Configurable Stack Size for Secure Partitions

Kevin Peng

August 4, 2022

Background

- The stack sizes for Secure Partitions are hard-coded in manifests
- Requirements to make them configurable for different use cases to better tune the size of TF-M

How to enable and use

- How to enable in a Secure Partition
  - Replace the value of the *stack_size* attribute with a configuration symbol

```
  "model": "SFN",                          13   "model": "SFN",
  "entry_init": "tfm_crypto_init",         14   "entry_init": "tfm_crypto_init",
- "stack_size": "0x1B00",                  15+  "stack_size": "CRYPTO_STACK_SIZE",
```

  - Set the default value of the configuration in *config_default.cmake*

```
                                           146+ set(CRYPTO_STACK_SIZE                "0x1B00"    CACHE STRING
                                                "The stack size of the Crypto Secure Partition")
```

- How to use
  - Just like any other existing configurations – override it in CMake files or command line

How is This Feature Implemented

Stack Generation

- The source file holds the stacks of Secure Partitions are generated by the manifest tool

Template

```
#include <stdint.h>

{% if config_impl['CONFIG_TFM_SPM_BACKEND_IPC'] == '1' or manifest.model == "IPC" %}
uint8_t {{manifest.name.lower()}}_stack[{{manifest.stack_size}}] __attribute__((aligned(8)));
{% endif %}
```

Manifest

```
  "entry_init": "tfm_crypto_init",
  "stack_size": "0x1B00",
```

Generated File

```
#include <stdint.h>

uint8_t tfm_sp_crypto_stack[0x1B00] __attribute__((aligned(8)));
```

Manifest

```
  "entry_init": "tfm_crypto_init",
  "stack_size": "CRYPTO_STACK_SIZE",
```

Generated File

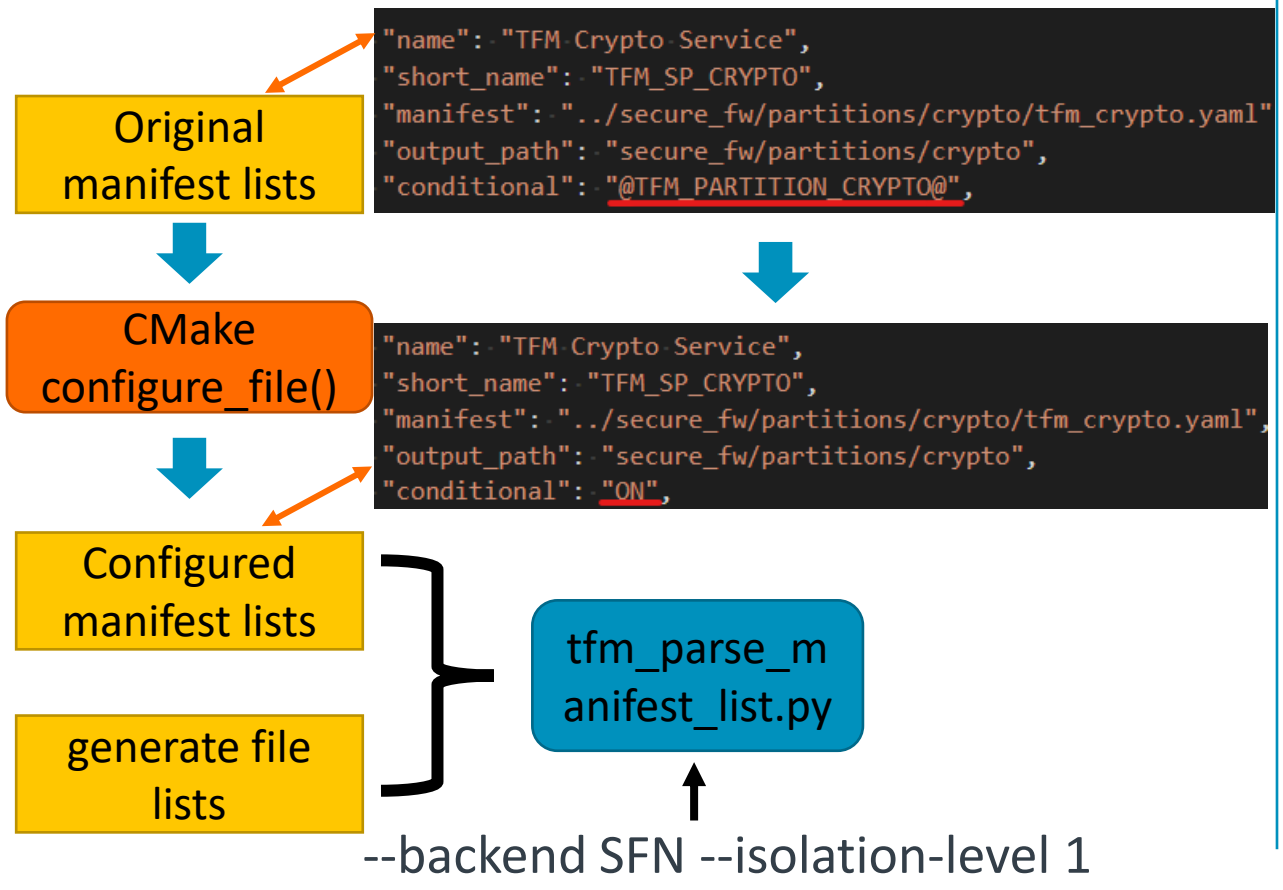
```
#include <stdint.h>

uint8_t tfm_sp_crypto_stack[CRYPTO_STACK_SIZE] __attribute__((aligned(8)));
```

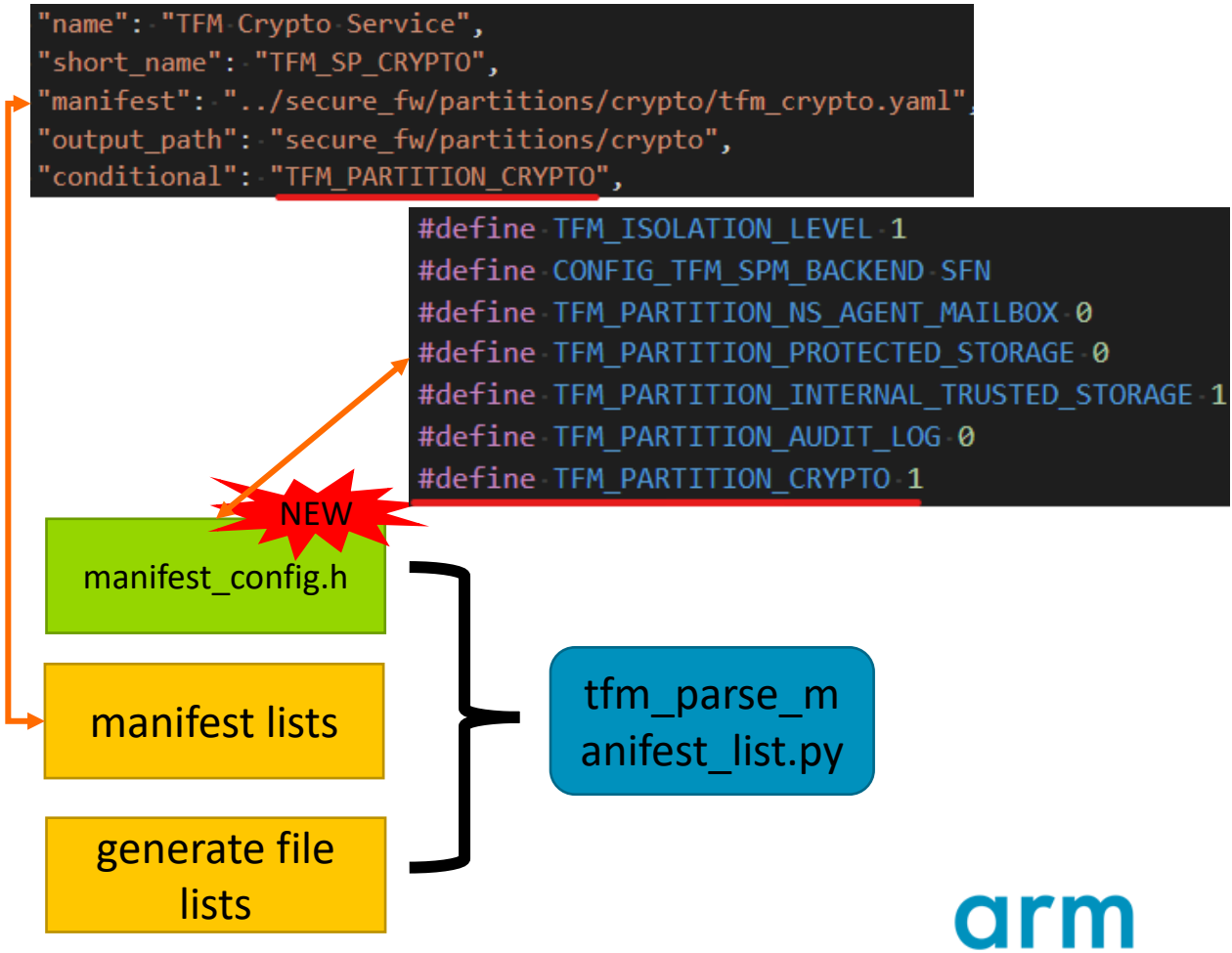
Pass Build Configurations to Manifest Tool

A recent change on the manifest tool

In the past

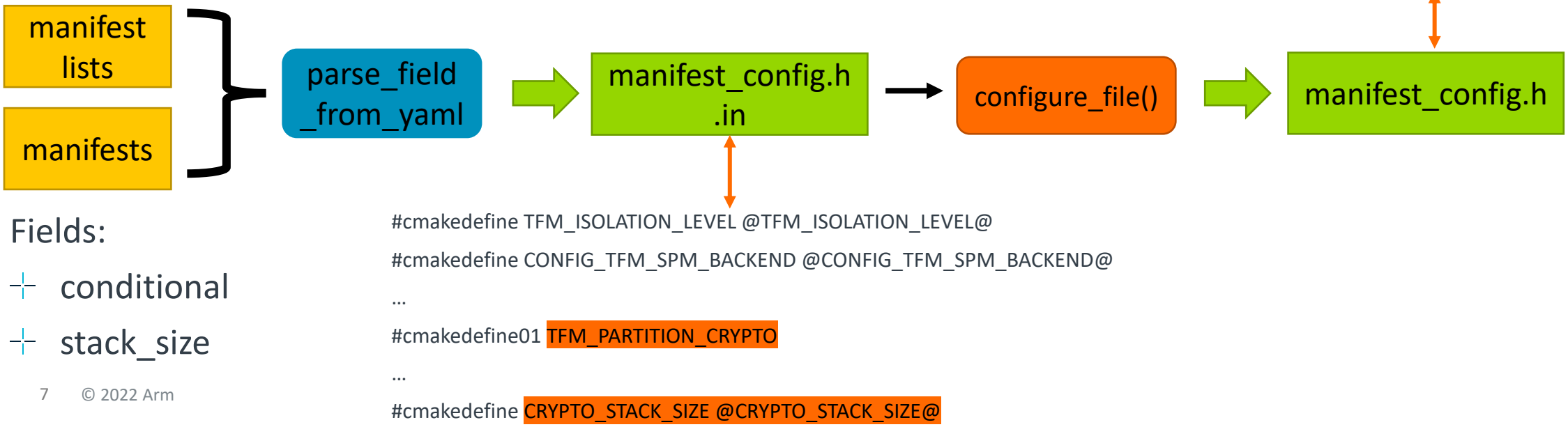


Now



How is The Config Header Generated in TF-M Build System

- Search for supported configurations in manifest lists and manifests
- Put them into a temporary file with the format
  - #cmakedefine01 config for BOOL types
  - #cmakedefine config @config@ for STRING types
- Do *configure_file()* on the above file



Fields:

- conditional
- stack_size

Other Use Cases of the New Feature Manifest Tool

- heap_size
- stateless_handle
- Everything in manifest or manifest lists in theory

Patches for Review

- Build: Support for configurable stack and heap size
  - Both stack_size and heap size are configurable

Alternatives to Solve the Stack Generation Issue

- Add compile definition to the target that builds the stack file
  - Very easy implementation
  - Issue for SFN backend – stack_size calculation and comparison required by manifest tool
- Set *stack_size* as *@CRYPTO_STACK_SIZE@* and *configure_file()* on the manifest
  - Couples manifests with build system
  - The configured manifests should not override the original one
  - Then it would be problem for the manifest tool to find the configured manifests because the manifest tool only knows how to find the original ones