Crypto API overview

Crypto API overview

Stable: 14.04.2020 - 11:45 / Revision: 31.03.2020 - 14:35

The Crypto API is a cryptography framework in the Linux® kernel. It is dedicated to the parts of the kernel that deal with cryptography, such as IPsec and dm-crypt.

Contents

1 Framework purpose

The purpose of this article is to introduce the Crypto API framework:

- general information
- main component/stakeholders
- how to use the Crypto API
- use cases

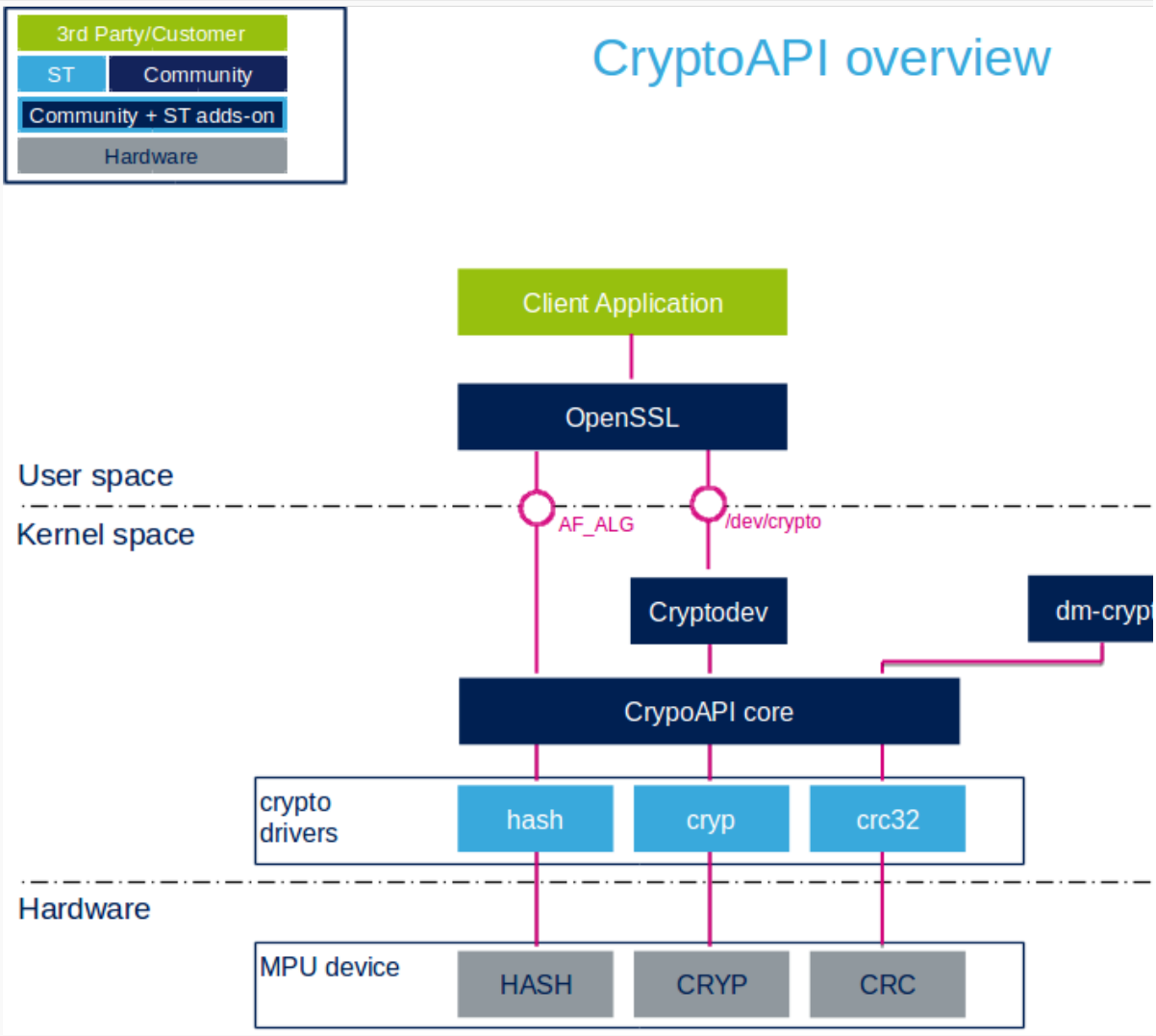
The Crypto API framework mainly includes all popular **hash** and **block ciphers** (encryption) functions.

A **hash** is a string or number generated from a text string. The length of the resulting string or number is fixed and widely varies with small variations of the input. The best hashing algorithms are designed so that it is impossible to turn a hash back into its original string. Hashing is particularly useful to compare a value with a stored value. However it cannot store its plain representation for security reasons. This makes hashing an ideal solution to store passwords.

Encryption turns data into a series of unreadable characters which length is not fixed. The encrypted strings can reversed back into their original decrypted form if the right key is not provided. Encrypting a confidential file is a good way to prevent anyone from accessing its content.

Drivers for CRYP (block cipher), HASH (hash) and CRC (cyclic redundancy check) are integrated within the Crypto API kernel service.

2 System overview



Crypto API

2.1 Description of the components

OpenSSL and dm-crypt are not part of the Crypto API framework but they are typical users of the Crypto API services.

From User space to hardware

- **OpenSSL** (User space)

OpenSSL[1] is a software library supporting the TLS and SSL protocols as well as cryptographic functions. Openssl is available in OpenSTLinux distribution.

- **dm-crypt** (Kernel space)

dm-crypt[2] is a kernel disk encryption subsystem. It is natively available in the standard Linux kernel.

- **Cryptodev** (Kernel space)

Cryptodev[3] is a device driver which provides a general interface for userland applications. Although it is not part of the standard Linux kernel, it is available in OpenSTLinux distribution.

- **CryptoAPI core** (Kernel space)

This layer represents the standard Linux kernel cryptographic framework.

- **hash**, **cryp** and **crc32** (Kernel space)

These are the cryptographic Linux drivers handling the HW blocks.

- **HASH**, **CRYP** and **CRC** (Hardware)

These HW blocks handle hash, ciphering, and CRC checksum.

2.2 API description

The Crypto API is documented in the Linux Kernel Crypto API section of the Linux Kernel documentation[4]. It offers both a kernel and a userland interface:

- kernel internal interface, used in particular by dm-crypt.
- userland algorithm interface (socket) named AF_ALG[5]. OpenSSL can use this interface.

In addition to the socket user interface, a more friendly interface, the cryptodev, can be used. It offers the /dev/crypto ioctl API. It is roughly described by the cryptodev.h[6] header file. OpenSSL can be configured to use this interface as an alternative to the historical AF_ALG interface.

3 Configuration

The Crypto API is activated by default in ST deliveries. Nevertheless, if a specific configuration is required, you can use Linux Menuconfig tool: Menuconfig or how to configure kernel and select:

```
[*] Cryptographic API  --->
    [*]   Hardware crypto devices  --->
        [*]   Support for STM32 crc accelerators
        [*]   Support for STM32 hash accelerators
        [*]   Support for STM32 crypto accelerators
```

4 How to use the Crypto API framework

The Crypto API framework can be used by other kernel modules.

The Crypto API documentation provides kernel code examples[7]:

- Symmetric-key cipher operation.
- Operational state memory with SHASH.

5 Use cases

- Disk encryption

This is a typical example of Crypto API framework usage. Refer to LUKS[8] for a standard disk encryption process.

6 How to trace and debug the framework

6.1 How to monitor

The list of available ciphers is given in /proc/crypto:

```
Board $> cat /proc/crypto
```

Output part showing that an STM32 driver provides with the CRC32 cipher:

```
...
name         : crc32
driver       : stm32-crc32
module       : kernel
priority     : 200
refcnt       : 1
selftest     : passed
internal     : no
type         : shash
blocksize    : 1
digestsize   : 4
...
```

6.2 How to trace

There are no specific traces for this framework.

6.3 How to debug

There are no specific debug means for this framework.

7 Generic source code location

- CryptoAPI core
- CryptoAPI interface
- stm32 crypto drivers

8 References

- OpenSSL a software library supporting the TLS and SSL protocols as well as cryptographic functions.
- dm-crypt a kernel disk encryption subsystem
- Cryptodev a device driver which provides a general interface for userland applications
- Linux Kernel Crypto API the official crypto API kernel documentation
- Crypto API Userland interface specification of the userland API
- cryptodev.h cryptodev header file specifying the userland API
- Crypto API kernel code examples some kernel code examples using the Crypto API framework
- LUKS (Linux Unified Key Setup) a disk encryption specification

Application programming interface

Cryptographic processor

Cyclic redundancy check calculation unit

GPIO alternate function