Проекты организации Trusted Firmware:

свободное системное ПО обеспечения безопасности на процессорах Arm

Эльвира Хабирова, Александр Анисимов

Открытая Мобильная Платформа

17 июня 2021

Открытая Мобильная Платформа

- Компания:
  - Основана в 2016 году
  - Офисы в Москве, Иннополисе, Санкт-Петербурге
  - 200+ человек
- Основные продукты:
  - ОС Аврора + Аврора SDK
  - Аврора Центр (Enterprise Mobility Management)
  - Аврора TEE

Кто мы такие

Разработчики доверенной среды исполнения Аврора TEE

- Александр Анисимов
  - Выпускник НИЯУ МИФИ; в компании с 2019 года
  - Занимается безопасным хранилищем ключей и доверенными сервисами
  - Разработчик встраиваемых систем
- Эльвира Хабирова
  - Выпускник МГУ; в компании с 2020 года
  - Занимается доверенной загрузкой и механизмами безопасности в ядре

План

Доверенная среда исполнения

Сообщество Trusted Firmware

- Миссия
- Проекты
- Устройство сообщества

Работа с сообществом

- От идеи до коммита
- Как делать вклад в OP-TEE
- Обсуждение
- Pull requests

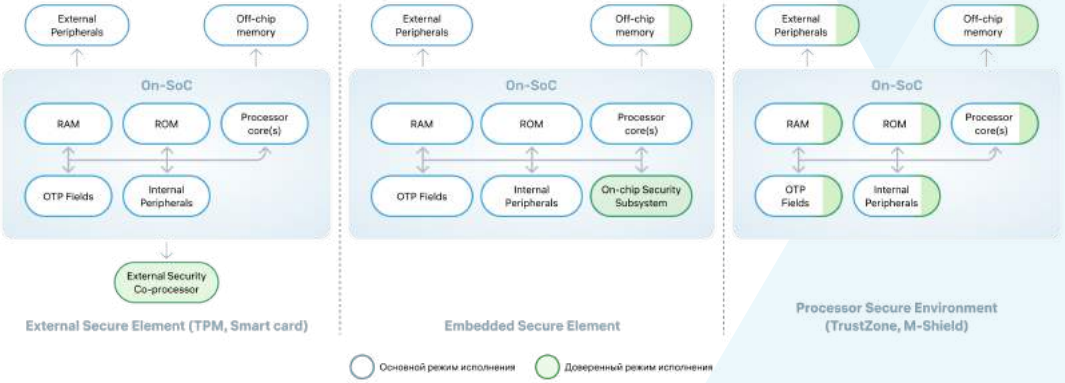
Почему Trust

- Это ПО, которому мы **доверяем**
- Уменьшаем trusted computing base

Применение

- Биометрическая идентификация
- Ввод паролей
- Доверенная криптография
- Хранилище ключей
  - Например, для банковских приложений

Доверенная среда исполнения

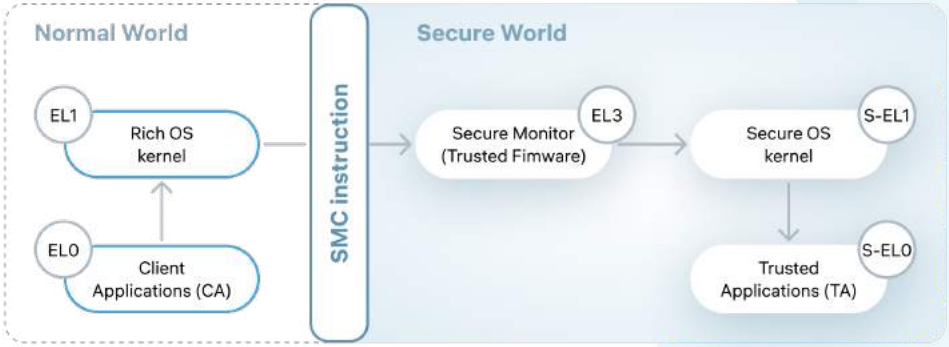


- Технология Arm TrustZone основывается на аппаратном разделении ресурсов между двумя состояниями процессора.

Подробнее про Arm TrustZone

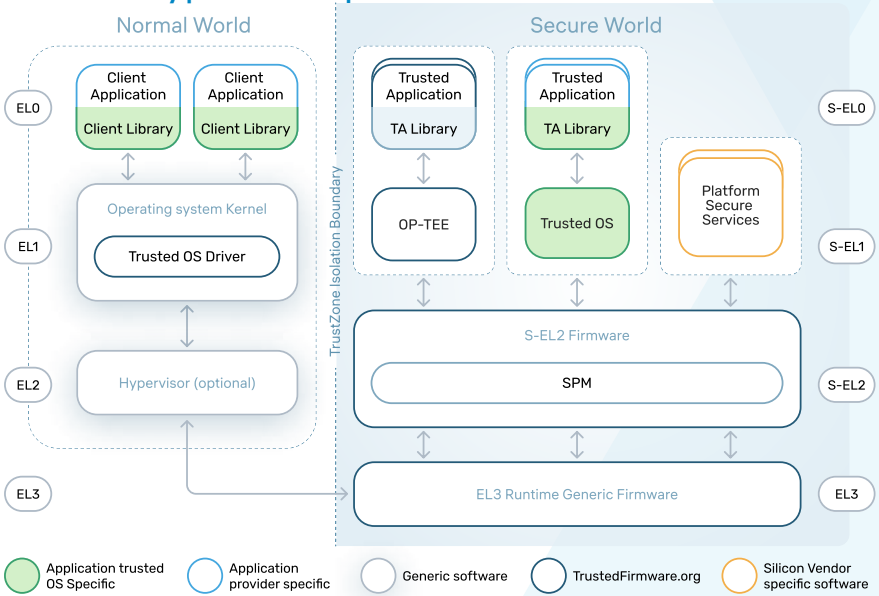
- 4 уровня исключений и 2 режима исполнения
- Разные адресные пространства
- TrustZone-aware периферия (GIC, CoreSight, SMMU, ...)
- TZASC (TrustZone Address Space Controller) и TZPC (TrustZone Protection Controller) — дополнительные IP-блоки программируемых компонентов контроллеров памяти и периферии для защиты памяти

Переключение

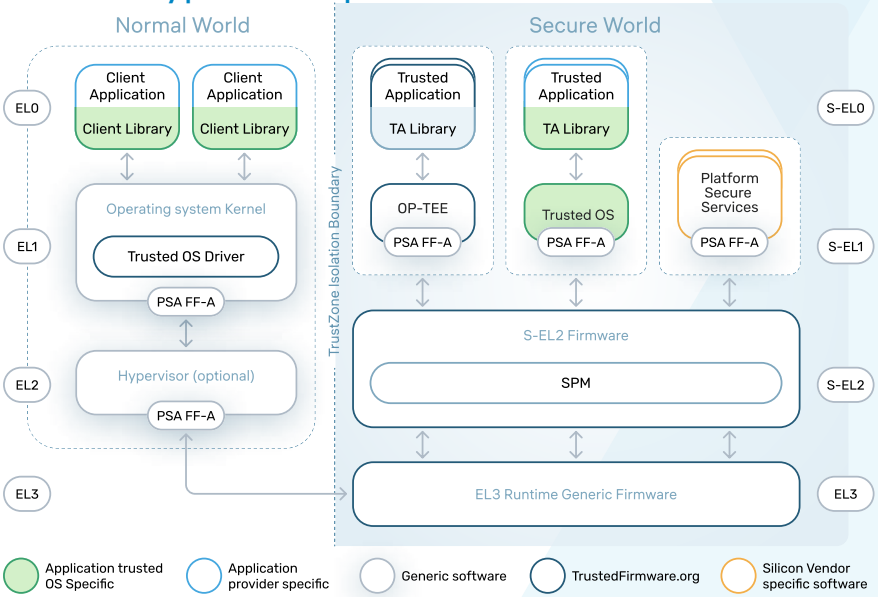


Переключение между основным и доверенными режимами осуществляется путём вызова специальной инструкции — SMC (Secure Monitor Call).

Общая схема уровней привелегий



Общая схема уровней привелегий



Члены Trusted Firmware

- Проект сообщества с открытым руководством

Миссия Trusted Firmware

- Поддерживать стек референсного ПО для доверенной среды исполнения
- Создавать и поддерживать связанные с этим стандарты
- Помогать производителям SoC использовать ПО Trusted Firmware
- Поддерживать порты на конкретные платформы

Проекты под управлением Trusted Firmware

- Trusted Firmware-A/M
  - Эталонная реализация монитора безопасности EL3
  - Загрузка доверенной ОС
  - Может реализовывать функции доверенной загрузки
  - Предоставление платформенных сервисов основной и доверенной ОС
  - Архитектура располагает к портированию кода на новые платформы
  - Реализует стандарты SMCCC, TBBR, PSCI, SCMI, и FF-A

Проекты под управлением Trusted Firmware

- OP-TEE
  - Доверенная среда исполнения для использования совместно с Linux в качестве основной ОС на процессорах Arm Cortex-A
  - Реализует стандарты GlobalPlatform API: TEE Client API Specification и TEE Internal Core API Specification

Проекты под управлением Trusted Firmware

- Hafnium
  - Гипервизор S-EL2
  - Эталонный менеджер безопасных партиций — SPM
  - Armv8.4-A+
  - Был передан в Trusted Firmware компанией Google в 2020 году
- Trusted Services
  - Фреймворк для разработки и развертывания доверенных сервисов в различных средах безопасного выполнения
  - Среды — доверенные приложения TEE, безопасные партиции, или безопасные анклавы
  - Самый новый проект в организации

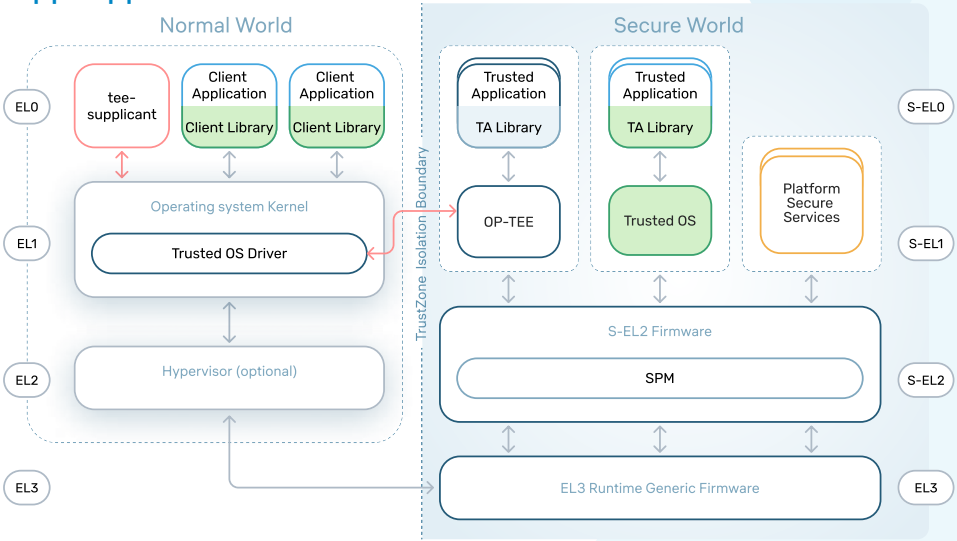
Проекты под управлением Trusted Firmware

- Mbed TLS
  - Реализация SSL/TLS и X.509, и соответствующих криптоалгоритмов
  - Подходит для встраиваемых систем
  - Распространяется под лицензией Apache 2.0
  - Используется в TF-A, TF-M и OP-TEE
  - До 2015 года именовался PolarSSL; в 2020 году стал частью Trusted Firmware

Устройство Trusted Firmware

- Два комитета: технический и исполнительный
- Члены сообщества имеют голос или долю голоса в комитетах
- См. также Project Charter и FAQ
- Также проводятся открытые технические форумы:
  - по TF-A — каждую нечетную неделю,
  - по TF-M — каждую четную,
  - и по OP-TEE — каждый месяц
  - https://www.trustedfirmware.org/meetings/

От идеи до коммита



- Проблема: tee-supplicant недостаточно расширяемый в плане специфичных для конкретной реализации функций

Как делать вклад в OP-TEE

- https://optee.readthedocs.io/en/latest/general/contribute.html

Разработка через Github

- Разработка OP-TEE ведется через Github
- Сначала принято обсуждение в форме issue, потом - PR
- Используется Developer Certificate of Origin (теги Signed-off-by:)
- Принято несколько элементов из традиционного стиля разработки: теги Tested-by:, Acked-by:, Suggested-by:, Reported-by:, ...
- Предложенные изменения сливаются в один коммит только в конце, вместе с проставлением тегов Acked-by:, Reviewed-by:

Обсуждение в issue

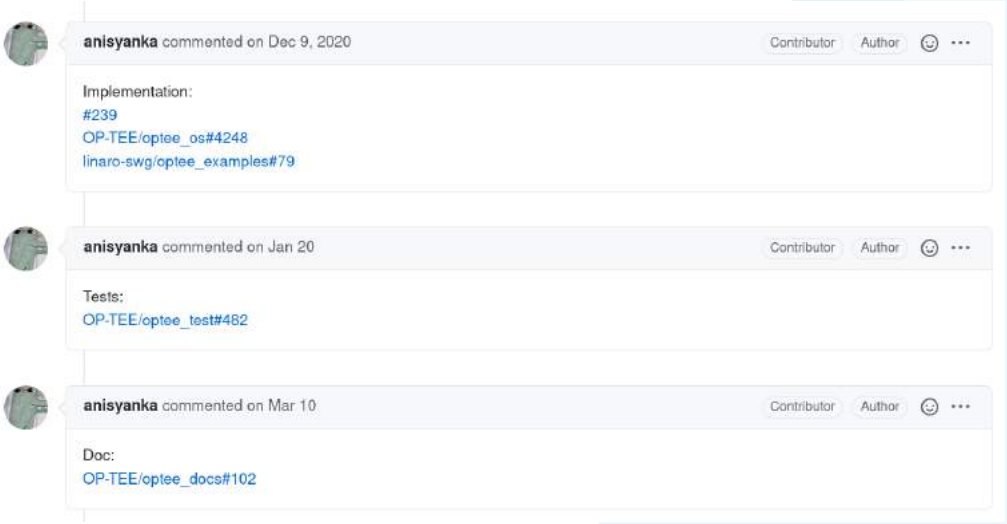
Обсуждение в issue

jforissier commented on Sep 18, 2020

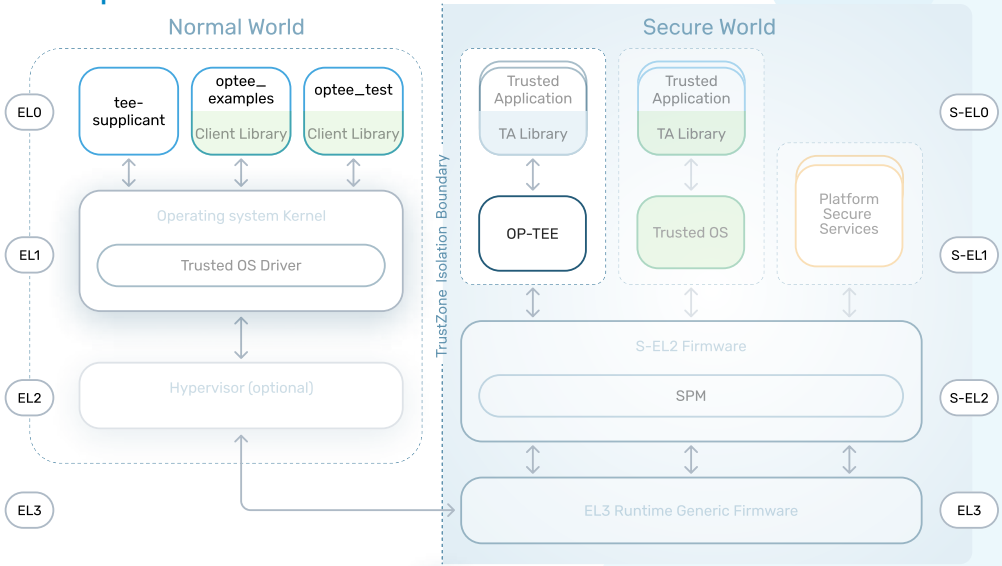
Hi **@anisyanka**,

> Hi,

Обсуждение в issue



Pull requests



Обсуждение в PR

tee-supplicant: add a framework for loadable plugins #239

Merged **jforissier** merged 1 commit into `OP-TEE:master` from `anisyanka:loadable-plugins` on Feb 4

Conversation 46 · Commits 1 · Checks 0 · Files changed 10

anisyanka commented on Dec 9, 2020 — Contributor

This framework makes the supplicant a bit more flexible
in terms of providing services. Any external TEE services
can be designed as a tee-supplicant plugin.
It makes it easy to:

- add new features in the supplicant that aren't needed in upstream,
  e.g. Rich OS specific services;
- sync upstream version with own fork;

To create a plugin developers should implement the
'struct plugin_method' and a bind function.
See public/tee_plugin_method.h file.
As an example this patch implements a 'syslog' plugin
and its Makefile. The plugin helps to log any
information from TEE to system journal.

Обсуждение в PR

tee-supplicant/src/optee_msg_supplicant.h

```
271 + * Invoke a tee-supplicant plugin.
272 + *
273 + * [in]     param[0].u.value.a        OPTEE_INVOKE_PLUGIN
274 + * [in]     param[0].u.value.b        uuid.d1
```

jenswi-linaro on Dec 16, 2020 Contributor

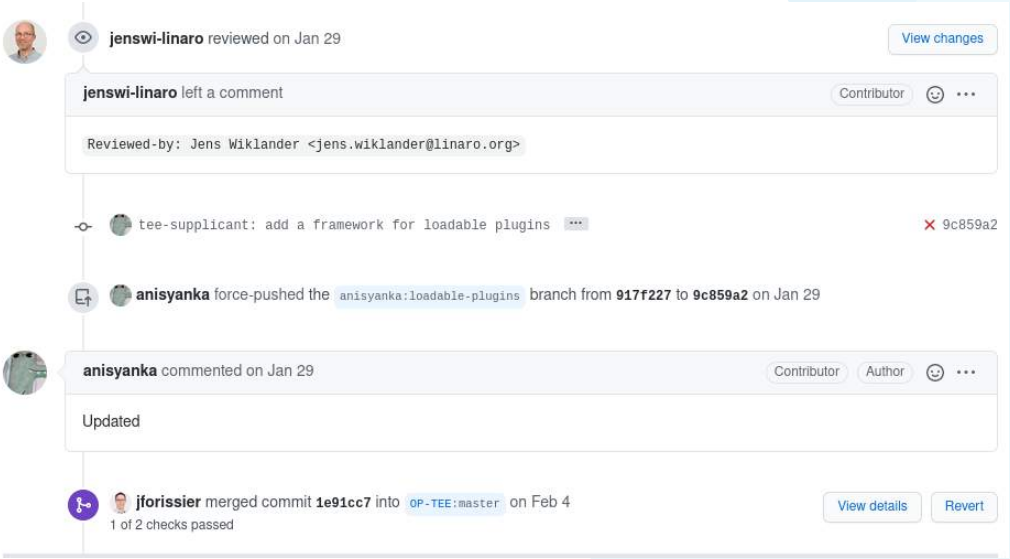
We need to define byte order of the UUID. Consider the case when Secure world is little endian but Normal world is Big endian. These value words will then be swapped (in the driver, not supported now by the way) before transmission.

I think this should to be defined as in https://developer.arm.com/documentation/den0028/c/ section "5.3 Unique Identification format".

1

- Будьте готовы выслушать критику и найти компромисс
- Регулярно обращаться к стандартам Arm приходится даже специалистам
- Обсуждение позволяет прийти к оптимальному решению

Обсуждение в PR



Итоги

- Рассказали об аппаратных возможностях обеспечения безопасности Arm
  - Уровни привелегий и их обязанности
- Trusted Firmware — сообщество с открытым руководством
  - ОМП — не только член, но и контрибьютор
- Описали проекты, входящие в Trusted Firmware
  - TF-A/M, OP-TEE, Hafnium, Mbed TLS, Trusted Services
- Рассмотрели процесс контрибьюта в один из проектов
  - Open Source — это важно

Запасные слайды

Аврора TEE

Доверенная среда исполнения Аврора TEE

- Обеспечение среды исполнения для доверенных сервисов
- Контроль целостности данных мобильной ОС
- Доверенный ввод-вывод (при загрузке)
- Биометрический датчик случайных чисел
- Безопасное хранилище данных
- Журналирование событий безопасности