NAME

aesni - driver for the AES and SHA accelerator on x86 CPUs

SYNOPSIS

To compile this driver into the kernel, place the following lines in your kernel configuration file:

```
device crypto
device cryptodev
device aesni
```

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

```
aesni_load="YES"
```

DESCRIPTION

Starting with Intel Westmere and AMD Bulldozer, some x86 processors implement a new set of instructions called AESNI. The set of six instructions accelerates the calculation of the key schedule for key lengths of 128, 192, and 256 of the Advanced Encryption Standard (AES) symmetric cipher, and provides a hardware implementation of the regular and the last encryption and decryption rounds.

The processor capability is reported as AESNI in the Features2 line at boot.

Starting with the Intel Goldmont and AMD Ryzen microarchitectures, some x86 processors implement a new set of SHA instructions. The set of seven instructions accelerates the calculation of SHA1 and SHA256 hashes.

The processor capability is reported as SHA in the Structured Extended Features line at boot.

The **aesni** driver does not attach on systems that lack both CPU capabilities. On systems that support only one of AESNI or SHA extensions, the driver will attach and support that one function.

The **aesni** driver registers itself to accelerate AES and SHA operations for crypto(4). Besides speed, the advantage of using the **aesni** driver is that the AESNI operation is data-independent, thus eliminating some attack vectors based on measuring cache use and timings typically present in table-driven implementations.

SEE ALSO

crypt(3), crypto(4), intro(4), ipsec(4), padlock(4), random(4), crypto(7), crypto(9)

HISTORY

The **aesni** driver first appeared in FreeBSD 9.0. SHA support was added in FreeBSD 12.0.

AUTHORS

The **aesni** driver was written by Konstantin Belousov <*kib@FreeBSD.org*> and Conrad Meyer <*cem@FreeBSD.org*>. The key schedule calculation code was adopted from the sample provided by Intel and used in the analogous OpenBSD driver. The hash step intrinsics implementations were supplied by Intel.