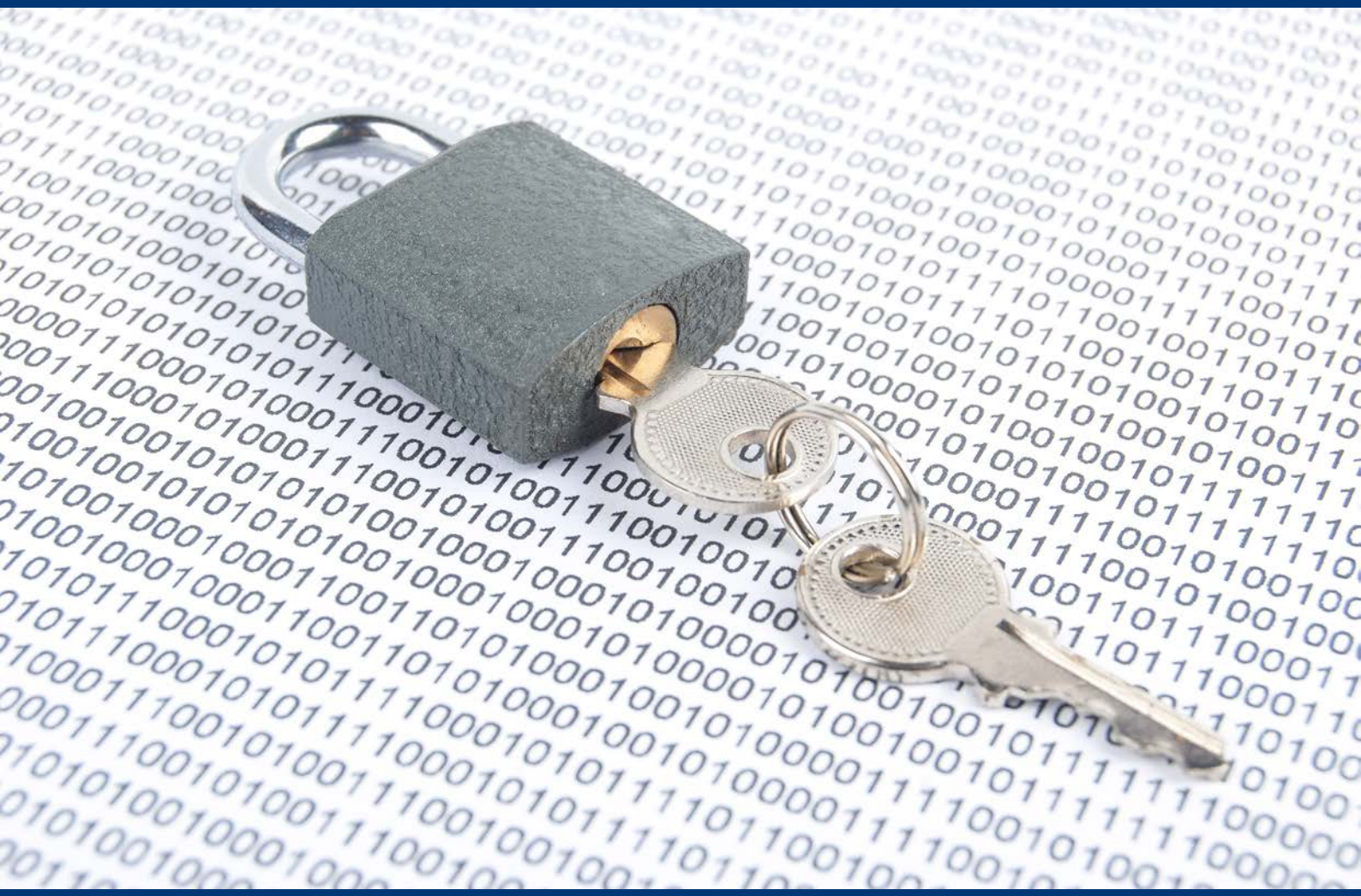
Ultra Safe Public Key Infrastructure

White Paper



Public Key Infrastructure (PKI) is now the principal way to assure trust; Hardware Security Modules (HSMs) are the predominant way to assure trust in PKI

Ultra ELECTRONICS | AEP

Public Key Infrastructure

The Foundation of Trust

Public Key Infrastructure (PKI) is now the principal way to assure trust; Hardware Security Modules (HSMs) are the predominant way to assure trust in PKI

About Trust

It is impossible to transact in the digital world without trust. In order to achieve trust, assurance is needed that shows:

- Entities with which transactions are being performed are actually the entity they say they are.
- There is no breach of privacy/confidentiality.
- The transaction that is intended is the transaction that actually takes place.
- The transaction can later be proven to have taken place.

Authenticity, Privacy/Confidentiality, Integrity and Non-repudiation have always been the four pillars of trust: but, while trust has always been required, the way that trust operates today has fundamentally changed from the way it was provided for in the past.

In our modern, global society, people often transact with people they have never met and may never encounter again. They may be on different sides of the planet; they may need to transact within the first few seconds of coming into contact with each other. In fact, the transactions may not even be between people at all, but between people and machines, or simply between machines. Examples are everywhere: automatic teller machines, electronic wire transfers, e-commerce, insurance claims processing, electronic filing of tax forms.

In a traditional environment, non-repudiation is provided by hand written signatures sometimes witnessed and notarised by third parties.

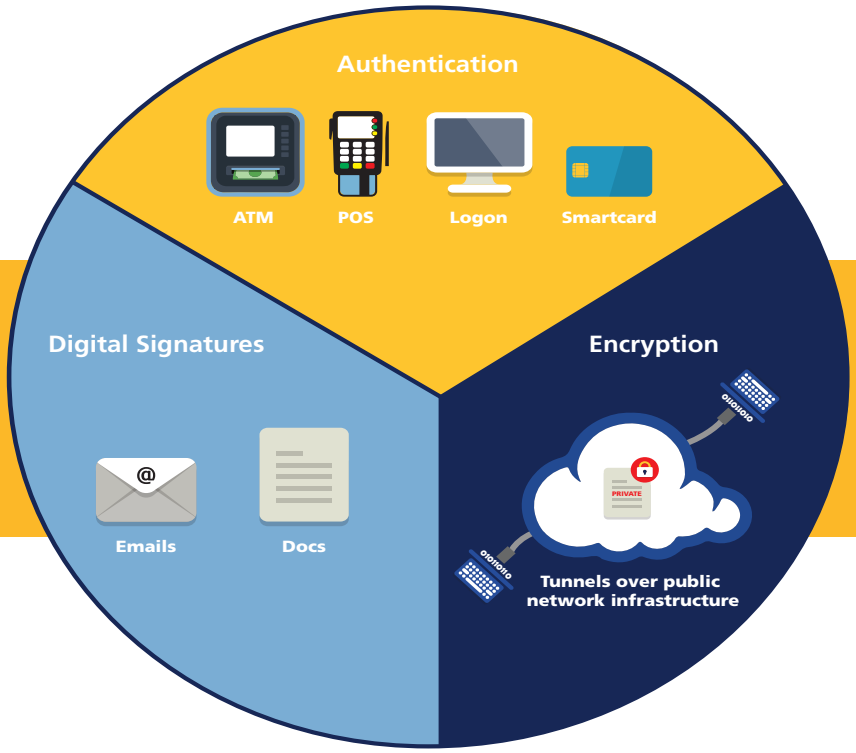
In an electronic environment however, digital signatures provide this same element of trust. Anywhere transactions are made, trust still needs to be established. But now it must be established over barriers that didn't exist in earlier times, or at least not to the degree they exist now: between anonymous parties, over great distances, with very high reliability, with great speed and efficiency and very often as a completely automated process.

Trust in today's world relies on PKI, this consists of three things:

1. Methodology
2. Technology
3. Infrastructure by which transactions can be performed

The method is relatively straightforward and relies on the presentation of trusted certificates between parties in a transaction. If you have a certificate that I trust, then I can also trust you. If I have a certificate that you trust, you can trust me.

Sources of trust are called Certificate Authorities, typically end entities are configured with one or more trust anchors which are then used as a starting point to validate a given certification path. The technology used is Cryptography and in particular Public Key Cryptography.



Public Key Cryptography

Cryptography is fundamentally the use of encryption algorithms or ciphers and keys to encrypt and decrypt data. Algorithms or ciphers are mathematical formulas or functions. Encryption algorithms are divided into two families based on the key type:

- Symmetric or secret key
- Asymmetric or public key encryption

In secret/symmetric key encryption both the sender and receiver use the same secret key, so named because the strength of the system relies on the key being known only to both parties. This creates a problem, namely that the secret key must be shared and thus securely communicated from one party to the other. In public/asymmetric key encryption, the sender and receiver each have a distinct but mathematically related key, which together constitute a key pair. Public key encryption requires that only one of the keys, referred to as the private key, must be kept secret and (usually) under the control of the owner. The other key, referred to as the public key, can be distributed freely for use by any party who wishes to participate in secure transactions with the owner. Public key encryption eliminates the secret key distribution problem. It is important to note, however, that secret key encryption still plays a major role in the implementation of a PKI.

Equally important as the advantages inherent in public key encryption, is the support for the properties of *authentication* (identification of the sender) and *sender non-repudiation* (the inability of a sender to refute that they signed something encrypted with their private key). Since anyone with the sender's public key can decrypt a message encrypted by the sender's private key, this type of encryption, called a digital signature, does not protect the confidentiality of the message. The sender is prevented, however, from denying that he was the originator of the information thus signed, unless the private key was compromised. The sender's key pair is used to create the digital signature. Although technically the operation performed is still encryption and decryption, the operations are termed *signing* and *verifying,* respectively.

Public Key Infrastructure

Methodology and technology alone, however, are not sufficient to create a PKI. By definition, a PKI also requires an infrastructure. That infrastructure consists of a source of trust called a Certificate Authority (CA) - the issuer of the root certificate which becomes part of all certificates granted to end users. Those end users request certificates from Registration Authorities (RAs) that act as intermediaries for the CA. Together, the CA, RAs and end-users comprise the PKI. These components of a PKI and the encryption keys and digital signatures in public certificates form a *chain of trust*. Many PKI deployments use a 'closed' system, that is, they own the whole *chain of trust*. At all three levels in the hierarchy security modules play key roles.

The CA

Public keys are distributed in the form of public key certificates or digital certificates. The CA is the very foundation of the PKI since it is the only component that can issue public key certificates. CAs are server computers with one or more security modules attached that issue certificates in response to requests from RAs on behalf of end-users. The CA is typically deployed in a hierarchical design. If the CA is compromised, then there is no trust. This might happen, for example, if an impostor were to issue certificates that fool credit card users into thinking they were dealing with a real bank when they were not. Equally, if the certificate is compromised, there can be no trust. Even if the CA is legitimate, the certificates it issues must still be protected so that they cannot be misappropriated. The CA must demonstrate an absolute level of trust. For example, it must ensure that keys are protected from theft, both physical and virtual. Rigorous procedures must be in place to ensure secure handling of keys and these procedures must be strictly enforced. The fact that standards are written down, and that procedures for handling key materials are audited (usually by an outside accounting firm) are examples of a CA's methodology and organisational infrastructure at work.

Although Certificate Signing Request (CSR) is not directly considered a component within a chain of trust its explanation is still beneficial due to its place within the certificate signing process. A Certificate Signing Request (CSR) is a base-64 encoded (PEM based) string which is generated using the public key along with a number of identity attributes such as DN, email, address etc. The CSR is then sent to the CA which it then uses to create a public certificate. The public certificate is then signed and sent back to the user. The benefit in using a CSR is that the private key never leaves the client.

The CA is also responsible for revoking certificates that have been compromised. The VA, or Validation Authority, is an optional component that a CA can delegate to publish certificate revocation lists (CRLs). A certificate revocation list is a list of certificates (or more specifically, a list of serial numbers for certificates) that have been revoked, and therefore entities presenting those certificates should no longer be trusted. Typically, the CA that issues a given set of certificates is also responsible for publishing revocation information associated with those certificates. However, it is possible for a CA to delegate that function to another entity.

The RA

The RA, or Registration Authority, is an intermediate layer in the infrastructure. As the name suggests, the RA's role is to verify the identity of end users that wish to register with the PKI and request certificates. While the RA is an optional component most CAs do not deal directly with end users, since to do so would be extremely time and resource-consuming; at any one time there might be many end users requesting certificates, and registration itself is typically a manual process. Another reason for offloading registration tasks is that different authorities have different registration procedures (for example, different identification requirements of the person requesting registration) and these can be more easily satisfied if each authority handles its own registration process. If the request is granted, the RA then receives the certificate back from the CA. This certificate includes the public key, the identity of the authorising CA and the identity of the user. The application then provides this information to the end-user, for example in the form of a smart card, a personal HSM. Note that although the RA can offload many functions from the CA, the RA can never be the issuer of a public key certificate.

Judicious deployment of RAs can provide two primary advantages. Firstly, RAs can help to reduce overall costs. This is especially true in large, geographically dispersed entities that require their users to be physically present before certain PKI related activities are permitted. A typical example would be end-user registration, but other PKI-related functions such as end-user initiated requests for certificate revocation or key pair recovery might also apply. There may also be other practical considerations, such as when an entity elects to outsource the CA service but retain control of the registration process. Secondly, offloading the administrative functions from the CA allows an entity to operate their CA off-line, which reduces the window of opportunity to mount remote attacks against that CA.

The End User

The third layer in the PKI hierarchy is the entity who ultimately uses the certificate to make *trusted transactions*. End Entities are sometimes thought of as end-users. Although this is often the case, the term End Entity is meant to be much more generic. An End Entity can be an end-user, a device such as a router or a server, a process, or anything that can be identified in the subject name of a public key certificate. End Entities can also be thought of as consumers of the PKI-related services. There are even cases when a provider of PKI-related services is considered to be an End Entity. For example, an RA is considered to be an End Entity from the point of view of the CA.

End Entities that will be bound to certificates, such as servers and end users, must enroll into the PKI before they can participate as members of the PKI. This involves an initial registration step followed by request creation, submission and certification. Upon receiving their certificates, including the public/private key pairs, end-users store them in their own security modules. Applications access the certificates programmatically - typically through interfaces that submit documents for digital signatures and/or encryption. End-users access the security modules often by using two factor authentication, similar to a bank ATM. Two factor authentication requires the end user to both have something (i.e. a smart card) and know something (i.e. a pin code). This security concept along with the *chain of trust* gives strong assurance that only the authorised end-user is making *trusted transactions*.

Dual Key Pairs

Support for two pairs of public-private keys is a fundamental requirement for some PKIs (for example, Entrust). One key pair is for data encryption and the other key pair is for digitally signing documents. Encryption key pairs and signing key pairs are a result of conflicting requirements. One such requirement is to support different algorithms for encryption and digital signature pairs and different validity periods. Another reason is to support data recovery, which requires the private keys for decrypting to be securely backed up, but non-repudiation, which requires the private keys for signing, not to be backed up. There might also be the requirement to support updating encryption key pairs and managing decryption key histories even though this conflicts with the requirement to securely destroy the private key used for signing when updating signing key pairs. Using two key pairs, an encryption key pair and signing key pair, solves these conflicting requirements.

Governance

To facilitate trust, a PKI must be operated with some level of oversight, and with policies, standards, and procedures in place to control how the PKI is managed. In addition physical security aspects should be considered. Prior to deploying any CA or issuing a certificate, define and agree upon the policy which governs the use of the PKI. A policy usually takes into consideration regulatory and industry requirements. The policy may also specify technical aspects of the PKI such as the cryptographic algorithms that must be used as well as operational controls for the CAs.

It is likely that other services either inside or outside of your environment will depend on the PKI, and the PKI policy should provide a clear guidance on what can be expected for certificate issuance, security, disaster recovery, etc. The policy does not need to be overly complex, but it is critical to develop and follow it from the beginning to have a level of assurance in the operated PKI.

In addition to PKI policy, you may need to develop CA-specific policies before implementing the PKI. For example a formal Certificate Policy/Certification Practice Statement that follows IETF Public Key Infrastructure X.509 Certificate Policy and Certification Practices Framework (RFC 3647) with accompanying standard operating procedures.

Security Modules in PKI

Modern transactions depend on trust, trust depends on PKI and PKI depends on the technology of security modules.

The function of a security module is to issue, validate and store keys in a protected environment. A security module can be one of two types: a Software Security Module (SSM) or a Hardware Security Module (HSM). The major difference is that an SSM is a program that runs on a general purpose computer, while an HSM is a dedicated computer specifically designed to function in a security role.

Function of Security Modules

Security modules provide the technological services for the PKI. When a system, such as a funds transfer system, requests a certificate to show to another bank that it can be trusted to execute a wire transfer, the security module is the device that actually generates the public/private key pair for the certificate. When the second bank receives the certificate (as part of the transaction), it also uses a security module and public key cryptography operations, this time to check the certificate and see whether the certificate can be trusted - i.e. that it came from a trusted source and that the certificate was not compromised along the way.

Security modules generate keys. Using public key encryption mechanisms, the security module generates *two keys* that have a mathematical relationship such that a message enciphered with one of the keys can *only* be deciphered by the other, and vice versa. One of these keys is kept private and the other made public. A public key is one part of a certificate - other parts include identity information such as the name of the entity issuing the certificate and the name of the entity holding the certificate.

Certificates and their keys also provide confidentiality using a combination of secret key and public key encryption. To decrypt a message encrypted by a specific sender (with their private key), simply use the sender's public key. To encrypt a secret key (one that only two parties to a message will share), encrypt it with the other person's public key. They (and only they) will be able to decrypt the secret key with their private key. Both parties can then use the secret key to exchange encrypted messages that no one else can read. This will not only ensure confidentiality, but also integrity, since if the message had been altered during transmission the key could not decode it. It also ensures authenticity, since only the holder of the secret key can decode the message encoded with that key.

Certificates also contain a special field called a digital signature. It is by signing the certificate (using a certificate's public key) that an end entity provides proof that it was party to a transaction - the concept of non-repudiation. Digital signatures use a similar technique to provide non-repudiation, as well as another layer of integrity and authentication.

To sign messages with digital signatures, security modules do the following:

1. Evaluate a message digest (a binary field of fixed length) using a hash generator algorithm
2. Encrypt the digest using the sender’s private key, included in the signature are the hashing algorithm name and the sender public key appended to the encrypted digest
3. Encrypt the whole message (the message itself and the signature) using the combination of secret key and public key cryptography

Authenticating the digital signature proceeds as follows:

1. The receiver's security module decrypts the secret key using the receiver's private key and the rest is decrypted using the secret key
2. The digest is decrypted using the sender's public key
3. Since hashing is a one way process the message digest is re-evaluated using the same hashing algorithm the sender used
4. The digests created from steps two and three are compared; if they are the same, the signature is authentic

The digital signature also provides a second layer of integrity assurance since any alteration of the message will mean that the digests would not match. It provides additional authentication since only the holder of the key can sign the message.

Finally, the keys must be generated, stored and used in a highly secure manner. This is especially true of the keys used by the PKI itself for the *trust anchor and chain of trust*.

Advantages of Hardware Security Modules

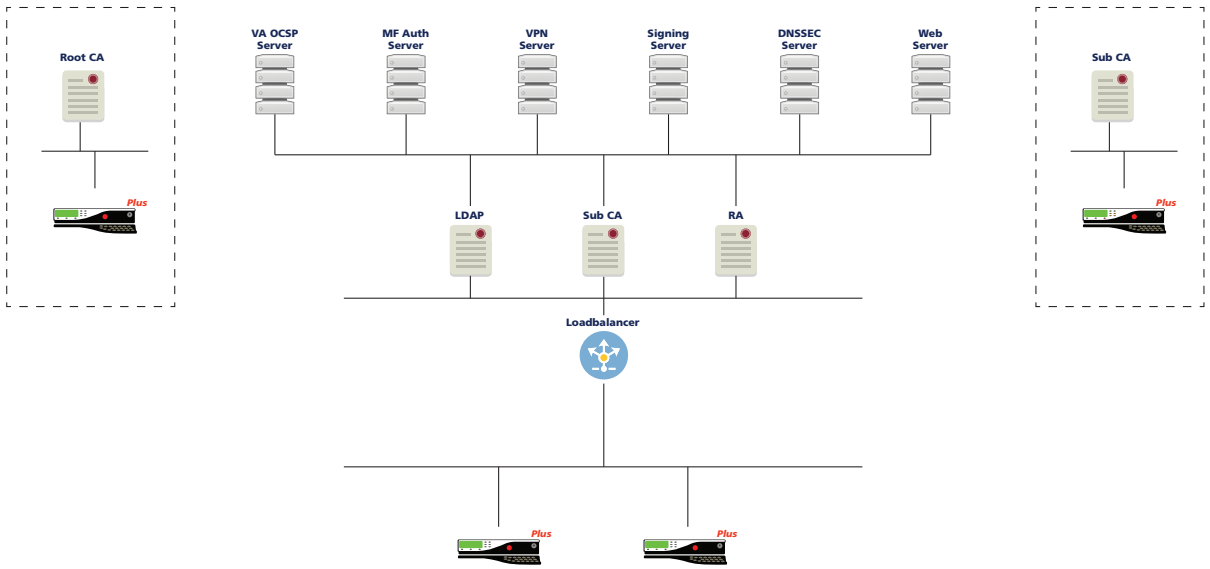
HSMs provide physical separation for cryptographic operations and key material. Another major advantage of HSMs is that, because their software and hardware is specifically dedicated to providing cryptographic operations, it can be specifically optimised for that purpose. HSMs perform cryptographic operations faster and with higher assurance than their software counterparts. For example, one of the processes at the heart of the generation of public/private key pairs for certificates is the generation of random numbers.

HSMs have dedicated hardware specifically designed to generate random numbers and they can therefore generate numbers that have greater entropy than would be the case if the hardware were not specifically designed for that purpose.

HSMs are standards compliant computing devices. An example of a technical standard is FIPS (Federal Information Processing Standard) 140-2, a U.S. government standard covering the implementation and assurance of security mechanisms such as algorithms and tamper protection.

Maximum Security PKI

Choose the HSM that has achieved the highest available level of security in order to achieve a maximum security PKI.



Keys Stored on Module

Surprisingly, not all HSMs fully exploit the inherent security advantages of using a separate hardware module for key generation and for key storage purposes. Even though they generate keys in hardware, they don't store them there, but outside of the security boundary on the general purpose host, encrypted under another key. Storing the keys on the host outside of the security boundary - even in encrypted form - means that a hacker can potentially remove them to another computer where they can be analysed and possibly compromised.

High Quality Keys

It is also important that the keys are of good quality, which requires a well-designed random number generator.

Connectivity

Connectivity is a key differentiator among HSMs. It determines how the module attaches to the rest of the PKI. In some cases, that connection is through a PCI interface card on the back of a server. This server is typically running applications (like those that register users) that access the module for PKI services (like encoding a message). Other modules, such as Keyper HSM, attach via an Ethernet connection. An Ethernet attachment provides a number of benefits that greatly enhance the module's ability to support security applications at all three levels of the PKI:

Security

One of the advantages of using an HSM is that keys can be stored off of the server in a more secure environment. It therefore matters greatly how the security module and host are connected physically. Connecting the two over a private Ethernet means that the security module can be effectively shut off from the outside world. Because packet forwarding is switched off, the only access is through the host.

Scalability

There are a limited number of PCI slots on the back of a computer. This limits the ability to scale cryptographic processing power (for example, when a CA needs to handle more certificate requests). An Ethernet connection has no such restriction.

Fault Tolerance

Since HSMs are connected over a network, if one goes off-line, the remaining HSMs can take over. Aggregate processing throughput for the entire network may fall, but at least the cryptographic operations themselves stay on-line.

Hot Swapping

Another advantage of network connections is that taking a module off the network does not crash the network. Removing a PCI card, on the other hand, requires a reboot. This limits the flexibility of administrators to service or upgrade equipment.

Load Sharing

This is a similar benefit to fault tolerance, except that a module does not have to go completely off-line before other modules step-in and share the processing load. In fact, the exact distribution of processing load over a number of modules ought to be an option the security administrator can configure - whether, for example, 10 hosts are allocated to one device or to 5 or to 10. What's important is that the security administrator has the flexibility to balance the load in a way that achieves the administrator's goals, such as to maximize the aggregate throughput or give the most important applications priority access to resources.

ACCE Reflects Core Competency

PKI technology providers enable cryptographic operations in hardware because hardware can offer very strong security, performance, manageability, scalability and fault tolerance capabilities. Case in point: ACCE - **Advanced Configurable Crypto Environment**. ACCE reflects over 100 man-years of combined engineering effort focused on developing highly optimized hardware architectures for encryption/ decryption, authentication, key management and key protection. FIPS 140-2 is suitable for protecting million dollar funds transfers, and life protecting data alike to thwart determined, skilled and well-funded attackers. Security Level 4* protects a cryptographic module against a security compromise due to environmental conditions or fluctuations outside of the module's normal operating ranges for voltage and temperature. Intentional excursions beyond the normal operating ranges may be used by an attacker to thwart a cryptographic module's defenses. ACCE includes special environmental protection features designed to detect fluctuations and zeroise CSPs. Like other ACCE products, it can also be upgraded with new software and algorithms, and can be configured remotely. It supports a range of key management options, with protected internal key store for over 15,000 keys, backed by secure key export and transport options. Where other HSMs only certify the hardware and the bootstrap code; all components within the FIPS 140-2 security boundary for ACCE, including firmware, are certified.

*Ultra Electronics AEP Keyper^{Plus} HSM

The Ultra Safe Keyper and Keyper^{Plus} HSM is the only Ethernet attached HSM in the world to achieve FIPS 140-2 Level 4 and works with all the major CA software vendors' stacks to provide the highest level of security for the cryptographic operations and keys that are required to underpin the security of the PKI.

Ultra Electronics
AEP
Knaves Beech Business Centre
Loudwater
High Wycombe
Buckinghamshire, HP10 9UT
Main Switchboard: +44 (0)1628 642 600
Email: info@ultra-aep.com
www.ultra-aep.com
www.ultra-electronics.com