Diamond Key Security Market Report

In our experience pitching new products ideas in both a large corporate environment and multiple startup environments, we have not seen as much excitement about the possibilities for a platform as we have seen for CrypTech. People continually hear about what CrypTech is and then ask something like, “That’s interesting, have you thought about using it for …?” There is a large range of use cases for an inexpensive cryptography engine, stand-alone, or embedded in other projects. This document captures some that we have vetted or heard from multiple sources as being interesting. We have potential customers (prototype users) for our Diamond Key HSM, Diamond-HSM™, in the DNSSEC space and the identity management space. We will elaborate on those and talk about some other uses that seem interesting.

I Existing Use Cases

At Diamond Key Security we set out to determine whether we could build a business on an open HSM at a low price point and what the initial use cases would be. We quickly settled on DNSSEC signing as our principal use case because it was so key to the work CrypTech was already doing, and shortly thereafter on SAML identity management because people approached us in search of an economically viable HSM solution for use in that space.

DNSSEC signing

We have spent most of our time talking to potential users of the CrypTech technology for the DNSSEC signing application. We have built a business case on this and believe there is enough of a market to get Diamond Key Security to sustainability here, and to begin to grow beyond sustainability to support additional application segments.

When we started collecting requirements for HSMs in DNSSEC there were over 200 ccTLD operators in the world. We have spoken with TLD operators from every continent and have had a good number of conversations. It’s still a sample, and not a very scientific sample. For perspective, our estimate is that we’ve talked to about 8 TLD operators in Asia Pacific, a couple in North America, 3 or 4 in Latin America, only one in Africa, and about 8 in Europe. The description here is a summary report of the findings of those conversations.

There are a couple of modes of operation of HSMs in the DNSSEC signing space: 1) keep the HSM in a safe and use it for key signing ceremonies 2) use the HSM for ongoing signing operations. We (Diamond Key Security) have chosen to implement a device that would be used in the second scenario.

HSM in a safe

People use HSMs for key signing ceremonies in a number of TLDs. Typically, they put the HSM in a safe, or multiple safes in different locations, powered down, and use them to generate KSKs and sign ZSKs a few (variable number of) times per year. The requirements we get for these are very low performance in terms of key signing, very high in terms of tamper resistance, and there are typically additional requirements in terms of signing ceremonies (interface to external devices for authorization, M of N operation). It is in deployments like this where people want very significant high-levels of security including either FIPS certification, or something equivalent, typically at Level 3. These security requirements drive complexity and cost in the HSM selected.

The requirements in terms of key management vary widely. Some TLDs want to duplicate the keys across all of these HSMs in geographically diverse locations, others want the keys never to leave the HSM. Most want at least a key backup to a different HSM even if it is kept in the same location.

It is this type of operation where the very low power operation is required – the ability to detect tamper and destroy keys while asleep for potentially months at a time.

Online signing

People have a variety of solutions for online signing, and a variety of methods of accomplishing the task. While there is variety in this use case, there are a few common requirements. Typically, people need a device that is continuously connected. Operators conveyed early in our requirements gathering that they wanted a network appliance or unit that was rack mountable and accessible via Ethernet. We have built such a device and all feedback we've received on that has been positive.

People want this box to be tamper resistant but the requirements are not as clear as those who are storing an HSM in a safe. Depending on the application, the specific view of vulnerability and the overall security paradigm of the network operations will drive the requirements for the HSM security in this case.

There is a wide range of performance requirements for these kinds of operations. We have spoken to operators who require signing performance of 10,000 signatures per second, and to operators who are content with 10 signatures per second. This is a surprisingly difficult requirement to get from a TLD operator. There seems to be great reluctance to share or possibly a lack of understanding of the number of signed records TLD operators have and what the signatures per second performance actually is. We are not building a device that can operate on the order of 10k sigs/sec. When discussing a device on the order of 100 sigs/sec, many think that is probably workable and acceptable. It would be nice to be able to say that if we could provide performance at 100 sigs/sec we could cover some known percentage of the market, but we have not been able to clearly quantify that.

The typical configuration we have heard from operators is that they use a 2048 bit key for the KSK and a 1024 bit key for the ZSK. Verisign announced at DNS-OARC (Oct 2018) that they planned to move to a 1280 bit key for the ZSK in all the TLDs they manage and since then other operators have told me they are investigating doing the same. .cz and .br now use ECDSA for signing, and several other operators are actively investigating that as well now.

Expectations based on existing HSM operations

It seems that a lot of DNSSEC signing operations are designed around capabilities of existing HSMs and modes of operation that are typical for them. It's not clear that there are requirements for this in many cases. There are certainly desires to do things that are atypical for existing HSMs. An example of this is response to tamper events. Multiple operators have HSMs that are functionally disabled on a tamper event and would like that not to be the case. They would prefer it if the HSMs would destroy the keying material stored in them on a tamper event, but then could be reset on site and continue to be used in operation. In at least one of these cases it's because an operational error triggered a tamper event (someone forgot to screw the unit into the rack….).

Miscellaneous requirements

We've had a couple of requirements from operators that it seems to be worth articulating although it's not clear yet whether there is anything really actionable on them. Many people would like it if there was an alternative to PKCS#11. One operator uses KMIP. A couple of folks have suggested it would be interesting to produce an alternative open standard to do what PKCS#11 does. One wonders whether it would be worthwhile to seek research funding for a research project to do an open source alternative to PKCS.

A few folks have shared a requirement for M of N authentication. This typically goes with the use cases of keeping an HSM inside a safe, but not always.

SAML identity management

We have engagement with a number of academic research networks who are interested in using HSMs for identity management, mainly for applications signing federation metadata (data at rest). We have had one conversation with a network who is also interested in using this for signing SAML assertions. Right now, the data is signed with RSA-2048 and uses the PKCS#11 interface. The signature requirement now is to support a peak load of around 800 signatures per second. It is conceivable that we could build an open solution based on a CrypTech device if it could support something on the order of 100 signatures per second. FIPS certification is not

a requirement here, but something that is more secure than what is used today. An open solution is highly desirable.

II New Use Cases

While doing business development and fund-raising for CrypTech we have heard many suggestions and ideas for additional uses for the technology. I'm not sure we have a comprehensive list. In this section we have documented some use cases that have come up on multiple occasions, or where we've had some more serious interest.

Hash-based signatures for Signed Code Updates

In 2018 CrypTech implemented hash-based signatures in its code base. A large Internet equipment vendor expressed interest in using CrypTech for signing software updates, with enough interest to acquire a CrypTech device and do some R&D using the device. Since that time the LAMPS working group has taken up the specification of hash-based signatures with the Cryptographic Message Syntax for use in signed distribution of software updates and in Internet of Things environments.

There are a few organizations participating in LAMPS who are determining whether there are concrete opportunities for deploying hash-based signatures. Some of those have their own implementation and related technology. We continue to explore opportunities for deployment of this technology.

RPKI

RPKI has been a target for implementation with CrypTech from early in the project. We have not spent a great deal of time pursuing it as a market opportunity because the number of operators is very low. An additional consideration has been that from our understanding there are a few implementations among this low number of operators each of which may require special customization. While we would be happy to provide a generic device that could be used, it doesn't seem likely that there is enough revenue or outside support for this to add much to the CrypTech effort.

In 2019 it is possible that this situation may change. NLnetlabs is developing an open source suite of tools for supporting RPKI in enterprise deployments. If their predictions about uptake are correct, this may lead to multiple operators using a single implementation, so supporting that implementation may be worth the investment. We have spoken to NLnetlabs about requirements for an HSM supporting their envisioned implementation and it seems doable from a functionality point of view. From a performance point of view, some increase on the

CrypTech hardware would help, but probably performance in the order of magnitude of 100 signatures per second would be more than adequate. NLnetlabs agreed to start exploring with their potential customers interest in using an HSM in their implementations and we look forward to additional feedback from their activities.

User portal

APNIC has shared a use case in addition to DNSSEC signing and RPKI for their user portal. The application would be to replace a soft HSM with a CrypTech-based device to manage keys in an openSSL based login system. The interface is PKCS#11 and uses RSA with 2048 bit keys. There may be some code customization on the interface side but from the requirements he has described it seems we have this covered already. Performance requirements are in reach with current performance measurements.

In-line encryption

At our F2F in 2017 in Stockholm, Nullvad attended with interest in using CrypTech for in-line encryption of VPN connections. At that time we had not done much (any?) work on exploring this as a target application. It is recorded here in order to put on the table the possibility of starting to address this as an application. We have not tried to find any additional users for this kind of application, but perhaps it is worth considering.

Related to this is a challenge that we have in funding. There is quite a bit of research funding available related to topics of privacy. In approaching a couple of these opportunities, the funders find it difficult to draw a strong enough connection between privacy use-cases and key-generation, storage, and signing operations. Perhaps it is worth considering exploring privacy applications as a way of getting additional funding for CrypTech.

Tor consensus

There is a small set of directory authorities in the Tor network. Each hour these directory authorities vote on which relays are part of the network and the resulting consensus document is signed by a medium-term key. Right now this signature is performed with an RSA-3072 key, but it would be interesting to investigate other alternatives (Ed25519 for example). In the nearterm the plan is to design a mounting bracket for the CrypTech Alpha that would allow an alpha to be mounted in a PCI slot (but not attached to the PCI bus).