Applying Enterprise Security Policy and Key Management

WHITE PAPER

Executive Overview

This white paper is the second in a series following An Enterprise Guide to Understanding Key Management which introduces different types of cryptography and keys used in modern data protection applications. In addition the guide provides a brief summary of the main key management elements used within the infrastructure. Please refer to it for background material to this white paper.

The information presented in this white paper discusses various approaches to cryptography and key management. Taking a proactive approach to data protection—planning, policies, and process—results in a smoother implementation and a positive return on investment. Unlike disparate, multi-vendor point solutions that can create limited "islands" of security, SafeNet's approach provides an integrated security platform with centralized policy management and reporting. This is ideal for seamless, cost-efficient management of encrypted data across databases, applications, networks, and endpoint devices. Centralized encryption and key management also provides a uniform and ubiquitous way of protecting data while reducing the cost and complexity associated with compliance and privacy requirements.

Each business has its unique network and operational requirements, which results in the need for tailored key management policies. While policies can be based on standardized specifications, it is a best practice to conduct a comprehensive risk assessment to reveal specific points to consider in designing key management policies and procedures.

End-to-End Security in the Infrastructure

Applications of Key Management

We begin this white paper by moving from the key management basics found in the white paper, An Enterprise Guide to Key Management where we introduced different types of cryptography and keys used in modern data protection applications and touched on the challenges associated with managing huge numbers of keys under a variety of security policies. It also provided a brief summary of the main key management elements used within the infrastructure. (They are referenced in Figure 1 below).

The application types presented in the following sub-sections are categorized as data-in-motion (with sub-categories of transaction-based and messaging applications) and data-at-rest protection. The applications are categorized in this way to focus on protection of the data content associated with them rather than protection of the media used to transport and/or store the data. The categories are distinguished as follows:

- Transaction-based applications involve the real-time exchange of data; this is commonly described as request and response. This category involves the automatic processing of data resulting in real-time changes to the state of the system. The data being exchanged is generally small in size. Data is typically protected based on the application handling the data.
- Messaging applications are sometimes referred to as "store-and-forward". In general, they are not real-time in nature and do not result in automated state changes. A side characteristic of this application category is that the data involved can be of any size and very large documents or messages are not uncommon. Data is typically protected for a specific set of authorized entities.
- Data-at-rest protection, as the name implies involves the protection of data confidentiality and/or integrity in a static or "stored" environment. This could be associated with very large storage solutions in data centers or Storage Area Networks or with disk and file protection on workstations or servers. Data can be of any size and could require protection for very long periods of time or could be shorter-lived (e.g., weeks or months). Data may be protected based on the application or on a set of authorized entities.

Key Management for Transaction-Based Applications

Two major transaction-based application types driving key management requirements are payments systems and Electronic Data Interchange (EDI). Until recently, key-management for transaction-based systems has exclusively involved symmetric keys. Symmetric keys are typically maintained by each organization on a per-trading partner or per-network link basis. For each connection (trading partner or processing network), organizations maintain at least two distinct symmetric keys – one for data encryption and one for data integrity and authentication (HMAC). In some cases, four keys are mandated – one for each traffic direction and for each purpose on each connection. When the number of connections becomes large, key management can quickly become extremely challenging.

Key Management for Messaging Applications

The term "messaging" in this case indicates a broad range of applications from e-mail to document processing and Web-based systems. Messaging security requirements are applied end-to-end at the application level. Because of the end-to-end nature of the applications and the need in many cases to establish keying relationships on an as-required basis, it is difficult to rely on a fixed set of pre-established symmetric keys similar to the transaction-based system described above. Key management is, therefore, largely based on asymmetric cryptographic techniques such as Diffie-Hellman key agreement or RSA-based key transport.

Key Management for Data-at-Rest Protection

Data at rest applications can vary greatly from localized file and disk encryption through to securing large database systems and storage networks. In this context, key management ensures the continual protection of data. Issues such as establishing and maintaining appropriate key assignments/allocations and optimal key rotation schedules can be crucial in large deployments.

Challenges of Key Management

- Security
- Scale
- Data Loss Protection
- Lack of Standard for Policy Definition
- Lack of Standard for Policy Enforcement
- Variety of End-point Solutions
- Variety of Implementation Technologies
- Satisfying Business Requirements
- Key Management Responses

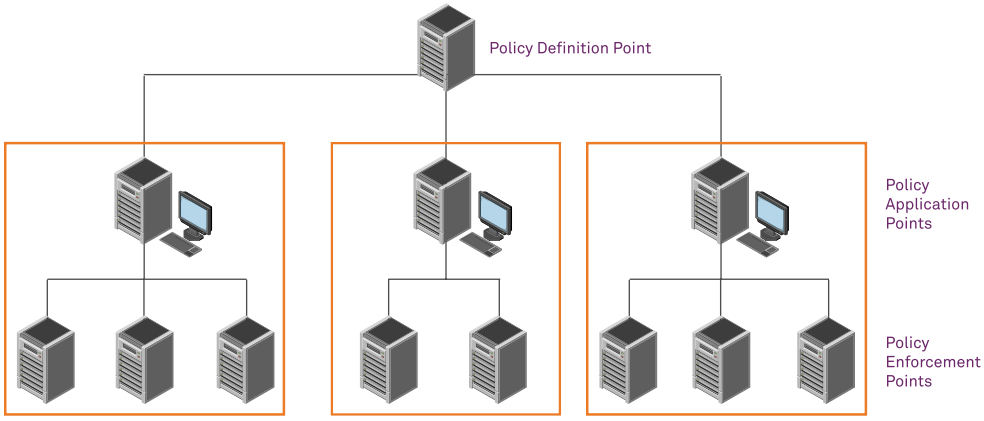

Figure 1 Key Management Elements

Key Management Challenges and Responses

The following sub-sections describe the challenges when considering the requirements for enterprise key management.

Key Management Challenges

Security

Establishing and maintaining adequate security throughout a key management system has been the number one challenge since the inception of cryptography. In the digital world, security challenges arise not only in the traditional areas of access control, personnel security and proper procedures, but also in the form of hackers/crackers searching the system for any sign of weakness and sophisticated cryptanalytic attacks on the algorithms used to protect the sensitive data. It is important, therefore, for the system to have the appropriate robust security controls and also be flexible to new developments in the technology.

Scale

One of the greatest challenges inherent to large numbers of end-points communicating or sharing data securely is the sheer number of keys requiring effective key management. Along with the numbers of endpoints involved, there is the potentially greater issue of managing the huge number of possible inter-relationships and their dynamic nature. The lack of scalability is a major limiting factor in the deployment of cryptographic security solutions. It is usually possible to deal with one dimension of the scalability challenge or the other – one can either limit the number and allow for a reasonable amount of freedom in the inter-relationships or one can allow large numbers by severely restricting the nature of the inter-relationships between end-points.

Data Loss Protection

The potential loss of data due to a failure in the key management system can cause serious concern. It is almost axiomatic that the data resources of most value to the enterprise are the ones most in need of cryptographic protection. Because of its value it is preferred this data must never be lost and it is incumbent on the design of the key management system to provide failure-proof (or, at least, fail-safe) mechanisms.

Lack of Standard for Policy Definition

An emerging challenge is the lack of, or the lack of agreement on, standards-based approaches allowing deployment of true enterprise key management solutions. In particular, policy definition is entirely proprietary, where it even exists. Since enterprise-level policy is fundamental to effective enterprise key management, it requires urgent attention. Policy definition must include the definition of assets, entities and access modes and the relationships between them in suitable to a highly dynamic environment. The policies must be defined in way that is intuitive to the security administrator/professional and the output must be readily interpreted by any lower level key management components.

Lack of Standard for Policy Enforcement

On the other side of the coin, even with standards-based policies being delivered to the lower level components, there is no approach to key management for policy enforcement that is agreed upon. For example, how and where do keys get generated? What is the standard that governs fundamentals such as random number generation, key storage and tamper response? What is the standard for key strengths to enforce the various policy levels? It is important for an enterprise to answer such questions before it can entrust the security of its most valuable assets to a centrally managed system. An important facet of this challenge is also related to the next point: For the lower level key management components and end-point solutions to properly enforce a specified security policy, they must be able to interoperate in a meaningful way. If one component can interpret and act upon the defined policy but others around it cannot cooperate in enforcing the policy, enforcement will be unreliable at best.

Variety of End-point Solutions

The challenges in this area come in two main categories:

- Different types of solutions. A typical system installation contains a number of different types of end-point solutions tailored for particular functions. Each one can only act upon a subset of the overall policy. How are the various policy subsets allocated to the various types? Is it possible to ensure that the coverage is complete – that there are no gaps in policy enforcement? This challenge exists even when the end-points are all provided by a single supplier.
- Different solution providers. Introducing solutions from multiple suppliers compounds the challenges in this area immensely. Can each of the end-point solutions be minimally managed at the device level, without even considering the implications for centralized key management policy enforcement? Without standards for policy definition, interpretation and enforcement, it is currently technically infeasible to bring the end-points together to enforce any sort of coherent enterprise-level key management policy.

Variety of Implementation Technologies

Another potentially complex challenge for enterprise-level key management results from the variety of implementing technologies employed in components at all levels. One could have, for example, dedicated hardware cryptographic modules, network appliances and software modules running on Windows servers all part of one system. The challenge is two-fold:

- Understanding the strengths and weaknesses of the different technologies and how each can best contribute to policy enforcement, and
- Finding a common, trusted means of communication between the different technologies.

Satisfying Business Requirements

Although this challenge is discussed last, it is certainly not least. Satisfying the business requirements is the first priority in any system deployment and the challenge is not limited to just security or key management systems. Security (and cryptography in particular) is however, not generally well understood and its place in enabling the delivery of critical business services is under appreciated. It is frequently perceived as being a stumbling block on the path to progress, cumbersome to use and expensive. For any enterprise key management system to succeed, it is vital that it be seen as a true enabler that is flexible, easy to use and understand and that can demonstrate an ability to reduce resource overhead.

Key Management Responses

Regardless of application type, there is a set of well-established techniques that have been developed over the years that can be applied to address most key management problems. Thus, it is not generally a question of developing a new key management solution in a particular situation; rather, it is a question of finding the most appropriate mix of the established solutions to satisfy the needs of a particular application or group of applications.

Key management has a long history, particularly with symmetric keys used in applications for data encryption and data integrity that has been punctuated by several spectacular failures – each of which has provided an opportunity to learn and improve. Based on the lessons learned, history has identified a number of factors paramount to successful key management. Each of them features prominently in describing the techniques presented in this section.

These critical factors are:

- High quality random number generation
    - This is fundamental to the security of key generation, particularly given the processing power available to potential attackers today.
- Tight control over the ways in which keys are used
    - Allowing keys to be used for more than one purpose, or allowing multiple keys to be used to encrypt a given plaintext, can result in inadvertent or deliberate disclosure of sensitive data.
- Multi-party control of key distribution and key entry
    - One of history's lessons is that it is simply too easy to compromise the security of the key management system if keys can be held and manipulated by one person acting alone.
- Secure storage and destruction of keys, particularly post-use
    - An important aspect of key management that can be easily overlooked is that the keys are often more valuable after they have been used than they are before.
- Strict accountability through each stage of the lifecycle
    - The management of keys can be very complex, requiring that a number of people all perform their individual tasks correctly in order to maintain the security of the system. In the event that something does go wrong, it is critical to be able to accurately re-create the sequence of steps leading to the failure.

The following sub-sections present some of the widely used key management techniques, particularly ones that are relevant to development of a policy-driven approach. The list is, therefore, meant to be representative but, by no means, exhaustive.

Common Key Management Techniques

Some of the techniques common to the management of both symmetric and asymmetric keys are described briefly below.

Centralized Generation: Centralized generation of symmetric keys and asymmetric keys pairs may be employed to meet one or both of the following requirements:

- It may be necessary to securely archive symmetric keys and asymmetric private keys for decryption purposes to ensure continuity of access to keys. Centralized generation simplifies the archival process by having just one process responsible for ensuring keys are securely archived.
- Many of the end-point solutions are not capable of generating symmetric keys or asymmetric key pairs of sufficient quality to meet the demands of the operational environment, typically because they lack either or both the processing power to efficiently perform key pair generation or a high-quality random number generator.

Key Transport: To make keys available throughout the infrastructure, it is necessary to transfer them from the generation site to the locations where they will actually be used. A number of techniques have been developed for this purpose, some of which are applicable to both asymmetric and symmetric keys and some are only used with symmetric keys. Briefly, they are:

- Key splitting and multi-person control and key entry. This technique is often used when initially configuring a site to use a master key or transport key in the techniques described below.
- Key wrapping (encryption). In many cases, the operational keys must be changed on a relatively frequent rotation schedule and multi-person key split transfers are simply not practical. In such cases, it is possible to encrypt the key values (either symmetric or asymmetric) using a previously established transport key or using public key-based techniques.

Key Agreement: In some cases, the requirements for cryptographic data protection are not only very dynamic but also somewhat ad hoc in nature; for example when the details of the communicating parties and their data protection requirements are not known before hand. In situations like this, key agreement techniques, such as Diffie-Hellman, allow symmetric keys to be negotiated “on the fly” between two end-points.

Key Derivation: Another method for generating symmetric keys on an as-required basis is key derivation. Symmetric keys may be derived from a previously agreed secret using a strong one-way algorithm (typically one of the accepted digest algorithms) and some other data that is unique to the particular keying arrangement (e.g., names, serial numbers, IP addresses). In many key management schemes, the shared secret is a master key that has been established at each end-point using one of the key transport techniques.

Key Usage Designation: Most cryptographic systems provide a capability to specify and enforce the usages permitted for each key. It is then left to the system developers and/or system managers to ensure that key usage settings are consistent and that multiple usages are not permitted in situations that could lead to leakage of key values. An extension to this concept available in some systems is the ability to enforce a higher-level policy to ensure that certain usages are not permitted or that multiple key usages are not permitted.

Dedicated cryptographic modules: As has been stressed several times, protection of the keys used in cryptographic data protection is critical to ensuring the protection of the data. The most common method of ensuring key protection is the use of dedicated cryptographic modules. The cryptographic modules are designed to isolate all aspects of the keys’ lifecycle from other applications and processes that could, accidentally or deliberately, compromise key security.

Common Key Managment Techniques

- Centralized Generation
- Key Transport
- Key Agreement
- Data Loss Protection
- Key Derivation
- Key Usage Designation
- Dedicated cryptographic modules
- Enforced Key Management Roles
- Centralized Trust

To ensure a high degree of isolation, hardware modules are often used. Whether the module is implemented in software or hardware, it is generally advisable to look for a third-party validation of its key protection and key management capabilities, in general (e.g., FIPS 140-2, PCI-HSM).

Enforced Key Management Roles: A well-defined set of user/administrator roles ensures that adequate operational controls can be enforced throughout the key management lifecycle. Roles allow the organization to, for example, impose multi-person control over critical operations. They also allow for a high level of accountability for the performance of key management operations.

Centralized Trust: The notion of trust is not always explicitly considered in key management. Whether trust is established through human, procedural means (e.g., key splits delivered by trusted personnel) or through electronic means (e.g., a PKI Certification Authority), it is critical to the proper management of key material and, ultimately to the proper operation of the system being supported. With a proper level of trust, there is an assurance that data will be delivered correctly, without unauthorized modification or disclosure, to the individuals or other systems that require it. When trust breaks down, that assurance is lost and, it is no longer possible to rely on the data being correct and authentic nor is it possible to rely on the fact that sensitive data has not been delivered to unauthorized entities. Because of the importance of maintaining trust, a central "trust anchor" is often employed to provide a highly assured reference point for the system.

Proposed Solution Approach

The goal of this paper is to offer a change from a technology-driven approach, focused around the individual end-point solutions, to a policy driven approach that provides a much needed framework for cryptography (regardless of the details of the end-point solutions. To do this, it is essential to approach the solution from the top down, by laying out a robust key management framework, and from the bottom up, by defining the mechanisms required for end-points to fully participate within the framework

Policy Definition and Application

Policy Definition

The policy definition element of the key management framework must extend policy elements (explained in the preceding An Enterprise Guide to Key Management white paper). Policy definition involves the following:

1. Asset definition. This must be flexible enough to permit definition of assets within a number of categories and at various levels of detail. The Asset categories should include the following, at a minimum:

a. Keys
b. Certificates
c. Documents and transactions
d. Stored data – unstructured
e. Stored data – structured
f. Connections – link layer
g. Connections – network layer
h. Connections – session layer

For each asset category, an attribute template will provide a means for capturing the meta-data associated with each category. A sample template for the generic key category, based on the PKCS #11 attribute set, is presented in Table 3-1. For each key class (and possibly by algorithm also), additional attributes may also be required. The IEEE KMI also provides a data template that would be suitable as the basis for the "Stored data – unstructured" category.

Attribute	Choices
Meta-data Attributes applied for all key objects	
Template Version	Integer
Key Classname	Public, private, secret
Key Algorithm	e.g., RSA, ECDSA, AES
Key Generation Class	Central generation, Local generation, Derived, Entered (if Entered then Key import must be allowed)
Key Derivation Class (applies if Key Generation Class is Derived)	Password-based, Key Agreement, hared Secret
Key Storage Class	Software, Local Hardware, Central Hardware
Key Export allowed	True/False
Key Import allowed	True/False
Key Import Class (applies if Key import allowed)	Manual, Manual – split key, Manual – M of N, Electronic (i.e., key transport)
Key Authentication Class	Nil, Password – per session, Password – per use, Strong Auth – per session, Strong Auth – per use
Key Validity Class	Rotation, Fixed period
Key Usage Class	Encrypt, Decrypt, Sign, Verify, Integrity, Authentication (multiple choices are possible, depending on the Key Class)
Attributes for the meta-data	
Can be modified	True/False
Can be copied	True/False
Attributes applied per key object	
Unique ID	Integer
Alias	String
Key owner	Entity Unique ID
Key creation date	Date-Time
Key rotation period (if Validity class = Rotation)	Hours, Days, Months
Key validity start date (if Validity class = Fixed Period)	Date-Time
Key validity end date (if Validity class = Fixed Period)	Date-Time
Key Value	Binary

Table 3-1Generic Key Attribute Template

2. Entity Definition. Entity definition will also be based on the specification of a number of attributes that capture the important characteristics of the various entities within the system. The entity definitions can be grouped via classname to create roles (or equivalent in devices and agents). Entity categories will include the following:

a. Person

b. Device

c. Agent

A sample template for the Person category is presented in the following table .

Attribute	Choices
Meta-data Attributes applied for all persons	
Template Version	Integer
Person Classname (i.e., Role)	e.g., Administrator, Security Officer, Finance, Auditor, etc.
Authentication Class	Nil, Password, Strong Auth
Authentication Locality	Local, Domain, Global
Access Class (this would primarily be used to impose access restrictions)	Encrypt, Decrypt, Sign, Verify, Create, View, Modify, etc.
Flow Control Class	Nil, Container, Labeled
Cryptographic Capability	Nil, Authentication, Integrity, Sign, Verify, Encrypt, Decrypt
Attributes for the meta-data	
Can be modified	True/False
Can be copied	True/False
Attributes applied per person	
Unique ID	Integer
Name	String
Creation date	Date-Time
Validity start date	Date-Time
Validity end date	Date-Time
Flow Control Attribute (if Flow Control Class specified)	Container ID, Sensitivity Label

Table 3 -2 Person Attribute Template

3. Access Mode Definition. Each system application will typically have its own set of access modes that reflect the nature of the application. The access mode definition is, therefore, left to the designers of the system policy based on a set of descriptive attributes. In a deployed solution, however, it could still be useful to provide an initial set of access modes that could be used for the system or just serve as examples or starting points for the actual definitions.

Attribute	Choices
Meta-data Attributes applied for all access modes	
Template Version	Integer
Access Classname	e.g., Create, View, Delete, Modify, Transfer, Connect, Sign, Encrypt, etc.
Authentication Class	Nil, Password, Strong Auth
Authentication Locality	Local, Domain, Global
Data Flow Class	Entity-Asset, Asset-Entity, Entity-Entity. Asset-Asset
Data Flow Restriction	Controlled Access, Controlled Flow
Flow Control Class (if Controlled Flow)	Nil, Container, Labeled
Cryptographic Protection	Nil, Authentication, Integrity, Signature, Encryption
Attributes for the meta-data	
Can be modified	True/False
Can be copied	True/False
Attributes applied per access mode	
Name	String
Creation date	Date-Time
Flow Control Attribute (if Flow Control Class specified)	Container ID, Sensitivity Label
Cryptographic Algorithms	e.g., AES-128, RSA-2048, etc.
Flow Control Attribute (if Flow Control Class specified)	Container ID, Sensitivity Label

Table 3.3 Access Mode Attribute Template

4. Relationship Definition. In many respects, Relationship Definition is the missing piece required to solve the key management puzzle. The ability to bring together assets, entities and access modes in a meaningful way is crucial to key management policy definition. It allows the enterprise to define and, as discussed in the later sections, utilize cryptography to enforce policies across the various applications addressing the organizations' business needs. Each relationship can be defined using a set of attributes, similar to what has been described for assets, entities and access modes. The relationship must specify which entity category and class can interact with which asset category and class through which access mode and using which cryptographic mechanism. Basic rule checking, such as, "Does the entity have the cryptographic capability that matches the cryptographic protection required by the access mode?" can be readily accomplished.

Policy Application

This is the level within the infrastructure where the abstract policy defined as described above is turned into reality based on the concrete details of the installation in question. The Policy Application Point is, effectively, a key management server that must take the policy definitions, allocate portions of the policy to the appropriate policy enforcement elements, communicate the policy sub-elements and provide services, such as key generation to the lower level infrastructure elements. In a complex system, there be many of these components, each responsible for interpreting a particular subset of the policy.

Policy application may be automated using a set of fundamental key management operations and messages as are described in KMIP, for example. It may also involve the use of manual, procedural techniques for such operations as initial master/transport key provisioning. The most important factor in the design of the Policy Application Points is ensuring that the entire policy is interpreted and applied.

An important early step in defining the Policy Application layer is to build upon the work done in KMIP and the IEEE KMI, for example, and extend the specifications of key management operations and messages as necessary to ensure that all applications, not only data storage protection, can be fully accommodated. Development of use cases covering as broad a range of data protection applications as possible will be an essential part of the definition process.

One of the important roles of this level is establishing and maintaining trust. Only at this level is it known what the actual key management techniques are for permitting the necessary trust management techniques to be employed. Infrastructure elements, such as Certification Authorities and Master Key generation are components of the Policy Application level.

Policy Enforcement

In many practical environments, the end-point solutions may not be able to interpret and act upon the key management messages produced by the various Policy Application Points. To accommodate this reality, the Policy Enforcement Points act as agents, receiving the appropriate key management messages, interpreting them and, using custom-developed interfaces, instructing the end-point to perform the required operation – for example, to perform a re-key or to accept and load a new data integrity key. Policy Enforcement Points must also act as interpreters of the trust infrastructure for the end-points to validate the authenticity of the communications from the Policy Application level and the links between end-points. To allow them to perform this role effectively, they must be able to act upon data regarding the trustworthiness of other elements in the infrastructure, including the ability to accept new trust anchor data. Managers and end points have to have a way to figure out how to and how much to trust each other.

End-point Implications

Most end-point solutions are currently not capable of participating in this type of policy-driven key management system. For the foreseeable, therefore, it will be the role of the Policy Enforcement agents to bridge the gap between the key management system and the functions provided by the end-points. In the future, however, it will be necessary for end-points to include specific handling capabilities within their interfaces to accept and act upon the standardized key management messages and to participate fully within the trust infrastructure that is essential to successful key management.

Conclusion

The key management approach described in this paper and the proposed solution implementation that has been presented directly address the pressing need for cryptographic key management that is focused on satisfying enterprise security requirements by protecting data throughout its lifecycle and wherever it may be within the infrastructure. By taking this approach enterprise-level key management becomes an important enabler to solving business problems and not simply another piece of security technology.

There remains much to be done to fully standardize the policy definition language, the management operations that can be applied and the communications protocols and messages necessary to properly manage arbitrary end-point solutions. SafeNet, however, is fortunate to be able to present a variety of end-point solutions covering most aspects of an enterprise's security infrastructure with the ability to begin the adoption of the key management approach today and to grow the capability over the next few years in a way that provides increasing value over time.

About SafeNet

Founded in 1983, SafeNet is a global leader in information security. SafeNet protects its customers' most valuable assets, including identities, transactions, communications, data and software licensing, throughout the data lifecycle. More than 25,000 customers across both commercial enterprises and government agencies and in over 100 countries trust their information security needs to SafeNet.

Contact Us: For all office locations and contact information, please visit **www.safenet-inc.com**
Follow Us: www.safenet-inc.com/connected