring the adding of an entry to an audit log and time-stamping?
>
> How can the Framework address scenarios that require synchronisation/consensus-checking across several autonomous CKMS providers redundantly processing the same transaction on behalf of a client? (Consider Diffie-Hellman-Lamport symmetric key exchange technologies[70] with *m* key distribution centers acting together, where the trust is distributed over the *m* KDC.)

[70] Diffie, W., and Hellman, M. E. Multiuser cryptographic techniques. In AFIPS '76: Proceedings of the June 7-10, 1976, national computer conference and exposition (New York, NY, USA, June 1976), ACM, pp. 109–112.
Available at http://doi.acm.org/10.1145/1499799.1499815.

34.22. QC: Section 6.4, page 34, Draft SP 800-130

Original text:

> *The CKMS design shall identify the integrity, confidentiality, source authentication, and source authorization services applied to each key and metadata management function implemented by the CKMS.*

RECOMMENDED additional requirements:

> The CKMS design **shall** identify the known risks and attack vectors that apply to each key and metadata management function implemented by the CKMS, and identify what countermeasures are taken (if any).
>
> The CKMS design **shall** perform a Safety Integrity Level risk analysis (ensure all operations fail safely and securely) on each key and metadata function implemented by the CKMS, with particular emphasis required on the correct operation if the CKMS is a distributed decentralised CKMS design.

34.23. QC: Section 6.4, page 34, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify the process for owner registration including the process for associating keys with owners.*

Suggested additional requirement:

> "The CKMS design **shall** specify the cryptographic technologies and human procedures used for identity management and enrolment, their security ratings, and what the known risks are."

34.24. QC: Section 6.4.4, page 35, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify how (e.g., manually or automatically based on the deactivation date-time) each key type is deactivated.*

Suggested additional requirement:

> "The CKMS design **shall** describe the performance of this operation, ensuring that adjustments to the deactivation date (forwards or backwards in time) can be performed rapidly across all active keys managed by the CKMS"
>
> For example, SHA-1 may become weaker much faster than anticipated, requiring every SHA-1 key in the CKMS to have its deactivation data moved closer to the present moment.

34.25. QC: Section 6.4.11, page 37, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify the circumstances under which bound metadata can be modified.*

Suggested additional requirement:

> The CKMS design **shall** specify how and under what circumstances a modify metadata command can be undone. (e.g., all key state / material stored for 90 days, allowing mass unauthorised modification to be 'instantly' recovered. This may be used to undo a malicious configuration change).
>
> The CKMS design shall specify how an “undo” operation is managed by ALL devices dependent on that metadata. (The master CKM device and dependent CKMS devices must maintain synchronisation).

34.26. QC: Section 6.4.12, page 38, Draft SP 800-130

Original text:

> *The CKMS design shall specify the circumstances under which the metadata bound with key can be deleted.*

Suggested additional requirement:

> The CKMS design **shall** specify if and how its constituent HSM's destroy key material and associated metadata under Tamper detection, and how the system remains "available and secure" if one HSM is tampered with. See comments below for section 6.4.13

34.27. QC: Section 6.4.12, page 38, Draft SP 800-130

Original text:

> *The CKMS design shall specify the technique used to delete bound metadata.*

Suggested additional requirements:

> The CKMS design **shall** specify the techniques used to delete bound metadata where-ever present in a (potentially multi-site) CKMS deployment. This may involve deleting the metadata from hot standby, tape archives, enrolled devices deployed in the field, and so on.
>
> The CKMS design **shall** specify how long deleted metadata is stored before actual zeroisation occurs under normal operating conditions.
>
> The CKMS design **shall** specify what mechanisms are in place to monitor the progression of the deletion cycle. (delete in all active HSM, deleted in 90% of active tokens, delete in 0% of archived tape stores. ETA for 100% deletion based on tape storage based on routine cycling is X years…)
>
> This section should explicitly cross reference back to section 6.4.9

34.28. QC: Section 6.4.14, page 38, Draft SP 800-130 (storage)

Original text:

> *Operational key storage involves placing a key in local storage for use during its cryptographic period without making a copy. Keys **should** be either physically or cryptographically protected when in storage*

Suggested amended text:

> Keys should be
>
> a) physically protected,
>
> b) cryptographically protected, or
>
> c) both physically and cryptographically protected when in storage.

Other relevant issues that should be addressed in the Framework:

> How are operational keys stored in local storage destroyed rapidly on tamper detection?
> How are keys in archive destroyed rapidly on HSM tamper detection or physical premises compromise?

34.29. QC: Section 6.4.14, page 38, Draft SP 800-130 (upgrade)

Original text:

> *The CKMS design shall specify how, where, and the circumstances under which keys and their bound metadata are archived.*

Suggested additional requirement:

> The CKMS design shall specify how the periodic upgrading of archived key material to a stronger cipher is achieved securely.

34.30. QC: Section 6.4.17, page 38, Draft SP 800-130

Original text:

> ***6.4.17 Key Retrieval***
>
> *Obtaining a cryptographic key from storage, a backup facility, or an archive is considered retrieval if done during normal CKMS operation. If there has been an environmental or man-made disaster and the key cannot be normally retrieved and used, the key may have to be recovered by special means or with special permission (see Section 6.4.19). The CKMS security policy should state the conditions under which a key may be retrieved normally.*
>
> *The CKMS design **shall** specify how, and the circumstances under which, keys and their bound metadata may be retrieved from a key database storage facility.*

Suggested additional requirements:

> {Note: A well designed CKMS that has hot and warm standby systems may not require the use of key retrieval.}
>
> The CKMS design **shall** specify how, if at all, the system avoids the need to perform key retrieval due to environmental or man-made disaster at one or more CKMS sites.
>
> The CKMS design **shall** specify the performance characteristics of key retrieval, what risks are associated, such as the loss of synchronisation or state between keys in the central key store and devices and data dependent on those keys. This should be query-able using the "online risk management systems". (That is, if we loose CKMS site x due to a catastrophic natural disaster, how much key material / metadata would be out of sync RIGHT NOW, how many critical systems would be impacted, and what is the financial cost to the dependent systems. This may require the system to make a logical snap-shot of both sites and compare the states.)

34.31. QC: Section 6.4.18, page 39, Draft SP 800-130

Original text:

> ***6.4.18 Key Escrow***
>
> ..
> *The CKMS design **shall** specify the security policy (e.g., continuous two-person control) for the protection of escrowed keys.*
>
> *The CKMS design **shall** specify how the security policy is implemented during the key escrow, i.e., how the confidentiality and multi-party control requirements are implemented during transport and storage of the escrowed key.*

Suggested additional requirements:

> The CKMS design **shall** specify any compartmentalisation techniques which permit key escrow to be performed within isolated domains co-existing in one CKMS system.
>
> Example case uses:
>
> - Sales escrow is independently managed to R&D escrow
> - Inter-organisation escrow, where each organisation controls their own escrow, while using the same common CKMS infrastructure, ensuring one organisation cannot compromise the security of the other organisation.

Suggested additional requirements:

> The CKMS **shall** identify how replication of key material is securely performed when supporting the exchange of key material between two devices enrolled within the CKMS.
>
> Following is an example key for a message/data element securely delivered to all sales staff and escrow agent assigned to sales within that enterprise with the assistance of a key translation centre.
>
> - key material is generated from a first source device and labelled with a target identifier (sales@company.com)
> - the key material and target identifier is received by the CKMS {from alice@company.com, to: sales@company.com, key: value }
> - the CKMS is responsible for identifying which accounts (including escrow accounts) the key material generated from the first source device must be relayed to.
>   - bob@company.com, gary@company.com, escrow_sales@company.com
> - relaying the key material and the original authenticated source field to each identified target.

34.32. QC: Section 6.4.18, page 39, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify how the security policy is implemented during the key escrow, i.e., how the confidentiality and multi-party control requirements are implemented during transport and storage of the escrowed key.*

Suggested additional requirements:

> The CKMS design **shall** specify how all dependent parties are sent audit logs/notification on escrow events, supporting accountability and the detection of unauthorised/abuse of escrow operations.
>
> In the context of on-demand key escrow requests, the CKMS design **shall** specify how a veto process may be used to protect against indiscriminate escrow. e.g. an authorised watch-dog entity (within or outside the Enterprise) may be given opportunity to conditionally prevent escrow operations if they observe localised or systematic abuse. This would require stipulation of a veto period before the content of escrow operations are released to the requester.

34.33. QC: Section 6.4.18, page 39, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify how, and the circumstances under which, keys and their bound metadata can be established.*

Suggested additional requirements:

> The CKMS design **shall** specify all protocols for key establishment.
>
> The CKMS design **shall** specify the risks and known attack vectors against those key-establishment protocols and list any techniques used to mitigate the known attacks. (Defence in depth).

34.34. QC: Section 6.4.21, page 40, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify how the integrity and confidentiality (if necessary) of the entered keys and bound metadata are protected and validated upon entry.*

Suggested additional requirements:

> The CKMS design shall specify the electronic and physical security mechanisms employed to protect key entry.
>
> The CKMS design shall specify the known risks and how these risks can be mitigated (like each party taking in their own trusted keyboard, using TEMPEST certified equipment, ...)

34.35. QC: Section 6.4.22, page 40, Draft SP 800-130

Original text:

> *If a private key, symmetric key, or confidential metadata is output in plaintext form, the CKMS design shall specify how the calling entity is authenticated before the key is provided.*

Suggested additional requirements:

> The CKMS design **shall** specify the fields and type of information recorded in the audit trail generated by key output routines. (it's schema)

34.36. QC: Proposed New Section: "Validate that the Public Key Certificate is well formed."

Observation:

> Ensure that known attacks "such as the presence of 0x00 in certificate fields" are detected. See [71] and [72]. Malformed public key certificates should be rejected BEFORE attempting to validate the public key certificate path chain.

Suggested requirement:

> The CKMS design **shall** specify how, where, and the circumstances under which, a public key certificate is determined to be well-formed. The CKMS design **shall** specify what known attacks against public key certificates are mitigated by the checking mechanisms employed by the CKMS design.

34.37. QC: Proposed New Section: "Compare the Public Key Certificate against prior information known within the system."

Suggested requirements:

> Check to see if this public key certificate is the one "normally" used for a given resource.
>
> Validate for multiple assertions of the same resource by different certificate authorities to check for existence of possible attacks.
>
> Validate the certificate authority is authorised to issue certificates regarding a resource. (This is conceptually different to validating the certificate path back to a root certificate).
>
> Also see the section 27 entitled "Improving internal security by checking the consistency of public key certificates assertions over resources" in this document. [page 69]

---

[71] Moxie Marlinspike, "Null Prefix Attacks Against SSL/TLS Certificates," Blackhat 2009, (last visited August 2009). http://www.blackhat.com/presentations/bh-usa-09/MARLINSPIKE/BHUSA09-Marlinspike-DefeatSSL-PAPER1.pdf

[72] Dan Kaminsky, "Something About Network Security," Blackhat 2009, (The video is no longer available online)

34.38. QC: Section 6.4.28, page 41, Draft SP 800-130

Original text:

> ***6.4.28 Validate Possession of Private Key***
>
> *...*
> *The CKMS design **shall** specify how, and the circumstances under which, possession of private keys and their bound metadata can be validated.*

Observation:

> Can we clarify how Key Confirmation is different to Proof of Possession (PoP) in the text? (See section 6.6.3 Key Confirmation)
>
> Some people have questioned the value of PoP technologies [73]. I could not find many papers talking about what attacks it prevents. Can the Framework point to a comprehensive (NIST?) paper that outlines clearly why, when, and how PoP should be used?

Additional text:

> According to Burt Kaliski [74]:
>
> - *If CA isn't sure of private-key possession, then it might issue a certificate with an adversary's name and someone else's public key*
>   - *Relying parties may make flawed assumptions as a result*
> - *If verifier isn't sure a signature public key is valid, then can't be sure either that signatures are hard to forge*
>   - *Dishonest signer may be able to repudiate on this basis*
> - *If sender isn't sure an encryption public key is valid, then can't be sure either that ciphertexts are hard to decrypt*
>   - *And in Diffie-Hellman, if encryption public key is combined with sender's long-term private key, sender's private key may be at risk due to small subgroup attacks*
>
> The CKMS design **shall** specify how the "testing" of possession of a private key cannot be used to force a device to authorise a transaction it did not intend. [a test should prevent possession of a key signing the digest of a message: "I authorise the transfer of $x from my account to account y"]
>
> See:
>
> *G.3.3 Active Authentication (Data Traces)*
>
> *In the challenge-response protocol used for Active Authentication, the chip signs a bit string that has been chosen more or less randomly by the inspection system. If a receiving State uses the current date, time, and location to generate this bit string in an unpredictable but verifiable way (e.g. using secure hardware), a third party can be convinced afterwards that the signer was at a certain date and time at a certain location.*
>
> *MTRD, "PKI for Machine Readable Travel Documents offering ICC Read-Only Access" 2004*

---

[73] Nokia, "On the usefulness of proof-of-possession", https://web.archive.org/web/20100610052242/http://middleware.internet2.edu/pki03/presentations/pop.pdf

[74] Burt Kaliski, et al. "Public Key Validity and Private Key Possession: Recent Developments", https://web.archive.org/web/*/http://www.rsa.com/rsalabs/staff/bios/bkaliski/publications/other/kaliski-public-key-validity-rsa-2004j.ppt

34.39. QC: Section 6.4.29, page 42, Draft SP 800-130

Original text:

> *The CKMS design shall specify all cryptographic functions that are supported.*

Suggested additional requirements:

> The CKMS design **shall** specify all cryptographic functions (including a break down on what modes of operation, with what key lengths, for how much data) that are supported.
>
> The CKMS design **shall** specify a comprehensive list of cryptographic papers and attacks known at date of publication against the cryptographic functions used by the design. (This may need to be an open wiki-like collaborative project to reduce repetition efforts by vendors).
>
> The CKMS designer **shall** list the attacks and evaluate if they pose a threat to the usage of that function in their CKMS. This will require at least a short statement and support evidence. (e.g. our CKMS only uses SHA-1 in the HMAC protocol. We only rely on SHA-1 in pre-image resistant modes of operation. General industry consensus is that using SHA-1 in this way is adequate in the short to mid term, see NIST paper…)
>
> (These requirements support section 11, section 11.2, and section 12.5 of SP800-130)

34.40. QC: Proposed New Section: "Cryptographic Key and Metadata Security: Within a HSM"

We propose a new section to be inserted around the original text:

> **6.5 Cryptographic Key and Metadata Security: In Storage**

Observation:

> Sometimes a HSM will have "Local Master Keys" which are stored on the chip in NVM. These LMK are used to encrypt ALL other keys processed within the HMS. On tamper detection the LMK can be zeroised rapidly, thereby invalidating all other keys on HSM.

Proposed requirement:

> The CMKS design **shall** specify the methods used to protect keys within a HSM. This may be as simple as pointing to the appropriate section in a FIPS 140-2 report.

34.41. QC: Section 6.6.1, page 44, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify the methods used to protect the confidentiality of symmetric and private keys during their transport.*

Suggested additional requirements:

> The CKMS design **shall** specify each key-transport scheme supported by the CKMS. (Sync with 6.6.2 key agreement). The CKMS design **shall** specify the methods, range of security ratings, and the risks present in each of the methods used.

34.42. QC: Section 6.6.1, page 44, Draft SP 800-130

Original text:

> *The CKMS design shall specify the methods used to protect the integrity of transported keys and how they are implemented to recover from **detected errors**.*

Suggested additional requirement:

> The CKMS design **shall** specify if an audit log entry is created so that abnormal number of corrupted keys can be detected.

34.43. QC: Section 6.6.1, page 45, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify if/how the identity of the key sender is authenticated to the receiver of transported keying material.*

Suggested additional requirement:

> The CKMS design **shall** specify the known risks and attacks regarding the identity of the key sender and what countermeasures, if any, are taken. (Defense in depth.)

34.44. QC: Proposed New Section: "The CKMS Identity management (support) system"

To be inserted around the original text:

6.7.1 The Access Control System (ACS)

Observation:

Identity management is intrinsically linked with CKMS. **The requirements/capabilities of IdM support need to be spelt out in the CKMS design.**

For example, a CKMS design may simultaneously support "multiple" IdM systems at different levels of assurance. OpenID for entry-level authentication, native enrolled smart card token based authentication, Federal PKI support, ….

The functionality, flexibility, usability and performance of the IdMS within a CKMS design is critical (think how can a global-scale CKMS design manage the access control requirements on a billion enrolled users over the entire life cycle of the CKMS deployment…)

We recommend significantly expanding the scope and detail on ACS within the NIST SP 800-130 document.

34.45. QC: Section 6.7.1, page 47, Draft SP 800-130

Original text:

*The CKMS design **shall** specify the capabilities of its ACS to accommodate, implement, and enforce various information protection policies.*

Suggested additional requirements:

The CKMS design **shall** specify how the ACS scales in a manageable way to accommodate it's rated users.

The CKMS design **shall** specify if it permits external sources outside of the CKMS proper to have veto control on certain transaction. (e.g. how does the ACS support programatic integration with other environments/systems that may be authorised to perform 'run-time' veto requirements. ACS operations may need to employ a publisher-subscriber 2-phase transaction based model.)

Are there any NIST/US Government standards for describing access control rules/state machines?

Comment:

We would like to see the scope/level of detail for section 6.7.1 expanded/increased.

34.46. QC: Section 6.7.5, page 48, Draft SP 800-130

Original text:

> *For each (n, k) key splitting system used, the CKMS design **shall** specify the rationale (logic, mathematics) as to why any k of the n components can form the key, but k-1 of the components provide no information about the key.*

Suggested additional requirements:

For each (n, k) key splitting system used, the CKMS design **shall** specify if the value of the key that is split into k parts is at any time known in full by the HSM device or some other party (e.g., does each party perform some part of the computation on their own trusted device within their own isolated address space, or are all parts supplied to one device which in turn exposes the symmetric key to the HSM, resulting in a potential single point of failure - the value of the key is no longer split). An accompanying risk analysis should be performed with regard to insider attacks (COTS vendor attacks), particularly if one 'trusted' device is permitted knowledge of all key material.

34.47. QC: Section 6.8, page 49, Draft SP 800-130

Original text:

> *When a CKMS compromise is detected*
>
> *a) The compromise should be evaluated to determine its cause and scope*
>
> *b) Compromise mitigation measures should be instituted to minimize the amount of data exposed*
>
> *c) Appropriate corrective measures should be instituted to prevent the reoccurrence of the compromise*
>
> *d) The CKMS should be returned to secure operating state.*

Observation:

> Our concern is that a "deliberately injected" CKMS vulnerability may become a strategic "denial of service attack".
>
> Let us consider a compromise in the context of a global scale CKMS. Should we take the entire global system down because of a compromise found in an edge CKMS site?
>
> If a vendor inserts a small but exploitable weakness into the system, should we shut down the entire CKMS?
>
> It's a complex question that stresses prevention and containment at the ARCHITECTURAL level being critical to prevent failures that could compromise the viability of the entire CKMS deployment. In addition to defensive coding, we require fault-tolerant and intruder-tolerant architectures. See the section number <> "Possibility of adopting the functional safety integrity within NIST SP 800-130?" in this document.

34.48. QC: Section 6.8.1, page 50, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify the range of acceptable cryptoperiods or usage limits of each type of key used by the system.*
>
> *(The usage of keys may be limited based on a criterion such as the amount of data processed using the key or the number of times the algorithm was initialized using the key.)*

Suggested additional requirements:

> The CKMS design **shall** specify how the range of acceptable cryptoperiods or usage limits of each type of key is enforced by the system. (e.g., tracking usage in the key meta-data).
>
> The CKMS design **shall** specify how the enforcement of key usage policies affects performance.
>
> The CKMS design **shall** specify what slack/tolerances/accomodations are present to improve performance (e.g. lazy update of key usage restrictions in a distributed CKMS design) and to support legacy devices where it is not possible to enforce or retrieve accurate measurement of key usage within an application (e.g. instead of measuring using how much ciphertext was generated, control the number of times a a key was issued for the purpose of initialising a cipher.).

34.49. QC: Section 6.8.2, page 51, Draft SP 800-130

Original text:

> *1) The CKMS design **shall** describe how physical access to cryptographic module contents is restricted to authorized entities.*
>
> *2) The CKMS design **shall** specify the approach to be used to recover from a cryptographic module compromise.*
>
> *3) The CKMS design **shall** identify any modules that are not vulnerable to non-physically invasive attacks.*
>
> *4) The CKMS design **shall** describe what non-invasive attacks are mitigated by the cryptographic modules used by the system and reference a description of how the mitigation is performed.*

Suggested additional requirements:

> The CKMS design **shall** describe how security of the entrusted key material is maintained in the event of the compromise of a HSM in the CKMS. (The CKMS design shall specify if there is a layer of defence that ensures that logical attacks mounted from within one HSM in the CKMS ecosystem are capable of compromising a user key) -- *This requirement is different from points 3 and 4 quoted above which assume attacks are prevented from compromising a system component.*
>
> The CKMS design **shall** specify if and how it employs hardware diversity in cryptographic modules to mitigate attacks.
>
> The CKMS design **shall** specify how the level of expose due to a compromised HSM that cannot enforce strict physical access control mechanisms (e.g. a user's smart card) is controlled. (Think credit-card systems that perform behavioural analysis to detect unusual spending patterns).

34.50. QC: Section 6.8.5, page 54, Draft SP 800-130

Original text:

> *For each role that is implemented, the CKMS design **shall** specify the training required for the CKMS security procedures.*

Question:

> Is this a check-list of actions that must be performed,
>
> or
>
> does the CKMS have to come with a comprehensive training manual?

34.51. QC: Section 6.8.6, page 54, Draft SP 800-130

Original text:

> *6.8.6 Personnel Compromise Recovery*
>
> *A security failure is any event that compromises the secure functioning of the CKMS. A CKMS **should** be designed to*
>
> *a) minimize the ability of humans to cause security failures,*
>
> *b) determine who or what caused the security failure, and*
>
> *c) mitigate the negative consequences of the failure.*

Suggested additional requirement:

> x) minimise the ability of humans to hide their actions that led to a security failure

34.52. QC: Section 6.8.6, page 55, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify procedures and design features for recovering from the compromise of personnel security involving accidental and intentional breaches of security.*

Proposed revision of text:

> CKMS design **shall** specify procedures and design features for recovering from the compromise of personnel security involving accidental, **negligent** and intentional breaches of security.

Comments/questions:

> Is the original design requirement quoted above asking the system to support different types of recovery process depending on the perceived nature of the breach?
>
> - Accidental (nobody's perfect)
> - Negligent (failure to apply rules; maybe under influence of behaviour modifying drugs [prescription or otherwise], etc)
> - Opportunistic (limited abuse of power for self gain)
> - Conspiratorial (potentially well thought out, wide-ranging problem)
>
> If yes, should there be a requirement that a recovery process started under one assumption (for instance based on the assumption that the compromise was an accident) should be revised / restarted / adjusted if the evidence began to suggest the compromise could be conspiratorial in nature?

34.53. QC: Section 7, page 56, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify all external interfaces to all applications and other CKMS in order to support **easy** replacement or update of external components (devices, software modules).*

Questions:

Is the requirement asking that for **every** component, there should be enough description of the protocol and state transitions for another vendor to clean-room[75] write and replace that component?

Does this mean a vendor **shall** expose all formal specifications and technical documents that they used to develop and maintain the system?

Should a vendor also expose the full source code, as this is the **definitive** finite state machine description of the CKMS component? (At least one **security** company provides this level of transparency in their commercial products[76]. Other companies, such as Sun/Oracle release most (all?) of their source code to their Java platform while retaining certain intellectual property rights.)

What constitutes “easy”? (Porting/recompiling source code might be considered relatively easy. This may be particularly important if the COTS vendor goes out of business and the software functionality of a component needs to be ported for use on hardware from a different vendor.)

Does this include "proprietary value-add extensions" over and above a minimum interoperability protocol. (That is, it is not sufficient to say: We are IPsec compliant, if you implement only a certain subset, or add extra functions)

Do patent/intellectual property rights around a component significantly impact the ability to ‘easily’ replace that component? The existence of intellectual property rights does not necessarily imply that the intellectual property dues are a proportionally significant component of the overall cost of implementing and running a CKMS deployment (hardware, software, staff, premises). In fact, in some cases the value the intellectual property adds may well reduce the overall cost of developing a system that did not license the technology. [Think power consumption costs over the life time of a system, think of savings due to a reduction in recovering from security vulnerabilities]. Certain licensing agreements may need to be in place to ensure their presence in the system is mutually beneficial for all parties involved over the life time of the CKMS deployment.

---

[75] http://en.wikipedia.org/wiki/Clean_room_design

[76] https://philzimmermann.com/EN/findpgp/

34.54. QC: Section 7, page 57, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify the physical security protections implemented by the CKMS components so that they are only accessible by authorized CKMS personnel.*

Suggested additional requirements:

> *The CKMS design **shall** specify how insider attacks by authorised/privileged CKMS personnel at one site are mitigated.*
>
> *The CKMS design **shall** specify how personnel are compartmentalised to ensure one authorised personnel acting unilaterally cannot physically compromise every site.*

34.55. QC: Section 7, page 57, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify all secure operating system requirements (including required operating system configurations) for the various CKMS components.*

Suggested additional requirements:

> *The CKMS design **shall** specify all secure hypervisor system requirements (including required hypervisor configurations) for the various CKMS components.*
>
> *The CKMS design **shall** specify all secure firmware requirements (including required firmware configurations) for the various CKMS components.*
>
> *The CKMS design **shall** specify all secure hardware root-of-trust requirements (including required configurations) for the various CKMS components.*

34.56. QC: Section 8.2.3, page 60, Draft SP 800-130

Original text:

> *In order to be effective, malware protection should be configured for the following:*
> *a) A daily scan,*

Suggested additional requirement:

> f) Once weekly the component should be taken offline and scanned by an alternate vendor's anti-virus tool. This provides increased anti-virus coverage, and stronger protection against root kits.

Question:

> Could the malware protection software become an attractive attack vector into compromising a system? The upgrade process for malware protection software could be an attractive attack vector to inject malware into the CKMS.

Question:

> What steps should be take after the discovery of the presence of virus / malware in a CKMS design to ensure system recovery? Is it enough to require checking of and following of anti-virus vendors recommendations? Do we need to revalidate the integrity of all software/firmware that could have been compromised by that virus/malware if it had executed? Should there be a protocol that accommodates different types and scales of CKMS deployment?

Observation:

> The performance of these security operations, and their impact on availability and quality of service level agreements, needs to be carefully managed. This may require specially modified versions of commercial off the shelf/open source anti-virus systems to ensure certain operational properties are maintained. (e.g. A multi-core computing platform, where anti-virus and security operations are bound to a single CPU core and there are certain disk-access / memory rate control limitations enforced.)
>
> *Antivirus software is very resource-intensive. There's been testing done by NIST (National Institute of Standards and Technologies) that showed that simply performing a virus definition update on an older control system processors can cause anywhere from a two- to a six-minute denial of service. That's just doing your daily virus definition update. There have been cases where installing antivirus software has shut down certain system control workstations.*
>
> *http://news.cnet.com/8301-27080_3-20004505-245.html , Joe Weiss.*
>
> **Therefore CKMS anti-virus/malware protocols and their impact on operations (availability etc) will require clarification and costing.**

1. QC: Proposed New Section: "Robust system maintenance with internal certification process"

Observation:

> Joe Weiss has observed that in some environments "upgrading to the latest security patches" can break the operation of a system. See: http://news.cnet.com/8301-27080_3-20004505-245.html
>
> For example, running the latest version of an operating system may change the underlying behaviour or compatibility with software and hardware devices.
>
> We need a way that allows new CKMS components to be inserted into a real system, and then ensure all regression testing performed flawlessly, so as to internally certify that the Primary and Secondary CKMS systems will remain operational after applying the upgrade.
>
> How is the order of component upgrades tested and checked, audited, managed, ... ??
>
> Should the Framework require CKMS designs to submit all new executables to application white-list checking[77] and also automated security vulnerability checking? [Concordia[78], Veracode[79]?]

34.57. QC: Section 9.5, page 60, Draft SP 800-130

Original text:

> *9.5 Scalability Testing*
> *…*
> *The CKMS design **shall** specify any scalability testing performed on the system.*

Suggested additional requirement:

> *The CKMS design **should** specify the measured (and projected) scalability properties of every function in the system. Certain classes of function may be grouped together under a single measurement if they have approximately equivalent performance characteristics. All performance projections **should** have an adequate level of justification.*

[77] https://web.archive.org/web/20101128215122/http://www.bit9.com/

[78] Daly et al, "Concordia: A Google for Malware", CSIIRW-6 2010.

[79] http://www.veracode.com/

34.58. QC: Section 9.6, page 65, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify the functional and security testing that was performed on the system and the results of the tests.*

Suggested additional requirements:

> *The CKMS design **shall** specify the functional safety (as distinct from generic functional) testing that was performed on the system and the results of the tests.*
>
> *The CKMS design **shall** specify the "Fuzz testing" [80] that was performed on each component in the system and the result of the tests.*
>
> *The CKMS design **shall** specify the "Fuzz testing" that was performed on the inner clear-text inputs received by a CKMS after successful authenticated decryption.*
>
> *The CKMS design **shall** disclose the number, severity and timing of faults identified through the development and operational life-cycle of the product/component/module so the maturity and stability of that part can be identified. [Does a product/component/module suffer from a perpetual chain of severe faults? Is this product ready for use in a production system? If we perform an independent code-audit, how many serious software errors are found?].*
>
> *The CKMS design **shall** disclose the nature and timing of changes to source code through the development and operational life-cycle of the product/component/module so the maturity and stability of that part can be identified. [This may be useful in a court-case to help establish if an independent code-audit is required on certain modules involved in a transaction under dispute to check for errors].*

34.59. QC: Section 10.3, page 66, Draft SP 800-130

Original text:

> *The CKMS design shall specify the minimum communications and computation redundancy needed to assure continued operation of services commensurate with the anticipated needs of users, enterprises and CKMS applications.*

Suggested additional requirements:

> The CKMS design **shall** identify the ability of the system to operate in the face of (degraded performance of full) network isolation between facilities.
>
> The CKMS design **shall** specify the abilities of the system to service transactions from both the primary and secondary site, when the communications between the primary and secondary site are temporarily offline (e.g. 1-hour, 24-hours, 48-hours, 1-week).
>
> The CKMS design **shall** specify how full re-synchronisation of the primary , secondary, ... sites occurs, including the techniques used to resolve all data-base inconsistencies. [e.g. forcing both sites to "rekey" to a new common key, …]

[80] http://en.wikipedia.org/wiki/Fuzz_testing

34.60. QC: Section 11.1.2, page 70, Draft SP 800-130

Original text:

> *11.1.2 Architectural Review*
> *The architecture review team should have expertise in cryptography, cryptographic protocols, secure system design, network security, and computer security.*

Proposed revised text:

> The architecture review team should have expertise in cryptography, cryptographic protocols, secure system design, network security, computer security, **human usability/accessibility, functional safety and distributed decentralised high availability system design.** (Do we need legal experts?)

Suggested additional requirement:

> *The CKMS **shall** specify all usability and accessibility testing that was performed to ensure the system was easy to use correctly.*
>
> [Is there some independent standard for certifying the usability and/or accessibility of a product ?]

34.61. QC: Section 11.1.2, page 70, Draft SP 800-130

Original text:

> *11.1.3 Functional and Security Testing*
>
> *Testing is typically performed before initial deployment, as part of the periodic security review, and in the event of a incremental security assessment. A variety of functional and security tests may be performed by the vendor, the information owner, or a trusted third party (see Section 9).*
>
> *The CKMS **shall** specify all testing that is required to be performed before initial deployment and specify the expected results.*

Suggested additional requirements:

> *The CKMS **shall** specify all system testing that is required to be performed as part of the periodic security review and specify the expected results.*
>
> *The CKMS **shall** specify all system self testing that is to performed routinely by the system during it's **normal operation** and specify the expected results.*

34.62. QC: Section 12.1.1, page 71, Draft SP 800-130

Original text:

> *12.1.1 Advantage of Standards*
>
> *…*
> *The CKMS design **shall** specify the federal, national, and international standards that are utilized by the CKMS and how conformance is tested for each.*

Suggested additional requirements:

> *The CKMS design **shall** specify the federal, national, and international **laws** that are observed by the CKMS implementation and how conformance is tested for each.*
>
> *The CKMS design **shall** specify which federal, national and international **laws** that should be observed by the CKMS (in the application it is designed for) but have not yet been implemented. The CKMS design **shall** advise which outstanding requirements can be manually achieved using user-configurable policies.*
>
> *The CKMS design **shall** specify if it has the ability to validate new policies associated with a key at run-time comply with the federal, national, and international **laws** that are observed by the CKMS. It will specify the capabilities and limitations of that coverage.*

34.63. QC: Section Section 12.5, page 73, Draft SP 800-130

Original text:

> *The CKMS design shall specify the expected security lifetime of each cryptographic algorithm used in the system.*

Proposed revised requirement:

> *The CKMS design shall specify the expected security lifetime of each cryptographic algorithm used in the system **against various adversaries with different security relevant capabilities.***
>
> *(This section should tie back into previous comments in section 2.1 "Rationale for Cryptographic Key Management" of* Draft SP 800-130 *).*
>
> (See also the text in section "16. Concerning security ratings" [page 51] in this document for more information.)

34.64. QC: Section Section 12.5, page 73, Draft SP 800-130

Original text:

> *b) Quantum Computing*
>
> *If large word size quantum computers could be built, then the security of integer factorization and discrete log-based public-key cryptographic algorithms would be threatened. This would be a major negative result for many CKMS which rely on these algorithms for the establishment of cryptographic keys.*

Suggested additional text:

> *It is known that it is not possible to retroactively protect the ciphertext dependent on the security of these at-risk cryptographic algorithms. This implies that all at-risk ciphertext archived by an adversary can be decrypted at will when that adversary has access to a large word size quantum computer. For this reason, it is important to ensure that the security lifetime of the algorithm used in the CKMS will cover the security lifetime of the information that it protects. Establishing the required security lifetime should involve representative consultation with all categories of stake holder (from the owner of the CKMS deployment, through to communities and individuals who entrust their private information to that CKMS deployment) and satisfy the most conservative legitimate interest.*

34.65. QC: Section Section 12.5, page 73, Draft SP 800-130

Original text:

> *Research is currently underway to find public-key algorithms that would be resistant to quantum computing (e.g., lattice-based public-key cryptography), but no widely accepted solution has yet been found.*

Quote from the US DHS:

> *Research strategies to achieve **a strong I&A architecture for the future include large-scale symmetric key infrastructures with key distribution a priori, federated systems of brokers to enable such a system to scale,** strategies for scaling symmetric creation of one-time pads, schemes of cryptography not reliant on a random oracle, and other schemes of cryptography not susceptible to attack by quantum computers (which seems possible, for example, with lattice-based cryptography).*
>
> *Page 52 of Department of Homeland Security's*
> *"A Roadmap for Cybersecurity Research". Nov. 2009.*
> *Available at http://www.cyber.st.dhs.gov/docs/DHS-Cybersecurity-Roadmap.pdf*

Suggested additional text:

> *Research is also currently underway to find scalable symmetric-key key distribution architectures that can use symmetric key algorithms that are already widely considered resistant to quantum computing (e.g. AES-256).*

Part 11:
Further observations, questions and suggestions regarding the text itself

35. Various Proposals and Questions on the text

35.1. VPQ: Section 2.1, page 11, Draft SP 800-130

Original text:

> *Cryptography is often used to protect information from unauthorized disclosure, to detect modification, and to authenticate the identities of system users.*

Observation:

> Data-centric cryptography is also used as a means for enforcing fine-grain access control to data elements, ensuring audit trails on access (on key retrieval/decryption/encryption/key updating), and for enforcing policy compliance.
>
> Also Cryptography can be used as a form of liability shifting, which should be discouraged[81].
>
> **The above original text might be expanded to cover those uses.**

35.2. VPQ: Section 2.1, page 11, Draft SP 800-130

Original Text:

> *Cryptography also provides a layer of protection for stored data (in addition to physical and computer security access controls) against insiders who may have physical and possibly logical (e.g., system administrator) access to, but not the authorization to know or modify, the information.*

Suggested additional text:

> "In this way, cryptography is routinely used to enable the use of transportation (Internet) and storage (cloud storage) facilities that are not owned by the organisation. The use of encryption between computer units (such as chip-to-chip encryption, or encryption between HSM) can reduce the exploitable attack area exposed to 'trusted' insiders."

35.3. VPQ: Section 2.1, page 11, Draft SP 800-130

Original Text:

> *This design principle is comparable to a design principle used in building safes and vaults: the designer builds the vault to a standard that would discourage the rational attacker from attempting entry;* ***the only way to open the safe is to open the safe door by trying possible combinations until the correct combination is selected.***

Proposed replacement text for the bold text above:

> **; if the designer is successful the fastest and most sensible way for an attacker to open the safe would be to systematically try all valid combinations until the correct combination was selected.**

---

[81] Ross J Anderson, "Liability and Computer Security: Nine Principles", http://www.formation.jussieu.fr/ars/2000-2001/UNIX/cours/5/COMPLEMENTS/DOC/why-cryptosystems-fail/liability.pdf

35.4. VPQ: Section 2.1, page 11, Draft SP 800-130 (continued)

Original Text:

> *Similarly, the only way to decrypt previously encrypted data (without knowledge of the correct key) is to test possible keys until eventually the correct key is used to decrypt the ciphertext to obtain the correct plaintext.*

Proposed replacement text:

> Assuming no layered defense and distributed trust mechanisms are applied, against brute force attacks, the protection provided by a safe, and the protection provided by a cryptographic algorithm, are both dependent on the number of possible combinations of their secret. Assuming the cryptographic primitive suffers from no security weaknesses, the only way to decrypt previously encrypted data (without knowledge of the correct key) is to test possible keys until eventually the correct key is used to decrypt the ciphertext to obtain the correct plaintext.

35.5. VPQ: Section 2.1, page 12, Draft SP 800-130

Original Text:

> *Other means of gaining access to the contents of the safe or to the information that has been encrypted may also exist. One can drill through the safe enclosure and one can attempt to find a short-cut method to crypt-analyze the cryptographic algorithm.*

Suggested replacement text:

> Other means of gaining access to the contents of the safe or to the information that has been encrypted may also exist. For example, one may try to listen to the sound of the combination mechanism of a safe to reduce the key search space. In the context of encryption, attackers may attempt to find a short-cut to breaking the cryptographic algorithm, by exploiting similar techniques using side-channel attacks.

35.6. VPQ: Section 2.1, page 12, Draft SP 800-130

Original Text:

> *Safe combinations and cryptographic keys both require protection.*

Proposed alternative replacement texts:

> *Safe combinations and cryptographic keys both require **physical** protection.*
>
> *Safe combinations and cryptographic keys both require **protection to be kept secret.***
>
> *Safe combinations and cryptographic keys are secrets that require appropriate protection against unauthorised parties.*

35.7. VPQ: Section 3.1, page 14, Draft SP 800-130

Original text:

> *A Framework is a description of the components (i.e., building blocks) that can be combined or used in various ways to create a "system" (e.g., a group of objects working together to perform a vital function).*

Question:

> Based on a full reading of the publication I understand that a CKMS design explicitly includes "non-computer" elements such as human procedures and the manual distribution of key material. Can the Framework flag this early on in the text in some way so that we know "components" isn't just referring to a piece of software or hardware but includes other elements?

35.8. VPQ: Section 3.1, page 15, Draft SP 800-130

Original text:

> *The CKMS designer should also study the potential users of the system. How many users will use the system for what purposes? Are the users mobile or stationary? Are the users knowledgeable of the CKMS or will it be transparent to them? Are users operating under stressful conditions where time is of the essence in getting the job done?*

Suggested additional text:

> Are the users operating in potentially mission-critical situations where certain "normal" requirements such as frequent password changing may prohibit the rapid response in a life-threatening situation[82].
>
> Flexibility in achieving appropriate security controls may need to be considered and evaluated. (This is inline with normal NIST standards with regard to selecting the choice of security controls [e.g. Do I need to have TEMPEST certified devices, or can I protect many lower cost COTS hardware in a TEMPEST enclosure], however **the CKMS design must be flexible enough to support the varying choices**).

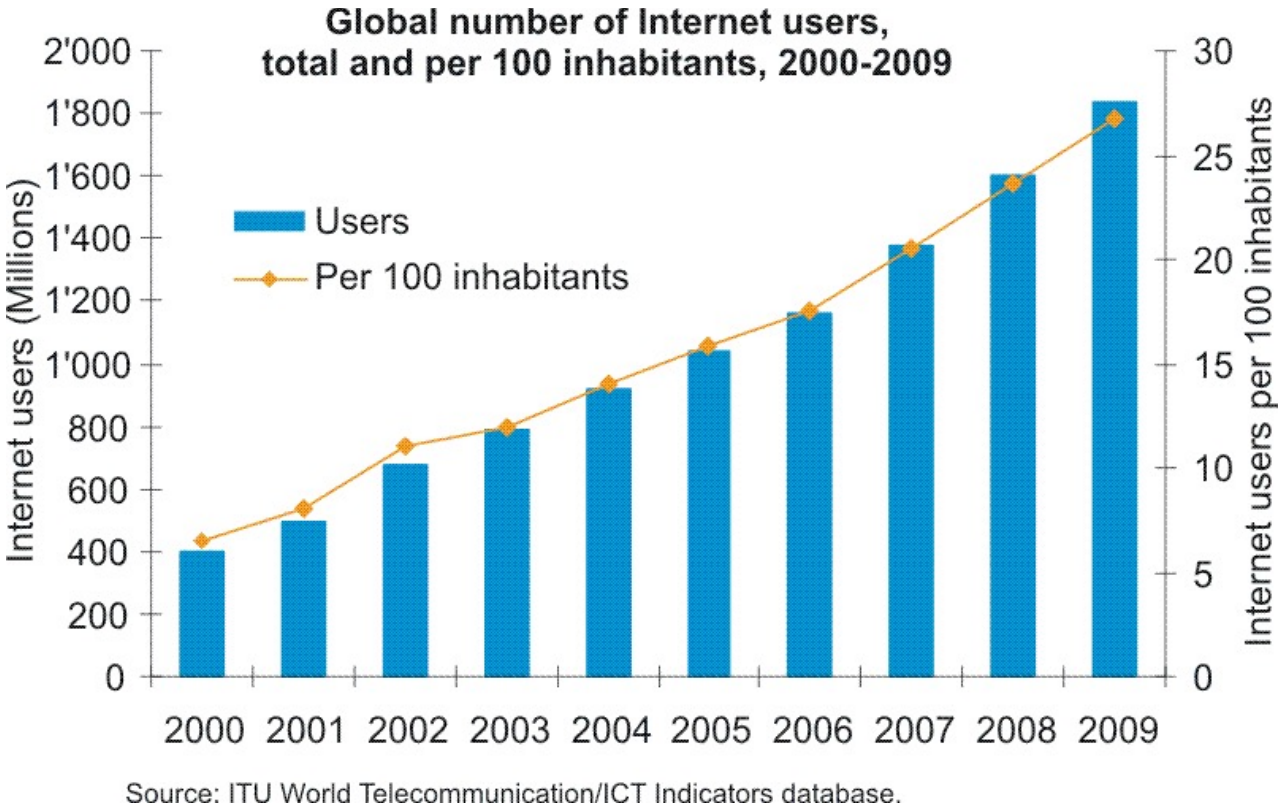
[82] Quote: "Policies as simple as requiring that default passwords be changed can be problematic. If you're in a very stressful situation, like the grid is going down or a power plant is in upset condition, it's been proven time and again that if people don't do what they're trained to do, they're going to do the wrong thing. If you force them to have a password they're not used to, they're not going to be able in a timely fashion to respond."
http://news.cnet.com/8301-27080_3-20004505-245.html , Joe Weiss.

35.9. VPQ: Section 3.3, page 16, Draft SP 800-130

Original text:

> *In the past, large key distribution centers often serviced a maximum of several thousand security subscribers. Now, **millions** of people use the Internet regularly with ever increasing demands, including new demands for keys.*

Observation:



Source: ITU World Telecommunication/ICT Indicators database.

Global scale CKMS will need to comprehensively address scalability from the onset. ITU states in 2009 approx there is **1.8 billion** Internet users world wide[83]. This is approximately 25% of the world population. 100% of the world population, without population growth is around 8 billion people.

A global-scale CKM will need to manage **at least** 8 billion enrolled users, and we may be able to assume that a system that can scale to that number should be able to scale beyond it as the population grows. Also, the designers will need to consider "how many keys per human" will be managed by the system.

Then we need to take into account the number of corporations. With a data-centric approach we need to then consider how many users they will have to manage secure connections for, and how many sensitive data elements in their databases will require unique keys.

Then we need to consider the “network of things” to support the humans and organisations. As a subset of the network of things, Eurosmart forecast[84] the existence of 20 billion Smart Secure Devices by 2020…

---

[83] http://www.itu.int/ITU-D/ict/statistics/
http://www.itu.int/ITU-D/ict/statistics/material/graphs/Internet_users_00-09.jpg

[84] https://web.archive.org/web/20111013161055/http://smartcardstrends.com/det_atc.php?idu=5403

35.10. VPQ: Section 4, page 18, Draft SP 800-130

Original Text:

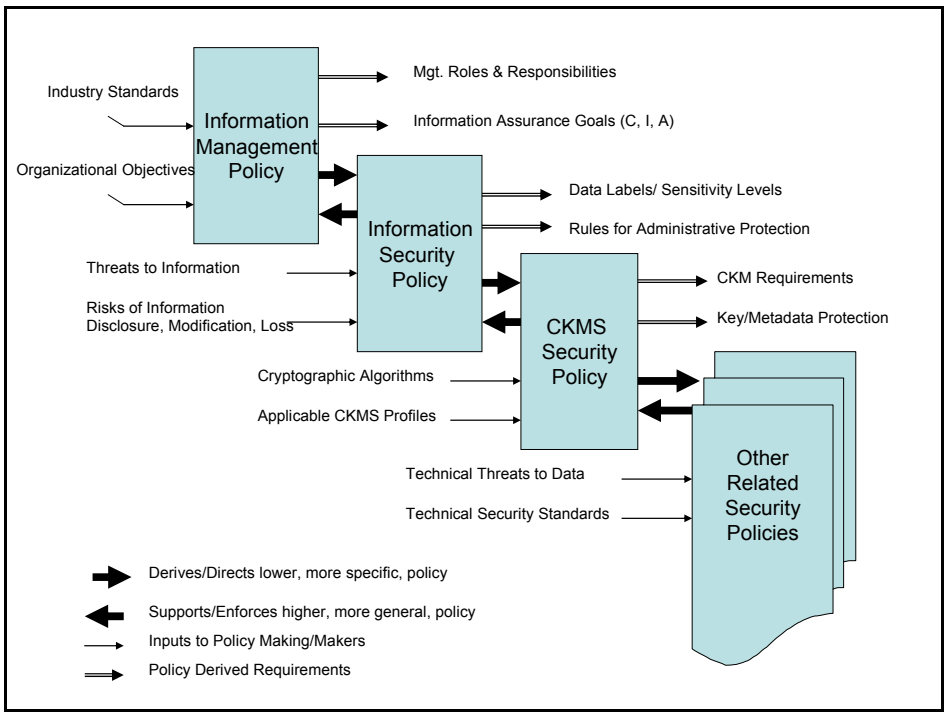


Figure 3: Related Security Policies

Question:

Where do legal requirements fit into figure 3? Is there some recommendation/advice on how the requirements as dictated by International Law (UN), various national Laws (US, European, UK, AU) are managed. e.g. if the organisational objective is to support a CKMS system that supports a corporation (or Civilian Government agency) operating in 100+ countries, how are the legal requirements that are then put on to the CKMS security policy managed? How are changes with international law managed? Is there a mechanism to test each policy within a CKMS against updated legal requirements? (How fast does it perform?) Does each organisation have to "rebuild this legal requirement framework" or is this going to be done once with a safe-harbour arrangement? **See our paper[85] discussing these issues.**

Is it possible to mitigate certain insider-attacks occurring at the policy level? For instance a CKMS design may be deliberately designed to leave open a weakness that may be exploited. (e.g. Management-level/ Privileged level technical staff have audit-free access to all information in a CKMS).

May we suggest this text: "An Organisation's information Management Policy should be independently vetted by security and legal experts that are independent of management to look for omissions and weaknesses". Is there some process where vetting may provide a safe-harbour with respect to meeting certain legal requirements? [Under the provision that all identified shortcomings are promptly correctly.]

May we suggest that the information management policy is routinely checked (every 24-48 months) and that a different organisation/group of people is used to vet it each time. [thereby increasing the number of fresh eyes looking at the guideline.]

---

[85] Synaptic Labs, "The need for the EC to fund the development of an electronic requirements management process to support the conversion of existing standards, existing policy guidelines and existing laws of several nations simultaneously in a unified requirements model that also supports national and regional variations.", media.synaptic-labs.com/pub/papers/TT/20100127-TT-D3-1b-P4.pdf

35.11. VPQ: Section 4.0 and 4.1, page 18, Draft SP 800-130

Original text:

> *(Section 4.) The CKMS must be designed in a manner that **supports the goals of the organization** that will using the CKMS. Therefore, several policies either influence, or are dependent upon, the CKMS for protecting the organization's information. Several of these policies and their relationships are depicted in Figure 3.*
>
> *...*
> *(Section 4.1) An organization's Information Management Policy specifies what information is be collected or created and how it is to be managed. The **senior executives of an organization** establish this policy using industry standards of good practices, legal requirements applicable to its information, and organizational objectives that must be achieved using the information that the organization will be collecting and creating.*

Questions:

Can we add: "and in a way that seeks to protect the legitimate interest of all stake holders (i.e. everyone that is in directly touched by the CKMS)"? Is there some notion that senior executives of an organisation should consult with representatives (tribal leaders, community leaders, widely respected organisations/ experts…) from all dependent stakeholders of the system when designing policies?

How does Data self determination and the Freedom Of Information Act tie-into CKMS designs? e.g. Making information known to stake holders on WHAT information is being managed by WHO, located WHERE, so they can submit well formed requests ASKING about that information through FOIA.

In a user-centric and data-centric environment, CKMS is used to log access to individual data elements, ... should this per-datum, per-access information also be reported to all dependent stake holders? Should this audit log then also report "who accessed, who authorised, and for what purpose, who was the result given to". "Manager X of Marketing company Y, authorised a data-mining operation to determine characteristic Z about you to give to all their current 9039 clients, to do this they accessed your CKMS protected data elements a, b, c, d, e, f, g, h, i, j, k, l, ...". Should there then also be "legal requirements" that this associated information is true, complete and honest with enforceable penalties for deceptive/ misrepresentation/inadequate reporting.

The current document focusses on a Client specifying their requirements. How should vendors that wish to offer COTS CKMS devices to an international audience show their product meet the needs of other countries and organisations ? What would the equivalent of figure 3 look like for Vendors? (Also see section 34.QC: Section 4.3, page 20, Draft SP 800-130 on page 81 of this analysis.)

How should a global-scale CKMS design be guided so it can support 1000's of different organisations goals? That is, a global-scale "CKMS as a service" provider for other organisations.

35.12. VPQ: Section 4.1, page 18, Draft SP 800-130

Original text:

> *It also specifies **what information is to be considered valuable and sensitive**, and how it is to be protected. In particular, this highest policy layer specifies what categories of information need to be protected against unauthorized disclosure, modification or destruction. These specifications thus form the foundation for an information security policy and **dictate the levels of confidentiality, integrity, and availability protection that must be provided for various categories of sensitive and valuable information.***

Observation:

> If we have different levels of confidentiality - integrity - availability, this means a security system will have a complex composite security rating assertion.
>
> It is no longer "The CKMS is 256-bit secure", it is "for data elements a, b, c we have 128-bit, for data elements d, e, f, ..."
>
> Is there any advice we can give to a compliant CKMS design specification on how to report: a) the security capabilities of the CKMS, b) the security properties of the CKMS as deployed.

35.13. VPQ: Section 4.2, page 19, Draft SP 800-130

Original text:

> *The inputs to this second layer of policy include, but are not limited to, the Information Management Policy specifications, the potential threats to the security of the organization's information, and the risks of the information to unauthorized disclosure, modification, and destruction or loss.*

Question:

> Can we explicitly include "lack of availability/responsiveness" in this sentence?

Question:

> To round out section 4.2 on information security policy, can we briefly address "information processing environment"? e.g., what data can be processed on standard computers, or wholly within the confines of programable hardware security modules, or within TEMPEST protected environments, etc.

35.14. VPQ: Section 4.3, page 19, Draft SP 800-130

Original text:

> *Inputs to this layer of policy include the selection of all cryptographic algorithms **and security techniques** to be used throughout the organization's automated information systems.*

Question:

> Does security techniques include non-cryptographic techniques based on behavioural analysis, rate limiting, and so on? Should the Framework explicitly state crypto + non-crypto security techniques?

35.15. VPQ: Section 4.3, page 19, Draft SP 800-130

Original text:

> *It is essential that the CKMS Security Policy support the goals of the organization's Information Management and Information Security Policies.* ***For example, if the Information Security Policy states that the confidentiality of the information is to be protected for up to 30 years, then the CKMS encryption algorithms and key management procedures must be selected to meet that requirement.***

Suggested replacement text:

> "For example, if the Information Security Policy states that the confidentiality of **each datum** is to be protected for up to 30 years, then the CKMS encryption algorithms, key management, **identity management procedures** and security controls in the processing environment **shall** be selected to meet that requirement".
>
> Can we also include text regarding how the "operational life cycle of a CKMS is intrinsically linked to the life cycle of the project it operates with. The operational life cycle of a key management system may or may not exceed the security lifecycle of each utterance of encrypted data it processes. For example a system may need to remain operational for 50 years, while ensuring that each utterance achieves at least 15 years security against adversaries that have access to the ciphertext, in which case either we select cryptographic primitives that remain secure for either a) the duration of the operational period of the system plus the security duration of the most sensitive utterance or b) we must explicitly **plan and budget** for the algorithms used in the system (and all dependent systems) to be upgraded at a given time in the future. The security of an algorithm and key-length is independent of the recommended operational use periods of any given instance of a key."
>
> See also section "16. Concerning Security Ratings" on page 51 of this analysis document.

35.16. VPQ: Section 4.3, page 20, Draft SP 800-130

Original text:

> *The CKMS Security Policy for a large enterprise **supporting multiple diverse organizations** must accommodate the security requirements and policies of each organization.*
>
> *This may require the protection of data having different security levels in different security domains, and may even involve processing and storing sensitive data in "mutually suspicious" domains. Organizational Information Security Policies and the CKMS Security Policies must accommodate any allowed information sharing, and the CKMS itself must be designed to help enforce how this sharing takes place.*

Observation:

> Is the document talking about a large enterprise (e.g. Government) that CONSISTS of multiple diverse organisations, or a large enterprise that must talk to autonomous organisations OTHER than itself and its subsidiaries?
>
> It appears the document is referring to managing inter-enterprise key management (that is, management of key material between two autonomous/competitive/mutually suspicious entities. This could mean CKM between government agencies of different nations).
>
> E.g. An original equipment manufacturer will have secure relationships with suppliers and merchants. Each supplier may have it's own different relationships with several OEM. How do we manage the complex CKM interrelationship requirements where each organisation may REQUIRE the ability to control and audit their own key material, and key material they have joint responsibility in maintaining.
>
> With regard to "mutually suspicious domains", one case use is symmetric key distribution ceremonies between two banks, where each bank may be responsible for managing "their part" of a 2 part secret... See Martin Fabians presentation at IEEE KMS 2010[86].

35.17. VPQ: Section 5, page 21, Draft SP 800-130

Original text:

> *One person or organization can perform multiple roles, and multiple individuals may perform a single role, but a CKMS often appoints different people or organizational components to **perform different roles for security and reliability purposes**.*

Suggested alternatives for revising the bold text above:

> "perform different roles for security and reliability purposes?" (no change)
>
> "perform different roles for security and availability purposes?"
>
> "perform different roles for security, reliability and availability purposes?"

[86] Fabian Martins, Crosscut Consulting / FIAP University "Practices and Difficulties of key management on the credit card market" (45 minutes), https://www.youtube.com/watch?v=BpcHaTVdl-g

35.18. VPQ: Section 5, page 21, Draft SP 800-130

Original text:

> *System Authority, System Administrator, System Designer, ...*

Observation:

> Roles and Responsibility is written assuming a CKMS is owned and operated by the same party. However, this may not necessarily be so: Inter Enterprise CKM, Outsourced use of "CKM as a service", … See Section 4.3, page 20, Draft SP 800-130 regarding mutually suspicious parties.

Question:

> Does the Framework need to talk about the CKMS software/hardware developer, and the need for audit trails on all code written by each programmer, involved in each component? With COTS equipment, clearly this falls outside the scope of a single Enterprise.
>
> What do we do where there is no audit trail on the software development? Should the Framework require that a few identifiable programmers "sign off" on a full independent code-audit?

35.19. VPQ: Section 5.5, page 21, Draft SP 800-130

Original text:

> *A key owner is an entity that is authorized to use a cryptographic key or key pair. For public-private key pairs, the association is typically established through a registration process. A symmetric key may have a single, specific owner or* ***may be shared by multiple owners.***

Question:

> What about split knowledge ownership of a private key in an asymmetric system? For example managing the private parts of the root asymmetric keys of certificate authorities?

35.20. VPQ: Section 5.7, page 22, Draft SP 800-130

Original text:

> *Audit Administrator*
> *An audit administrator is responsible...*

Question:

> Should a CKMS design be required to have some way to "escalate" an audit administrators activity, for example in the cases of:
>
> - Previous failures with a specific CKMS deployment / or other CKMS deployments by the same vendor
>
> - Complaints of Operator confusion due to "inaccurate" operator displays

35.21. VPQ: Section 5.9, page 22, Draft SP 800-130

Original text:

> *A key recovery agent is allowed to recover escrowed keys from storage after identity verification and authorization of the requesting entity is performed in accordance with the CKMS security policy.*

Question:

> Is a sole key recovery agent granted "system wide CKMS access" or are they constrained to domains within an CKMS, and in this or some other way, such as dual agents, limiting the damage from an insider attack by the key recovery agent.
>
> This is particularly important in international global-scale CKMS. No country will want any country to have carte blanch escrow rights on it's Government or citizens. However some countries might consider creating a co-operative escrow process for addressing large-scale criminal activities (such as arms-trafficking, international white-collar crime, …).

35.22. VPQ: Section 6.1, page 23, Draft SP 800-130

Original text:

> *Keys Types*

Comments:

> Is "symmetric data encryption key" something that would be different to "Symmetric authentication key". Normally data encryption/decryption key is different to the key used for message integrity.

Comments:

> According to private correspondence with Fabian Martins, the credit-card market systems sometimes use a "one time use symmetric transport key". This is conceptually different to a Symmetric Key Wrapping Key which is used to encrypt several keys. A CKMS may generate and consume ephemeral symmetric transport keys but they should not archive those one time use keys. These one-time use symmetric transport keys are used to encrypt zone master keys when they are split and physically couriered between two HSM.

Comments:

> Does the Framework need to specify a key type for symmetric keys that are generated and used by HSM to encrypt all their content internally, where the value of that key must NEVER leave or be shared with any other party/device?

35.23. VPQ: Section 6.2, page 24, Draft SP 800-130

Original text:

> *This section lists and describes the metadata that can be bound with the various types of keys. Key metadata is defined as information associated with a particular key that specifies the secure and appropriate usage and management of the key. The metadata that is appropriate for binding with a key should be selected by the CKMS designer based upon a number of factors including the key type, the key life cycle state, and the CKMS security policy.*

Comments:

In some cases metadata can be "independently" validated by a party other then the entity supplying the metadata into the system. E.g. Has the domain name in a certificate been registered with the company that is registered with the certificate making assertions over that domain name?

Has the CKMS validated those independent assertions, and has the user who is consuming that metadata from the CKMS checked? See also “6.2 Key Metadata”, page 28 of Draft SP 800-130: “ii. How metadata is vetted”

Comments:

Meta data appears to be envisaged as this “single logical database that is synchronised perfectly between primary and secondary sites”. What if certain key material is maintained at a central back-office location, and there is an edge HSM that is actively doing rolling keys/re-encrypting a database. Presumably the edge HSM may need to be loosely coupled with the central back-office location to ensure high-speed throughput. [I.e. it may perform key-rolling on it’s local copy of 10,000 keys very quickly and then slowly up-load the new key values back to the central key store. Is there some way to indicate a key (or group of keys) are currently being updated by a given HSM and you should go talk to it regarding the latest state for a specific query? Or should the CKMS automatically forward the request to the correct location? Can a device processing a forwarded request then communicate directly with the requester, avoiding the need for relaying?]

Comments:

Keys and their meta-data may have context specific legal applications placed on them. For instance, this key is used to protect data that must remain secure for “A period of X years till the contract ends + 7 years” starting from date Y. The meta-data may need to remain retrievable within the system until X+Y+7. We need to capture this information, and ensure that the algorithms selected ensure the ciphertext is rated to achieve this result.

35.24. VPQ: Section 6.2, page 24, Draft SP 800-130

Original text:

> *A CKMS need not bind all applicable metadata with a given key and a CKMS may not bind any metadata with some or all of the keys.*

Comments:

> Is there benefit to providing guidance that a CKMS probably should bind all applicable meta data with the key to limit exposure to unforeseen meta data attacks (security difficulties with complex systems)?
>
> [The security vulnerability that broke[87] EMV protocol was because a particular command "Verify" was not cryptographically authenticated. It's tempting to optimise a design to remove cryptography to the point where the system is exposed to unanticipated security vulnerabilities.]

35.25. VPQ: Section 6.2, page 24, Draft SP 800-130

Original text:

> *b) Key Identifier: This text string is used by the CKMS to select a specific key from a collection of keys. A key identifier is generally unique.*

Comments:

> For symmetric keys, can the key identifier be randomly generated, ensuring that randomly generated identifier is a unique identifier within the system? Is there a case-use that requires ID's to be short numbers, or can we use a 256-bit randomly generated identifiers without limitation?

35.26. VPQ: Section 6.2, page 24, Draft SP 800-130 (states)

Original text:

> *c) Key Life Cycle State: A key life cycle state is one of a set of finite states that describe the permitted conditions of a cryptographic key. Possible states of a key include: Pre-Activation; Active; Deactivated; Compromised; Destroyed; Destroyed Compromised; and Revoked. All compromised keys should be revoked.*

Comments:

> Do we need additional states of:
>
> - Archived, is that the same as deactivation, or is archived less of a life-cycle state and more about just having a backup that we can retrieve if we have to? If a CKMS system is "distributed" across multiple sites, do we need to store online metadata about how the backup can be retrieved? Is there a case use where that might be useful?
>
> - Revoking: CKMS has notified *x* out of *y* dependants on the revocation status? We may have a list of parties we need to notify regarding revocation, and we may want to know what % of the revocation notices have received confirmation receipts, as well as be able to query certain critical dependants

[87] Murdoch, S. J., Drimer, S., Anderson, R., and Bond, M. Chip and PIN is Broken. In IEEE Symposium on Security and Privacy (May 2010). http://www.cl.cam.ac.uk/~sjm217/papers/oakland10chipbroken.pdf

35.27. VPQ: Section 6.2, page 25, Draft SP 800-130 (security strength)

Original text:

j) Security Strength of the Key

Comments:

See section “16. concerning security ratings” on page 51 of this document.

VPQ: Section 6.2, page 25, Draft SP 800-130 (applications)

Original text:

l) Appropriate Applications for the Key

Comments:

Is there a standard reference list of applications we can use? We will need such a list so it is possible to ensure interoperability between CKMS (particularly for transitioning between CKMS systems). For example, does this requirement relate to the Mitre “Common Platform Enumeration”[88] and the NIST “Security Content Automation Protocol (SCAP) Version 1.0”[89] [90] in any meaningful way? If so, can this document point to these standards as an example and provide further guidance.

Do we really want to limit applications by product name and version? What about “Appropriate **type** of applications for the key”, so that it is also more abstract. Something like “key can be used for digital signatures in secure email applications”. Rather than limiting it to just S/MIME or some specific protocol which may change.

Are we going to have problems with the order of applications enabled/disabled, in the way that some Firewalls enable/disable ports according to a sequential interpreted script[91]?
e.g.

disable all ports ->
enable port 80 ->
enable port 43 when condition x, y and z are satisfied for a period of time w...

Should the Framework state: "What OPERATIONS may be performed"? .. So key exchange yes, digital signature no, ... This may already be covered by "Modes of Operation".

[88] http://cpe.mitre.org/

[89] http://dx.doi.org/10.6028/NIST.SP.800-126

[90] http://dx.doi.org/10.6028/NIST.SP.800-117

[91] http://en.wikipedia.org/wiki/Iptables

35.28. VPQ: Section 6.2, page 25, Draft SP 800-130 (continued)

Original text:

m) Security Policies Applicable to the Key:

Comments:

Is there a standard interpreted/formal language for security policy description and execution?

Is there a notion that a CKMS security policy may be software that consults other systems?

a CKMS has an application programming interface,
the CKMS stores an Enterprise Java Bean as a policy for a key,
the state of the bean maintains connections to back-end databases not normally considered part of the CKMS system (such as access to department of motor vehicle records)
...

How do we assign “generic policies” to classes of key, so we don’t have to encode the same policy over and over again for each key instance? How complex is policy compliance going to get, and how can we manage interoperability / transition between vendors?

35.29. VPQ: Section 6.2, page 26, Draft SP 800-130

Original text:

Key Access Control List (ACL)

Comments:

This section may need to specify what level of authentication assurance is required by a user in ACL (that is, it is not just sufficient to be the "person", but also to authenticate at a given assurance level. See IDABC (Interoperable Delivery of European e-government services to public Administrations, Businesses and Citizens) AAL (Authentication Assurance Levels)[92] and the and the US OMB Memorandum M-04-04, E-Authentication Guidance for Federal agencies, December 16, 2003, available at: http://www.whitehouse.gov/omb/memoranda/fy04/m04-04.pdf .

35.30. VPQ: Section 6.2, page 26, Draft SP 800-130

Original text:

Parent Key

Comments:

This is an excellent requirement. We agree that it is important that all key material be internally managed by the CKMS run-time. The ability to map inter-relationships is particularly important when considering revocation management and risk assessment.

[92] http://events.oasis-open.org/home/sites/events.oasis-open.org.home/files/Moulinos.pdf

35.31. VPQ: Section 6.2, page 26, Draft SP 800-130

Original text:

> *Key Protections:*
> *Metdata Protection:*
> *Metadata Binding Protection:*

Comments:

> Should protections indicate the "known risk" factor? i.e. there are 8 (fed pki) or 40+ (civilian) root certificate organisations, any of which can generate a certificate over the same resource ...
>
> If we captured this information maybe we would have to capture how many active CA were in the system that had authority over that name-space **at the time the operation was performed?**
>
> e.g., 1:40 <-- civilian internet where any 1 out of 40 identity providers is sufficient, 3:40 would be a split authority scheme where any 3 of the 40 identity providers must be in agreement regarding an identity.
>
> The Framework may also need a flag to indicate if the name space system is internally co-ordinated between all Identity Providers in a way that ensures only authorised identity providers can make assertions regarding a specific set of identities / mapped to specific name spaces. See here[93] for more information regarding co-ordinating root certificate authorities.

35.32. VPQ: Section 6.2, page 28, Draft SP 800-130

Original text:

> *v. Rekey date (The date-time that a key was replaced with a new key that was generated so that it is completely independent of the key being replaced.)*

Comments:

> What is meant by completely independent?
>
> - a new key may be 256-bits of entropy generated from a RNG, and that is supplied to the device in an information theoretically secure way.
> - a new key may be 256-bits of entropy generated from a RNG, and that key is wrapped using a key-encrypt-key. The security of the new key is dependent on the security of the key-encrypt-key, even thought the entropy is completely independent of the old key.
> - a new key may be 256-bits of entropy generated from a RNG, the old 256-bit key and new 256-bit key concatenated and supplied to a cryptographic hash function, where 256-bits of the digest is used as the key. This is “independent” in that there is fresh 256-bits of entropy, however it is ‘dependent’ in that the security of the key cannot be any weaker than the old key, making it a favourable construction in some circumstances depending on the attack model.
> - …

[93] Synaptic Labs, “We need to explore new distributed decentralised trust models that remove the current system-wide single point of trust failure”, NSTIC Idea Scale, (No longer available online)

35.33. VPQ: Section 6.2, page 28, Draft SP 800-130

Original text:

> *viii. Revocation date (The date-time that a key was revoked)*

Comments:

> Can we store more useful information so that certain audit operations are possible? e.g. what was the window of opportunity for an attack to take place based on exposed key material (particularly for authentication mechanisms which may require high priority notification levels). The window could be mapped as follows:
>
> Revocation Notification Date
>
> Revocation 25% Complete Date
>
> Revocation 50% Complete Date
>
> Revocation 75% Complete Date
>
> Revocation 87% Complete Date
>
> Revocation 100% Complete Date

35.34. VPQ: Section 6.2, page 28, Draft SP 800-130

Original text:

> *Revocation Reason:*

Comments:

> Is there an existing revocation standard a CKMS designer may be able to adopt regarding "standardised revocation reasons"?
>
> Maybe, something along the line of Java's Exception handling mechanism[94], where there is a base "Exception" class, and then you can inherit off that to create more specialised subclasses "Application Exception", "Runtime Exception" … But in this case you would start with "CKMS Exception" and create subclasses relevant to CKMS implementations.
>
> This would permit automated management of certain classes of exception, while still supporting detailed reporting of case-specific information with standard schemas.

[94] https://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html

35.35. VPQ: Section 6.2, page 28, Draft SP 800-130

Original text:

> *For each key type used in the system, the CKMS design shall specify, all bound metadata elements and the circumstances under which the metadata is created and bound to the key.*

Comments:

> With reference to section 6.4.22 “Key Output”, is there some way of storing metadata indicating if the value of the key has been output as cleartext, and under what conditions (as a split key scheme? as direct output?), and to whom. **This audit trail may need to be readily accessible during a legal investigation.**
>
> e.g. In a banking context, a key that is physically transported using a split key scheme has been compromised. We may need to know the identity of the people performing the courier operation. Do we also need to know what hardware security modules were involved in the transaction, what version of the operating system, … to identify and hold insider attackers accountable?

35.36. VPQ: Section 6.2, page 28, Draft SP 800-130

Original text:

> *For each key type, the CKMS design shall specify all applicable metadata elements from the list below (even if they are not bound elements):*

See also:

> *With reference to section 6.8.1 “Key Compromise”: “A CKMS should limit the exposure of key compromises by establishing a cryptoperiod or usage limit for each key that it uses”*

Comments:

> Do we need a counter for "how many more 'protection' invocations until the key must be refreshed"? --> thinking along the lines of "NIST key length transitioning document" and also a broad range of cryptanalytic attacks (differential analysis, …) which require large number of plaintext-ciphertext pairs under a fixed-key to be effective.

35.37. VPQ: Section 6.2, page 29, Draft SP 800-130

Original text:

e) Method of Distribution

i. Internal module key (i.e., key is created and used within the module only)
ii. Manual
iii. Electronic

Comments:

Method of distribution needs a "Protocol Name" and "parameters".

As mentioned above, we may need quite detailed information about the parameters:

- Were there trusted third party relays/couriers involved?
- Over how many paths was the key material split?
- Was the key material randomised (All or nothing transformation) to ensure the parts did not directly correlate to part of the key value?
- Were the individual key parts securely "stored" in a safe, "destroyed" by fire, or maybe we don't know where these parts are any more because the courier may have had the opportunity to make a copy.

35.38. VPQ: Section 6.2, page 29, Draft SP 800-130

Original text:

Applications that may use the key (e.g., TLS, SCL, EFS, S/MIME, IPSec, PKINIT, SSH, etc.)

Comments:

Can you provide citations for each technology here?
What is an application, and what is the best way to manage this?

See our feedback in "34. QC: Section 3.1, page 15, Draft SP 800-130" on page 73.
This also relates to "35.27 VPQ: Section 6.2, page 25, Draft SP 800-130" on page 122.

VPQ: Section 6.2, page 29, Draft SP 800-130 (continued)

Original text:

j) Key Assurances

i. Symmetric key assurances

Suggested additional assurance questions:

Was it performed in an information theoretically secure way? (ITS against WHO with WHAT capabilities?)

Was it performed in a TEMPEST enclosure?

Was it derived in a computationally secure (classical / post quantum) way from a key that was negotiated in an information theoretically secure way?

What is the possible exposure level to different classes of insider attacks?

35.39. VPQ: Section 6.2, page 29, Draft SP 800-130

Original text:

iv. Public Key Validity Check

** Who performs it*

** Circumstances under which it is performed*

** How it is performed*

Comments:

Over and above “mathematical correctness tests”, and “revocation checking” have we checked internally to see if there is more than one certificate assigned to resources claimed to be under control by this public key?

35.40. VPQ: Section 6.2, page 29, Draft SP 800-130

Original text:

*For each key type, the CKMS design **shall** specify the protections (including binding techniques) that are applied to key and bound metadata. The CKMS design **shall** specify when these protections are applied and (if appropriate) when they are verified.*

Comments:

Not directly related to the above text, but how do we manage “resource contention control of key meta-data”?

How should we manage “mutual exclusive locks” / “multiple readers, one writer locks”, … Could key locking be used in controlling access, the number of concurrent accesses and for synchronising remote systems when a key value is updated? How does the CKMS ensure atomicity, consistency, isolation, durability (ACID)?

35.41. VPQ: Section 6.3, page 29, Draft SP 800-130

Original text:

> ***Key Life Cycle States and Transitions***
>
> *A key may pass through several states between its generation and its destruction. This section is a modification of Section 7 Key States and Transitions from [SP 800-57-part1].*

Comments:

Do we need some notion of "authorised mode of operation", "normal operation", "crisis operation with lower security controls and increased interoperability"?

How might this affect cryptographic audit logging, and then 'recovery' after the event through systematic key reinforcement of all keys used during crisis mode that were exposed to lower security measures than normally permitted?

See also section 6.7.1 The Access Control System (ACS) of SP 800-130.

35.42. VPQ: Section 6.3.1, page 29, Draft SP 800-130

Original text:

> ***Key States***

Comments:

There seems to be the need to enrol dependants on the state of the key, and to notify a subset of those dependants concerning some state transitions. (e.g. a transition to compromised may require formal notification to an oversight body). How does the system ‘track’ the notification operations?

In one case use, stateless devices (devices without a reliable clock source) such as RFID may require explicit revocation on certificate expiration. This means we may need "revoked with a status code/priority".

Even access to key material may require certain subscribers to be notified to support behavioural analysis security controls.

35.43. VPQ: Section 6.3.1, page 29, Draft SP 800-130

Original text:

> ***Key States***

Comments:

What about "generation state"?:-

> “I am in the process of establishing the meta-data around the key generation process, such as describing the protocols that must be used to negotiate this key when I’m told to generate”…
>
> “I am getting the key generation step authorised by some party”,
>
> "I am about to generate a random number by calling a RNG",
>
> "I am in the process of negotiating the mutual creation of key by exchanging nonces with another device",

e.g. A central CKMS may create a key entry, which requires two other devices to negotiate their key. A central CKMS may only be “aware” that a key exchange has taken place between two devices, where the two devices locally have associated records linking back to the central CKMS.

35.44. VPQ: Section 6.3.1, page 30, Draft SP 800-130

Original text:

> ***c) Suspended state:*** *The use of a key may be suspended for a period of time. Individual modules may locally suspend the use of a key without reporting the suspension beyond the users of the module. A suspended key may be restored to an active state at a later time. A suspended key is suspended for all use unless re-activated. Eventually the suspended key is either activated or deactivated.*

Comments:

Can a suspended state go directly to Revoked or compromised state without going via "active"?

Scenario: “Hi I *think I* have lost my credit card. Please suspend my card. If I don't find it in 48 hours and call you back, can you then revoke my card and issue me another one?" [reduce risk, while opening up potential to avoid token re-issuance costs]

Once suspended, I may need to NOTIFY all registered dependants that the key material has been suspended.

35.45. VPQ: Section 6.3.1, page 30, Draft SP 800-130

Original text:

> ***Compromised state:*** *Generally, keys are compromised when they are released to or determined by an unauthorized entity.*

Comments:

Can we “subclass” compromised state to provide additional information? e.g. “Adversary Has key”, “User lost control of smart card”, “HSM is no longer responsive”.

Compromised state should also take into account meta-data of the key exchange.E.g. I have lost my smart card, it's possible an adversary has it, and given the make and model they may be able to extract the key in estimated X hours of reverse engineering at $Y cost IF they wanted to. There are 100 resources at risk, …

This information should be registered with the “Runtime Risk Management System”

Including quantifiable information about properties of a compromise risk can enable certain policy operations to be enforced e.g. what priority level must I apply to compromise notification messages, and who do I need to send “heads-up” warnings to, so they can perform retro-active behavioral analysis with regard to certain resource access (is this a very likely breach, are the resources this token accessing high value)

35.46. VPQ: Section 6.3.2, page 32, Draft SP 800-130

Original text:

> ***Transition 4:*** *Keys transition from the pre-activation state to the active state when the key becomes available for use. This transition may be activated after reaching an activation date or by an external event. In the case where keys are generated for immediate use, this transition occurs immediately after entering the pre-activation state.* ***This transition marks the beginning of a key’s cryptoperiod.***

Comments:

If the key is generated internally within a device using a RNG, then yes the transition to the active state marks the beginning of a key’s cryptoperiod, definitely.

If the key is negotiated between two devices using an information theoretically secure technique (such as within the protective confines of a tempest enclosure), then yes.

However, if the key is negotiated using a public key transaction, then the ciphertext potentially exposed to an attacker during pre-activation has sufficient information to allow the key material to be mathematically attacked. So if a key is stored in pre-active state for say 10 years, it may be the that the key has already been compromised before the ‘beginning of a key’s cryptoperiod’.

35.47. VPQ: Section 6.3.2, page 32, Draft SP 800-130

Original text:

> ***Transition 5:*** *An active key may transition from the active state to the compromised state when the integrity of a key or the confidentiality of a key requiring confidentiality protection becomes suspect. Generally, keys are compromised when they are released to or determined by an unauthorized entity.*

Comments:

Can we include "or when the authorised party reported loosing physical control over a device storing key material with the CKMS"?

35.48. VPQ: Section 6.3.2, page 32, Draft SP 800-130

Original text:

> ***Transition 7:*** *An active key may transition to the deactivated state if it is no longer to be used to apply cryptographic protection to data or no longer intended to be used to process cryptographically protected data. A key may transition from the active state to the deactivated state if the key is replaced or at the end of the key's cryptoperiod.*
>
> ***Transition 10:*** *A suspended key may also transition to the deactivated state if that key is no longer to be used to process data.* ***All appropriate users should be notified that the key has been deactivated****.*

Comments:

Might we need to notify dependants on Transition 7? Notification requirement should be stated consistently throughout this section. See transition 10 above for an example of notification.

35.49. VPQ: Section 6.3.2, page 33, Draft SP 800-130

Original text:

> ***Transition 13:*** *Assuming that a key is not determined to be compromised while in the deactivated state, a key may transition from the deactivated state to the destroyed state. In general, a key transitions to the destroyed state as soon as it is no longer needed*

Comments:

This transition may need to take into account data-retention laws, etc. Data Retention laws vary based on jurisdiction. How does the CKMS determine what jurisdictions have authority over this key metadata, and might that change based on run-time properties?

Can keys managed with a "global-CKMS" be restricted to storage within HSM in certain countries to avoid "legal overlap". e.g. If a key is replicated in 100 countries, does access of that key from any one of those 100 countries invoke the (potentially contradicting) laws of all 100 countries simultaneously?

Likewise, does the CKMS may need to consider the law of the country the key is accessed from, plus the law of the country applying to the client requesting and receiving that key material? Does the CKMS need to be able to have the ability to autoselect the least restrictive country to access the replicated key material from? [Key is stored in 10 countries, you are in country x, you can access key from these 3 countries].

35.50. VPQ: Section 6.4, page 34, Draft SP 800-130

Original text:

> ***6.4 Key and Metadata Management Functions***
>
> *The functions described in this section are performed on keys or metadata for management purposes.*

Comments:

> In safety critical systems, might we need to be able to veto a key state change?
>
> For example, "I'm the control system of a nuclear power station. I'm currently managing a crisis situation with my reactor. Please avoid my routine/non-critical key rolling operations at this time. Please don't ask/ force the instructors to perform a routine password change just now!"

35.51. VPQ: Section 6.4.1, page 34, Draft SP 800-130

Original text:

> ***6.4.1 Generation***
>
> ***When a user requires a key, the user may request that the key be generated by the CKMS. The user may need to specify the type of key and other necessary parameters, including some metadata, when requesting this function.***

Suggested Text:

> "When a user requires a key, there are many ways this can be achieved. The user may request that the key be generated by one HSM in the CKMS, negotiated by two HSM, the manual insertion of a secret key (such as in the case of password for a website, or those issued by a third party), registration of a public key (as in the case of a certificate authority service) and so on. Some key generation schemes explicitly require that the key is NOT released from the HSM (as in the case of non-repudiation services)."

Comments:

> Is there a distinction between key registration (public key inserted into CKMS), key generation (where the random generator within CKMS is used), and key distribution (where a key is securely distributed between two HSM, such as with mirroring / backup / high availability services)?
>
> see 6.4.20 Key Establishment

35.52. VPQ: Section 6.4.1, page 34, Draft SP 800-130

Original text:

> *Key generation techniques typically depend on the specifications of the cryptographic algorithm associated with the key.*

Suggested revised text:

> **At a lower level of abstraction,** key generation techniques typically depend on the specifications of the cryptographic algorithm associated with the key.

35.53. VPQ: Section 6.4.1, page 34, Draft SP 800-130

Original text:

> *Key generation for asymmetric algorithms involves the generation of a key pair.*

Suggested revised text:

> Key generation for asymmetric algorithms involves the generation of **a mathematically related** key pair.

35.54. VPQ: Section 6.4.3, page 34, Draft SP 800-130

Original text:

> *The activation function provides for the transition of a cryptographic key from the pre-activation to active state. This function may automatically activate the key. Alternatively, this function may generate a date-time metadata value that indicates when the key becomes active and can be used. A deactivation date-time may also be established using this function.*

Observation:

> In some cases, activation may require explicit notification to subscribers. For example behavioural security analysis engines, etc.

35.55. VPQ: Section 6.4.4, page 34, Draft SP 800-130

Original text:

> ***6.4.4 Deactivation***
>
> *This function transitions a key into the deactivated state. ...*

Question:

> Do we also need to discuss the concept of "deactivation approaching" so that dependants can begin re-actively organising their key-update cycle? [Your token will expire in 3 months, please go here to update your contact and payment details and authorise us to send you a new token…] This may be a requirement regarding the capability of the “CKMS policy engine” and the “user centric interfaces” which may be a online web portal, letter mailed in the post, a phone call to the client, or ...

35.56. VPQ: Section 6.4.4, page 34, Draft SP 800-130

Original text:

> *A cryptographic key is generally **given a deactivation date and time** when it is created and distributed..*

Question:

> How should this section tie into the counter for the maximum number of protection operations?

35.57. VPQ: Section 6.4.6, page 36, Draft SP 800-130

Original text:

> *A key may be temporarily suspended*

Question:

> Can we state in the text that suspension is different to "locking and synchronisation" controls on key material? e.g. A key should not enter into suspended state as a locking and synchronisation mechanism.

35.58. VPQ: Section 6.4.8, page 36, Draft SP 800-130

Original text:

> *A key can be updated by transforming it in a deterministic and synchronized manner everywhere it is needed. Key update has the possible security exposure that an adversary who obtains a predecessor key and knows the update transformation can update that (predecessor) key to the new key.*

Observation:

> If fresh nonce material is exchanged between two parties (a first device and a second device), that nonce material could be mixed in with the otherwise deterministic update function. In some cases, assuming a specific adversary does not have the capability to monitor ***all*** communications between first and second device (e.g. an adversary on the LAN attacking a smart card token that roams between various internet access points), these updates add fresh entropy (with regard to that adversary) into the state update function and can "recover" from a security compromise. See Ross Anderson's paper on "smart trust for smart dust" on the exchange of entropy **in the clear** as a low-cost security function against adversaries with limited ability to monitor ALL network communications [95].

[95] Anderson, R., Chan, H., and Perrig, A. Key infection: Smart trust for smart dust. In ICNP '04: Proceedings of the 12th IEEE International Conference on Network Protocols (Washington, DC, USA, Oct. 2004), IEEE Computer Society, pp. 206–215. Available at http://www.cl.cam.ac.uk/~rja14/Papers/key-infection.pdf

35.59. VPQ: Section 6.4.9, page 37, Draft SP 800-130

Original text:

> *6.4.9 Destruction*
>
> *Keys and their bound metadata should be destroyed when they are no longer to be used.*

Observation:

> How about data-retention laws?
>
> How about the ability to AUDIT the occurrence of a transaction, even if we can't recover the key material?
>
> It is possible that after the value of a key is zeroised we need to then archive the bound meta-data and destroy it after 7 years? (and refuse to roll the key during archival re-encryption).
>
> This text should explicitly link back into section 6.4.12 (Delete Metadata).

35.60. VPQ: Section 6.4.10, page 37, Draft SP 800-130

Original text:

> ***Alternatively,*** *physical protection can be provided to the key and its bound metadata so that parts of the combination cannot be replaced without authorization and the key itself cannot be disclosed to unauthorized entities.*

Suggested text:

> **Alternatively and/or in addition,** physical protection…

35.61. VPQ: Section 6.4.15, page 38, Draft SP 800-130

Original text:

> ***6.4.15 Backup Key Storage***
>
> *Backup key storage involves placing a copy of a key in a safe facility so that it can be retrieved if the original is lost or modified. Backup copies of keys may be located in the same or a different facility than the operational keys to assure that the keys can be retrieved when needed even after a natural or man-made disaster.*

Observation:

> Are the requirements for Key Backup and Key Archiving negated by having Primary and Secondary standby sites?
>
> Active backup key storage could be achieved online by mirroring keys in another HSM located in the same or different facility to ensure high-availability and freshness of key material, and to also support improved performance in unexpected peak periods (such as crisis situations involving first responders). This would facilitate availability and better synchronisation of key materials across the CKMS deployment.
>
> Clearly key material may be actively accessed in multiple locations. How is this data managed and consistency maintained in the event of partial wide-area network failure? (that is, resynchronise against attacks against an ISP / the internet).

35.62. VPQ: Section 6.4.16, page 38, Draft SP 800-130

Original text:

> ***6.4.16 Key Archive***
>
> *Key archive involves placing a key in a safe long-term storage facility so that it can be retrieved when needed. Key archiving usually requires provisions for moving the key to new storage media when the old media are no longer readable because of aging of, or technical changes to, the media readers.*

Observation:

Might the process of key archiving invalidate the policy requirements that are active on certain keys?

What are the performance requirements on key-archive to support deletion operations? Must a Key Archive be capable of deleting keys and meta-data 4 weeks after a deletion request is received ?

Are we thinking that key-archiving involves tapes?

Archives themselves may need to be encrypted, and the movement of data from one archive to another may require secure re-encryption to larger / stronger cryptographic primitives. The Framework should require that the CKMS design should specify how it does this without exposing archived data to insiders.

Is Key Archiving intended simply to be a low-cost version of Key Backup that has limited operational capability to serve CKMS requests? (a slave device to a CKMS system that only talks with the CKMS servers) In this way key-archiving enables "additional site location" to be used to store key material at lower cost??

Observation:

Offline key archiving of post quantum secure **Merkle Tree private keys** can result in a catastrophic security failure in the scheme if the signing leaf-nodes are reused. **Special care must be taken for state-maintaining public key schemes** (as opposed to stateless randomised schemes). **A single re-use of a leaf-node results in the entire {public, private} key pair being compromised[96].**

35.63. VPQ: Section 6.4.26, page 41, Draft SP 800-130

Original text:

> ***6.4.26 Validate Symmetric Key***
>
> *This function performs certain tests on the symmetric key and its bound metadata. These tests might involve checking for the proper length and format of expected parameters. This command may also verify any error detection/correction codes or integrity checks placed upon the key and its bound metadata.*

Observation:

This may also include checking for known weak keys, such as in the case of DES.
Care with weak keys needs to be taken. The 'check' for weak keys on the CKMS **needs to** be suitable for the selected 'target' device. That is, some smart cards may be more restrictive on the value of a key than the CKMS test. Consider 3DES. Should 3DES apply DES weak key checks on each of the three 56-bit long keys? How does every vendor product handle it?

---

[96] Coronado, C. Provably secure and practical signature schemes. Doctoral thesis (elib.tu-darmstadt.de/diss/000642), Technische Universit¨at Darmstadt, Nov. 2005. Available at http://tuprints.ulb.tu-darmstadt.de/epda/000642/carlosDiss.pdf

35.64. VPQ: Section 6.4.27, page 41, Draft SP 800-130

Original text:

> ***6.4.27 Validate Private Key (or Key Pair)***

Observation:

> Merkle tree algorithms may randomly select an unused node, internally sign a message and validate the message to check that the public / private pair of that tree appears 'correct'.
>
> If the Merkle tree is fully expanded (as in the context of high-performance signing applications) it may be possible to randomly validate internal nodes to incrementally validate the correctness of that tree.

35.65. VPQ: Section 6.4.30, page 42, Draft SP 800-130

Original text:

> ***6.4.30 Manage Trust Anchor Store***

Observation:

> Conceptually, a trust anchor may be a unique pair-wise symmetric key shared between a token/device and the central CKMS. Using that symmetric trust anchor, the (CPU constrained) device may ask for a digital signature and certificate to be validated on behalf of that device by the CKMS.

35.66. VPQ: Section 6.5, page 43, Draft SP 800-130

Original text:

> *The associated private key that is used to decrypt the keys should also be protected in some manner, e.g., using physical security,* ***that usually does not involve encryption.***

Proposed alternative text:

> The associated private key that is used to decrypt the keys should also be protected in some manner, e.g., splitting the key onto three or more smart cards, and storing the smart cards in three different secure locations.

35.67. VPQ: Section 6.5, page 43, Draft SP 800-130

Original text:

> *All keys require integrity protection, because a garbled key will not correctly perform its intended function.*

Suggested additional text:

> Also, some cryptographic attacks can be performed if an adversary is permitted the ability to arbitrarily choose related keys. (related key attacks).

35.68. VPQ: Section 6.5, page 43, Draft SP 800-130

Original text:

> *A key may be garbled, lost, or destroyed to the extent that it cannot be recovered by error correcting codes. If the key is a symmetric key or a private decryption key, this could result in the loss of the data protected by the key.* ***A CKMS should employ methods*** *for backing-up, archiving, and recovering keys as necessary to provide for the recovery of valuable data. Appendix B of [SP 800-57-part1] provides guidance on the recovery procedures for various key types.*

Proposed variation:

> ***A CKMS should employ methods for*** online mirroring, online backing-up, offline archiving and recover keys as necessary to ensure the correctness, integrity and availability of valuable data without loss in the advent of a component/site failure.

Observation:

> The original text suggests that backup and archiving are desirable for disaster recovery. Conceptually having a primary facility with Operational, Hot Standby, warm standby, cold-standby components synchronised real time with geographically separated secondary and tertiary facilities with equal compute and availability properties could provide greater over-all system availability and integrity than resorting to offline (and always partially out-dated) backup/archiving.
>
> That is, we think it is important to ensure that the CKMS system always remains operational in the advent of any subset of the m-1 of the facilities falling to disaster. This may require new storage systems (RAID) to be taken to the remaining operational facility, synchronised over LAN, the RAID moved to the new facility, put online, and then the remaining synchronisation performed over WAN.

35.69. VPQ: Section 6.6.1, page 44, Draft SP 800-130

Original text:

> *6.6.1 Key Transport*
>
> *When cryptographic keys and metadata are transported (distributed) from one secure location (data sender) to another (intended data receiver), they should be protected.*

Suggested additional text:

> Key transportation schemes should ideally be randomised, such as with an All-or-nothing-transformation[97], to ensure that if the value of a given key is transported twice, the encoded value in transit is always different.
>
> This is particularly important for split key transport schemes where one or more relays may be involved in transporting the same key several times. If AONT is not used, a single courier/relay may discover the value of n parts of the key after it is transmitted >= n times.

35.70. VPQ: Section 6.6.1, page 44, Draft SP 800-130

Original text:

> *A manually distributed key could be physically protected by a trusted courier, or a physically protected channel could be used. Very often, the keys are sent electronically over networks that are susceptible to data eavesdropping and modification.*

Proposed revised text:

> A manually distributed key could be physically protected **by one or more trusted couriers**, or **a physically protected channel (such as a smart card, TEMPEST enclosure, or point-to-point quantum key distribution channel) could be used**.

35.71. VPQ: Section 6.6.1, page 44, Draft SP 800-130

Original text:

> *If cryptography is used to protect the confidentiality of symmetric and private keys during transport, then a key establishment technique involving either a symmetric key-wrapping-key or, one or more asymmetric key-transport-key pairs is used.*

Suggested additional text:

> A symmetric key wrapping key scheme may be split path, with one or more relays on each path. e.g DHL[98] symmetric key exchange, or an ad-hoc wireless mesh network protocol.

---

[97] http://en.wikipedia.org/wiki/All-or-nothing_transform

[98] Diffie, W., and Hellman, M. E. Multiuser cryptographic techniques. In AFIPS '76: Proceedings of the June 7-10, 1976, national computer conference and exposition (New York, NY, USA, June 1976), ACM, pp. 109–112.

35.72. VPQ: Section 6.6.1, page 44, Draft SP 800-130

Original text:

> *All transported keys require integrity protection because a garbled key will not correctly perform its intended function.*

Proposed revised text:

> All transported keys require integrity protection because a garbled/corrupted key will not correctly perform its intended function, and some cryptanalytic attacks can exploit the ability for an adversary to control the key if they can gain access to the corresponding ciphertext. Removing the ability to garble keys from an adversary may improve the security/durability of the system.

35.73. VPQ: Section 6.6.2, page 45, Draft SP 800-130

Original text:

> *6.6.2 Key Agreement*
>
> ***Two entities, working together, can create and agree on a cryptographic key without the key being transported from one to the other.*** *Each entity supplies some information that is used to derive a common key, but an eavesdropper obtaining this information is not able to determine the agreed-upon key.*

Proposed revised text:

> *Two entities, working together, can exchange entropy and agree on the value of a cryptographic key. Because this process is collaborative, the value of the resultant key is not generated from one party and transmitted to the other.*

35.74. VPQ: Section 6.6.2, page 45, Draft SP 800-130

Original text:

> *6.6.2 Key Agreement*
>
> *This is known as key agreement. Cryptographic algorithms employing key-agreement keys are used by each entity.*

Suggested additional text:

> An advantage of key-agreements where both parties contribute entropy to the final value of the key is that it can mitigate the presence of cryptographic weakness in a RNG at one party. Additionally, the exchange of fresh entropy between two parties in the key agreement protocol can help protect against a broad range of replay attacks.
>
> Long-lived keys negotiated using key agreements may later be used in key-transportation agreements (e.g. mirroring, archiving, backup). Key agreement may be used to generate a one-time-use transport key that is then used to perform key transport e.g. Quantum Key Distribution systems.

35.75. VPQ: Section 6.6.3, page 45, Draft SP 800-130

Original text:

> *6.6.3 Key Confirmation*
>
> *When keys are established between two entities, each entity may wish to have confirmation that the other party did in fact establish the correct key. Key confirmation schemes are used to provide this capability. [SP 800-56A] and [SP 800-56B] specify key confirmation schemes for use in Federal CKMS. Other methods may also be appropriate.*

Questions:

> Can we clarify how Key Confirmation is different to Proof of Possession in the text? (See 6.4.28 Validate Possession of Private Key.)
>
> Is there a NIST standard for key confirmation for symmetric key systems?

35.76. VPQ: Section 6.7, page 45, Draft SP 800-130

Original text:

> *The security of a CKMS depends on the proper sequence and execution of the key management functions described in Section 6.4. The execution of these functions may be driven by time, an event, a human, or some combination of these options.* ***Therefore, an access control system is required to*** *assure that key management functions are only performed in response to requests (calls) by authorized entities and are appropriate for the key state.*

Proposed revised text:

> Therefore, an **identity management system and** access control system is required to assure …

35.77. VPQ: Section 6.7, page 45, Draft SP 800-130

Original text:

> *Even if the calling entity is authorized to call a key management function, the call may be refused for some reason. For example, the metadata may indicate that the function is inappropriate under the existing conditions.*

Suggested additional text:

> *In times of crisis (or high work load), low-priority operations such as key-rolling may be temporarily suspended. Calls may also be refused on account of behavioural control measures that limit the rate of access to key material by certain parties, or due to veto operations by other parties on escrow operations, and so forth.*

35.78. VPQ: Section 6.7, page 45, Draft SP 800-130

Original text:

> *6.7.1 The Access Control System (ACS)*

Observation:

> *Usability with regard to scalability of this section is critical.*
>
> *For example, if you have a billion uses, how do you manage the ACS? How do you compartmentalise the domains of control (e.g. each organisation may have it's own ACS domain within a single global scale CKMS deployment). In a user-centric design, each user may have veto-rights over which organisations can process their data.*

35.79. VPQ: Section 6.7.2, page 48, Draft SP 800-130

Original text:

> *Keys requiring output from the module may be transported using a key transport scheme. A symmetric encryption/decryption key may then be output and transported in encrypted form using the public key of the receiving entity.*

Proposed revised text:

> *A symmetric encryption/decryption key may then be output and transported in encrypted form using the (public or symmetric) key of the receiving entity.*

35.80. VPQ: Section 6.8, page 49, Draft SP 800-130

Original text:

> ***6.8 Compromise Recovery***
>
> *In an ideal situation, the CKMS would protect all keys and sensitive metadata so that data requiring confidentiality protection is never compromised, and data requiring integrity protection is never modified by unauthorized parties.*

Proposed revised text:

> *In an ideal situation,* ***collectively the CKMS and all dependent devices processing sensitive key material*** *would protect all keys and sensitive metadata, so that data requiring confidentiality protection is never compromised, and data requiring integrity protection is never modified by unauthorized parties.*
>
> *(This raises the ontological question: does a Enterprise CKMS system boundary include all devices and entities that are dependent on that Enterprise CKM solution?)*

35.81. VPQ: Section 6.8.2, page 50, Draft SP 800-130

Original text:

> ***6.8.2 Cryptographic Module Compromise***

Questions:

> This section seems to be written from the perspective of an Enterprise CKMS product, as opposed to the use of a smart card or HSM in the field.
>
> Does it make sense to talk about an Enterprise CKMS and users of the CKMS in independent but related sections?

35.82. VPQ: Section 6.8.2, page 50, Draft SP 800-130

Original text:

> *Cryptographic modules can be compromised either physically (obtaining direct access to the keys within the module) or by non-invasive methods (obtaining knowledge of the keys within the module by some external action).*

Question:

> What if it is discovered that the compromise occurs by the Vendor exploiting a back-door they implemented?
>
> Can we make the text around “non-invasive methods” clearer? Are you referring to side-channel attacks?

35.83. VPQ: Section 6.8.4, page 52, Draft SP 800-130

Original text:

> *6.8.4 Network Security Controls and Compromise Recovery*

Question:

> Should we encourage use of different network security appliance vendors to protect against a single insider attack compromising all security functions?
>
> e.g. two sets of firewall by different vendors with 'logically' identical configurations managed by two independent teams? [this is for the purpose of global-scale CKMS as opposed to small enterprise security, that is, security proportional to the value of the system].

35.84. VPQ: Section 6.8.4, page 53, Draft SP 800-130 (2 Factor)

Original text:

> *If passwords are compromised, the passwords should be replaced. The users may need further training in selecting the password, in understanding password entropy, in changing passwords frequently, and in maintaining the confidentiality of written-down passwords. An examination should also be made of the authentication protocols to determine if password sniffing, online dictionary attacks or offline dictionary attacks are feasible.*

Observation:

> Two factor authentication should be recommended to mitigate damage of weak password entropy (or password compromise) and reduce the burden on maintaining password frequency.

35.85. VPQ: Section 6.8.4, page 53, Draft SP 800-130 (OS)

Original text:

> *If the platform operating system is compromised, one or more of the following actions should be considered, and appropriate corrective measures taken:*
>
> *a) Make sure that all the latest operating system security patches are installed.*
>
> *b) Ask the operating system vendor if there is a patch for the compromise.*
>
> *c) Determine if a device configuration change or if blocking some protocols will prevent future attacks of the same nature as the one that caused the compromise.*

Question:

> What if there is a connection between the operating system back-door and the person exploiting the attack (same person)? Should a CKMS design move to another operating system vendor? What if you standardise on a proprietary platform (require non-standard API's) and there is no other alternative vendor?

35.86. VPQ: Section 6.8.4, page 53, Draft SP 800-130 (net app)

Original text:

> *If the network security application is compromised, one or more of the following actions should be considered, and appropriate corrective measures taken:*
>
> *a) Make sure that all the latest network security patches are installed.*
>
> *b) Ask the application vendor if there is a patch for the compromise.*
>
> *c) Determine if a device change, an application configuration change, or the blocking of certain protocols will prevent future attacks that allowed or caused the compromise.*

Question:

> What if the attack is performed by an insider from the network security application vendor?

35.87. VPQ: Section 6.8.7, page 55, Draft SP 800-130

Original text:

> *Once security is breached, the integrity of the entire breached area should be suspect. The CKMS should inform the appropriate entity as specified in the security policy of the breach so that mitigation actions can be taken. In addition, it may not be sufficient to replace all sensitive data within the breached area, because the attacker could have modified or added to the logic within the area so that the new keys and sensitive information could also be compromised in the future.*

Observation:

> Can we make it clearer that the Primary CKMS and the Second CKMS systems, each operating with different sets of keys, need to be operating from physically DIFFERENT sites, to prevent **all** keys in the system being compromised from various security breaches.... That is, to protect against system wide single point of trust failure.
>
> The use of primary and secondary CKMS may require special key-handling with regard to keys used to encrypt data-at-rest. This means data may need to be encrypted "while the primary server is online" and then later "when the secondary server is online". See section 5.3 "Visually illustrating the role of Primary and Secondary Facilities" in this document.

35.88. VPQ: Section 6.8.7, page 55, Draft SP 800-130

Original text:

> *Thus, a smooth transition may require the capability to support the use of at least two algorithms simultaneously. In that case, the cryptographic protocols should be designed to identify and negotiate which algorithm will be used in a particular key establishment transaction.*

Suggested additional text.

> The option to enable Legacy support should be supported by policies that can also restrict legacy access to a specific set of associated cryptographic credentials. In this way, as time progresses and the up-take of the new protocol / algorithm increases in the community of interest, it is then possible for a CKMS instance to reduce the exposure to adversaries port-scanning for vulnerable implementations of a protocol over the Internet.

35.89. VPQ: Section 7, page 56, Draft SP 800-130

Original text:

> ***If security is improved in one CKMS component, but the component is no longer interoperable with peer components having older security mechanisms, the new component will generally not be accepted in the marketplace****. For example, if a new encryption algorithm is installed at some entity in a network, then only the entities with the new algorithm capability will be able to communicate with the new technique. Other entities will likely continue to communicate using the older algorithms. Unless accommodation is made for the smooth transition to the new algorithm (e.g., by allowing the use of legacy algorithms where necessary), the transition will be slow even when the new algorithm offers significant benefits. This is especially true because the security lifetime of a cryptographic algorithm is only an estimate and the old algorithm may actually be secure for additional years.* ***Thus, a smooth transition may require the capability to support the use of at least two algorithms simultaneously.*** *In that case, the cryptographic protocols should be designed to identify and negotiate which algorithm will be used in a particular key establishment transaction.*

Quote:

> *If the adversary has access to the ciphertext data and can determine the key, then the data no longer has reliable confidentiality protection. That is, the owner of the sensitive information should consider the information to no longer be protected (i.e., the information should be considered as being in plaintext form). Several scenarios need to be considered when evaluating whether or not the information is or will remain secure.*
>
> *...*
>
> ***If the ciphertext data is re-encrypted*** *or rewrapped*[99] ***using a stronger algorithm or key length, then the confidentiality of the sensitive information will remain valid as long as the stronger algorithm remains secure.***
>
> *Draft NIST SP 800-131 June 2010*

Recommended additional text:

> Additionally, legacy deployments should consider the use of secure-tunnels and wrap-arounds with more modern security primitives to maintain interoperability (using the weaker algorithm) while providing increased protection (using a stronger algorithm in the secure tunnel) where viable. Such an approach can be used to 'upgrade' legacy devices in the field, WITHOUT changing any software / hardware in the weaker device, by using a software/hardware "bump-on-the-wire" approach.
>
> **Recommendation: Compliant CKMS implementations should be designed to facilitate "future" bump-on-the-wire functionality by supporting the configuration of a "secure proxy server"[100]. (Future proofing the system).**

[99] Decrypted or unwrapped using the original algorithm and key to produce the original plaintext, and then encrypting or wrapping the plaintext using another algorithm and key.

[100] http://en.wikipedia.org/wiki/Proxy_server

35.90. VPQ: Section 7, page 57, Draft SP 800-130

Original text:

> *A secure operating system is the foundation for securing a computer system. Without ensuring that the underlying operating system is secure, the security of CKMS components and the data running on the computer system cannot be assured. A secure operating system has the following security features:*

Proposed variation of the above text:

> A secure operating system **is an essential requirement for** securing a computer system. Without ensuring that the underlying operating system is secure, the security of CKMS components and the data running on the computer system cannot be assured. **Furthermore, if the operating system is running on a hypervisor, that hypervisor must also be secure. Likewise, this dependency continues down through the secure firmware, to a trusted hardware platform, which may include the use of trusted platform modules[101].**

35.91. VPQ: Section 7, page 57, Draft SP 800-130

Original text:

> *Note that CKMS components that perform dedicated security functions and do not provide a general-purpose CKMS component development, loading, or processing capability, may have reduced or minimal operating system requirements. As an example, consider a special-purpose appliance loaded with firmware and/or software to perform intrusion detection functions. This appliance may not have an operating system, and hence has no operating system security requirements. Another example is a firewall or intrusion detection system built on a "locked down" operating system so that the capability to load other CKMS components is not available.*

Proposed variation on the text:

> *Note that CKMS components that perform dedicated security functions and do not provide a general-purpose CKMS component development, loading, or processing capability, may have reduced or minimal operating system requirements.* ***These designs may employ JustEnoughOS[102] or have custom boot logic that is purpose-built exclusively for that platform and tied exclusively to one application (such that the OS and application logic are linked together in one executable binary).***

[101] http://en.wikipedia.org/wiki/Trusted_Platform_Module

[102] http://en.wikipedia.org/wiki/Just_enough_operating_system

35.92. VPQ: Section 8.2.4, page 57, Draft SP 800-130

Original text:

> *8.2.4 System Monitoring*
> *In order to protect the integrity and confidentiality of the data files of the CKMS, system monitoring tools may be deployed. These tools execute on the platform being monitored or on another platform dedicated to monitoring various hosts. These monitoring tools can detect modifications to system files or their access control attributes and post alerts and audit events (see Section 6.8.3).*

Observation:

The text regarding system monitoring seems very short and may not adequately describe the full breadth of what the author appears to be thinking of. E.g. In addition to monitoring the integrity and confidentiality of the data files of the CKMS (message digest checks on files?), does System Monitoring also include per event tracking of audit logs and other events generated by logic within the CKMS directly?

Might it be desirable to encourage the use a common network monitoring and event logging protocol such as Simple Network Management Protocol, and/or Java Management Extensions (JMX API)?

How is compartmentalisation achieved within the System Monitoring of the CKMS?

- Is there a single Super Authority capable of monitoring all events?
- Is there a way for stake holders to monitor data related to them?
- How do we protect against un-authorised information leakage between mutually suspicious enterprises who are both using the same system?
- Is there generic system behaviour information that can be made to the public?
  - up time / unscheduled down time, number of requests processed by the system, number of enrolled users, number of keys within the entire system, average response time for revocation notifications, and other useful “marketing” / “confidence building” information
- How is “multi-site” system monitoring managed?
  - Are there aggregate views?
  - Are there different administrators for each site?
  - …
- Is there a way to conditionally set “debug event logs” for specific communities to facilitate problem solving, without turning **debug_level = full** across the entire system?
- ?

35.93. VPQ: Section 8.3, page 61, Draft SP 800-130

Original text:

> *Networked CKMS components are protected using a mix of firewalls and intrusion detection and prevention systems.*

Questions:

Is it better to have all services in one hardware appliance? Are there any recommendations (or need to report) on the use of vendor diversity? Is it appropriate to run network traffic through two different network security appliances of equivalent function from different vendors to protect against insider attacks from one vendor?

35.94. VPQ: Section 8.3, page 61, Draft SP 800-130

Original text:

> *Boundary control devices (such as firewalls, filtering routers, VPN, IDS, IPS, etc.) should be hosted on computer systems (see Section 8.2) or should be implemented in dedicated hardware devices.*

Observation:

> There is increasing distrust in relying exclusively on "perimeter" fire walls. The concern is due to attacks mounted from within a local area network against that computer. See The Open Group Jericho Forum[103].

Proposed revised text:

> Boundary control devices (such as firewalls, filtering routers, VPN, IDS, IPS, etc.) should be **implemented in dedicated hardware devices and also hosted on computer systems (see Section 8.2) to provided layered defense-in-depth security, and to protect against Local Area Network bound attacks, where appropriate.**

35.95. VPQ: Section 8.4, page 63, Draft SP 800-130

Original text:

> *The CKMS design **shall** identify the cryptographic modules that it uses and their respective security policies.*

Question:

> How does the CKMS operator achieve assurances that the HSM doesn't have a back door that can be exploited by privileged employees of the HSM Vendor? What should a CKMS do if they cannot get satisfactory answers to this question?

[103] http://www.opengroup.org/jericho/

35.96. VPQ: Section 8.4, page 63, Draft SP 800-130

Original text:

> *9. Testing and System Assurances*
>
> *In this section, the term "CKMS device" may refer to any component of a CKMS or to an entire CKMS itself. A CKMS device may be composed of hardware, software, firmware or any combination thereof. A CKMS device may undergo several types of testing to ensure that it has been built to conform to its design, that it conforms to various standards, that it continues to operate according to its design, that it is interoperable with other CKMS devices, and that it can be used in larger systems for which it is intended.*

Proposed revised text:

> A CKMS device may undergo several types of testing to ensure that it has been built to conform to its design, that it conforms to various standards, that it continues to operate according to its design, **that it does not perform additional functions not permitted by the design requirements (malware), that it fails safely,** that it is interoperable with other CKMS devices, and that it can be used in larger systems for which it is intended **under all foreseen or prescribed operating conditions.**

35.97. VPQ: Section 9.3, page 63, Draft SP 800-130

Original text:

> *9.3 Interoperability Testing*

Observation:

> It is important that interoperability testing includes both binary compatible and semantic interoperability tests. See section 12, "Binary and Semantic Interoperability" in this analysis for more information.

35.98. VPQ: Section 9.4, page 64, Draft SP 800-130

Original text:

> *9.4 Self-Testing*

Observation:

> Periodic Rebooting.
>
> Ageing related bugs in a system are such that their probability of causing a failure increases with the length of time the system is up and running. A proactive recovery approach is to clean the system internal state to reduce the failure rate. This kind of preventative maintenance is known as "Software Rejuvenation". This can provide the opportunity for regular self-testing. See here[104] for more information and additional citations.
>
> We recommend the SP 800-130 talks about ageing related bugs and their prevention as part of the self-testing life cycle.

[104] K. Trivedi et al, "Achieving and Assuring High Availability", http://srejuv.ee.duke.edu/HighAvailability.pdf

35.99. VPQ: Section 9.6, page 65, Draft SP 800-130

Original text:

> *9.6 Functional Testing and Security Testing*

Observation:

> Section 9.6 does not discuss functional **safety** testing.
>
> Please see section "13. Possibility of adopting the Functional Safety Integrity levels within NIST SP 800-130?" on page 44 in this analysis for more information.

35.100. VPQ: Section 9.7, page 65, Draft SP 800-130

Original text:

> *Since testing is restricted to a finite number of cases that is typically significantly less than the total set of possibilities, testing does not guarantee that a device or system is correct or secure in all situations.*

Suggested outline of additional text:

> With this limitation in mind, code coverage is one consideration in the safety certification of avionics equipment. The standard by which avionics gear is certified by the Federal Aviation Administration (FAA) is documented in DO-178B[105]. Appropriate safety standards **shall** be applied to testing mechanisms deployed in the CKMS design.
>
> To improve assurance levels with regard to testing, the level of code coverage[106] attained by testing should be reported. Modern testing suites provide the ability to determine how much source code was tested using a regression test suite. Such testing should demonstrate that all exception handling, error messaging and audit log functions are operational.
>
> Comprehensive regression tests (with known answer tests) for each COTS component **shall** be made available by the vendor to the CKMS customer so they can perform on-demand regression testing when one or more CKMS components change within a system.

35.101. VPQ: Section 9.7, page 65, Draft SP 800-130

Original text:

> *The CKMS design shall specify the environment under which it is to be used.*

Question:

> Can NIST provide a reference to an appropriate (NIST/military) standard that could be used to specify the environment under which a system was tested, and how to perform adequate testing in that environment?

---

[105] http://en.wikipedia.org/wiki/DO-178B

[106] http://en.wikipedia.org/wiki/Code_coverage

35.102.VPQ: Section 10.1, page 66, Draft SP 800-130

Original text:

> *10.1 Facility Damage*
>
> *A CKMS should be located in physically secure and environmentally protected facilities. In addition, the CKMS should provide for backup and recovery in the event that damage to the CKMS occurs. The backup and recovery facilities should be designed, implemented, and operated at a level commensurate with the value and sensitivity of the data and operations being protected.*

Question:

> It may be beneficial to discuss the distance between the primary and secondary CKMS sites, and request that this information is specified by the CKMS design. The amount of distance is clearly important, given two facilities are located 1 kilometer away offer much less assurance against a natural disaster then co-locating in disjoint states (e.g. California, Maryland).
>
> Can NIST SP 800-130 provide a reference to an appropriate standard for co-location of service provisioning?

Suggested Requirement:

> The CKMS synchronisation and backup technologies **shall** be able to scale efficiently with regard to communications latency and the physical distance between co-location sites.

35.103.VPQ: Section 10.2, page 66, Draft SP 800-130

Original text:

> *The CKMS design **shall** specify the minimum electrical, water, sanitary, heating, cooling, and air filtering requirements for the primary and all backup facilities.*

Question:

> Can NIST SP 800-130 provide a reference to an appropriate standard / template for describing these requirements?

35.104.VPQ: Section 10.5, page 67, Draft SP 800-130

Original text:

> *Software failures may be minor, major or catastrophic in consequences. Minor errors may be due to undetected software errors (bugs) or due to temporary failures. Such errors or failures should be investigated and repaired before the CKMS is used. Major failures may be intentionally caused by corrupting the CKMS data or software. These failures should be investigated and repaired, perhaps by returning to a known secure state that was previously stored in a backup facility.*

Question:

If a privileged insider (a programmer involved in writing the CKMS code) injected an exploitable vulnerability in the design, exploits that vulnerability to run malware, and that malware is capable of corrupting the primary site, what protection mechanisms, if any, are available to protect against that same insider routinely corrupting the second site, particularly if the software at the first site and second site are written by the same organisation (privileged insider)?

Scenario:

- Primary Site Active (first set of keys are active)
- Adversary remotely triggers malware in primary site
- System administrators identify problems with first site
- System administrators trigger system-wide transition to Secondary Site and activate second set of keys.
- First site is taken offline.
- Adversary remotely triggers malware in secondary site
- ….
- Leading to the entire system being corrupted and taken offline

35.105.VPQ: Section 10.5, page 67, Draft SP 800-130

Original text:

> *Catastrophic errors should be investigated, and a backup facility used until the primary system can be completely reloaded from a known secure state.* ***In such situations the CKMS data created since the last secure state was saved may be lost.*** *A CKMS should be implemented and operated under the assumption that a catastrophe will eventually occur. Therefore, it is recommended that full secure-state system backups are made on a regular basis, and that the latest CKMS secure state can be reloaded into a repaired and ready CKMS.*

Question:

It is one thing to require that a database be returned to an internally consistent state. However, this state may not be consistent with the changes that have been made in systems outside of the CKMS database. Forcing a recovery could be a logistics nightmare and cause many security breaches (reactivation of tokens known to be compromised...)

If key and meta-data changes are lost, it is possible that:

- Policies are no longer enforced
- Devices can no longer communicate (compromise of availability)
- Encrypted data can no longer be retrieved
- ...

What mechanisms are available to determine “what” systems have been impacted so corrective operations take place?

35.106. VPQ: Section 10.7, page 68, Draft SP 800-130

Original text:

> *A major disaster would imply that large numbers of operational keys and metadata were lost or corrupted beyond recovery from primary storage. If a key retrieval or key recovery system exists, then the keys and metadata could be restored. However, if the keys were not backed-up or escrowed, then they would have to be replaced with new keys and the information that the original keys protected may be lost.*

Observation:

The above text assumes that online real-time mirroring is not performed.

A major disaster at **one** site could result in a large number of operational keys and metadata being lost or corrupted beyond recovery in a HSM at that location. However, if online real-time mirroring of HSM at two locations is performed, this means that no information may be lost. [Transaction based systems can ensure that keys are NOT USED before they are committed to multiple sites. Same with Policy changes. Client software can be designed to “re-submit” a metadata policy request if it did not commit. Zero loss.]

35.107.VPQ: Section 12.2, page 72, Draft SP 800-130

Original text:

> *12.2 Ease of Use*
>
> *Possibly the most significant constraint to the use of a CKMS is the difficulty that some systems present to the untrained user. Since most users are not cryptographic security experts and security is only a secondary goal for them, the CKMS needs to be as transparent as possible.*

Suggested additional text

> *User interfaces that coach users incrementally as they begin to use the CKMS could be of assistance. For instance, for new users/administrators, a wizard configuration[107] process could indicate: "What it's doing, why this question is important, what it means if the user try to subvert it / select something weak, ..." for each input request. However, it is important that fully trained administrators can perform the day-to-day routine tasks without constant lecturing and hand-holding by a Wizard tool.*

35.108.VPQ: Section 12.2.1, page 72, Draft SP 800-130

Original text:

> *12.2.1 User Perceptions, Prejudices, and Premonitions*
>
> *Ease of use is very subjective. Something easy or obvious for one person may not be easy or obvious for another. Designers should keep in mind that users are not usually security experts so they may not understand the purpose of the security feature that they are operating.* ***Security is not usually the primary purpose of the product****. Past experiences, perceptions, and prejudices may taint a person's evaluation of a product. A large segment of the potential user population needs to be satisfied with a security product, including that it is easy to use, for it to be widely procured and used.*

Proposed revised text for the bold text selected above

> *Security may not be the primary motivation of a user. Security may in fact be perceived as a significant barrier to the user achieving some other more interesting objective. Security may be no more than a 'tick the box exercise' for that user. In some unfortunate cases the pre-tense of employing security may be more about liability shifting[108], than actually achieving security for all stake-holders in practice.*

35.109.VPQ: Section 12.5, page 73, Draft SP 800-130

Original text:

---

[107] http://en.wikipedia.org/wiki/Wizard_(software)

[108] Anderson, R. J. Liability and computer security: Nine principles. In ESORICS '94: Proceedings of the Third European Symposium on Research in Computer Security (London, UK, Nov. 1994), vol. 875 of LNCS, Springer-Verlag, pp. 231–245. Available at http://www.cl.cam.ac.uk/~rja14/Papers/liability.pdf

12.5 Technological Challenges

A CKMS should implement cryptographic algorithms as modules that can be replaced and updated without significantly affecting the rest of the implementation. In particular, block cipher parameters like key length and block length should be variable so that they may be increased if necessary.

Proposed revised text

A CKMS should implement cryptographic functions as modules that can be replaced by fundamentally different technologies without significantly affecting the rest of the implementation. e.g. A system using public key techniques for key exchange should be designed so symmetric techniques can be used instead. Alternatively a layered defence-in-depth strategy of employing both techniques should be used. With components such as data privacy, the system should be able to shift between block cipher and stream cipher mode of operations, support variable parameters such as key length and block length, allowing these things to be adjusted as necessary to support new primitives that may not exist yet.

END.

We Need Assurance!

Brian Snow
U. S. National Security Agency
bdsnow@nsa.gov

Abstract

When will we be secure? Nobody knows for sure – but it cannot happen before commercial security products and services possess not only enough functionality to satisfy customers' stated needs, but also sufficient assurance of quality, reliability, safety, and appropriateness for use. Such assurances are lacking in most of today's commercial security products and services. I discuss paths to better assurance in Operating Systems, Applications, and Hardware through better development environments, requirements definition, systems engineering, quality certification, and legal/regulatory constraints. I also give some examples.

1. Introduction

This is an expanded version of the "Distinguished Practitioner" address at ACSAC 2005 and therefore is less formal than most of the papers in the proceedings.

I am very grateful that ACSAC chose me as a distinguished practitioner, and I am eager to talk with you about what makes products and services secure.

Most of your previous distinguished practitioners have been from the open community; I am from a closed community, the U.S. National Security Agency, but I work with and admire many of the distinguished practitioners from prior conferences.

I spent my first 20 years in NSA doing research developing cryptographic components and secure systems. Cryptographic systems serving the U.S. government and military spanning a range from nuclear command and control to tactical radios for the battlefield to network security devices use my algorithms.

For the last 14 years, I have been a Technical Director at NSA (similar to a chief scientist or senior technical fellow in industry) serving as Technical Director for three of NSA's major mission components: the Research Directorate, the Information Assurance Directorate, and currently the Directorate for Education and Training (NSA's Corporate University). Throughout these years, my mantra has been, "Managers are responsible for doing things right; Technical Directors are responsible for finding the right things to do."

There are many things to which NSA pays attention in developing secure products for our National Security Customers to which developers of commercial security offerings also need to pay attention, and that is what I want to discuss with you today.

2. Setting the context

The RSA Conference of 1999 opened with a choir singing a song whose message is still valid today: "Still Haven't Found What I'm Looking For". The reprise phrase was . . . *"When will I be secure? Nobody knows for sure. But I still haven't found what I'm looking for!"*

That sense of general malaise still lingers in the security industry; why is that? Security products and services should stop malice in the environment from damaging their users. Nevertheless, too often they fail in this task. I think it is for two major reasons.

First, too many of these products are still designed and developed using methodologies assuming random failure as the model of the deployment environment rather than assuming malice. There is a world of difference!

Second, users often fail to characterize the nature of the threat they need to counter. Are they subject only to a generic threat of an opponent seeking some weak system to beat on, not necessarily theirs, or are they subject to a targeted attack, where the opponent wants something specific of theirs and is willing to focus his resources on getting it?

The following two simple examples might clarify this.

Example 1: As a generic threat, consider a burglar roaming the neighborhood wanting to steal a VCR. First, understand his algorithm: Find empty house

(dark, no lights) try door; if open, enter, if VCR – take. If the door is resistant, or no VCR is present, find another dark house.

Will the burglar succeed? Yes, he will probably get a VCR in the neighborhood. Will he get yours? What does it take to stop him? Leave your lights on when you go out (9 cents a kilowatt-hour) and lock your door. That is probably good enough to stop the typical generic burglar.

Example 2: As a targeted threat, assume you have a painting by Picasso worth $250,000 hanging above your fireplace, and an Art thief knows you have it and he wants it. What is his algorithm? He watches your house until he sees the whole family leave. He does not care if the lights are on or not. He approaches the house and tries the door; if open, he enters. If locked, he kicks it in. If the door resists, he goes to a window. If no electronic tape, he breaks the glass and enters. If electronic tape is present, he goes to the siding on the house, rips some off, then tears out the fiberboard backing, removes the fiberglass insulation, breaks though the interior gypsum board, steps between the studs, and finally takes the painting and leaves.

It takes more effort to counter a targeted threat. In this case, typically a burglar alarm system with active polling and interior motion sensors as a minimum (brick construction would not hurt either). With luck, this should be enough to deter him. If not, at least there should be increased odds of recovery due to hot pursuit once the alarms go off.

There is no such thing as perfect security; you need to know how much is enough to counter the threat you face, and this changes over time.

3. What do we need?

NSA has a proud tradition during the past 53 years of providing cryptographic hardware, embedded systems, and other security products to our customers. Up to a few years ago, we were a sole-source provider. In recent years, there has come to be a commercial security industry that is attractive to our customers, and we are in an unaccustomed position of having to "compete." There is nothing wrong with that. *If* industry can meet our customer's needs, so be it.

Policy and regulation still require many of our customers to accept Government advice on security products. However, they really press us to recommend commercial solutions for cost savings and other reasons. Where we can, we do so. However, we do not do it very often because we still have not found what *we* are looking for – assurance.

Assurance is essential to security products, but it is missing in most commercial offerings today. The major shortfall is absence of assurance (or safety) mechanisms in *software*. If my car crashed as often as my computer does, I would be dead by now.

In fact, compare the software industry to the automobile industry at two points in its history, the 1930s and today. In 1930, the auto industry produced cars that could go 60 mph or faster, looked nice, and would get you from here to there. Cars "performed" well, but did not have many "safety features." If you were in an accident at high-speed, you would likely die.

The car industry today provides air bags, seat belts, crush zones, traction control, anti-skid braking, and a host of other safety details (many required by legislation) largely invisible to the purchaser. Do you *regularly* use your seat belt? If so, you realize that users *can* be trained to want and to use assurance technology!

The software security industry today is at about the same stage as the auto industry was in 1930; it provides performance, but offers little safety. For both cars and software, the issue is really assurance.

Yet what we need in security products for high-grade systems in DoD is more akin to a military tank than to a modern car! Because the environment in which our products must survive and function (battlefields, etc.) has malice galore.

I am looking forward to, and need, convergence of government and commercial security products in two areas: assurance, and common standards. Common standards will come naturally, but assurance will be harder – so I am here today as an evangelist for assurance techniques.

Many vendors tell me that users are not willing to pay for assurance in commercial security products; I would remind you that Toyota and Honda penetrated U.S. Markets in the 70's by differentiating themselves from other brands by improving reliability and quality! What software vendor today will become the "Toyota" of this industry by selling robust software?

4. Assurance: first definition

What do I mean by assurance? I'll give a more precise definition later, but for now it suffices to say that assurance work makes a user (or accreditor) more confident that the system works as intended, without flaws or surprises, even in the presence of malice.

We analyze the system at design time for potential problems that we then correct. We test prototype devices to see how well they perform under stress or when used in ways beyond the normal specification. Security acceptance testing not only exercises the product for its expected behavior given the expected

environment and input sequences, but also tests the product with swings in the environment outside the specified bounds and with improper inputs that do not match the interface specification. We also test with proper inputs, but in an improper sequence. We anticipate malicious behavior and design to counter it, and then test the countermeasures for effectiveness. We expect the product to behave safely, even if not properly, under any of these stresses. If it does not, we redesign it.

I want functions *and* assurances in a security device. We do not "beta-test" on the customer; if my product fails, someone might die.

Functions are typically visible to the user and commanded through an interface. Assurances tend to be invisible to the user but keep him safe anyway.

Examples would be thicker insulation on a power wire to reduce the risk of shock, and failure analysis to show that no single transistor failure will result in a security compromise.

Having seat belts in a car provides a safety function. Having them made of nylon instead of cotton is the result of assurance studies that show nylon lasts longer and retains its strength better in the harsh environment of a car's interior.

Assurance is best addressed during the initial design and engineering of security systems – not as after-market patches. The earlier you include a security architect or maven in your design process, the greater is the likelihood of a successful and robust design. The usual quip is, "He who gets to the interface first, wins".

When asked to predict the state of "security ten years from now," I focus on the likely absence of assurance, rather than the existence of new and wonderful things.

Ten years from now, there will still be security-enhanced software applications vulnerable to buffer overflow problems. These products will not be secure, but will be sold as such.

Ten years from now, there will still be security-enhanced operating systems that will crash when applications misbehave. They will not be secure either.

Ten years from now, we will have sufficient functionality, plenty of performance, but not enough assurance.

Otherwise, predicting ten years out is simply too hard in this industry, so I will limit myself to about five years. Throughout the coming five-year span, I see little improvement in assurance, hence little true security offered by the industry.

5. The current state of play

Am I depressed about this state of affairs? Yes, I am. The scene I see is products and services sufficiently robust to counter many (but not all) of the "hacker" attacks we hear so much about today, but not adequate against the more serious but real attacks mounted by economic enemies, organized crime, nation states, and yes, terrorists.

We will be in a truly dangerous stance: we will think we are secure (and act accordingly) when in fact we are not secure.

The serious enemy knows how to hide his activities. What is the difference between a hacker and a more serious threat such as organized crime? The hacker wants a *score*, and bragging rights for what he has obviously defaced or entered. Organized crime wants a *source*, is willing to work long, hard, and quietly to get in, and once in, wants to stay invisible and continue over time to extract what it needs from your system.

Clearly, we need confidence in security products; I hope we do not need a major bank-failure or other disaster as a wake-up call before we act.

The low-level hackers and "script-kiddies" who are breaking systems today and are either bragging about it or are dumb enough to be caught, are providing some of the best advertising we could ask for to justify the need for assurance in security products.

They demonstrate that assurance techniques (*barely*) adequate for a benign environment simply will not hold up in a malicious environment, so we *must* design to defeat malice. Believe me – there is malice out there, beyond what the "script-kiddies" can mount.

However, I do fear for the day when the easy threats are countered – that we may then stop at that level, rather than press on to counter the serious and pernicious threats that can stay hidden.

During the next several years, we need major pushes and advances in three areas: Scalability, Interoperability, and Assurance. I believe that market pressures will provide the first two, but not the last one – assurance.

There may or may not be major breakthroughs in new security functions; but we really do not need many new functions or primitives – if they come, that is nice. If they do not, we can make do with what we have.

What we really need but are not likely to get is greater levels of assurance. That is sad, because despite the real need for additional research in assurance technology, the real crime is that we fail to

use fully that which we already have in hand! We need to better use those confidence-improving techniques that we do have, and continue research and development efforts to refine them and find others.

I am not asking for the development of new science; the safety and reliability communities (and others) know how to do this – go and learn from them.

You are developers and marketers of security products, and I am sorry that even as your friend I must say, "Shame on you. You should build them better!" It is a core quality-of-implementation issue. The fact that teen-age hackers can penetrate many of your devices from home is an abysmal statement about the security-robustness of the products.

6. Assurance: second definition

It is time for a more precise definition. Assurances are confidence-building activities demonstrating that

1. $ The system's security policy is internally consistent and reflects the requirements of the organization,
2. $ There are sufficient security functions to support the security policy,
3. $ The system functions meet a desired set of properties and *only* those properties,
4. $ The functions are implemented correctly, and
5. $ The assurances *hold up* through the manufacturing, delivery, and life cycle of the system.

We provide assurance through structured design processes, documentation, and testing, with greater assurance provided by more processes, documentation, and testing.

I grant that this leads to increased cost and delayed time-to-market – a severe one-two punch in *today's* marketplace; but your customers are growing resistive and are beginning to expect, and to demand, better products *tomorrow*. They are near the point of chanting, "I'm mad as hell, and I'm not going to take it anymore!"

Several examples of assurance techniques come to mind; I will briefly discuss some in each of the following six areas: operating systems, software modules, hardware features, systems engineering, third party testing, and legal constraints.

7. Operating systems

Even if operating systems are not truly secure, they can at least remain benign (not actively malicious) if they would simply enforce a digital signature check on every critical module prior to each execution. Years ago, NSA's research organization wrote test code for a UNIX system that did exactly that. The performance degraded about three percent. This is something that is doable!

Operating Systems should be self-protective and enforce (at a minimum) separation, least-privilege, process-isolation, and type-enforcement.

They should be aware of and enforce security policies! Policies drive requirements. Recall that Robert Morris, a prior chief scientist for the National Computer Security Center, once said: "Systems built without requirements cannot fail; they merely offer surprises – usually unpleasant!"

Given today's common hardware and software architectural paradigms, operating systems security is a major primitive for secure systems – you will not succeed without it. This area is so important that it needs all the emphasis it can get. It is the current "black hole" of security.

The problem is innately difficult because from the beginning (ENIAC, 1944), due to the high cost of components, computers were built to share resources (memory, processors, buses, etc.). If you look for a one-word synopsis of computer design philosophy, it was and is SHARING. In the security realm, the one word synopsis is SEPARATION: keeping the bad guys away from the good guys' stuff!

So today, making a computer secure requires imposing a "separation paradigm" on top of an architecture built to share. That is tough! Even when partially successful, the residual problem is going to be covert channels. We really need to focus on making a secure computer, not on making a computer secure – the point of view changes your beginning assumptions and requirements!

8. Software modules

Software modules should be well documented, written in certified development environments, (ISO 9000, SEI-CMM level five, Watts Humphrey's Team Software Process and Personal Software Process (TSP/PSP), etc.), and *fully* stress-tested at their interfaces for boundary-condition behavior, invalid inputs, and proper commands in improper sequences.

In addition to the usual quality control concerns, *bounds checking* and *input scrubbing* require special attention. For bounds checking, verify that inputs are of the expected type: if numeric, in the expected range; if character strings, the length does not exceed the internal buffer size. For input scrubbing, implement reasonableness tests: if an input should be a single word of text, a character string containing multiple words is wrong, even if it fits in the buffer.

A strong quality control regime with aggressive bounds checking and input scrubbing will knock out the vast majority of today's security flaws.

We also need good configuration control processes and design modularity.

A good security design process requires review teams as well as design teams, and no designer should serve on the review team. They cannot be critical enough of their own work. Also in this world of multi-national firms with employees from around the world, it may make sense to take the national affinity of employees into account, and not populate design and review teams for a given product with employees of the SAME nationality or affinity. Half in jest I would say that if you have Israelis on the design team put Palestinians on the review team; or if Germans are on one, put French on the other. . . .

Use formal methods or other techniques to assure modules meet their specifications exactly, with no extraneous or unexpected behaviors – especially embedded malicious behavior.

Formal methods have improved dramatically over the years, and have demonstrated their ability to reduce errors, save time, and even save dollars! This is an under-exploited and very promising area deserving more attention.

I cite two examples of formal methods successes: The Microsoft SLAM static driver verifier effort coming on line in 2005, and Catherine Meadows' NRL Protocol Analyzer detecting flaws in the IKE (Internet Key Exchange) protocol in 1999. You may have your own recent favorites.

As our systems become more and more complex, the need for, and value of, formal methods will become more and more apparent.

9. Hardware features

Consider the use of smartcards, smart badges, or other hardware tokens for especially critical functions. Although more costly than software, when properly implemented the assurance gain is great. The form-factor is not as important as the existence of an isolated processor and address space for assured operations – an "Island of Security," if you will. Such devices can communicate with each other through secure protocols and provide a web of security connecting secure nodes located across a sea of insecurity in the global net.

I find it depressing that the hardware industry has provided hardware security functionality (from the Trusted Platform Group and others) now installed in processors and motherboards that is not yet accessed or used by the controlling software, whether an OS or an application.

10. Security systems engineering

How do we get high assurance in commercial gear?

a) How can we trust, or

b) If we cannot trust, how can we safely use, security gear of unknown quality?

Note the difference in the two characterizations above: *how we phrase the question may be important*. For my money, I think we need more focus on how to use safely security gear of unknown quality (or of uncertain provenance).

I do not have a complete answer on how to handle components of unknown quality, but my thoughts lean toward systems engineering approaches somewhat akin to what the banking industry does in their systems. No single component, module, or person knows enough about the overall transaction processing system to be able to mount a successful attack at any one given access point. To be successful the enemy must have access at multiple points and a great deal of system architecture data.

Partition the system into modules with "blinded interfaces" and limited authority where the data at any one interface are insufficient to develop a complete attack. Further, design cooperating modules to be "mutually suspicious," auditing and alarming each other's improper behavior to the extent possible.

For example: if you are computing interest to post to accounts there is no need to send the complete account record to a subroutine to adjust the account balance. Just send the current balance and interest rate, and on return store the result in the account record. Now the interest calculating subroutine *cannot* see the data on the account owner, and therefore cannot target specific accounts for theft or other malicious action. We need to trust the master exec routine, but minimize the number of subroutines we need to trust. Yes, I know this is over-simplified, but you get my drift.

In addition, to guard against "unintended extra functionality" within given hardware modules or software routines, the development philosophy needs to enforce something akin to "no-lone zones" in that no single designer or coder can present a "black-box" (or proprietary?) effort to the system design team that is tested only at its interfaces and is then accepted.

Review all schematics and code (in detail, line by line) for quality and "responsive to stated requirement" goals. This review should be by parties independent of the designer. This is expensive, but not

far from processes required today in many quality software development environments to address reliability and safety concerns.

This of course requires all tools (compilers, CAD support, etc.) used in the development environment to be free of malice; that can be a major hurdle and a difficult assurance task in and of itself (remember the Thompson compiler in "Reflections on Trusting Trust, CACM 1983)!

The "Open Source" movement may also provide value in this area. There are pluses and minuses with open source, but from the security viewpoint, I believe it is primarily a plu**s.**

Further architectural constraints may be imposed to make up for deficiencies in certain modules. Rather than (or in addition to) encryption in application processes prior to transmission to other sites which could be bypassed or countered by a malicious operating system, you might require site-to-site transmissions to go through an encrypting modem or other in-line, non-bypassable link encryptors.

Link encryption in addition to application layer encryption is an example of a "Defense in Depth" strategy that attempts to combine several weak or possibly flawed mechanisms in a fashion robust enough to provide protection at least somewhat stronger than the strongest component present.

Synergy, where the strength of the whole is greater than the sum of the strength of the parts, is highly desirable but not likely. We must avoid at all costs the all-too-common result where the system strength is less than the strength offered by the strongest component, and in some worst cases less than the weakest component present. Security is so very fragile under composition; in fact, secure composition of components is a major research area today.

Good *system* security design today is an art, not a science. Nevertheless, there are good practitioners out there that can do it. For instance, some of your prior distinguished practitioners fit the bill.

This area of "safe use of inadequate components" is one of our hardest problems, but an area where I expect some of the greatest payoffs in the future and where I invite you to spend effort.

11. Third party testing

NIST (and NSA) provide third-party testing in the National Information Assurance Partnership Laboratories (NIAP labs), but Government certification programs will only be successful if users see the need for something other than vendor claims of adequacy or what I call "proof by emphatic assertion – Buy me, I'm Good."

If not via NIST or other government mechanism, then the industry must provide *third-party* mediation for vendor security claims via consortia or other mechanisms to provide *independent* verification of vendor claims *in a way understandable by users*.

12. Market/legal/regulatory constraints

Market pressures are changing, and may now help drive more robust security functionality. The emergence of e-commerce in the past decade as a driver for secure internet financial transactions is certainly helpful, as is the entertainment industry's focus on digital rights management. These industries certainly want security laid on correctly and robustly!

I hope citizens will be able to use the emerging mechanisms to protect personal data in their homes, as well as industry using the mechanisms to protect industry's fiscal and intellectual property rights. It is simply a matter of getting the security architecture right.

I wonder if any of the industry consortia working on security for digital rights management and/or electronic fiscal transactions have citizen advocates sitting on their working groups.

Lawsuits might help lead to legal "fitness-for-use" criteria for software products – much as other industries face today. This could be a big boon to assurance – liability for something other than the quality of the media on which a product is delivered!

Recall that failure to deliver expected functionality can be viewed, in legal parlance, as providing an "attractive nuisance" and is often legally actionable.

One example is a back yard swimming pool with no fence around it. If a neighbor's child drowns in it, you can be in deep trouble for providing an attractive nuisance. Likewise, if you do a less than adequate job of shoveling snow from your walk in winter (providing the appearance of usability) you can be liable if someone slips on the ice you left on the surface. Many software security products today are attractive nuisances!

All you need do is to Google "Software Quality Lawsuits" or a similar phrase, and you can find plenty of current examples of redress sought under law for lack of quality in critical software. Do not attempt to manage defects in software used in life-critical applications. Remove them during the development and testing processes! People have died due to poor software in medical devices, and the courts are now engaged; the punitive awards can be significant.

One example of a lawsuit already settled: *General Motors Corp. v. Johnston* (1992). A truck stalled and was involved in an accident because of a defect in a PROM, leading to the death of a seven-year old child. An award of $7.5 million in punitive damages against GM followed, in part due to GM knowing of the fault, but doing nothing.

There are social processes outside the courts that can also drive vendors toward compliance with quality standards.

One of the most promising recent occurrences in the insurance industry was stated in the report of Rueschlikon 2005 (a conference serving the insurance industry). Many participants felt that, "The insurance industry's mechanisms of premiums, deductibles, and eligibility for coverage can incent best practices and create a market for security . . . This falls in line with the historic role played by the insurance industry to create incentives for good practices, from healthcare to auto safety . . . Moreover, the adherence to a set of best practices suggest that if they were not followed, firms could be held liable for negligence."

Bluntly, if your security product lacks sufficient robustness in the presence of malice, your customers will have to pay more in insurance costs to mitigate their risks.

How the insurance industry will measure best practices and measure compliance are still to be worked out, but I believe *differential* pricing of business disaster recovery insurance based in part on quality/assurance (especially of security components) is a great stride forward in bringing market pressure to bear in this area!

13. Summary

In closing, I reiterate that what we need most in the future is more assurance rather than more functions or features. The malicious environment in which security systems must function *absolutely requires* the use of strong assurance techniques.

Remember: most attacks today result from failures of assurance, not failures of function.

Rather than offer predictions, try for a self-fulfilling prophecy – each of us should leave this conference with a stronger commitment to using available assurance technology in products! It is not adequate to *have* the techniques; we must *use* them!

We have our work cut out for us; let's go do it.

In closing, I would like to thank Steven Greenwald, Brad Martin, and Greg Shipley for their insights and help in preparing this article.