CrypTech 2015

End of Year Report

The CrypTech project was formed in response to indications, based on the Snowden revelations, that hardware implementations of key cryptographic algorithms and functions have been systematically targeted in an effort to weaken and subvert their utility. The goal of the project is to create an open source design for Hardware Cryptographic Modules (HSMs) and an associated reference implementation that allows anyone to deploy and audit a secure, low cost cryptographic engine in their environment.

The vision of CrypTech is that key security infrastructure, like DNSSEC, RPKI, TOR Consensus, PGP, Lets Encrypt CA, etc. should not have to depend on closed HSM products that not only cannot be audited but where there is real possible cause to distrust the functions implemented in the product.

To that end the **CrypTech Mission Statement** is:

- Put hardware crypto capability in the hands of as many people as possible by
  - making it low cost - under $1k for a real HSM
  - allowing diverse HSM manufacture by multiple vendors, anywhere
  - enabling good crypto at the edge (i.e., in applications where it is out of scale and out of price range today)
- Making HSMs more trustable by
  - diverse design and development
    - away from the DC beltway, NSA, GHCQ, Beijing, Tel Aviv
    - cooperation of a diverse set of (US, SE, RU, DE) engineers
    - diverse testing: US, RU, ...
- Being open source so it can be inspected

CrypTech High Level Goals

The fundamental goal of the CrypTech project is to create an open reference platform for a hardware security module (HSM). This platform includes:

- Source code for both the hardware and software designs. For hardware, FPGA HDL code, schematics, board layout and BOM are provided.
- Tools, documentation and examples to allow anyone to implement (or have someone implement for them) an HSM that is tested and evaluated to establish trust in the HSM suitable for the users need.

The HSM being developed supports a number of use cases, both in terms of functionality and performance. The HSM also improves security with a well defined trust boundary and support for physical tampering detection and response.

The specific functions being developed to meet these requirements are:

- A state of the art random number generator.
- Symmetric and asymmetric cryptographic implementations for AES, ChaCha20, RSA, GOST, and Elliptic Curve Cryptography.
- Standards based secure key wrapping and key management.
- Active tamper detection with master key obliteration.

The reference platform will not be sold as a commodity HSM by the CrypTech project. A few will be produced for prototyping and analysis. Instead interested parties that need a HSM will have to assemble and manufacture their own HSMs, or have a third party do the manufacturing. The open licenses used for the reference platform allow unrestricted use of the reference design, in part or as a whole, including manufacturing and selling HSMs and/or other commercial and non-commercial uses. The license used for all code and other technical artifacts is a mix of 2- and 3-clause BSD.

The open design principles of CrypTech provide several specific benefits:

- All code is open and available under open and unrestricted license.
- Open, transparent, auditable, and traceable development process.
- Open design that allows for customization, observation and testability - in development as well as during operation.

The primary focus for the development is to produce a design of an HSM with better security than that of a commercial HSM. A secondary goal is to develop a trustworthy tool chain, test programs and documentation needed to implement and verify a HSM based on the reference platform. Unfortunately, the project does not currently have the resources to focus on things such as the tool chain, enclosure, usability, and geek-free user interfaces.

Current Technical Status

The current prototyping version of CrypTech is based on the Novena platform [NOVENA] with the software using libtfm, while the HAL API is loosely modeled on libtomcrypt, and of course the crypto hardware inside the FPGA on the Novena. The Novena based CrypTech HSM is a network connected HSM that supports DNSSEC zone signing using PKCS#11 commands. The CrypTech software and hardware is available as Debian packages for the Debian Linux running on the Novena.

The FPGA contains support for key cryptographic primitives, including modexp for RSA, AES, ChaCha20, SHA256 and SHA512. ECDSA is nearing completion.

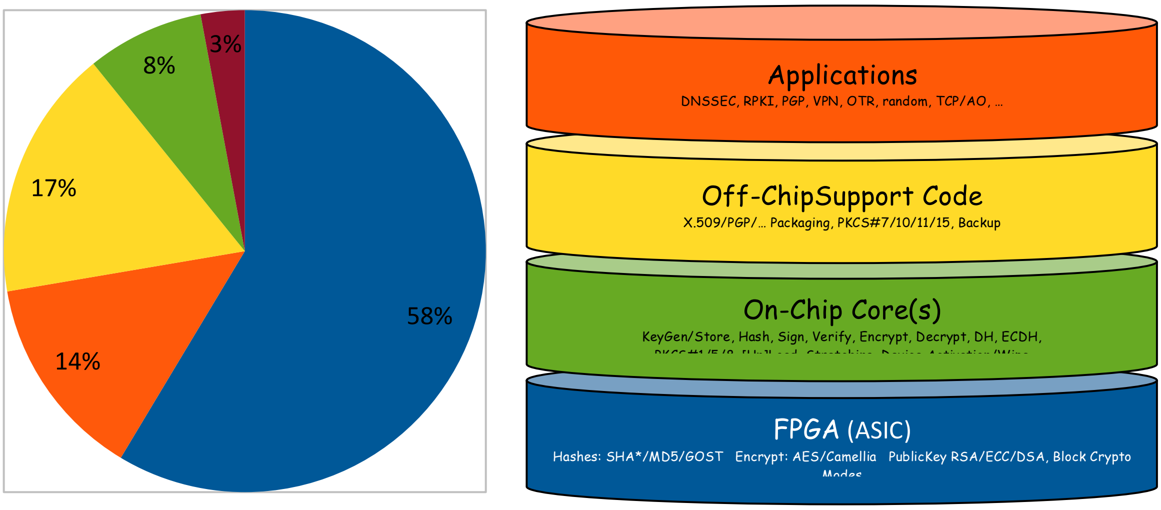
The current development effort is focused on moving from the Novena to the CrypTech "Alpha" board. This board is a CrypTech specific design that provides more FPGA resources than the Novena, a software execution environment with less features and thus reduced attack surface than the Novena, and without all the Novena's laptop appliances. The Alpha board also has hardware support for implementing active tamper detect mechanisms.

The Alpha hardware includes an ARM Cortex-M4 MCU, an Altera Artix-7 FPGA, and minimal peripheral modules. In parallel support for elliptic curve (EC) acceleration and the next generation of the CrypTech random number generator are nearing completion. The hardware-based TRNG is in late stage testing and will be included on the Alpha board.

Software will be developed to include management access, continuous monitoring, and management of hardware primitives, as well as tamper detection and response. Certain aspects of an operational HSM environment such as operator panel functionality including key injection, key pad, display etc. are currently out of scope but expected to be provided by a commercial offering based on the CrypTech technology. While CrypTech does not plan to have the prototype Alpha board certified, CrypTech assumes that boards based on the CrypTech design will go for FIPS and CC.

End of Year Status for 2015

Expenditure By Layer

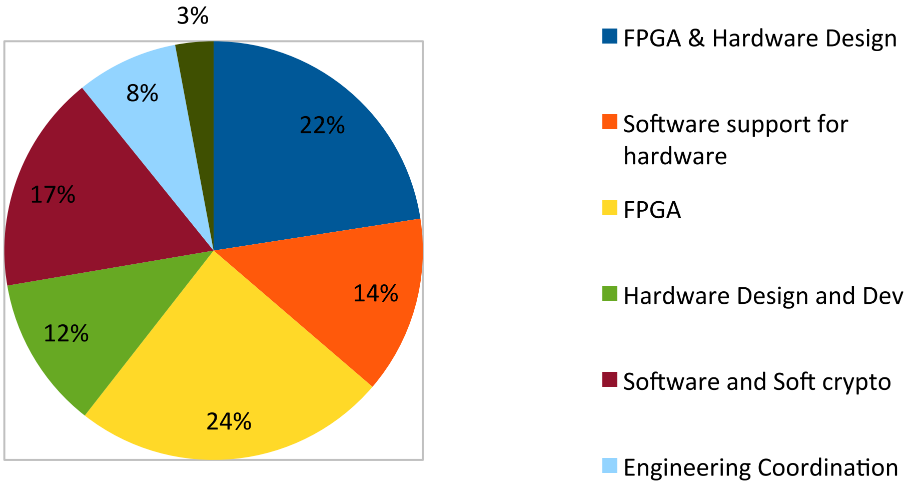


The colors map to the colors in the "CrypTech layer cake": Blue layer is FPGA, Green layer is off-FPGA hardware including a CPU for communicating with the FPGA, tamper circuits, memory, host interfaces etc. Yellow layer is embedded code in the Green layer hardware. Finally Red layer is application integration. Purple denotes administration, coordination and project management costs.

This picture shows where the CrypTech project has been using resources: most have been spent on work in the lowest "Blue" layer which includes hardware design, FPGA programming - notably design of the RNG subsystem where the project has spent a lot of time and energy during 2015 - producing excellent results. Work in the Blue layer also included implementation of several cryptographic primitives including ECDSA, modexp for RSA, ChaCha20, AES etc.

Work in the Green layer has included software implementations of RSA, EC, etc., as well as interfaces to the hardware versions, and the implementation of a hardware abstraction layer (HAL) that to enable the CrypTech codebase to run on multiple hardware implementations.

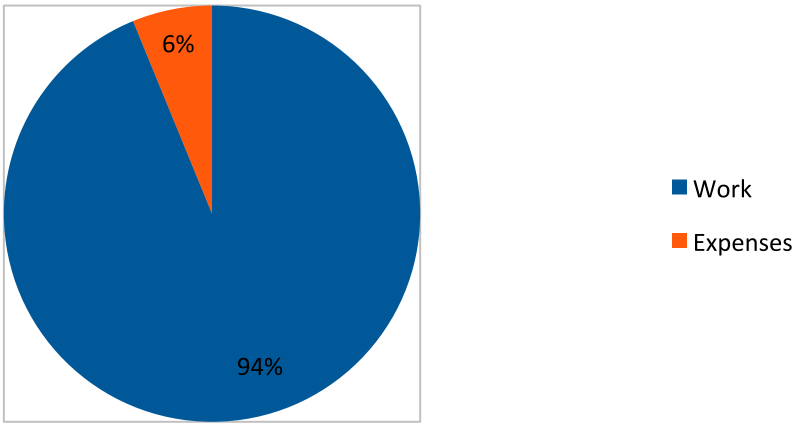
Finally work in the Yellow layer has focused on the PKCS#11 implementation which was used in the Praha workshop to demonstrate fully working integration with OpenDNSSEC.



A more detailed view of expenditure by engineering focus area also shows the same pattern.

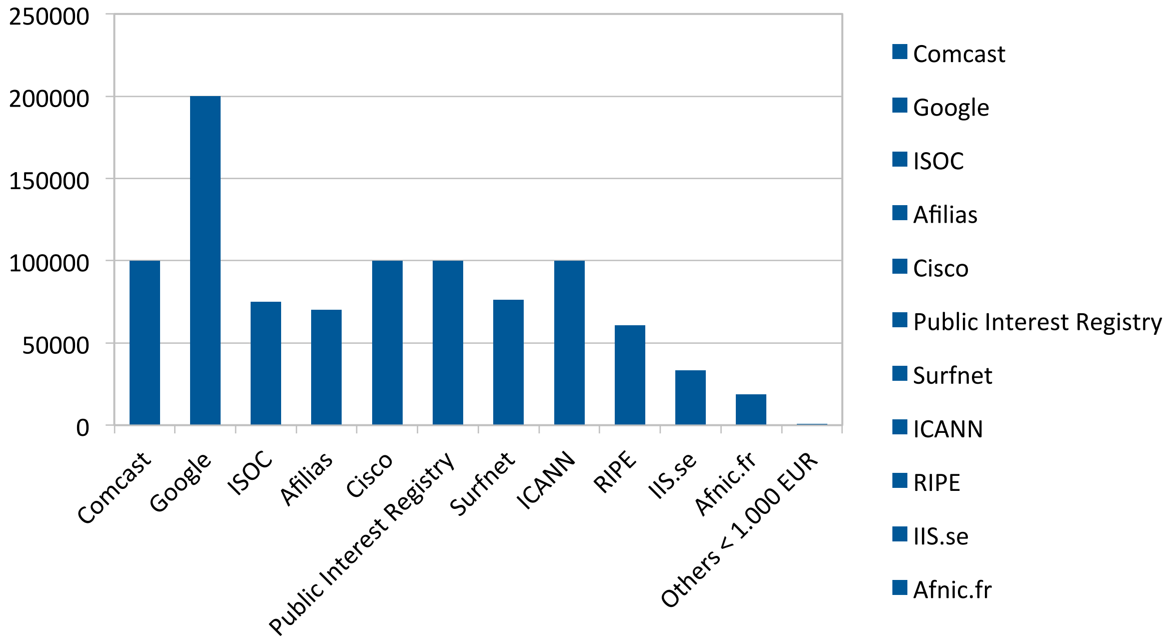
Expenditure By Type (work/expense)

The CrypTech projects spends around 94% of donated resources on engineering. The remaining six mostly covers travel for the (few) project meetings. We try to combine project meetings with travel for other events in an effort too minimize travel costs for the project.



Donations

Donations to the CrypTech project is capped at $100k USD per donor per year. The graph below aggregates data for 2014 and 2015. To date donations totaled $870k, less than half of the planned budget. This has slowed development significantly. At this writing, half of the engineers have been on hold for a few months due to lack of funds.



2016 Goals

Assuming full funding, our goals for 2016 are as follows:

- Produce the Alpha board
- More validation of the random number generator
- Complete and test ECDSA in both software and hardware
- Add Ec25519 in software and hardware
- Finish GOST hardware
- Sign and validate DNSSEC in software and hardware
- Sign and validate RPKI in software and hardware
- Sign 'Let's Encrypt' CA
- Demo a 'Let's Encrypt' client with very slow TLS
- Hold workshops/hackathons to shake out use cases and usability
- Prepare to help a few folk make HSM product in 2017
- Prepare to help a few folk deploy and use applications in 2017