CrypTech

Building a More Assured HSM and Obsessing About the Tool-Chain

Randy Bush <randy@psg.com>

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
  - Corporate Authentication
  - Encryption / Decryption
  - VPNs
  - Source of Randomness
  - Your use goes here

The Need

- Every week a new horror about Crypto/Privacy
- *der Spiegel's* revelation of the "SpyMall Catalogue"
- Compromises of and trojans in most network devices, servers, firewalls, ...
- **We are relying on HSMs designed and made by 42-eyes government contractors**
- Many people are not comfortable with this


OBAMA MEETS WITH CHINA'S DICTATOR
ANY COMMENT ON YOUR OUTRAGEOUS CYBER-SPYING?
IS THAT QUESTION FOR ME, OR HIM?
ASSOCIATED PRESS

Origins

- This effort was started at the suggestion of Russ Housley, Jari Arkko, and Stephen Farrell 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 <u>reference design</u> for HSMs**
- 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
- And **diverse**, engineering and funding

Funding (so far) From

SUNET

SURF NET

Internet Society

Google

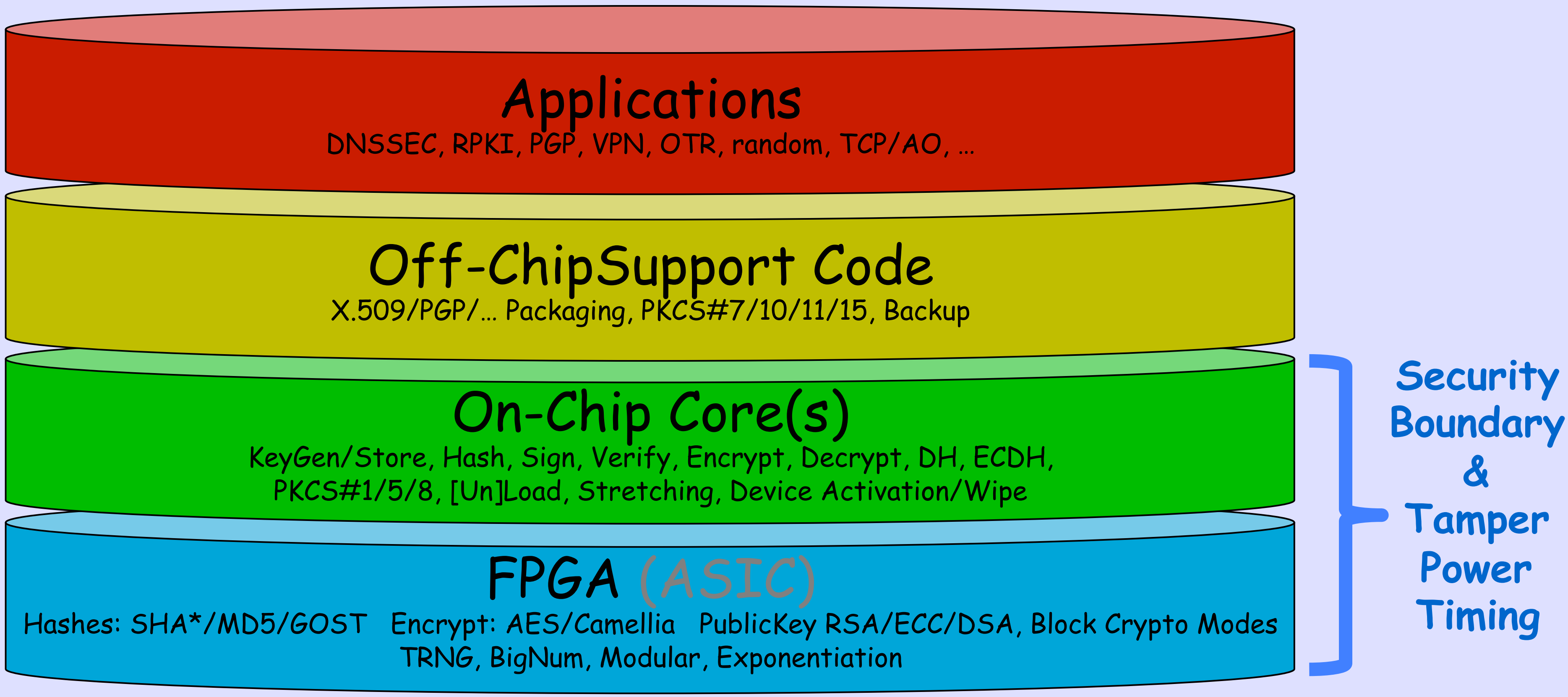
A Few Private Donations

Your Public Interest Registry

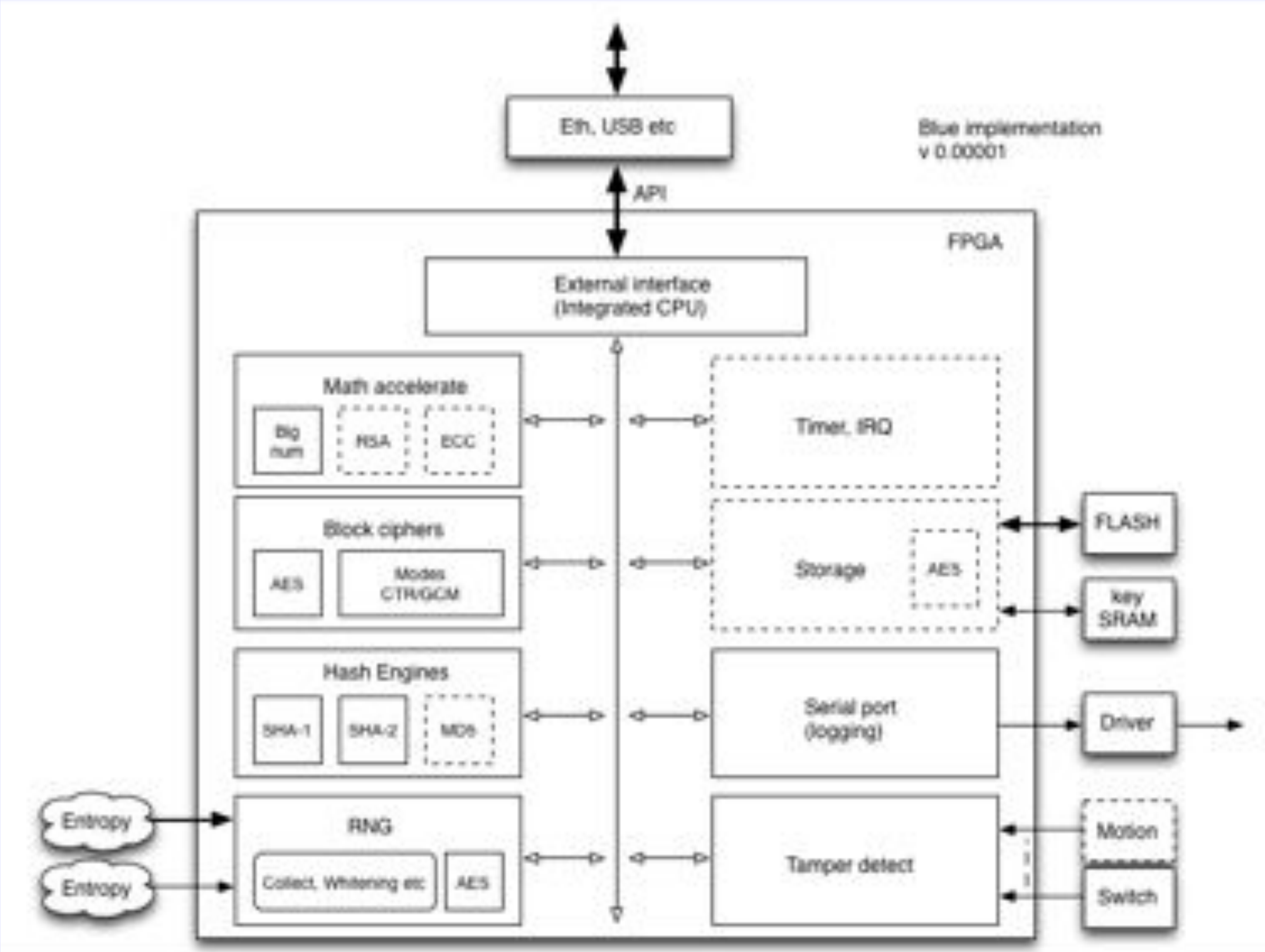
iana

COMCAST

Layer Cake Model

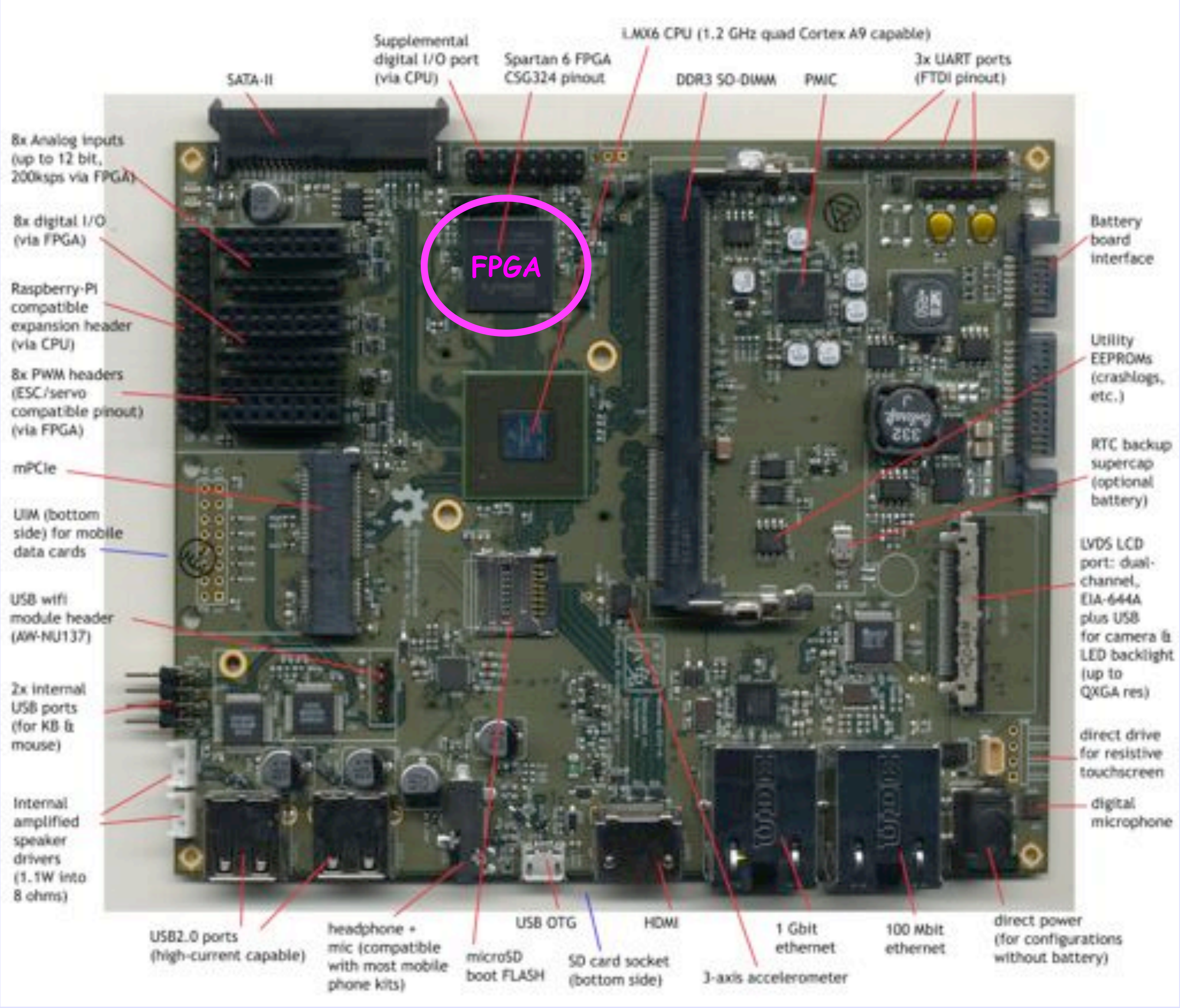


FPGA Cat Video



A Prototyping Board

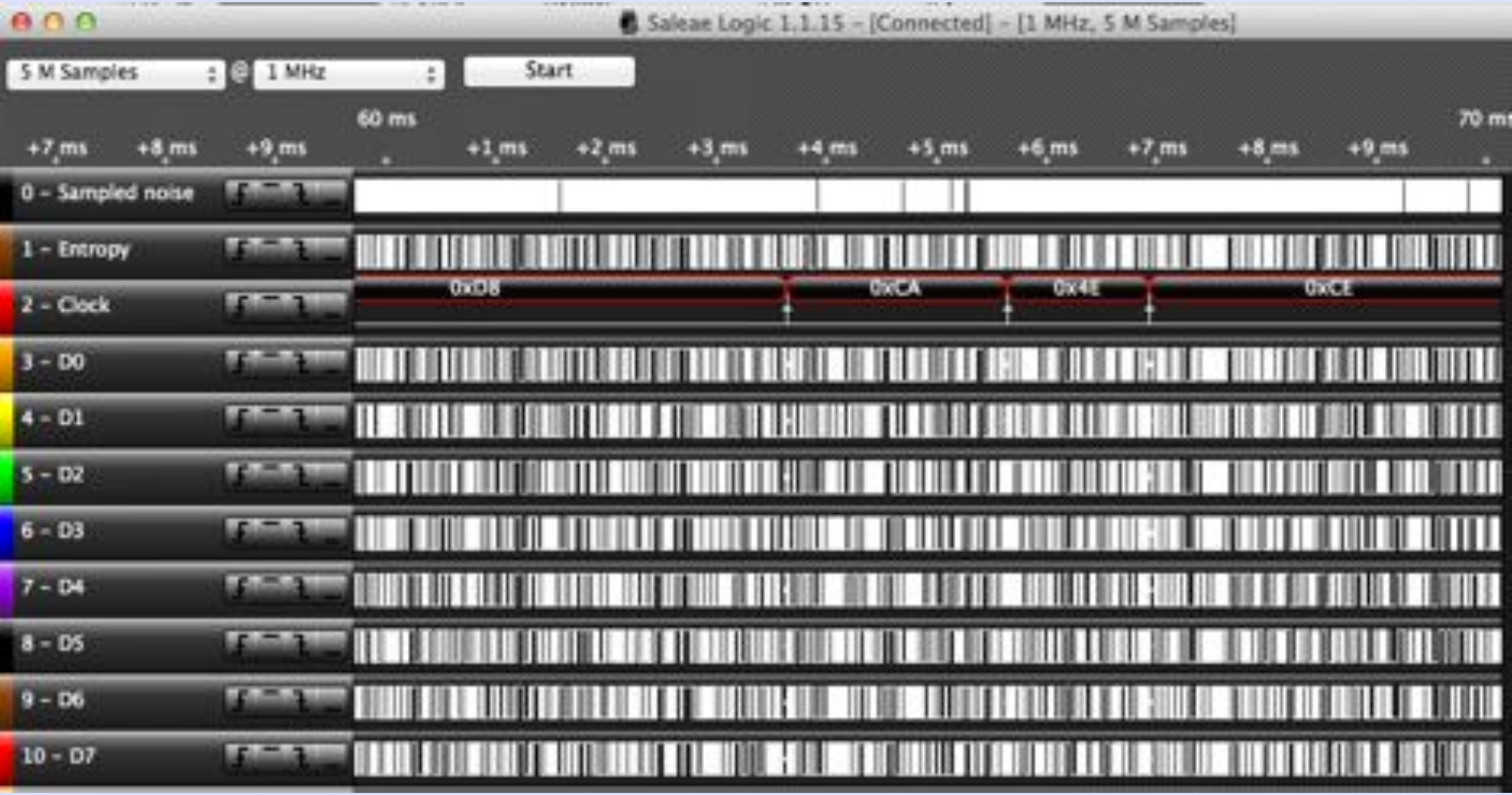
Novena Spartan 'Laptop'



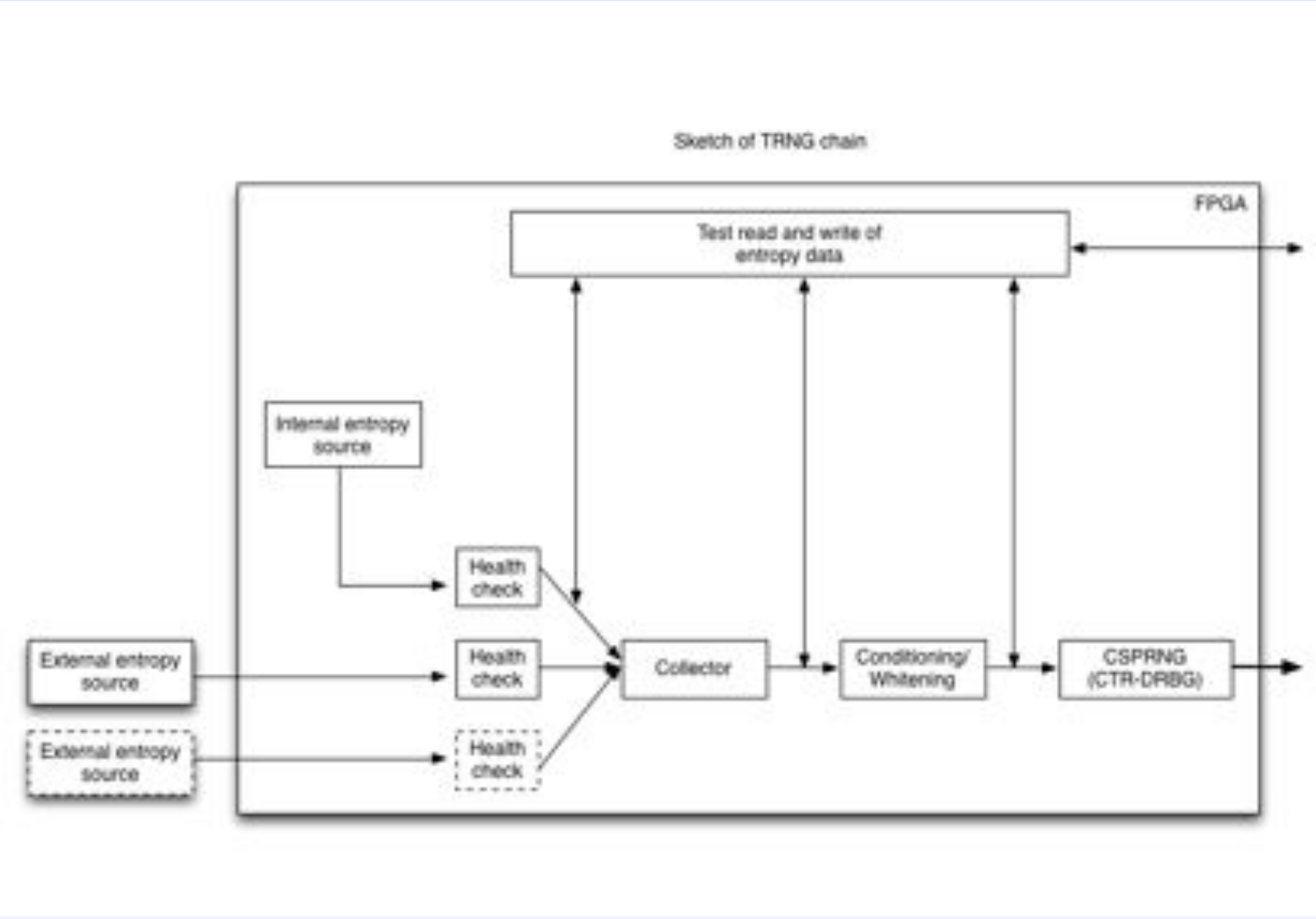
Entropy with Pi Pin-Out

Entropy on Novena

Entropy Porn

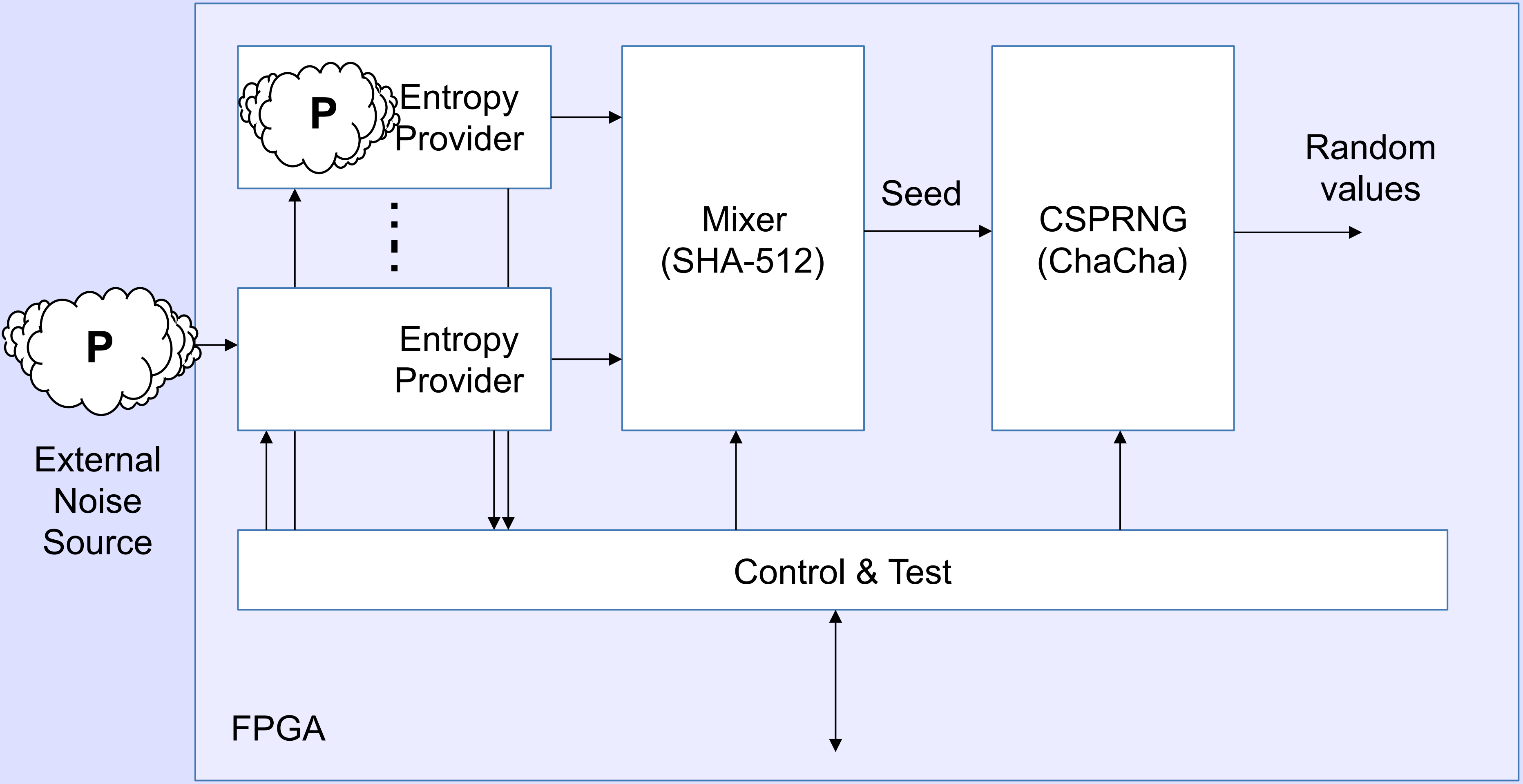


TRNG Chain

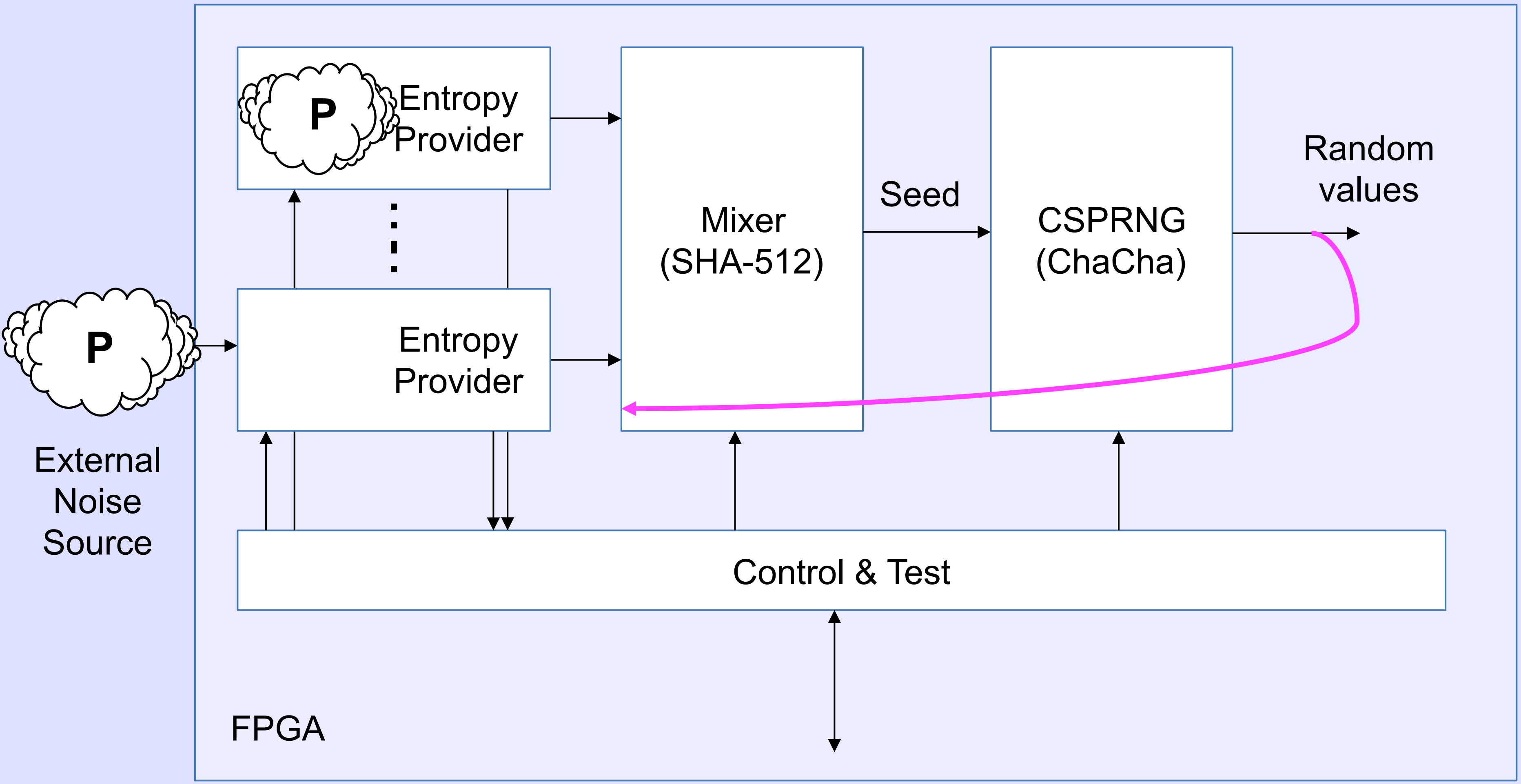


Sketch of TRNG chain
FPGA
Test read and write of entropy data
Internal entropy source
Health check
External entropy source
Health check
External entropy source
Health check
Collector
Conditioning/ Whitening
CSPRNG (CTR-DRBG)

The TRNG Architecture



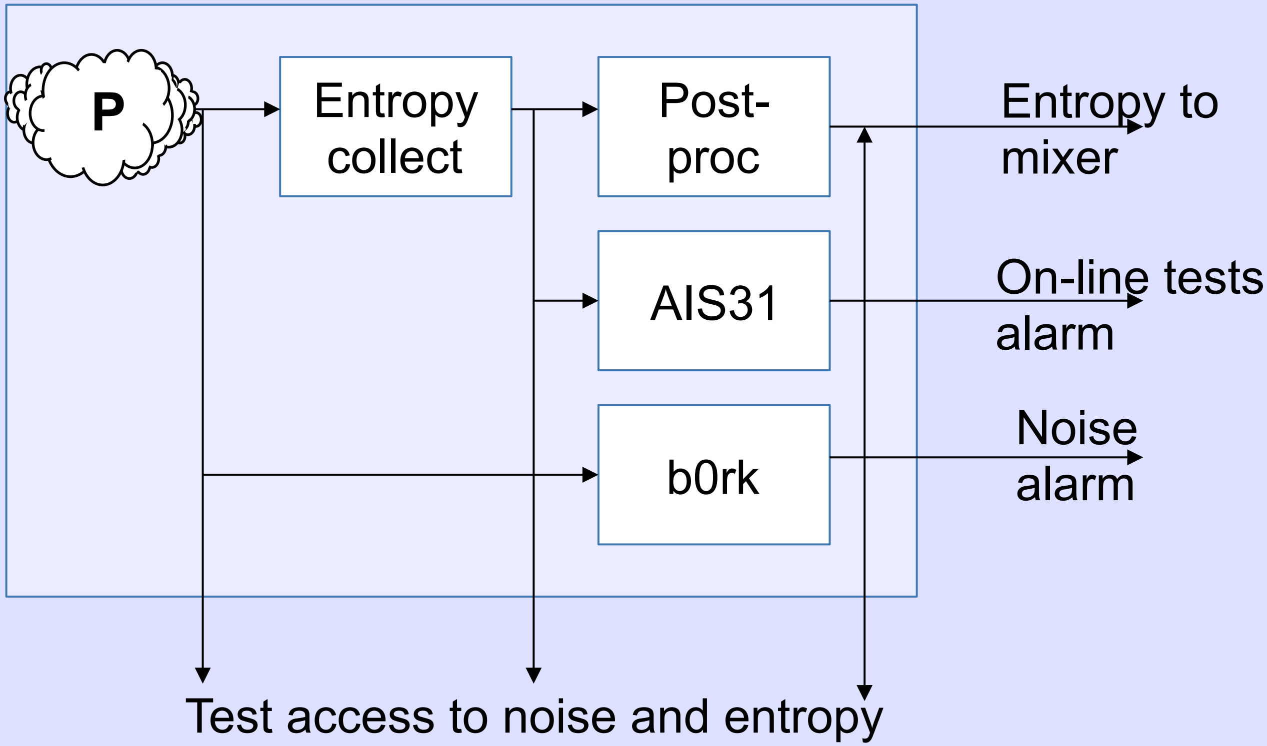
Or Maybe


P
Entropy Provider
P
Entropy Provider
External Noise Source
Mixer (SHA-512)
Seed
CSPRNG (ChaCha)
Random values
Control & Test
FPGA

Test and Observability

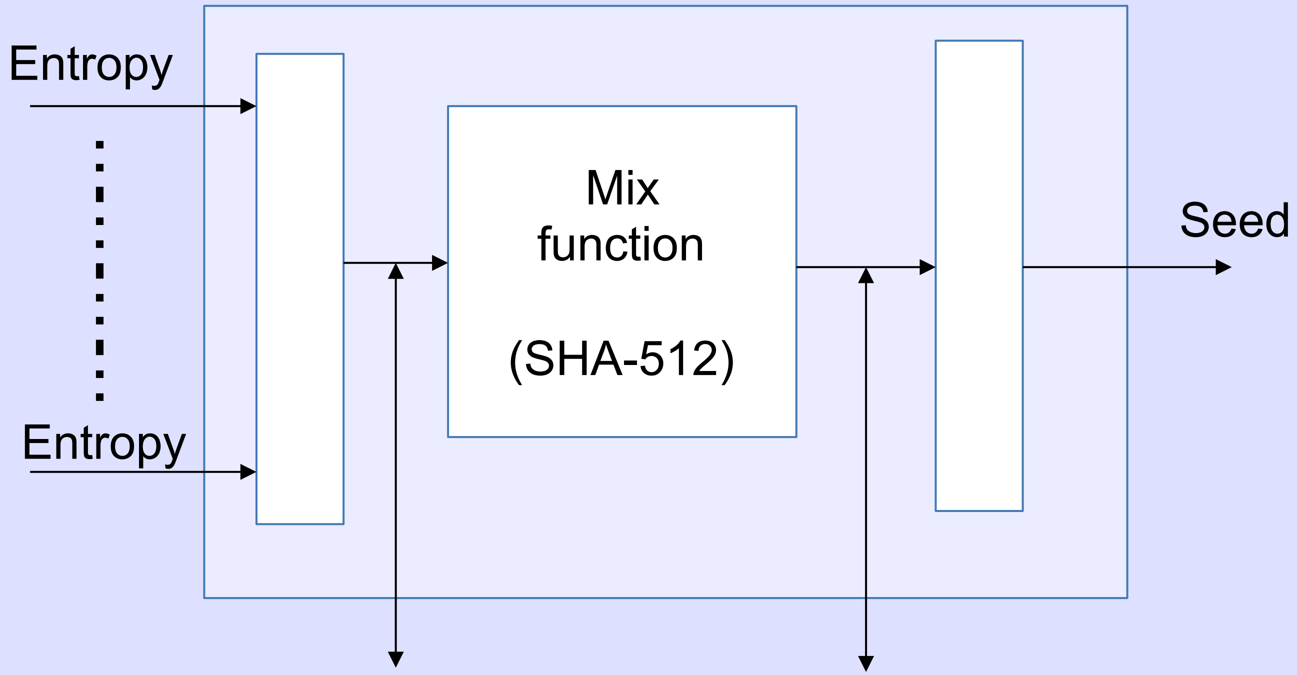
- Two modes
  - Production Mode (PM) and Test Mode (TM)
- Observability of entropy sources in PM
- Continuous on-line testing in PM
- Injection in stages and complete chain in TM
- Generation of a small number of values in TM
- Allows test of all digital functionality including continuous tests.
- Full restart when going between TM and PM

Observability & Test of Entropy Sources



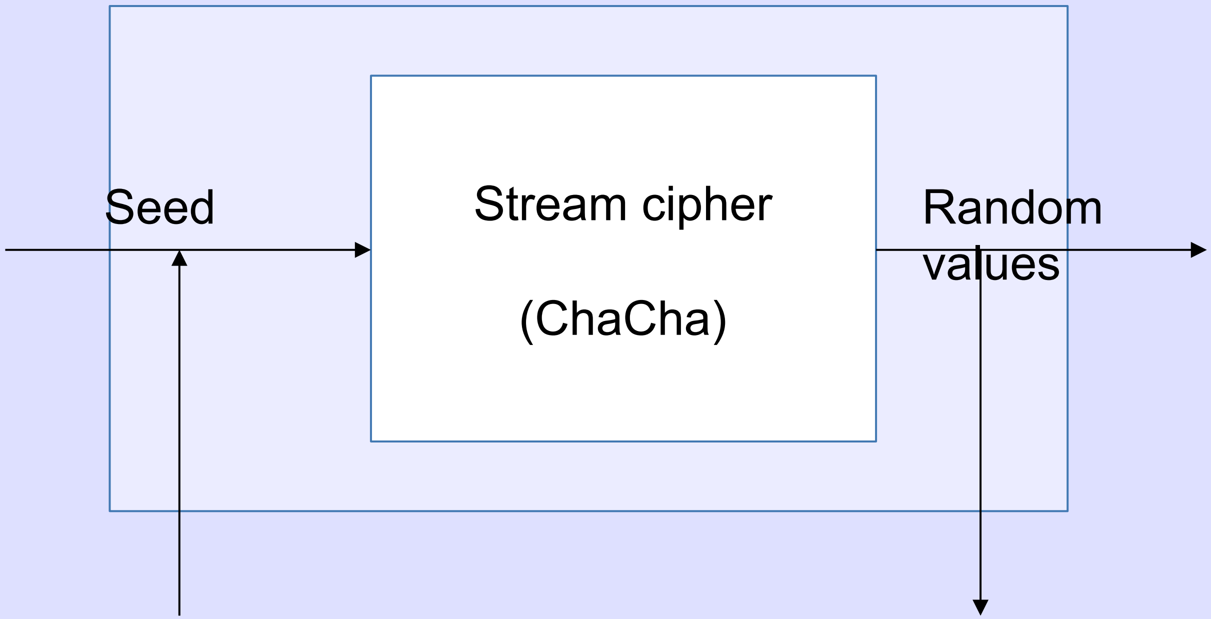
- Extract for off-line comprehensive testing
- Inject for functional testing in test mode

Observability & Test of Mixer



Inject for functional testing in test mode

Observability & Test of CSPRNG



Inject for functional testing in test mode

Some of the Fears

- ToolChain Poisoning
- Device Poisoning
- Side-Channel Attacks
- How can you tell if your vendor actually implemented CrypTech, and correctly?

The Tool Chain

- When my laptop's fan goes on, I think it is the NSA, GCHQ, Israelis, Chinese, … are 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!

Critical Tool-Chain

- C compilers audited and built using DCC
- Audited kernel, libc, …
- Audited whole darn UNIX or Linux
- Audited Verilog compiler
- Audited FPGA download tools
- Audited test tools
- Trojan prevention & detection

HDL / Verilog

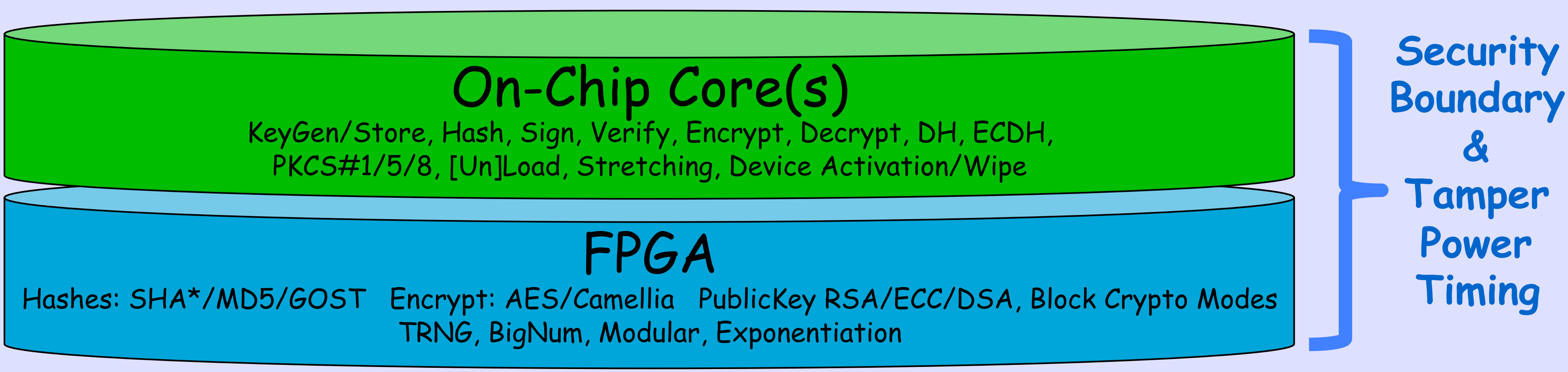
- But FPGAs/ASICs are programmed in a Hardware Definition Language, Verilog
- It is very hard to get an open Verilog compiler
- Verilog can not compile itself, so DDC is not applicable here, just a DCC C compiler
- We are working on methods of gaining trust in the FPGA tool chain

Side Channel & Tampering

- Exponentiation circuit **timing leaks** are exploitable remotely
- **Power leakage** is exploitable locally
- **Physical attack** detection critical
- Wipe key store if tampering detected
- Side-Channel attacks are the subject of entire conferences

Potting Boundary

- The FPGA/ASIC and accompanying Core(s) (ARM, whatever) are within the physically protected boundary of the chip carrier potting.
- On-board battery/capacitor to buy the time to wipe all data if unplugged from power
- 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.

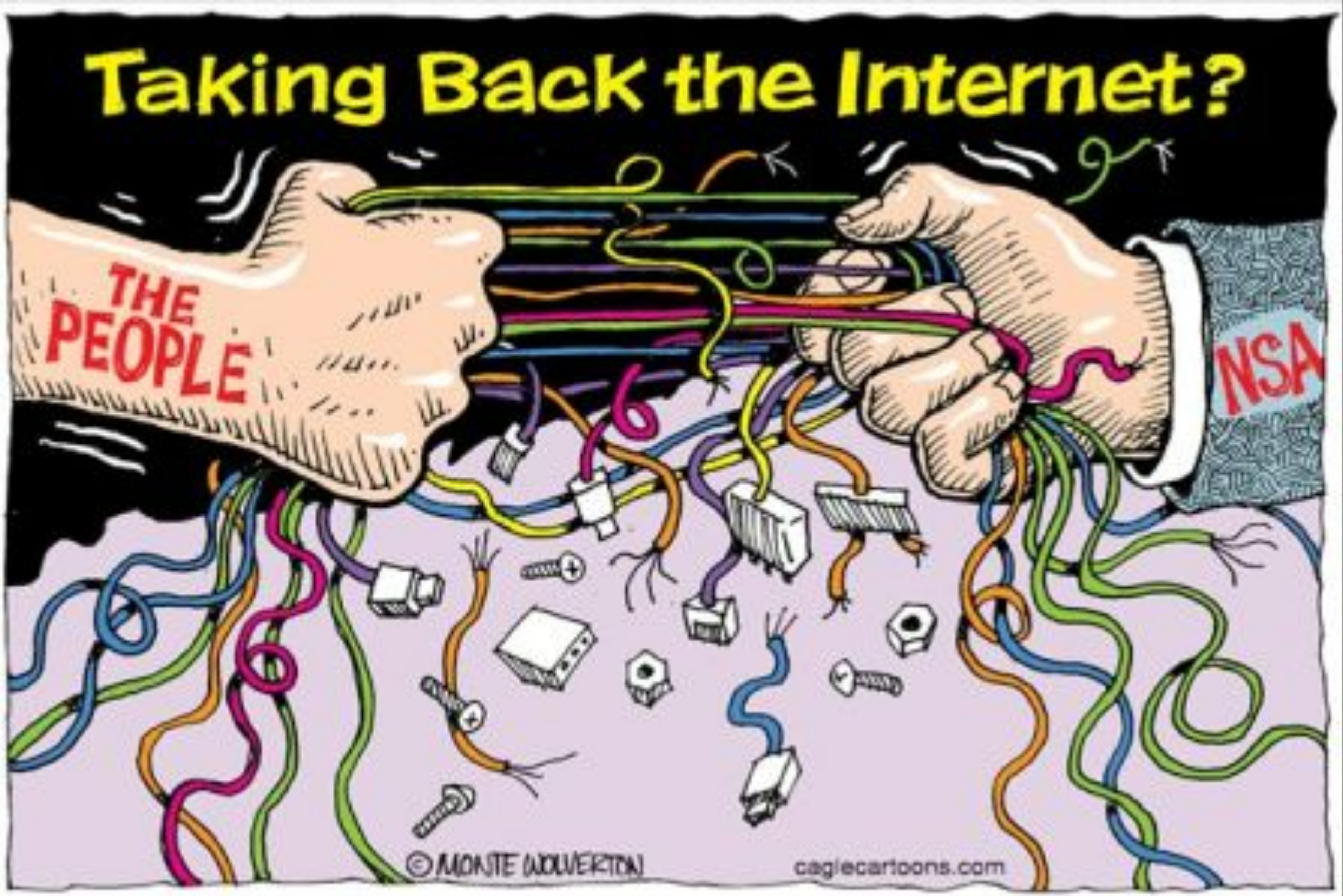


Some Phases

- 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.

A Few Related Projects

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


Taking Back the Internet?
THE PEOPLE
NSA
© MONTE WOLVERTON
caglecartoons.com

https://cryptech.is/