Cryptographic Hardware Accelerators

A Cryptographic Hardware Accelerator can be

- integrated into the SoC as a separate processor, as special purpose CPU (aka Core).
- integrated in a Coprocessor [https://en.wikipedia.org/wiki/Coprocessor] on the circuit board
- contained on a Chip on an extension circuit board, this can be connected to the mainboard via some BUS, e.g. PCI
- an ISA extension [https://en.wikipedia.org/wiki/Template:Multimedia_extensions] like e.g. AES instruction set [https://en.wikipedia.org/wiki/AES instruction set] and thus integral part of the CPU (in that case a kernel driver in not needed)

The purpose is to load off the very computing intensive tasks of encryption/decryption and compression/decompression.
As can be seen in this AES instruction set [https://en.wikipedia.org/wiki/AES instruction set] article, the acceleration is usually achieved by doing certain arithmetic calculation in hardware.

When the acceleration is not in the instruction set of the CPU, it is supported via a kernel driver (/dev/crypto). There are two drivers offering /dev/crypto in OpenWRT:

- Cryptodev-linux [https://github.com/openwrt/packages/tree/master/utils/cryptodev-linux] kernel module, which utilizes the Linux kernel crypto drivers
- OCF (OpenBSD Crypto Framework), which utilizes the OpenBSD crypto drivers

Both ways result to a /dev/crypto device which can be used by userspace crypto applications (e.g., the ones that utilize openssl or gnutls).

Performance

Depending on which arithmetic calculations exactly are being done in the specific hardware, the results differ widely. You should not concern yourself with theoretical bla,bla but find out how a certain implementation performs in the task you want to do with it! You could want to

- you could attach a USB drive to your device and mount a local filesystem like ext3 from it. Then you want to read from and write to this filesystem from the Internet over a secured protocol. Let's use `sshfs`. You would set up a sshfs.server on your device and a sshfs.client on the other end. Now how fast can you read/write to this with and without Cryptographic Hardware Accelerators. If the other end, the client, is a "fully grown PC" with a 2GHz CPU, it will probably perform fast enough to use the entire bandwidth of your Internet connection. If the server side is some embedded device, with let's say some 400MHz MIPS CPU, it could benefit highly from some integrated (and supported!) acceleration. You probably want enough performance, that you can use your entire bandwidth. Well, now go and find some benchmark showing you precisely the difference with enabled/disabled acceleration. Because you will not be able to extrapolate this information from specifications you find on this page or on the web.
- you could want to run an OpenVPN or an OpenConnect server on your router/embedded device, instead of using WEP/WPA/WPA2. There will be no reading from/writing to a USB device. Find benchmarks that show you exactly the performance for this purpose. You won't be able to extrapolate this information from other benchmarks.
- think of other practical uses, and find specific benchmarks.

Enabling /dev/crypto

Run `make menuconfig` and select

With cryptodev-linux

- kmod-crypto-core: m
  - kmod-cryptodev: m

With OCF

This must not be combined with cryptodev-linux.

Kernel modules → Cryptographic API modules

- kmod-crypto-core: m
  - kmod-crypto-ocf: m

Utilities

- ocf-crypto-headers: m

Adding /dev/crypto support to crypto libraries

Libraries → SSL

- libopenssl: m
  - Crypto acceleration support: y
- libgnutls: m
  - enable /dev/crypto support: y

Note that there are some known issues with openssl's /dev/crypto support [http://rt.openssl.org/Ticket/Display.html?id=2770&user=guest&pass=guest].

Enabling specific hardware driver

Soekris vpn1411

- http://www.soekris.com/vpn1401.htm [http://www.soekris.com/vpn1401.htm]

Run `make menuconfig` and select

Kernel modules → Cryptographic API modules

- kmod-crypto-core: m
  - kmod-crypto-aes: m
  - kmod-crypto-des: m
  - kmod-crypto-hw-hifn-795x: m

Marvell CESA

Cryptographic Engine and Security Acceleration

- PDF download [http://www.google.com/search?sclient=psy&hl=en&source=hp&q=site%3Awww.marvell.com+cesa&btnG=Search]
- Seagate Dockstar
- 2.6.32: AES commit [http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=commitdiff;h=85a7f0ac5370901916a21935e1fafbe397b70f80]
- 2.6.35: SHA1 and HMAC-SHA1 commit [http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=commitdiff;h=750052dd2400cd09e0864d75b63c2c0bf605056f]
- r22877 [https://dev.openwrt.org/changeset/22877]: [kirkwood] Add kernel package for the mv_cesa crypto module
- r23145 [https://dev.openwrt.org/changeset/23145]: [kirkwood] Fix mv_cesa module dependencies and .ko file location Thanks KanjiMonster & Memphiz
- r23229 [https://dev.openwrt.org/changeset/23229]: [packages/kernel] Make mv_cesa crypto module available on Orion as well.
- r23383 [https://dev.openwrt.org/changeset/23383]: [package] kernel: underscores in package names are bad, rename kmod-crypto-mv_cesa to kmod-crypto-mv-cesa
- r26406 [https://dev.openwrt.org/changeset/26406]: kernel: add a missing dependency for the mv_cesa crypto driver
- r26407 [https://dev.openwrt.org/changeset/26407]: kernel: mv_cesa depends on CRYPTO_BLKCIPHER2 and CRYPTO_HASH2
- r26413 [https://dev.openwrt.org/changeset/26413]: kernel: remove double definition of depends in crypto-mv-cesa and make it look like the other entries. Thank you Maarten

Geode AES engine

- using.the.geode.s.aes.engine

VIA padlock

- VIA Padlock security engine [http://www.via.com.tw/en/initiatives/padlock/hardware.jsp]
- Padlock_(disambiguation) [https://en.wikipedia.org/wiki/Padlock_(disambiguation)]
- http://fijam.eu.org/blog/?p=198 [http://fijam.eu.org/blog/?p=198]

Historical

Note: If you want to learn about the current situation, you should search the Internet or maybe ask in the forum. This is outdated. Especially if you want to know, how fast a copy from a mounted filesystem (say ext3 over USB) over the scp is, you should specifically search for such benchmarks. Some models of the BCM47xx/53xx family support hardware accelerated encryption for IPSec (AES, DES, 3DES), simple hash calculations (MD5, SHA1) and TLS/SSL+HMAC processing. Not all devices have a hw crypto supporting chip. At least Asus WL500GD/X, Netgear WGT634U and Asus WL700gE do have hw crypto. However, testing of a WGT634U indicates that a pin under the BCM5365 was not pulled low to enable strong bulk cryptography, limiting the functionality to single DES.

- How did you find that out?
  - Do you get an interrupt when sending a crypto job to the chip and limiting the request to DES only?)

The specification states the hardware is able to support 75Mbps (9,4MB/s) of encrypted throughput. Without hardware acceleration using the blowfish encryption throughput is only ~0,4MB/s. Benchmark results that show the difference between software and hardware accelerated encryption/decryption can be found here [http://www.danm.de/files/src/bcm5365p/bench/]. Due to the overhead of hardware/DMA transfers and buffer copies between kernel/user space it gives only a good return for packet sizes greater than 256 bytes. This size can be reduced for IPSec, because network hardware uses DMA and there is no need to copy the (encrypted) data between kernel and user space. The hardware specification needed for programming the crypto API of the bcm5365P (Broadcom 5365P) can be found here [http://voodoowarez.com/bcm5365p.pdf].

- The crypto chip is accessible through the SSB bus (Sonics Silicon Backplane). A Linux driver for SSB is available in OpenWRT's kernel >= 2.6.23 (Kamikaze)
- An example about how to communicate with the crypto chip can be found here [http://www.danm.de/files/src/bcm5365p/] (file b5365ips.tar.bz2).
- An OCF Linux driver that works with the ASUS WL500gP can be found in Trunk (SVN) or here [http://www.danm.de/files/src/bcm5365p/] and is called **ubsec_ssb**. Only OCF-enabled applications can be accelerated. That means, if you want e.g. an accelerated OpenSSH you have to manually enable cryptodev in OpenSSL. The driver is still considered experimental.
- Links to mailing-list posts with references to more recent and working version of Linux driver for Broadcom crypto chips here [http://marc.theaimsgroup.com/?l=openssl-dev&m=110915540208913&w=2] and here [http://www.mail-archive.com/openssl-dev@openssl.org/msg18804.html].
- Sun Crypto Accelerator 500 and 1000 (X6762A) cards are based on BCM5821. Might be worth checking Solaris references as well. Here [http://src.opensolaris.org/source/xref/crypto/quantis/usr/src/uts/common/crypto/io/] is OpenSolaris driver for Broadcom crypto chips.
- Asus WL-700gE sources come with patched FreeSwan to utilize ubsec.
- Closed-source binary included in Asus Wl-700gE sources do support AES based on headers.
- There's a Linux port [http://ocf-linux.sourceforge.net/] of the OpenBSD Cryptographic Framework (OCF) but the ubsec driver (Broadcom 58xx PCI cards) is not ported yet. If you compile OCF with the /dev/crypto device driver, userspace applications and libraries such as OpenSSL can be accelerated. There are patches for Openswan as well.
- Discussion [http://forum.openwrt.org/viewtopic.php?id=5032] about hardware accelerated crypto.
- Various versions of old BCM5820 driver sources [http://sukkamehulinko.romikselle.com/openwrt/bcm5820/].
- BCM5801/BCM5805/BCM5820 Security Processor Software Reference Library http://www.broadcom.com/products/access_request.php?category_id=0&id=7&filename=5801-5805-5820-SRL101-R.pdf [http://www.broadcom.com/products/access_request.php?category_id=0&id=7&filename=5801-5805-5820-SRL101-R.pdf]
- Cisco PIX VAC+ Encryption module is 64-bit PCI card based on Broadcom BCM5823. Another similar card is Checkpoint VPN-1 Accelerator Card II, III and IV from Silicom [http://www.silicom.co.il/].

| SoC / CPU | Accelerated Methods | Datasheet |

| BCM94704AGR | WEP 128, AES OCB AES CCM | 94704AGR-PB00-R.pdf [http://www.broadcom.com/collateral/pb/94704AGR-PB00-R.pdf] |

| BCM?4704P | WEP 128, AES OCB, AES CCM, VPN | 94704AGR-PB00-R.pdf [http://www.broadcom.com/collateral/pb/94704AGR-PB00-R.pdf] |

| BCM5365 | AES (up to 256-bit CTR and CBC modes), DES, 3DES (CBC), HMAC-SHA1, HMAC-MD5, SHA1 and MD5. IPSec encryption and single pass authentication. | 5365_5365P-PB01-R.pdf [http://www.broadcom.com/collateral/pb/5365_5365P-PB01-R.pdf] |

| BCM5365P | AES (up to 256-bit CTR and CBC modes), DES, 3DES (CBC), HMAC-SHA1, HMAC-MD5, SHA1 and MD5. IPSec encryption and single pass authentication. | 5365_5365P-PB01-R.pdf [http://www.broadcom.com/collateral/pb/5365_5365P-PB01-R.pdf] |

Tags

crypto