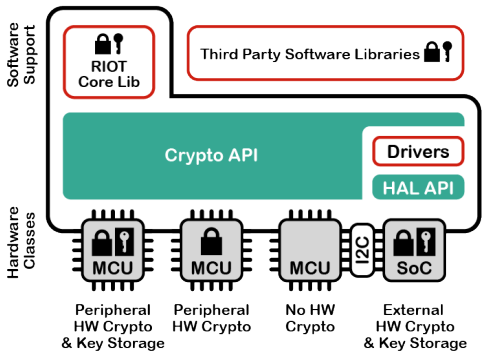
Integration of the PSA Crypto API with Configurable Hardware and Software Backends in RIOT

Lena Boeckmann

November 4, 2021

Cryptographic Backends

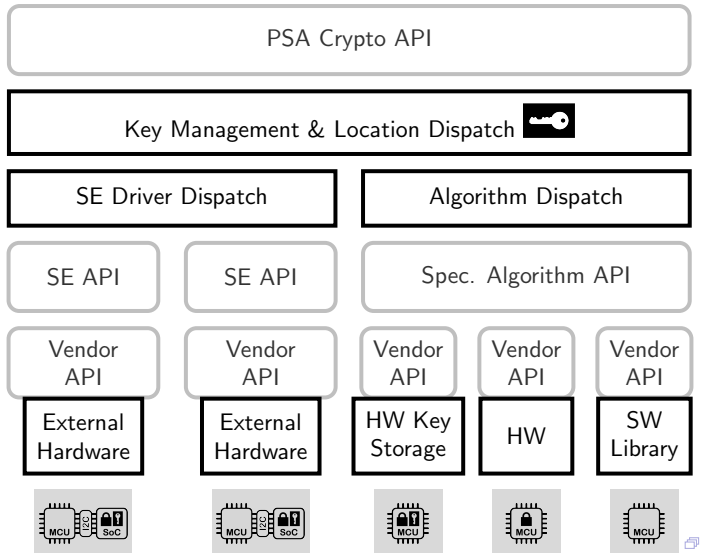


Challenges

- Platforms with varying hardware crypto capabilities need to be supported
- Both protected and unprotected key storage need to be supported
- Hardware and software backends should be exchanged transparently beneath a unified API

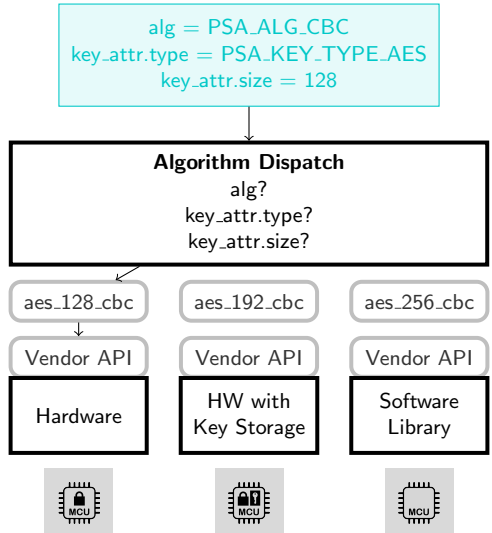
PSA Crypto Implementation Structure

Backend Selection Based On Key Location



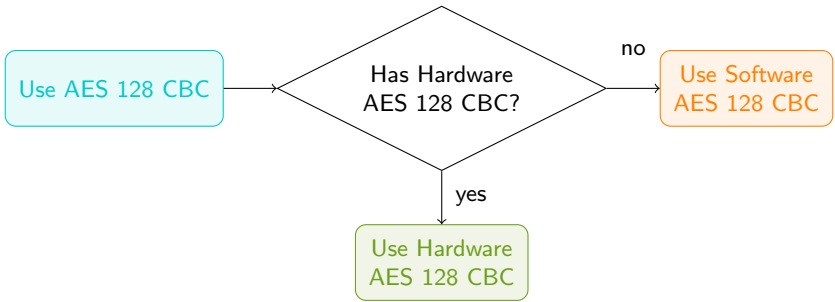
Algorithm Dispatch

Backend Selection Based On Algorithm & Key Attributes

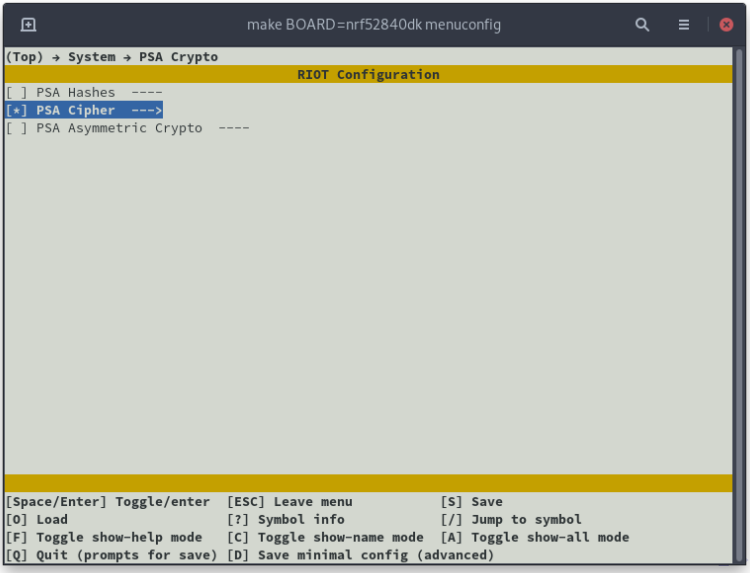


Configuring Crypto Backends with Kconfig

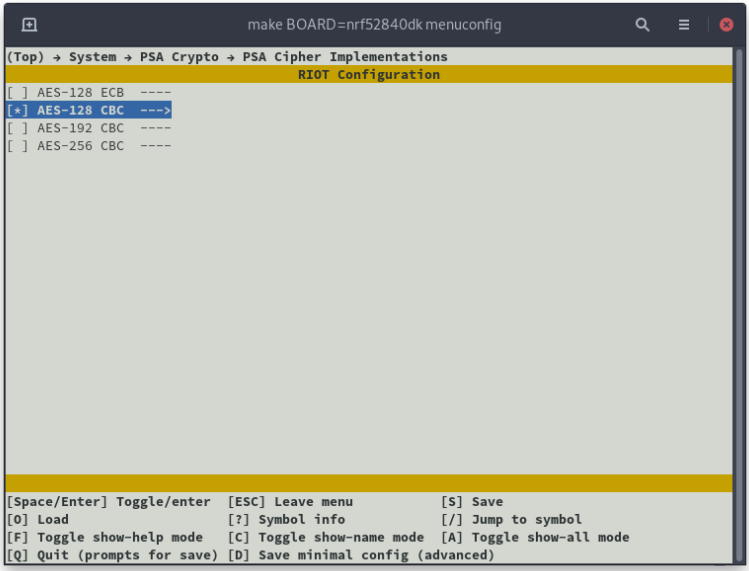
- Selection-based configuration system
- Symbols are selected in config files to enable or disable features
- Hardware is default when CPU defines symbol (e.g. HAS_HW_AES_128_CBC)
- User may change default configuration if desired



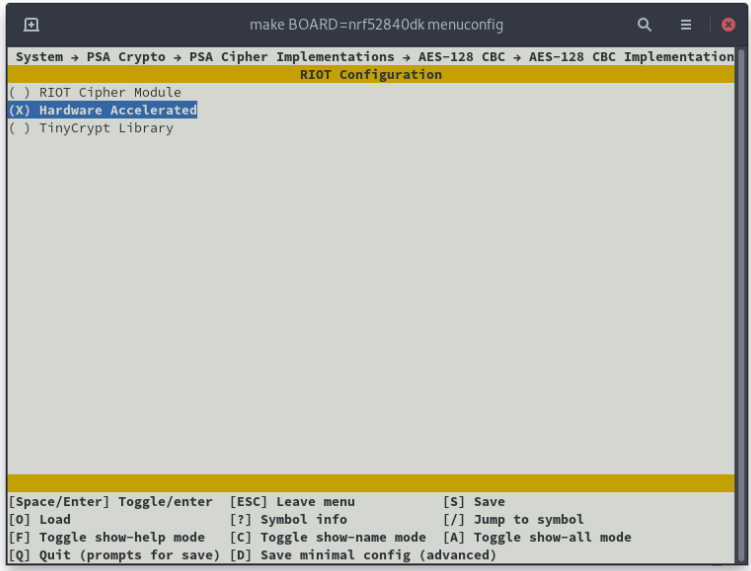
Menuconfig Walkthrough



Menuconfig Walkthrough



Menuconfig Walkthrough



Implementation Status

Component	Status	Description	Next Steps
Volatile Key Management	✔	Volatile keys in local memory and SEs can be handled	Add support for persistent keys
Cryptographic Operations	✔	Multiple Backends for Hashes and Ciphers, some elliptic curve support	Extend support for cryptographic operations
Secure Element Handling	✔	Multiple devices can be handled	Add support for other devices

What's challenging?

Fine grained configurations and dispatch based on key attributes make the implementation very complex!