CrypTech

Building a More Assured Hardware Security Module with a More Assured Tool-Chain

Hardware Security Module

Hardware Security Module

From Wikipedia:

> A hardware security module (HSM) is a physical computing device that safeguards and manages digital keys for strong authentication and provides crypto processing. These modules traditionally come in the form of a plug-in card or an external device that attaches directly to a computer or network server.

HSMs Are Used For

- Principally, Lock-box for Private Keys for
  - DNSsec
  - RPKI
  - PGP
  - ssh
  - Your use case goes here
- Also,
  - Encryption / Decryption
  - VPNs
  - Source of Randomness

The Need

- Every week a new horror about Crypto/Privacy
- *der Spiegel's* revelation of the "SpyMall Catalogue"
- Compromises of most network devices, servers, firewalls, ...
- We are relying on HSMs which are designed and made by government contractors
- Many people are not comfortable with this

Origins

- This effort was started at the suggestion of Russ Housley, Stephen Farrell, and Jari Arkko of the IETF, to meet the assurance needs of supporting IETF protocols in an open and transparent manner.

- But this is NOT an IETF, ISOC, ... project, though both contribute. As the saying goes, we work for the Internet.

Goals

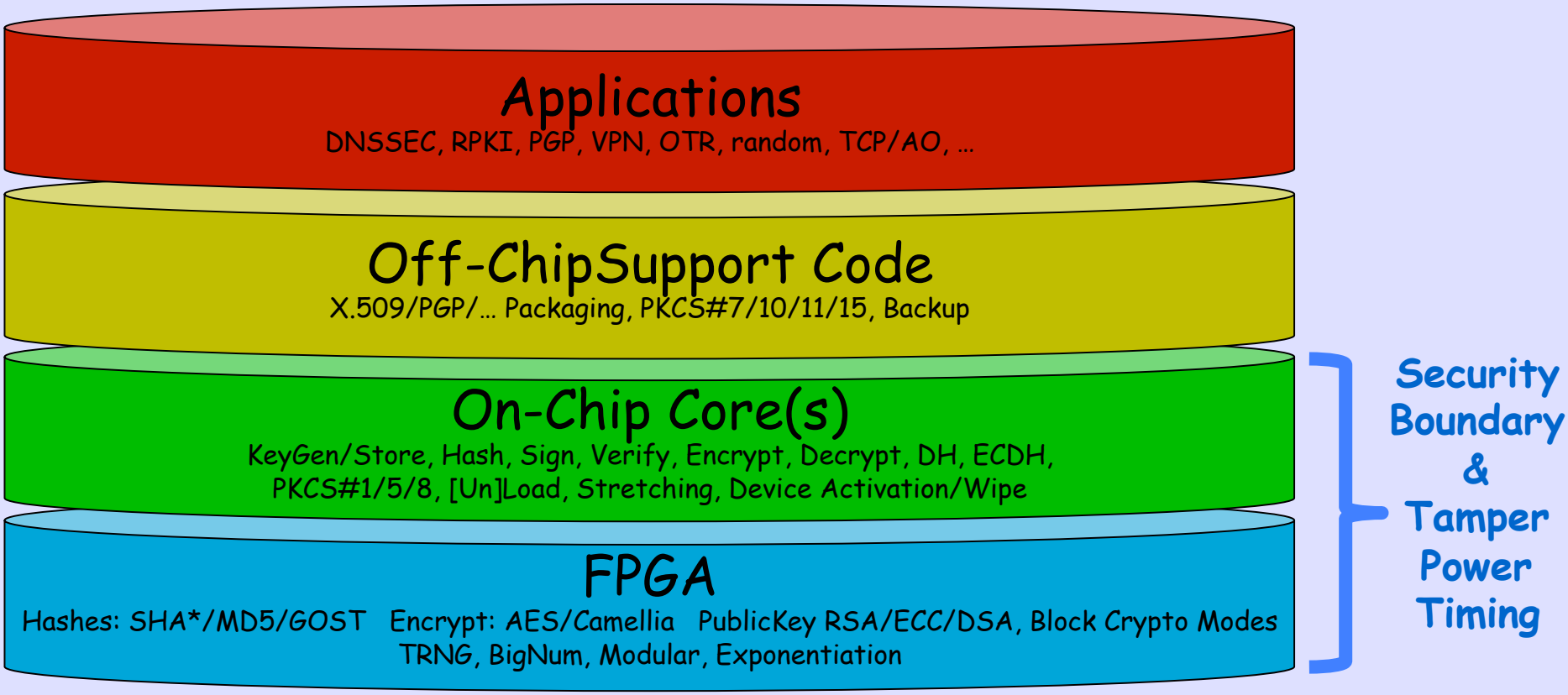
- An open-source reference design for HSMs, not a product
- Scalable, first cut in an FPGA and CPU, later allow higher speed options
- Composable, e.g. "Give me a key store and signer suitable for DNSsec"
- Reasonable assurance by being open, diverse design team, and an increasingly assured tool-chain

Open and Transparent

- The project is being run in a maximally open, transparent manner with traceability for all decisions etc.

- We do this in order to build trust in the project itself

Layer Cake Model

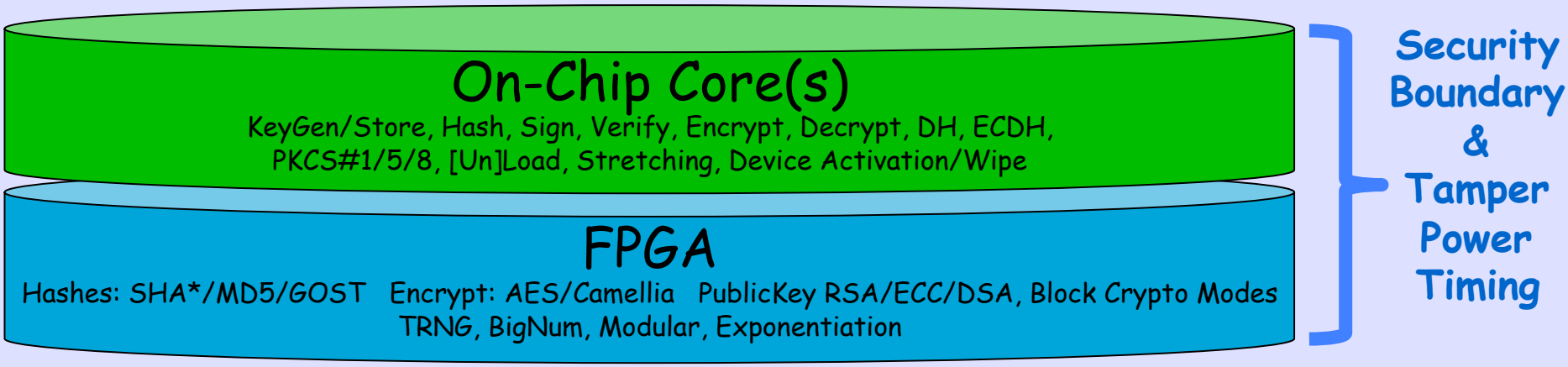


Side Channel & Tampering

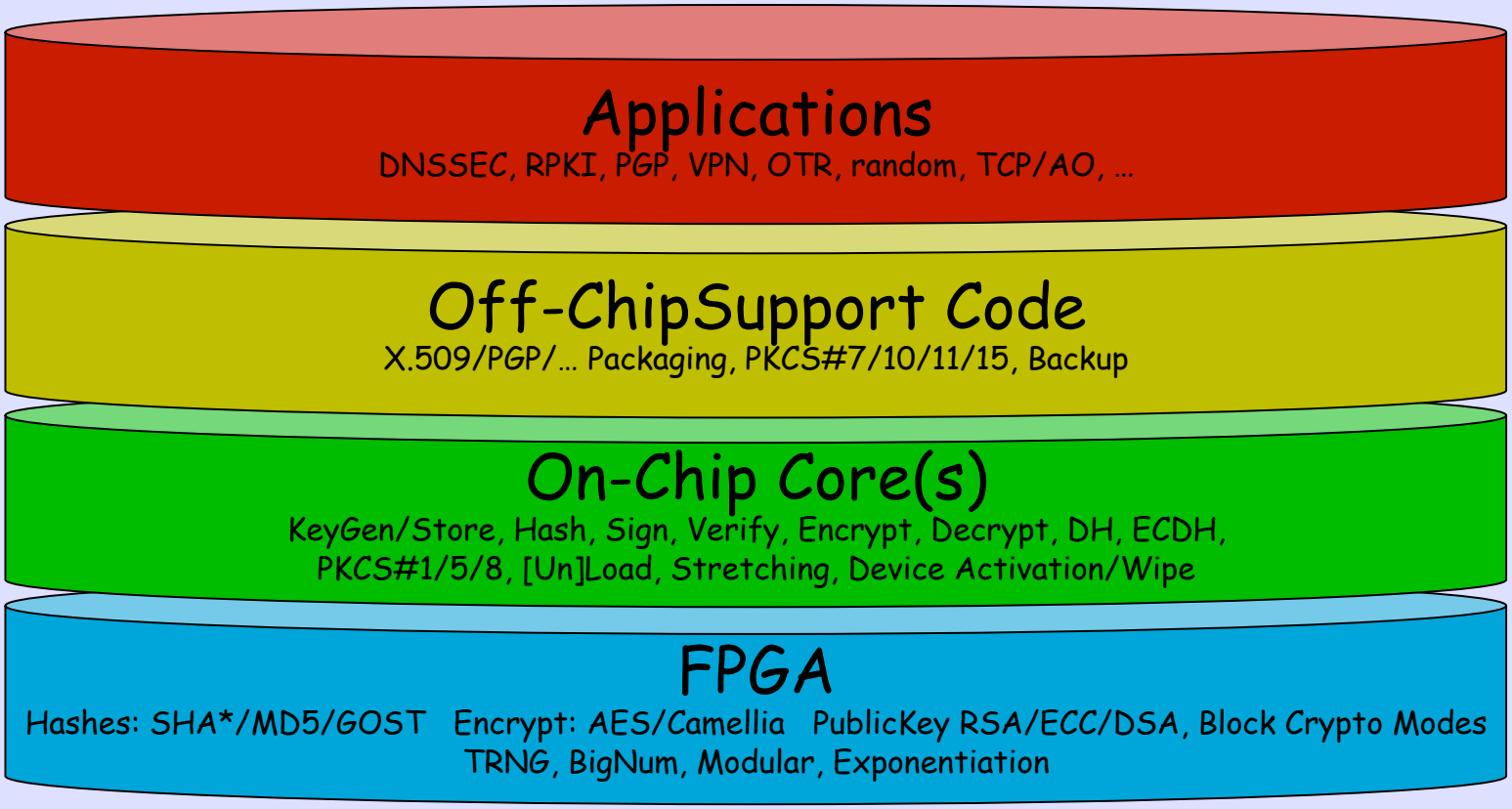
- Exponentiation circuit timing leaks are exploitable remotely
- Power leakage is exploitable locally
- Physical attack detection critical
- Wipe signal to chip
- On-board battery/capacitor to buy the time to wipe if unplugged from power

Potting Boundary

- The FPGA/ASIC and accompanying Core(s) (ARM, whatever) are within the physically protected boundary of the chip carrier potting.
- We worry about side channel attacks, i.e. information leakage from how power is used, RF, data-dependent time to do an operation, etc.
- We worry about tampering, what if the chip is opened and attacked? So the potting includes tampering sensors and code to wipe all keys if tampering is detected.



Writing 'Code'



C or HLL
e.g. Python

C or HLL
e.g. Python

C &
Assembler

HDL, e.g.
Verilog

The Tool Chain

- When my laptop's fan goes on, I think it is the NSA, GCHQ, Israelis, Chinese, … fighting to see who will own me
- We have NO ASSURANCE of our tool set, from CPU to Kernel to Compiler to …
- When constructing assurance-critical tools, we need to maximize assurance in the tools used to build them

The Compiler

- Ken Thompson's 1984 Turing Award paper *Reflections on Trusting Trust*

- A self-reproducing trap in the C compiler which "would match code in the UNIX "login" command. The replacement code would miscompile the login command so that it would accept either the intended encrypted password or a particular known password." You have been owned!

Double-Diverse Compilation

- In his 2009 PhD dissertation, David Wheeler explained how to counter the "trusting trust" attack by using the "Diverse Double-Compiling" (DDC) technique
- We can use this on GCC and clang to get somewhat assured compilers
- But you still have to inspect the source!

HDL / Verilog

- But FPGAs/ASICs are programmed in a Hardware Description Language, Verilog
- It is very hard to get an open Verilog compiler
- Verilog can not compile itself, so DDC is not applicable here, just a DCC C compiler
- We are working on methods of gaining trust in the FPGA tool chain

That is Just the Start

- Of course this is all behind an "air gap"
- So one can use the compiler to make a kernel and operating system
- But who audits the kernel, libc, … source?
- Can double comparison (Intel, AMD, …) give us some trust in the hardware?
- And all this really needs diverse auditing.

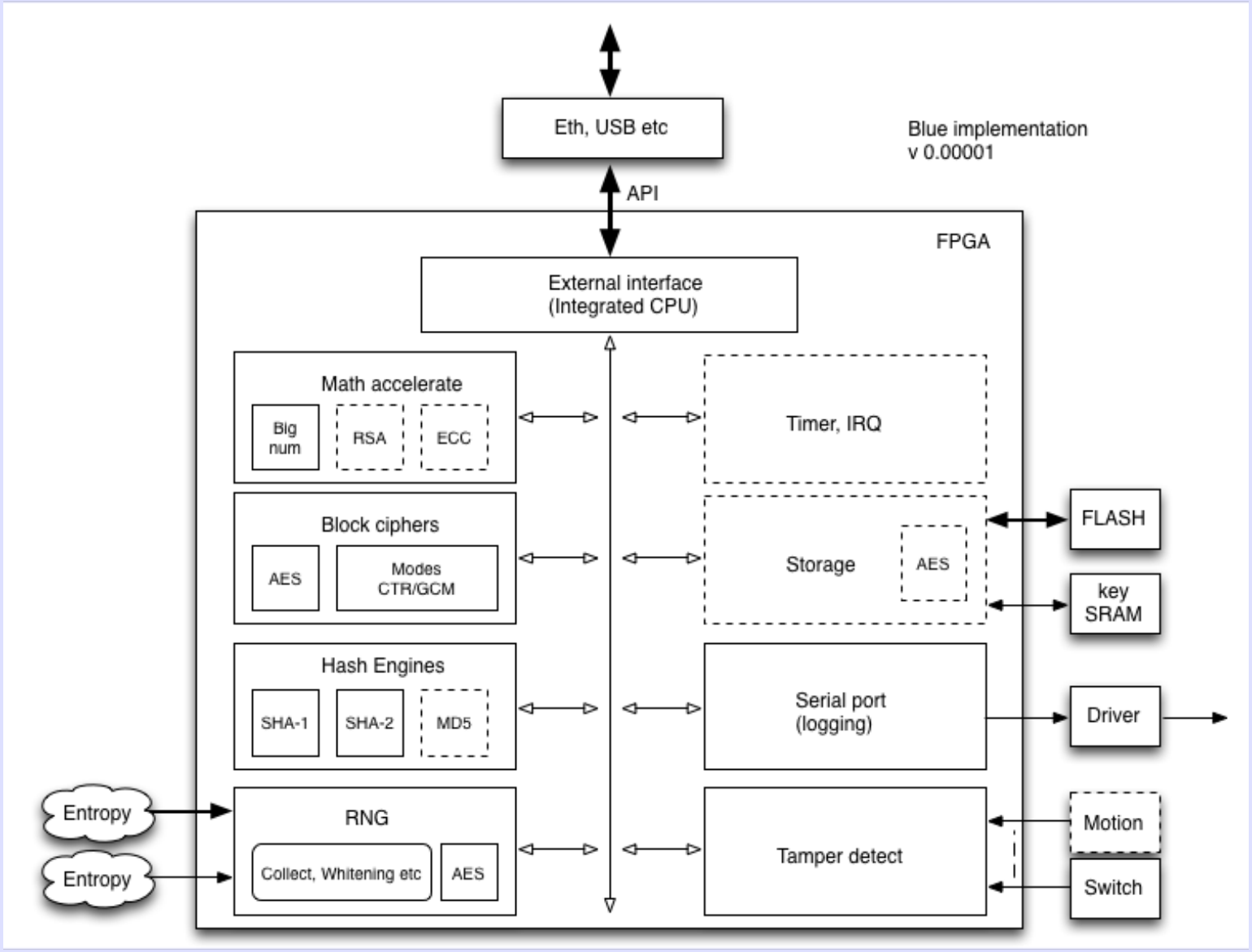
Critical Tool-Chain

- C compilers audited and built using DCC
- Audited kernel, libc, …
- Audited whole darn UNIX or Linux
- Hardware tools are mostly proprietary!
- Audited Verilog compiler
- Audited FPGA download tools
- Audited test tools
- Hardware trojan detection

It Takes Time

- First Year: Tool-chain, Basic Design, not all cyphers, not all protocols, prototype implementations on FPGAs and boards
- Second Year: Better Tool-chain, all needed cyphers, hashes, crypting, … and integration with some apps, DNSsec, RPKI, TLS, PGP, Tor
- Third Year: Solid packaging, ability to compose designs for use cases, etc.

FPGA Cat Video



A Few Related Projects

- Truecrypt audit: `http://istruecryptauditedyet.com`
- OpenCores: `http://opencores.org`
- Icarus Verilog: `http://iverilog.icarus.com`
- ModelSim compiler/simulator
- Valgrind: `http://valgrind.org`
- clang+llvm: `http://clang.llvm.org`

Diversity & Transparency

We have two main means of increasing assurance

- Diversity: finance, engineers, and reviewers from many places, cultures, and politics
- Transparency: all lists open, all code open, all finances open, …

Diversity Improves Trust

- Diverse Technical Team - from diverse countries / environments
- Transparent Development - code, designs, documentation all public
- Auditable and Audited – Please help audit

Diverse Funding

- Multinational and Multi-Stakeholder
- Industry, Academe, Social, …
- Diversity is critical, no donor > 10%
- No anonymous donations

No Project is an Island

- We'll steal from anybody! ☺
- We'll share with anybody!
- We incite others to help, copy, clone, …
- If donors are generous, we will finance others working on related/needed work
- We desperately want to further diversify and to build trust

We Seek Review

- We seek, expect, and encourage any form of open and transparent review
- We will not (soon) seek certification as cost/benefit is high
- But we encourage/expect others to take the designs down that path
- We will document all security claims

Minimal Organisation

- Finances at the non-profit in Nordic R&E network, NORDUNET
- Administration at Swedish R&E network, SUNET, Leif Johanssen and Maria Häll
- Technical: a cooperative of very senior folk with coordination, not management
- Fund Raising – All of us, you too!

Running Code

We do have cores running on proto boards

- SHA-1 hash
- SHA-256 hash
- SHA-512 hash
- Start of RNG processing (no sources)

But we're just getting going