UNBOUND

THE UNBOUND NEXTGEN VHSM®

Prof. Yehuda Lindell,
Chief Executive Officer and Co-founder

WHITEPAPER

Introduction

The days of encryption and cryptography being a niche technology solving marginal problems are way in the past. The goal today is to encrypt everything, everywhere, all the time. This is a significant challenge and far from being realized, but is well accepted as the direction that organizations are headed. Digital signatures have become a central tool in securing our infrastructure, powering authentication, secure software updates, document signing and digital transactions.

The ubiquity of cryptography brings many business, operational and security challenges. Businesses today need to be able to react quickly to changing needs and roll out new solutions as fast as possible. Security measures are often inhibitors of business, and certainly slow them down significantly. This is especially true of legacy hardware solutions for cryptographic key management and protection that typically involve a lengthy procurement and integration process, and do not have the flexibility of modern virtualized environments. These solutions can also be very expensive, affecting the business's ability to be cost effective.

Operationally, large organizations need to work with many different key management solutions, each integrated with only a subset of applications and environments, and often providing very limited capabilities. This results in a lack of visibility across the organization and makes deploying company-wide policies difficult. This increased complexity not only has financial ramifications but also impacts security. Managing physical HSMs at different data centers across the world, along with keys in cloud HSMs and cloud KMS environments is extremely difficult. This is challenging both for administrators, as well as developers who need to consume cryptographic services in different environments, often without a unified API.

In order to address today's computing and security needs, we need cryptographic solutions that remove the burden from the administrator and developer, are both easy to use and manage, and work in every environment. Virtualized environments are the norm, and it is time for cryptographic infrastructure to catch up.

In this white paper, we will describe the Unbound NextGen Virtual HSM®, and how it addresses the aforementioned challenges and many others.

The Problems with Existing Solutions

Legacy solutions for protecting cryptographic keys rely on physical security in the form of an HSM. An HSM (Hardware Security Module) provides a secure environment where cryptographic operations take place, without ever exporting the keys. The security of an HSM is derived from the fact that no other code runs inside the HSM, the API is limited to cryptographic operations (and so is easier to protect than a general-purpose machine), and physical protections prevent physical access to the disk and memory, and side-channel attacks from extracting the keys. HSMs have been used for decades and are considered the gold standard for protecting keys. Although they are far from perfect (see information about a devastating attack on a top vendor's HSMs in this brief post, and in the presentation at Real-World Crypto 2020), they are in general a good security solution, but one that is problematic in today's computing environments. Primarily, this is due to the fact that they are a physical anchor when everything else is virtualized. They typically require physical access to administer, require manual operations to synchronize keys amongst multiple HSMs, and they work the way legacy hardware works in contrast to the way that modern software works. The goal today is to simultaneously operate computing environments in on-prem or private data centers and in multiple clouds (i.e., hybrid environments), but operating HSMs in such environments is extremely painful. Not all clouds provide HSM services, and when they do, they work differently from on-prem HSMs. In addition, different HSMs have very different behaviors, and large organizations have multiple different management tools for their HSM fleet. Finally, HSM procurement is a slow process, and whereas people are used to immediate allocation of new computing resources in the cloud or virtualized environment, HSMs can take months to purchase, install and deploy. As such, even when HSMs provide the appropriate security properties needed, they inhibit business and business processes.

Due to the above, there is great interest in providing software-based solutions for key protection. Straightforward software solutions in the form of a hardened server that carries out the operations suffer from a single point of failure; an attacker breaching that machine or stealing the administrator's credentials (or likewise, insider threat) can steal all of the cryptographic keys on the machine with disastrous results. Likewise, whitebox cryptography – an attempt to obfuscate keys – have extremely poor security. In a competition held by the CHES conference in 2017, all but two of the whitebox submissions were broken in just a few days (and the longest lasted approximately two weeks). A more advanced solution utilizes recent secure enclave and trusted execution environment technology (e.g., Intel SGX, ARM Trustzone, etc.). Unfortunately, unlike HSMs where cryptographic code runs in isolation, trusted execution environments share resources with other code running on the same processor and therefore suffer from extremely effective software side-channel attacks. These attacks can easily extract cryptographic material, from even the best known "side-channel proof" code; see this survey paper for more information. Having said this, trusted execution environments are an excellent secondary security measure, making it much harder to steal secrets than from plain memory. As such, they are valuable in adding defense in depth, but not good enough to constitute the primary security measure for protecting assets like valuable cryptographic keys, as I explain here.

A New Paradigm for Software-Defined Cryptography

3.1 Secure Multiparty Computation for Software-Defined Cryptography

In order to achieve a software-based solution that is also secure, a different security model is needed. In particular, there must be no single point of failure, and this means that the secret or cryptographic key cannot reside in memory at any single machine at any given time. This may appear to be an impossible task – how can one carry out a cryptographic operation such as decryption or transaction signing, without holding the key? Fortunately, technology called secure Multi-Party Computation (MPC), also known as threshold cryptography in the context of protecting keys, is able to do exactly this. The secret key is split into two or more parts called shares, such that all shares are needed in order to get any information about the key. Then, the different shares are placed on different servers and devices, so that an attacker would have to breach them all in order to steal the key. MPC protocols enable the different machines holding key shares to interact (running an MPC protocol) so that they receive the result of the operation without revealing to each other anything whatsoever about the key. This means that the key remains fully protected, even while in use. MPC has been studied in academia for over three decades and has a strong and well-founded theory. MPC protocols have mathematical proofs of security, guaranteeing that an attacker who is unable to breach all machines is unable to learn anything whatsoever about the key, even if the attacker knows the protocols being used and can run arbitrary malicious code in its attempt. More basic information about MPC can be found in a Basic introduction to MPC and a Technical Primer to MPC. A more in-depth introduction to MPC for a general computer science audience appears here, and resources for in-depth study of MPC can be found at the MPC alliance page and here.

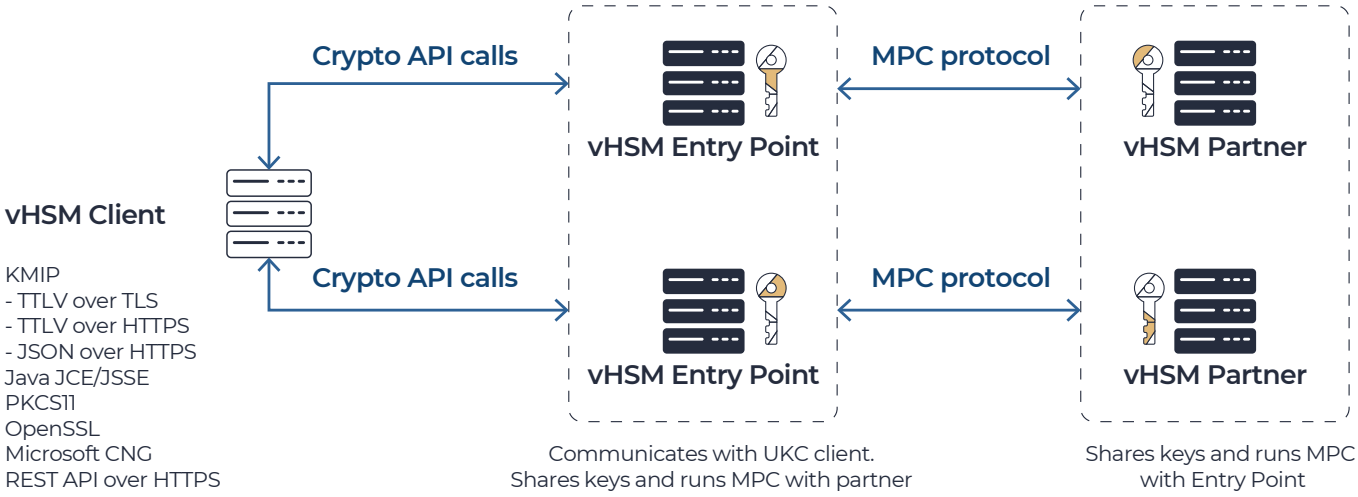
3.2 Unbound's Academic & Research Background

Unbound was founded by MPC researchers. Both Professor Nigel Smart and myself have been researching MPC for many years, and have collectively written approximately 100 papers on the topic (see Nigel's publication list and my publication list). In addition, some of Unbound's MPC protocols are the fruit of internal research at the company (for just one example, the two-party protocol for ECDSA published at CRYPTO 2017). Unbound's team of cryptographers ensure that the MPC solutions that we provide undergo rigorous inspection and evaluation internally, before deployment.

As cryptographers and researchers, we strongly believe in transparency and independent review. All novel protocols by Unbound have been published in open academic conferences and have been peer reviewed. In addition, an independent review of all of Unbound's protocols was undertaken by Professor Victor Shoup of NYU, and a separate independent code review that the actual code matches the specification reviewed by Prof. Shoup was also carried out. Finally, as part of our belief in transparency, all information about our protocols (and even the code) is available to customers, under NDA.

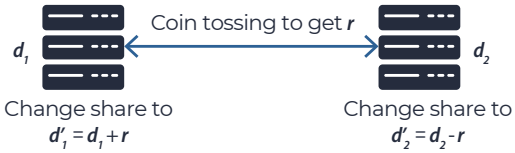
3.3 The Unbound Virtual HSM®

The Unbound virtual HSM (vHSM) uses MPC in order to split the key between two machines, and to carry out computations without ever uniting the key shares. Typically, multiple pairs of machines are used, and each pair holds an independent sharing of the key. Furthermore, all administration and key operations (e.g., adding a key) are automatically synchronized between each pair. In each pair, one of the machines is called the "entry point" – this is the machine that receives the request to carry out a cryptographic operation. The other machine is called the "partner" and communicates only with the entry point, running the MPC protocol to carry out the operation.



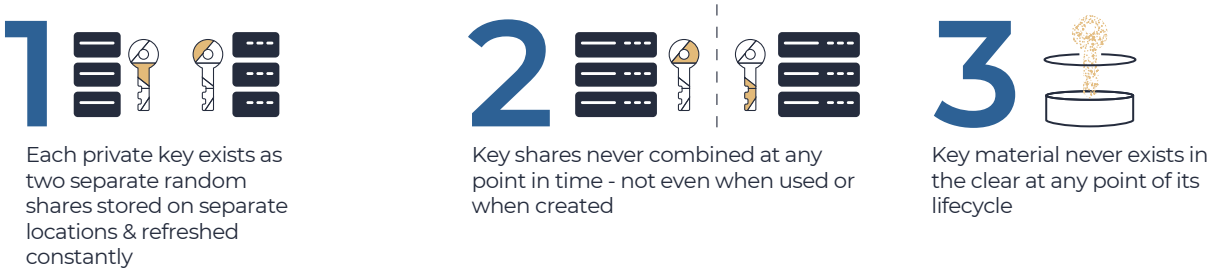
In addition to splitting each key into shares (parts) that are never united, the key shares are refreshed at frequent intervals (every hour, by default) so that although the key remains the same, the material held by each machine is completely rerandomized. This means that an attacker has to simultaneously breach both machines in order to learn anything (stealing one share before a refresh and the other share after a refresh reveals nothing at all about the key).

- Initial key sharing of d: random values d_1 and d_2 so that $d_1 + d_2 = d$
- The refresh process:



- Note that $d'_1 + d'_2 = d$ and so is a valid sharing
- Note that given d_1 and $d'_2 = d_2 - r$, nothing can be learned about d

The security model offered here is completely different to that of hardware. Instead of holding keys in a single place and using physical (and software) hardening to prevent access, the keys are distributed between two location so that simultaneous access to both is needed in order to learn anything. By ensuring strong separation between these devices (e.g., different administrator credentials and environments), this is extremely difficult for an attacker, and provides a very strong security guarantee without compromising on the functionality, flexibility and benefits of software. We remark that only the entry point communicates with applications and needs to be accessed by administrators. Thus, the partner machine speaks only to the entry point and after being set up never needs to be accessed by an administrator. This means that it can be a hardened machine with credentials that are not used anywhere else in the system. This type of deployment adds further to the security of the vHSM.



Loyal to the defense in depth approach, various software and hardware security technologies must be used in synergy and not considered mutually exclusive. For example, running the MPC protocol inside a secured environment like IBM LinuxOne (or even inside a trusted execution environment), one can make it significantly harder for an attacker to break the system. Similarly, in environments where a physical HSM is present, it can be leveraged as a hardware anchor, adding yet another layer. A good security solution will effectively leverage these multiple layers of security, in a simple and seamless manner.

Due to the rich capabilities of MPC, the vHSM supports all standard cryptographic operations and algorithms: RSA decryption and signing, ECC – ECDH, ECDSA and EdDSA, AES in multiple modes, HMAC, and so on. MPC changes only the way the computation is carried out, but not the algorithm itself. As a result, MPC-based cryptographic libraries can receive FIPS 140-2 certification, and the Unbound vHSM is the first (and currently only) MPC solution to have received this. Specifically, the Unbound vHSM has received FIPS 140-2 Level 1 and FIPS 140-2 Level 2 Certification, with design assurance of level 3.

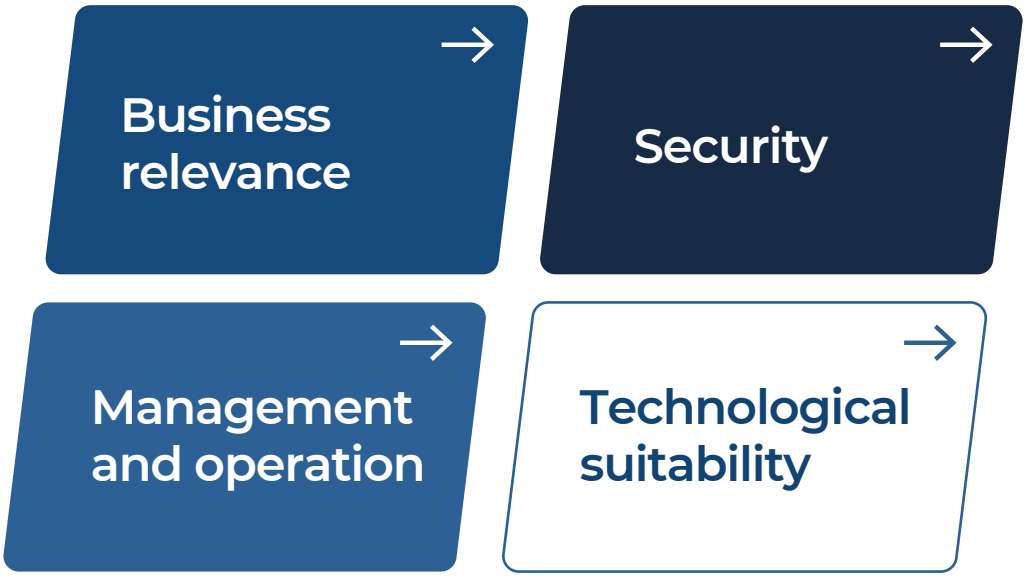
The Unbound vHSM achieves the notion of software-defined cryptography, in the sense that it constitutes a solution that previously required hardware. This transition to a software-only solution means that vHSMs can be added and removed at will, and that general-purpose and virtualized computing can be utilized instead of special-purpose hardware. As we will see in the next section, this has far reaching implications.

Benefits and Features of the Unbound Virtual HSM®

At first sight, the Unbound virtual HSM is "just" an HSM, but in software. Indeed, it requires advanced cryptographic techniques in the form of MPC to be able to achieve this securely, but the result is "just" software. However, this misses the point, and is analogous to saying that virtualization and the cloud economy is "just" having a service provider run your data center. It is well accepted and understood today that virtualization and the cloud have fundamentally changed the way modern computing works and how software is served and consumed. In a time when all of an organization's computing is virtualized, the only physical remnant is an HSM. This has a profound impact on how it works and how it interacts with the rest of the computing environment.

For just one illustrative example, consider the replication of a fully virtualized data center in one geolocation to another fully virtualized data center on the other side of the world. This is actually an easy task, since no changes need to be made to the software and everything can be done remotely. However, if HSMs are used to protect the cryptographic keys in the first data center, then this is no longer an easy task. In particular, physical HSMs need to be procured and physically installed at the other data center. The keys in the HSM at the first data center need to be replicated to the other HSM, which requires physical access, and is a non-trivial task. If the same task were to be attempted to replicate a private data center to the cloud, the challenges would be even greater. In contrast, if a virtual HSM was in use instead, there would be no difference between this and any other application or software in the data center. A virtual HSM enables you to secure your cryptographic infrastructure in the same way that you run all other software, and that is a fundamental difference – a software defined vHSM.

In this section, we will describe the benefits and features of the Unbound vHSM, from four different perspectives:



4.1 Business Relevance

The tradeoff between usability and security is a well-known one, and in many cases it is inherent. In the real world, you truly cannot have your cake and eat it too. However, this does not mean that the friction cannot be reduced, so that the "pain" of deploying security solutions is significantly lower. This same tension also exists between the security organization and business units and operators in an enterprise. In general, a successful business needs to move quickly and cost effectively. However, security and compliance requirements place limitations on what is allowed, and can both slow down deployment and increase cost. Therefore, security solutions that support and enable the business have a big advantage over those that hinder it.

The Unbound software-only virtual HSM is aligned with modern business needs, and significantly reduces the pain associated with protecting cryptographic keys. The following properties of the Unbound NextGen vHSM® demonstrate this fact:

- **Business Agility:** Given the ubiquity of cryptography today, almost all applications require encryption, signing or other services that utilize cryptographic keys. If an HSM needs to be purchased in order to protect these keys, then a lengthy procurement and deployment process is needed. Many enterprises report that this process takes months, and this significantly slows down the roll-out of new business applications. Attempting to solve this by purchasing physical HSMs in anticipation of future business needs is usually not an option, both financially and technically as the required HSM model, licensing and physical location is highly dependent on specific business requirements. Thus security organizations end up slowing down business initiatives. In contrast, virtual HSMs can be deployed in the same way that new virtual machines are spawned. This can even be an automatic process, with the IT or security department providing an on-demand HSM service. Once the virtual HSM platform is initially installed, new applications can be allocated HSM services within minutes.
- **Platform & Environment Agnostic:** In order to reduce cost and maintain high availability and robustness, organizations are working in hybrid environments including private data centers and multiple clouds. Physical HSMs are very challenging to use in such environments. First, not all cloud providers have HSM services, and when these are used they essentially lock the customer in to that cloud vendor. Second, on-prem and cloud HSMs offer services quite differently. Third, the administration of keys (generation, deletion, rotation, etc.) needs to be carried out separately for each environment. All of this makes it extremely hard to work in these environments. In contrast, a virtual HSM works just like all other virtual machines. In particular, virtual HSMs can run in any environment and in any geolocation. Furthermore, all virtual HSMs in all environments are automatically synchronized and so the administration of keys is straightforward.
- **Cost Effectiveness (Software vs Hardware):** Due to its very nature, there is always an additional cost when purchasing additional single-purpose hardware. In contrast, the cost of virtual HSMs decreases the more it is utilized. As the use of cryptography increases more and more, this is an important consideration.
- **Developer Ease of Use:** In order to support business needs, the integration and use of cryptographic keys must be smooth. The Unbound vHSM supports all standard cryptographic libraries (PKCS11, KMIP, CNG, OpenSSL, Java crypto, etc.) and also includes a simple and modern REST API in order to remove from developers the burden of selecting cryptographic functions. In addition, vHSM comes with broad cryptographic support, including advanced techniques like PII protection via tokenization and format-preserving encryption.

- **Compliance:** As we have mentioned above, the Unbound virtual HSM is FIPS 140-2 Level 1 and 2 certified, with Level 3 design assurance (but not Level 3 certification since it does not have physical protection, by design).

In summary, the Unbound virtual HSM is closely aligned with the business needs of modern enterprises. This is one of the stated design goals of the vHSM, since aligning security with business is the best way to help organizations improve the security of their systems.

4.2 Security

Being a software based security solution enables the Unbound NextGen vHSM® to provide all of the benefits described above (and more). However, this is only acceptable if high security is preserved. The days of checkbox security are long gone, and organizations today understand that they need real security. Indeed, more and more organizations view cyber-attacks as an existential threat.

As we have described, the security behind the Unbound vHSM is based on secure multiparty computation, a subfield of cryptography that has been studied for over 30 years. Stated simply, the (mathematically proven) security guarantee is that even if one of the two servers is breached and completely controlled by an attacker (and even if that attacker knows all of the code being used and can run arbitrary attack code), it cannot learn anything about the key. Thus, the single point of failure that occurs when a key is located on a single machine is removed. As discussed above, this paradigm can be used to achieve a very high level of security. One interesting and important property of this is that by not allowing any single administrator to have access to both machines in the vHSM pair, there is no insider who has access to cryptographic keys. This mitigation of insider threat is very significant given the high percentage of breaches that are due to insider negligence or more.

Another important point to observe regarding security is what constitutes “key protection”. The classic model of key protection is to prevent keys from being stolen. This is indeed a major concern since all cryptography is rendered useless if the key is stolen, and this is therefore the security model in all HSMs. Note, however, that if an attacker can breach a machine that is authorized to use the HSM, then they can carry out any operation that the authorized machine can. Nevertheless, in many settings, this goes a long way to mitigating the threat. Consider the case of a database with 10 million encrypted credit card numbers. If the attacker steals the encrypted database and key, then they can just decrypt everything offline and steal all credit card numbers. However, if the key cannot be stolen (e.g., it is stored in an HSM), then the attacker will have to decrypt the credit card numbers one at a time by running malware in the victim system. Such an attack is easier to detect, and can be mitigated using anomaly detection and the like. Although prevention from key theft is an important measure, there are also many cases where it is far from sufficient. Consider, for example, an attacker wishing to steal the plans to Lockheed Martin’s F-35 fighter plane. In such a case, it is sufficient for the attacker to use a protected key once in order to decrypt the file and obtain it in plaintext. For another example, consider the use of cryptographic keys for code signing, to protect software and firmware updates. A single fraudulent use of the key suffices to deploy malware that is accepted by all users as fully legitimate. Thus, in a growing number of cases, what is really needed is protection from key misuse and not just protection from key theft. In a software-based virtual HSM, protection from key misuse can be achieved via a combination of multiple means:

- MPC can be used to require authorization from multiple entities for an operation, with cryptographic enforcement. Specifically, the vHSM can be configured to allow an operation for a key only if the instruction is signed by a key that is split amongst multiple approvers. These approvers use an MPC protocol to generate a signature, so that a signature can only be generated if a full quorum approves the operation (such a quorum can also be threshold, meaning that any large enough subset of the approvers can generate the signature). It is important to stress that these approvers all hold shares of the key, so this authorization cannot be bypassed. In the example of code signing, these approvers can include human and non-human entities participating in SSDLC (secure software development lifecycle) processes – build machines, QA, security scanners, R&D personal, etc.
- Both machines holding key shares can detect misuse. The advantage of this over classic HSM models is that such misuse prevention is done by the vHSM itself.
- Risk-based policies regarding key usage can also be enforced by the vHSM. These policies can include things like rate limiting (how many operations per second a certain key is allowed to be used for), day of week and time of day (e.g., only allowing operations during work hours, or raising a warning at any other time), and so on.

Note that some of the above measures for protection against key misuse require tailoring to the specific use case and should not be considered out-of-the-box functionality.

The Unbound vHSM includes tamper-proof auditing, in the sense that every operation is logged on both vHSM machines and signed by them (so an attacker would need to breach both in order to tamper with the log). This log can be automatically connected to any SIEM (Security Information and Event Management) system like Splunk, QRadar, ELK, etc. in order to track operations and system behavior automatically, including detecting anomalies.

In the classic HSM model, the HSM itself has very limited policy enforcement on allowed key usage (e.g., it can define what operation is allowed, but not which client can use it beyond the use of partitions, nor how often it is rotated and so on). The policies on allowed key usage are mainly enforced by a separate key management system that is connected to the HSM. The disadvantage of this model is that if the key manager is bypassed by an attacker, then the policy enforcement is also bypassed. In contrast, the Unbound vHSM's unified approach to key management and key protection carries out key policy enforcement itself; both machines running the MPC inside the vHSM verify the policies and it is not a separate unit.

Finally, there are important additional properties that are achieved by software, and typically not by hardware. First, software can be updated quickly, achieving cryptographic agility. This means that new cryptographic standards can be incorporated quickly (e.g., new Elliptic curve groups as needed for example in Blockchain applications, new modes of operation, and post-quantum cryptography), and any newly discovered cryptographic flaws or bugs in the system itself can be quickly and easily fixed. In contrast, updating physical hardware is painful and slow, and sometimes extremely difficult (e.g., updating TPMs and smartcards after the ROCA attack). In addition, the model of security by obscurity is long gone, and transparency is well-accepted to be extremely important for security. Although not an essential part of the model, HSM vendors have typically been extremely opaque regarding how security is achieved in their HSMs, and what code is run. In contrast, Unbound's MPC protocols are all open to scrutiny, and customers are welcome to review all details of the security architecture of vHSM under NDA, and even to review the code itself.

4.3 Management & Operation

The Unbound vHSM includes built-in key management, and is not a separate product. The aim of the key management layer is to provide a true single pane of glass, including visibility, control and compliance for all keys, independently of where they are. This is achieved trivially by using vHSM for all keys, since as we have discussed, the same system can be used for keys that are used (simultaneously) in private data centers and any cloud. The vHSM key management system is gradually adding support for managing keys in the cloud (e.g., AWS, Azure). This greatly simplifies work processes for administrators, as it enables the use of a single management system for all of these keys (meaning that policies like key rotation for all keys can be defined and deployed from a single place). Going forward, Unbound aims to manage cryptographic keys and secrets of all types, even those stored inside physical HSMs in private data centers.

One of the difficulties of managing physical hardware, like HSMs, is that much of the administration requires physical access to the machine (e.g., using a PED). With a virtual HSM, all administration can be carried out remotely, and admin quorums can be defined in order to authorize operations like adding users, clients, definition of enterprise-wide crypto policies and so on. In addition, the Unbound vHSM comes with a modern GUI and command-line interface, and supports scripting and automation. This means that new vHSMs can be deployed almost instantaneously and managed even from home, streamlining crypto operations.

As we have described regarding preventing key misuse, our vision is to include the ability to implement rich maker-checker work flows. This is currently supported for specific use cases, which will be expanded to cover any scenario. In general, it will be possible to define a policy that a cryptographic operation requires a certain flow (set of people/machines) to approve, and this approval process is enforced cryptographically using MPC (i.e., each approver, or maker and checker holds a cryptographic key share). Unlike basic maker-check work flows, with vHSM it will be possible to define sets (e.g., 3-out-of-5 of a set of people need to check and approve the transaction). Enforcing this cryptographically prevents failures due to misunderstandings, desires to cut corners, and so on.

An important part of operations is ensuring business continuity in the time of a disaster. The Unbound vHSM can be deployed in multiple data centers and clouds in different geolocations, and all administration and other operations (including adding, deleting, rotating keys, adding clients, and so on) are automatically synchronized across all instances. In addition, the Unbound vHSM's clients automatically redirect their requests to alternative vHSM instances, in case the ones in their area are down. This provides high availability and redundancy without the pain of installing physical machines at multiple locations and manually synchronizing them. In addition, the Unbound vHSM has a built-in backup system (while ensuring no single point of failure) for the unlikely case that all instance pairs fail. Note that at the very minimum, two pairs of virtual HSMs should be installed to ensure high availability.

Another important feature on the operations side is the ability to scale up and scale down vHSM machines, as needed. This is much more cost effective than always having the maximum possible number of HSMs as required at peak time. In this sense, virtual HSMs are part of the cloud economy, and throughput can always be increased by just adding additional pairs. Due to the fact that each vHSM pair works independently of the other, the scale-up of adding additional pairs is truly linear. For this reason, the most cost effective deployment of vHSM is typically multiple pairs of single-core VMs.

4.4 Technological Suitability

When deploying new solutions today, it is crucial that the solution works seamlessly with modern technologies. The Unbound vHSM supports true multi-cloud computing, with the vHSM pairs themselves being in different clouds (e.g., the entry point in one cloud and the partner in another, to achieve strong separation) or within the same cloud but agnostic to it (and so the same system can protect applications in AWS, Azure, GCP, IBM cloud and so on). Currently, the Unbound vHSM can be used in any cloud IaaS deployment. Furthermore, Unbound is working with GCP to enable the Unbound vHSM as an external key management partner, meaning that the keys used by Google to secure BigQuery or the Compute Engine are actually stored inside a vHSM. This means that customers will no longer have to hand over their keys to Google and can maintain control while still utilizing the advanced services that the cloud provides.

Given the fact that it's pure software, a virtual HSM could support serverless (FaaS – function as a service) use cases, enabling organizations to reduce computing costs and scale on the fly, in true as-a-service fashion.

Summary

In summary, the Unbound Next-Gen vHSM works in the way that you expect all of your other software and applications to work. It can take advantage of everything that modern computing has to offer and is fully aligned with modern business needs. This can all be achieved without compromising on security due to the underlying MPC technology which enables very strong key protection via separation (rather than attempting to physically fortify a single machine). The result is the ultimate combination of maximum security, best experience and highest cost effectiveness.

Take a Test Drive of the Unbound NextGen vHSM®

GO TO THE INTERACTIVE DEMO