CrypTech

Building a More Assured Hardware Security Module

Randy Bush <randy@psg.com>
HSMs Are Used For

- Principally, Lock-box for Private Keys for
  - DNSsec
  - RPKI
  - PGP
  - Tor
  - Corporate Authentication

- 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 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?
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 reference design 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
Funding (so far) From

A Few Private Donations

Your Logo Goes Here
Layer Cake Model

Applications
DNSSEC, RPKI, PGP, VPN, OTR, random, TCP/AO, ...

Off-Chip Support Code
X.509/PGP/... Packaging, PKCS#7/10/11/15, Backup

On-Chip Core(s)
KeyGen/Store, Hash, Sign, Verify, Encrypt, Decrypt, DH, ECDH, PKCS#1/5/8, [Un]Load, Stretching, Device Activation/Wipe

FPGA (ASIC)
Hashes: SHA*/MD5/GOST  Encrypt: AES/Camellia  PublicKey RSA/ECC/DSA, Block Crypto Modes TRNG, BigNum, Modular, Exponentiation
A Prototyping Board
Novena Spartan 'Laptop'

1. MX6 CPU (1.2 GHz quad Cortex A9 capable)
2. 8x Analog inputs (up to 12 bit, 200ksp/s via FPGA)
3. 8x digital I/O (via FPGA)
4. Spartan 6 FPGA CSG324 pinout
5. Supplemental digital I/O port (via CPU)
6. DDR3 SO-DIMM
7. PMIC
8. 8x PWM headers (ESC/servo compatible pinout) (via FPGA)
9. nPCIe
10. UIM (bottom side) for mobile data cards
11. USB wifi module header (AW-NU137)
12. 2x internal USB ports (for KB & mouse)
13. Internal amplified speaker drivers (1.1W into 8 ohms)
14. USB 2.0 ports (high-current capable)
15. MicroSD boot FLASH
16. SD card socket (bottom side)
17. USB OTG
18. HDMI
19. 1 Gbit ethernet
20. 100 Mbit ethernet
21. 3-axis accelerometer
22. Battery board interface
23. Utility EEPROMs (crashlogs, etc.)
24. RTC backup supercap (optional battery)
25. LVDS LCD port: dual-channel, EIA-644A plus USB for camera & LED backlight (up to QXGA res)
26. Direct drive for resistive touchscreen
digital microphone
27. Direct power (for configurations without battery)
Entropy with Pi Pin-Out
Entropy on Novena
The TRNG Architecture

Entropy Provider

Mixer (SHA-512)

CSPRNG (ChaCha)

Control & Test

FPGA

External Noise Source

Random values
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

Seed → Stream cipher (ChaCha) → Random values

Inject for functional testing in test mode
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.

---

**On-Chip Core(s)**
- KeyGen/Store, Hash, Sign, Verify, Encrypt, Decrypt, DH, ECDH, PKCS#1/5/8, [Un]Load, Stretching, Device Activation/Wipe

**FPGA**
- Hashes: SHA*/MD5/GOST
- Encrypt: AES/Camellia
- PublicKey: RSA/ECC/DSA, Block Crypto Modes
- TRNG, BigNum, Modular, Exponentiation

---

**Security Boundary & Tamper Power Timing**
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.
Taking Back the Internet?

https://cryptech.is/