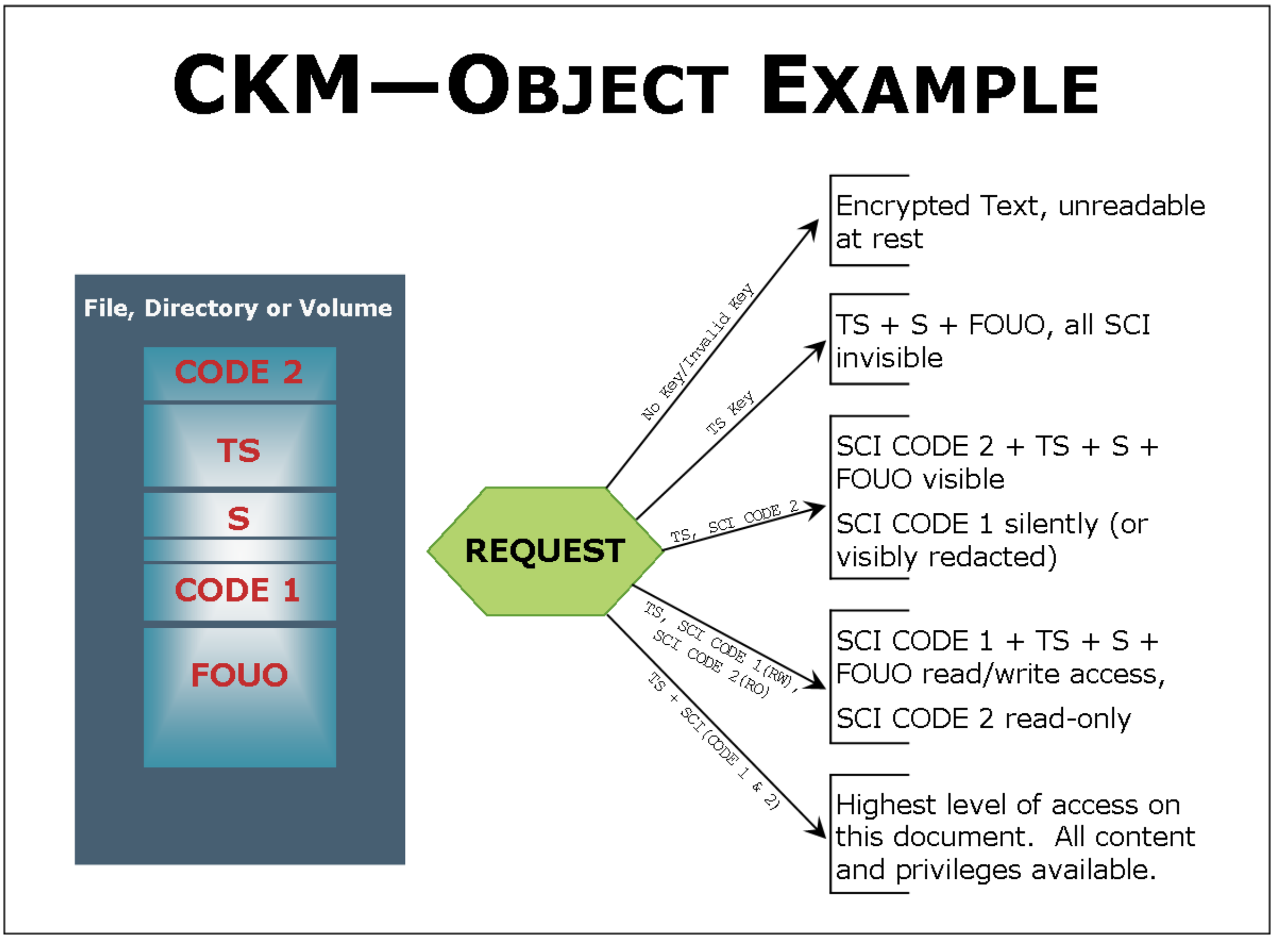
TEC SEC

Providing the Transition
from Communication Security
to Information Security

Persistent Protection of Objects with Object Oriented Key Management

Persistent protection with encryption of data itself is a logical next step for firewall network enhancement. Encryption can be viewed in various means and has surfaced as an essential element for protecting information exchanges, for policy enforcement, and for differentiated attribute accesses.

In general, protecting data may be found in traditional secure network tunneling with a Public Key Infrastructure providing the key management support. A movement to persistent protection encryption would entail creating self-protecting data objects. The financial services, in the form of an ANSI x9 standard, has published x9.73 which sites a dynamic key management schema called Constructive Key Management® CKM®. The inherent Object Oriented Key management encryption schema is a U.S. Navy SPAWAR SBIR phase 3 award-winner.


CKM—Object Example
File, Directory or Volume
CODE 2
TS
S
CODE 1
FOUO
REQUEST
No Key/Invalid Key
TS Key
TS, SCI CODE 2
TS, SCI CODE 1(RW), SCI CODE 2(RO)
TS + SCI(CODE 1 & 2)
Encrypted Text, unreadable at rest
TS + S + FOUO, all SCI invisible
SCI CODE 2 + TS + S + FOUO visible
SCI CODE 1 silently (or visibly redacted)
SCI CODE 1 + TS + S + FOUO read/write access,
SCI CODE 2 read-only
Highest level of access on this document. All content and privileges available.

In the above example, a file/directory/volume or any sub-object is illustrated. At rest, the object is encrypted and no care/maintenance is required on the bits [it is only encrypted once, though the data may have users at many different access levels]. Only one instance/copy of the object is needed for all supported access levels, rather than multiple instances at each possible access level. In the field, this could be a storage unit in a forward-operating base – if the base were overrun, no specific data destruction policy would need to be completed.

Upon a request (authorized or unauthorized) a specific sequence of key descriptors is applied before a key is ever requested. This sequence of key descriptors is basically an "any-to-any" set (multidimensional matrix) of field selectors that are configured a priori. In a traditional model we could describe hierarchical access modes like "CONFIDENTIAL, SECRET, TOP SECRET, SCI" and modes like SCI where various levels are exclusive and non-additive are difficult to enforce and can have a heavy key management burden. Upon expiration of a key, or revocation of access, all data has to be re-encrypted to the new level of access.

The key descriptors (attributes) add an important additional functionality, access modes that were heretofore unimaginable at an object level (time of day, NATO, NOFORN, geographic/gps location, multiple party/shared key, etc) can be enforced at the time the data object is accessed in any logical method (such as AND, OR). Roles or rules can be applied to the attributes.

Totally disparate security models may intersect – for example Agency A and B are sharing information but not with C (e.g. posse comitatus). Even algorithms that perform the encryption can be set at matrix creation time (e.g. AES for NOFORN and EC for NATO). The attributes associated with the encryption schema can further differentiate access.

On a large scale, such as an Agency SAN, this level of protection would be available for every object at all times. Data visible to an Executive might show all aspects of a project. When an engineering team reviews it, accounting data might be automatically redacted (or vice-versa). HR data might be invisible to the line executive but not to an HR executive.

When implemented as part of a file system, the encryption schema becomes transparent as it is absorbed into an existing, heterogeneous environment. The files can be copied/moved/backed up without concern for whom or how. Identity security can be treated separate from access control. Existing encrypted channels can continue to exist, but with the added encrypted files. Since a single object can serve many views, significant data duplication is eliminated and version control is done from a single document.

In summary, the CKM schema provides several objectives: 1) an encryption environment which can support multiple forms of collateral on a common platform at the granularity of an object; 2) Compartmentation by algorithmic access with hierarchical access defined by attributes; and 3) Self-protecting data objects are created which are data label aware, and services can be exerted based on that awareness.