WHITE PAPER

Key Security Domains for Enterprises

Enabling Robust Cryptographic Key Isolation Strategies

ENTRUST

SECURING A WORLD IN MOTION

Contents

Introduction

This white paper outlines how the Entrust Security World architecture and Entrust nShield hardware security modules (HSMs) enable enterprises to implement robust cryptographic key isolation strategies.

It discusses key concepts such as module-protected keys, access control lists, softcards, and operator card sets (OCSs) and explains how these mechanisms can be combined to enforce policy-driven sub-domains for secure key management.

Cryptographic keys within an enterprise are used to identify people and machines, secure internal and external communications, encrypt and tokenize data at rest, and sign messages and documents, as well as for other use cases. It is therefore vital for any business relying on cryptographic keys to have assurances and enforceable policies surrounding key usage. The nShield family of hardware security modules (HSMs) provides the ability to achieve that level of assurance.

By using the Security World key management framework, supported by the nShield HSM family, an organization can create for itself a structured key infrastructure that meets today's dynamic and fluid demands. This paper demonstrates how it is possible to easily configure Security World to permit sub-domains of cryptographic keys.

nShield is a family of multi-purpose HSMs that provide a trusted environment for secure cryptographic processing, key protection, and key management.

Business Model

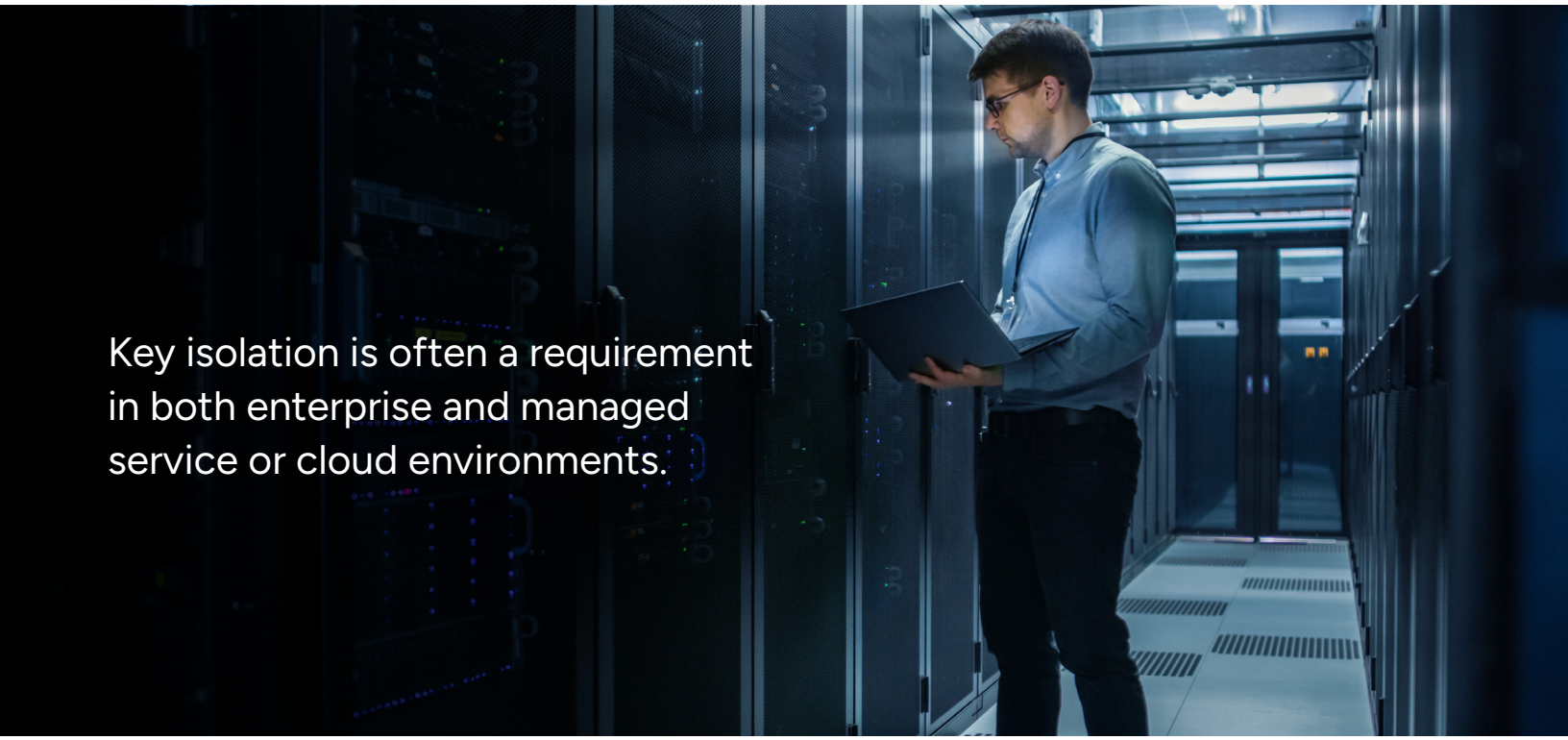
Security World key management addresses the primary business requirement where customers want to share cryptographic infrastructure resources between applications or departments within the same enterprise, while keys should be separated to preserve the necessary isolation between applications.

Designing and implementing a policy to meet business requirements is a straightforward process.

In addition to the requirements of the business model, there are several factors to consider when assessing techniques for key isolation:

- **Object Types**: Usually users, applications, or keys.
- **Scale**: This can range from one or two enterprise users or applications to millions of keys or customers.
- **Security**: What determines the true level of security? What authentication policies are protecting the application key material? How are physical security controls mapped to logical controls, and vice versa?
- **Accessibility**: What access does the hosting organization have to customers' material? Within an enterprise, a provider may want to provide a super-user or an administrative quorum with access to all the keys. However, customers will have more trust in public service if the provider can't access their keys.

With all these factors in mind, how can Entrust nShield HSMs assist in the development of isolated systems for the control of keys?



Security World

To understand how Entrust nShield HSMs can be deployed to support flexible isolated environments, we first need to have a clear understanding of some Entrust Security World architecture principles.

To alleviate the developer from the burden of creating a key infrastructure, Entrust provides the Security World architecture, which is a simple yet flexible way to create and manage application keys, protected in various ways, while also providing integrated load-balancing and disaster recovery functionality. Entrust also provides industry-standard APIs to the Security World architecture such as: PKCS#11, Microsoft CAPI-NG, JCE, and RESTful like.

It is assumed that nShield HSM users leverage this standard key infrastructure, whether they are integrating with existing standard interfaces or custom applications.

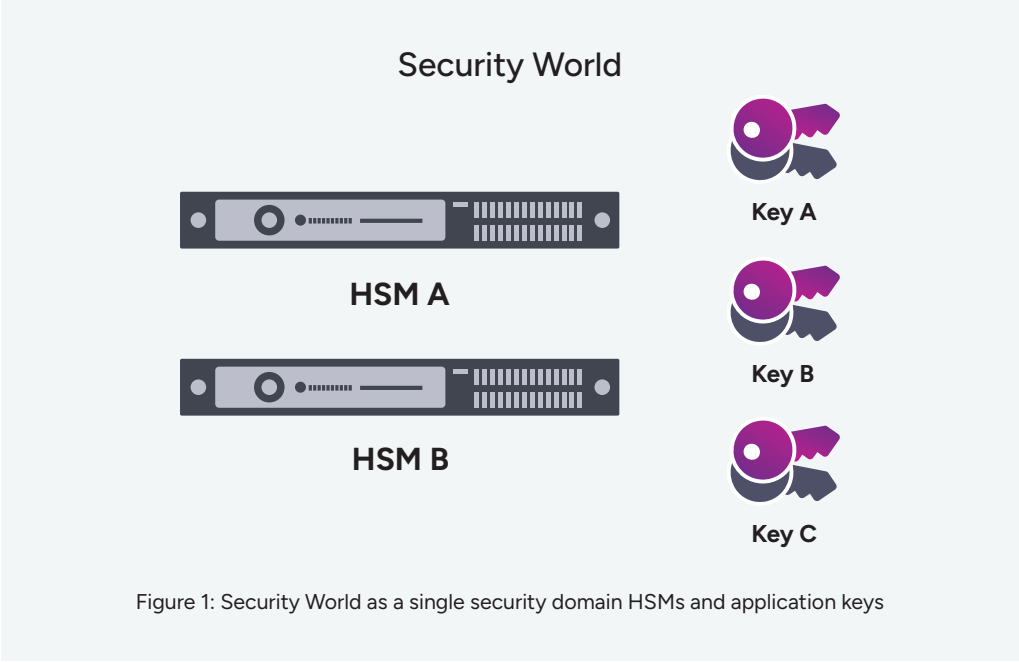


Figure 1: Security World as a single security domain HSMs and application keys

Practically speaking, a Security World creates a security domain for keys and objects to be managed across many HSMs and clients. An nShield HSM can only be configured with a single Security World at a time, but this white paper defines how it can also support multiple security sub-domains.

A Security World is the cornerstone for understanding the layers of isolation that can be achieved using nShield HSMs. The creation of a Security World begins with the creation of the Module Key. This key is encrypted under a logical token that is generated inside the HSM at the time of generating the Security World. The token is then written in parts across a set of smart cards, called the administrator card set or ACS. A standard secret sharing algorithm is used to build the parts of the ACS token. A minimum number of these cards, called a quorum, is required to access and reconstruct the Module Key and subsequently perform certain restricted administrative functions, such as adding the module key to another HSM. The notation used to describe the quorum of the card set is “K of N,, where N is the total number of cards in the card set and K (K>0) is the number of cards required to form a quorum. Note that adding an HSM to an existing Security World requires only the ACS quorum to configure a new HSM by adding their module key to it. This operation can also be performed to an existing HSM by resetting it and adding it to a new Security World, per the process described above.

Application Key Tokens

The most basic administrative function in (and the fundamental purpose of) a Security World is the generation of application keys, which are used by the authorized clients of a specific application to access the HSM, and requesting services, such as encryption/decryption, signing, etc. In general, authorized HSM users can generate new application keys; however, an additional authorization is required in case of strict FIPS or Common Criteria operating mode. When generating an application key within Security World on an nShield HSM, it is imperative that the raw key material is protected by the certified hardware module at all times. It is also important that the key can be loaded by authorized clients and backed up using industry best practice guidelines.

Security World mechanisms take the raw application key material, along with various metadata about how the key can be used – the access control list (ACL) – and cryptographically "wrap" it (encrypt it), using the module key. These application key tokens can then be distributed to and stored on all authorized HSM clients so that they can load the key at a later date. The wrapped keys can be safely backed up onto industry standard recovery media, since the key is encrypted with the module key, and can only be decrypted inside an nShield HSM belonging to the same Security World.

These module-protected keys are the standard and simplest domain of key isolation in a Security World. Each application that needs the services of the HSM can have its own module-protected key created. Only authorized clients for that key can load it to an HSM in the Security World, and the ACL, defined by the business rules of the application owners, governs how the clients can use the key, with varying levels of granularity.

Note that, since the application keys are stored in the data files of the authorized clients, the volume of keys that a Security World can protect is virtually unlimited – the limit is based on the external storage availability, not on the HSM memory. This neatly addresses the topic of "scale" when considering key isolation principles.

Also note that all module-protected keys are recoverable by the ACS.

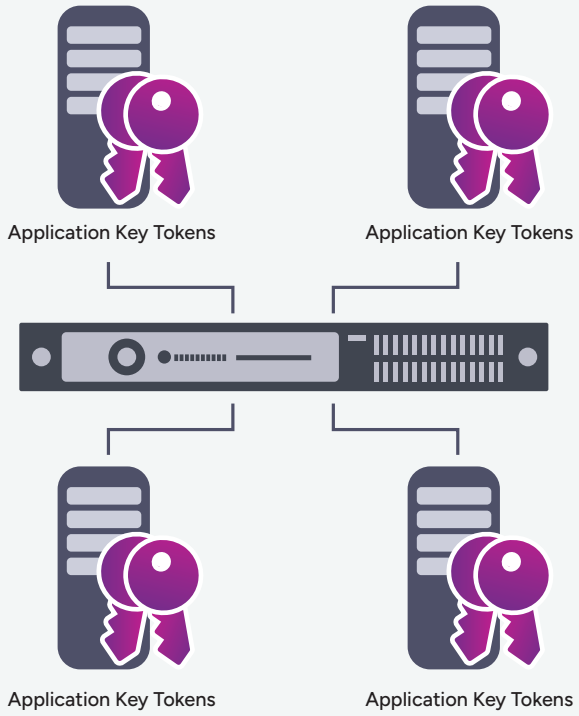


Figure 2: Application key tokens stored on hosts

Security World addresses the age-old challenge of providing strong protection for keys while simultaneously ensuring they are available for use by authorized applications deployed over high-scale, redundant, and distributed servers.

More on ACLs

As described briefly above, access control lists (ACLs) form a significant part of the metadata associated with a key. They are securely wrapped along with the key when the key is generated and are protected to the same high standards as the key itself. An nShield application key token's access control list defines the specific cryptographic operations permitted for that key and the authorizations required to perform them. An ACL can specify a wide range of properties and conditions, from simple to highly complex. As such, it is a powerful tool for isolating one application key from another.

The ACL determines what the key can be used for, including a combination of operations or restricting it to a single purpose. These include:

- Encryption and decryption
- Whether a key can generate and verify digital signatures
- Whether new keys can be created from existing ones, including what mechanisms can be used and whether the key can be extracted
- Whether a key can be used to securely encrypt (wrap) or decrypt (unwrap) other keyss
- Whether a key can be exported from the HSM in plaintext (note: in FIPS 140-3 Level 3 and Common Criteria CC/CMTS compliant modes, this is allowed for public keys only.)

The ACL also enforces the specific conditions that must be met before an operation can be performed. It can define hierarchical requirements involving various tokens or certificates. The ACL can require the presentation of specific authentication tokens. More complex ACLs can establish a dependency chain, requiring other specific keys to be loaded first before an operation can be carried out.

In addition to defining permissions and authorizations, the ACL can impose constraints on key usage, such as:

- Specifying a time limit for how long a key remains loaded in the HSM after being authenticated
- What cryptographic mechanism(s) are allowed
- Other cryptographic restrictions to prevent the misuse of keys protected by the HSM

These ACL policies are all managed, unwrapped, and enforced by the HSM natively whenever a key is loaded and, as such, cannot be compromised by an attacker. The ACL for a key is set when the key is generated and is not normally modifiable after that.

Additional Key Isolation Methods

When even stronger isolation (above and beyond basic module protection and ACLs) needs to be implemented for safeguarding application keys, softcards (passphrases) or operator card sets (OCSs) can be used to introduce an extra layer of cryptographic isolation and authorization required to use load those keys. Physical OCSs and logical softcards are collectively referred to as "authentication tokens." An authentication token is associated with an application key when the key is generated by the ACS. The authentication token introduces an extra layer of encryption and authentication to these keys to build logically separated sub-domains.

The application key must be accompanied by the presentation of the authentication token and validated before the key can be loaded onto an HSM. Once an application key has been loaded into an HSM, it can be used as often and according to any other restrictions permitted by the ACL for ACL-approved cryptographic operations before being unloaded. A single authentication token can be used to protect multiple application keys.

Softcards

When it has been determined that an application key requires some additional security controls, a simple authorization method is available in the form of softcards. A softcard is a file containing a logical token that cannot be loaded without a passphrase; this logical token encrypts all application keys protected by it and must be loaded first, to authorize the subsequent loading of any key protected by the softcard.

Softcards are comprised of a single passphrase, but they enforce an additional layer of isolation for the protected keys, making them loadable only by people with access to the softcard and knowledge of the passphrase. They can be a convenient alternative when physical access to a smart card reader is impractical (see Operator Card Sets below).

Softcards add some additional controls over when an application key is loaded and by whom. The passphrase is also used as a seed component for the application key, tying together its cryptographic roots and furthering the isolation/uniqueness of the application key from other keys in the system. Softcards can be used to protect multiple keys. Keys are essentially "assigned" to the protection of an existing softcard. Also, keys protected by a softcard may be either recoverable or non-recoverable. If nobody with access to a softcard can recall the passphrase to use it, the softcard is broken. Recoverable keys protected by a softcard can be recovered by the ACS and assigned to a new or existing alternate softcard. However, unrecoverable keys cannot. This can add an additional layer of security by ensuring application keys must not be deceptively "recovered." In the case of a "lost" (non-recoverable) application key, the application owners/operators would have to agree to generate a new application key. This means rekeying the application. So deciding whether a key should or should not be recoverable is an important commitment that requires careful consideration before selection.

Softcard passphrases may be set up as recoverable or non-recoverable at the time the softcard is established. A recoverable passphrase allows the creation of a new passphrase for the softcard under the control of the ACS cardset.

Also note that softcards are persistent, meaning the application key loaded by a softcard will remain loaded until removed programmatically by conditions in the ACL, by a command to unload the key by the softcard owner, or by the reset of the HSM.

Operator Card Set (OCS)

nShield HSMs can also make use of operator card set (OCS) smart cards to provide an extra layer of encryption and a recommended two-factor authentication to load application keys. An OCS is not typically a single smart card (although it could be). Like the ACS, an OCS is a logical token that encrypts the application keys it protects and is shared in parts across a set of smart cards that represent an authorized group. When created, the necessary quorum of these cards is also set. This is the minimum number of cards from the total set that must be presented to authorize the use of the keys protected by the card set. This is the minimum number of cards from the total set that must be presented to reconstruct the token and load any keys protected by the card set.

Since individual cards are normally allocated to authorized members of a group of users (each smart card with its passphrase, if set up), when a card set is authorized within the HSM, this does not represent a single user's authorization, but rather the authorization of the group to perform the requested action. Essentially, a "key ceremony" is required to load the key encrypted by the card set.

The notation used to describe the quorum of the card set is "K of N," just as in the ACS card set.

Technically, an OCS card set can have a 1 of N quorum. A special property of a 1 of N OCS is that only a single card needs to be presented (along with its passphrase, if set up). This means that no physical switching of cards in slots needs to take place, which can be a practical advantage in certain conditions where you not only want to protect where an application key is loaded but also want to retain the advantage of restricting when it is loaded.

Note that a 1 of N OCS quorum is inferior to multiple cards being required for a quorum. In effect, the reconstruction and loading of the OCS-protected key is down to one cardholder. In a K of N quorum, there is the additional layer of encryption used to isolate the application key domain – the sharding of the OCS key – and it requires collusion among multiple cardholders to reconstruct and load the key in "out of policy" conditions.

Optional Features of OCS Quorums

Passphrases for OCS Cards:
You will note that each time we referenced an OCS card, above, we also said "along with its passphrase, if set up." Each card within an OCS card set can optionally be assigned a passphrase. It is highly recommended that the passphrases be used. As with softcards, the passphrase(s) become additional seed material for the application key and the passphrase introduces a second factor of authentication – something the user has (the card) and something the user knows (the passphrase). This further isolates the OCS-protected application key sub-domain by ensuring that possession of the card(s) alone is insufficient to load the key(s) into the HSM.

As with softcards, keys protected by an OCS can be generated as recoverable or non-recoverable. Should the OCS become unable to assemble a quorum, non-recoverable keys will be lost and new keys must be generated for those applications under a new or another pre- existing card set.

OCS passphrases can also be configured as recoverable or non-recoverable, just like softcard passphrases. They must be set up when the OCS is generated. In addition, at setup time, you can choose to have a single passphrase across the entire card set or individual passphrases for each card. Each additional step can improve security but also increases the need for physical protection of the card sets and securing the passphrases.

You can also generate a new OCS card set for the protection of the application keys in question but this will encrypt those keys with a new shared token when the prior card set is replaced. The old card set and passphrases will no longer work. Like with an ACS card set, this would likely only be performed if the card set was becoming increasingly likely to be unable to construct a quorum (lost, damaged cards or individual card passphrases).

Persistent vs. Non-Persistent Authorization:
OCS card sets can be configured as either persistent or non-persistent. In a persistent configuration, once the required quorum of cards has been presented (with their passphrases, if set up) and authorization is granted, the authorization remains valid even if the cards are subsequently removed from the local or remote smart card reader. This is useful in scenarios where continuous physical presence is impractical. In contrast, a non-persistent OCS requires that the last card inserted remains inserted; removing the last card in a non-persistent OCS immediately revokes the authorization. This provides an even higher level of control and is ideal for environments demanding strict enforcement of key access policies.

Policy Enforcement

The concepts of Security World application key ACLs, softcards, and OCS quorums can be tightly bound together to meet the most demanding security requirements and key domain isolation and control policies for any given application.

With this in mind, we can see that a key loading and usage policy is enforced by three factors:

- **Access to the Application Key Token:** If you don't have the application key token on your application server, you simply cannot load that key onto a target HSM. This policy is enforced outside the HSM by careful and deliberate synchronization of specific application key tokens across the application server estate.
- **Access Control Lists (ACLs):** Once the key is loaded, it can only then be used for specific purposes and under specific conditions described in the ACL that is bundled in the application key token. Again, this policy is enforced inside the HSM.
- **Token Authorization:** If a key is protected by an authorization token, such as a softcard or an OCS, then you must present that token before you are permitted to load the key into the HSM. This policy is enforced inside the HSM. As we have demonstrated, the effect of token authorization is to create a sub-domain of key space, restricted for users that possess the key token, the authorizing token, and its associated authenticator(s): smart cards loaded into a local or remote smart card reader (and the associated passphrase) presented for OCS or the passphrase for softcard protection. If only authorized users of the application key have access to the authorization token and the associated passphrase, that application key is completely isolated from all other application keys protected by the HSM. We have also demonstrated that, in the case of OCS tokens, an additional layer of encryption (sharding and secret sharing algorithm for the card set) is added to the way the application keys are isolated.
- Lastly, note that key recoverability can be used to further isolate keys protected by token authenticators as well. Non-recoverable keys protected by softcards or an OCS will be lost forever if the required authentication can no longer be presented. This requires establishing a new application key and assigning it to a new or existing authenticator. As such, making keys non-recoverable should be carefully considered in light of the need for application key isolation/protection versus the cost of rekeying or loss of data due to the non-recovery of the application key.

The ACL associated with an application key defines the key policy in a form that an nShield HSM can strongly enforce.

CONCLUSION

Enable Secure Key Management With Security World and nShield HSMs

By combining nShield Security World facilities for module-protected keys using ACLs and additional authorization tokens such as softcards and OCSs, application security sub-domains can effectively be created. Logical isolation can be supplemented by additional layers of encryption, specifically using OCS with a N>1 quorum and passphrases. Together, these policies create Security World structures that not only meet a static security environment but also coexist with the unique key isolation and protections dictated by enterprise application security owners within the same HSM or groups of HSMs. In addition, another layer of isolation can be added through establishing keys that are non-recoverable to increase security, but this must be used with discretion to avoid rekeying applications.

- Security sub-domain keys can be recoverable to guarantee flexibility and reliability of keys.
- Security sub-domain keys can be non-recoverable to increase security and isolation.
- When added to the operational efficiencies of unlimited keys protected by an HSM, simplified addition of HSMs to a Security World for improved performance/redundancy, no specialized backup equipment or software needed, and speed upgrades requiring no new hardware, these key isolation capabilities make nShield HSMs a formidable alternative to traditional HSM subdividing or "partitioning."

Security World avoids the need for expensive backup tokens and manual key cloning.

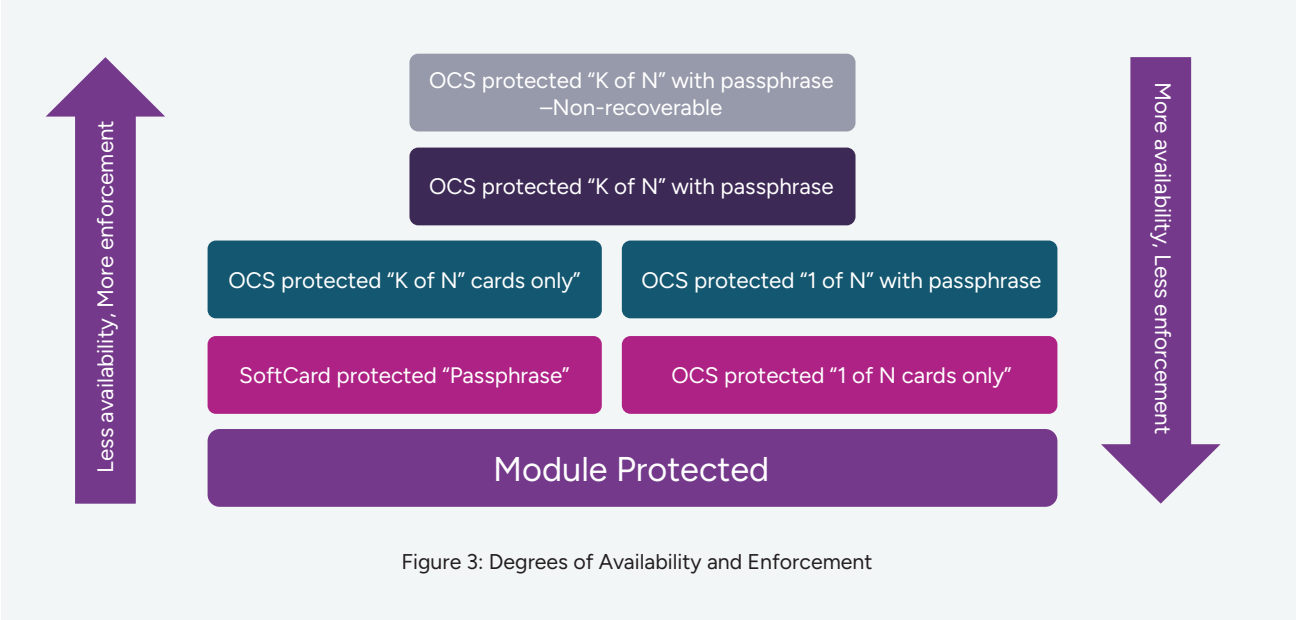


Figure 3: Degrees of Availability and Enforcement

ABOUT ENTRUST

Entrust fights fraud and cyber threats with comprehensive identity-centric security that protects people, devices, and data. Our solutions help enterprises and governments safeguard critical systems from every angle, enabling secure onboarding and issuance, providing everyday identity protection, and empowering them with 360-degree visibility and orchestration across keys, secrets, and certificates so they can transact and grow with confidence. Building on our decades as a pioneer and innovator in establishing trust, Entrust has a global partner network and supports customers in over 150 countries.

For more information, visit entrust.com.

entrust.com Toll-Free: 888.690.2424 | International: +1.952.933.1223 | sales@entrust.com

ENTRUST
SECURING A WORLD IN MOTION