Crypto Acceleration on FreeBSD

Philip Paeps
philip@FreeBSD.org

The FreeBSD Project

meetBSD 2008 — Mountain View, CA, USA
16 November 2008
Outline

1. Cryptography in FreeBSD
   - Userland Applications
   - Kernel Subsystems
   - Drawbacks and Problems

2. The opencrypto Framework
   - History and Purpose
   - Kernel and Userland Interface
   - Hardware Acceleration
   - Use in Applications

3. Performance Measurements
   - glxsb(4) on a Soekris
   - hifn(4) on a Fast AMD64

4. Future Directions
Outline

1. Cryptography in FreeBSD
   - Userland Applications
   - Kernel Subsystems
   - Drawbacks and Problems

2. The opencrypto Framework
   - History and Purpose
   - Kernel and Userland Interface
   - Hardware Acceleration
   - Use in Applications

3. Performance Measurements
   - glxsb(4) on a Soekris
   - hifn(4) on a Fast AMD64

4. Future Directions
Userland Applications

- OpenSSL in the base system
- GnuTLS and others in ports
- Homegrown implementations
Kernel Subsystems

- IPSEC
- Block Devices
  - GBDE
  - GELI
- ZFS
Drawbacks and Problems

- Many CPU-intensive operations
- Limited parallelism
- Lots of scary code-duplication
Outline

1 Cryptography in FreeBSD
   - Userland Applications
   - Kernel Subsystems
   - Drawbacks and Problems

2 The opencrypto Framework
   - History and Purpose
   - Kernel and Userland Interface
   - Hardware Acceleration
   - Use in Applications

3 Performance Measurements
   - glxsb(4) on a Soekris
   - hifn(4) on a Fast AMD64

4 Future Directions
History and Purpose

- Ported from OpenBSD in 2002
- Consistent software and hardware interface
- Originally particularly intended for IPSEC
- Fairly modular and extendable design
Kernel and Userland Interface

- Asynchronous session-oriented interface
- Kernel systems use `<opencrypto/cryptodev.h>`
- Userland uses `ioctl` interface on `/dev/crypto`
Hardware Acceleration

- Device drivers register callbacks with framework
- Support one or more algorithms
- Limited support for flow-control
- Caveat: acceleration can sometimes slow things down!
Supported Devices

- glxsb(4) — AMD Geode
- hifn(4) — Hifn
- padlock(4) — VIA Padlock
- safe(4) — SafeNet
- ubsec(4) — Broadcom/Bluesteel
Use in Applications

- Most kernel subsystems use `crypto(9)`
  - ... Notable exception: GBDE
- OpenSSL `cryptodev` ENGINE
  - Not used automatically
  - Fairly easy to use
  - Work in progress (patches)
Outline

1 Cryptography in FreeBSD
   - Userland Applications
   - Kernel Subsystems
   - Drawbacks and Problems

2 The opencrypto Framework
   - History and Purpose
   - Kernel and Userland Interface
   - Hardware Acceleration
   - Use in Applications

3 Performance Measurements
   - glxsb(4) on a Soekris
   - hifn(4) on a Fast AMD64

4 Future Directions
Simple openssl speed Test

```
% openssl speed -evp aes-128-cbc
[...]
The 'numbers' are in 1000s of bytes per second processed.
type          16 bytes  64 bytes  256 bytes  1024 bytes  8192 bytes
aes-128-cbc   4936.66k  5476.96k  5648.11k  5693.51k  5701.87k

% openssl speed -evp aes-128-cbc -engine cryptodev
genre "cryptodev" set.
The 'numbers' are in 1000s of bytes per second processed.
[...]
type          16 bytes  64 bytes  256 bytes  1024 bytes  8192 bytes
aes-128-cbc   5850.39k  23944.46k  118509.23k  416638.93k  3879235.74k
```
Encrypting a Large File

```
% dd if=/dev/random of=cryptme bs=1M count=350

% /usr/bin/time -h openssl enc -e -aes-128-cbc [...] 
1m11.57s real 1m7.69s user 3.34s sys

% /usr/bin/time -h openssl enc -e -aes-128-cbc [...] -engine cryptodev 
18.41s real 1.51s user 16.75s sys
```
Simple openssl speed Test

% openssl speed -evp aes-128-cbc
[...]
The ’numbers’ are in 1000s of bytes per second processed.
type 16 bytes 64 bytes 256 bytes 1024 bytes 8192 bytes
aes-128-cbc 50014.57k 55329.90k 57058.55k 57505.75k 57578.37k

% openssl speed -evp aes-128-cbc -engine cryptodev
engine "cryptodev" set.
The ’numbers’ are in 1000s of bytes per second processed.
[...]
type 16 bytes 64 bytes 256 bytes 1024 bytes 8192 bytes
aes-128-cbc 367.92k 1525.02k 5146.43k 11861.38k 20413.72k
Encrypting a Large File

% dd if=/dev/random of=cryptme bs=1M count=350

% /usr/bin/time -h openssl enc -e -aes-128-cbc [...] 8.47s real 7.44s user 1.01s sys

% /usr/bin/time -h openssl enc -e -aes-128-cbc [...] -engine cryptodev 21.33s real 0.34s user 2.82s sys
Outline

1 Cryptography in FreeBSD
   ■ Userland Applications
   ■ Kernel Subsystems
   ■ Drawbacks and Problems

2 The opencrypto Framework
   ■ History and Purpose
   ■ Kernel and Userland Interface
   ■ Hardware Acceleration
   ■ Use in Applications

3 Performance Measurements
   ■ glxsb(4) on a Soekris
   ■ hifn(4) on a Fast AMD64

4 Future Directions

Philip Paeps
Crypto Acceleration on FreeBSD
Future Directions

- Reduce code-duplication in acceleration drivers
- Enable cryptodev ENGINE by default in OpenSSL
Questions? Comments?