Разработка виртуального HSM

Карташов Д. А.

Кафедра математических и информационных технологий
Санкт-Петербургский Академический университет

Введение

Криптография в приложениях:

- хранение секретных данных
- вычисления с их использованием

Проблемы безопасности:

- компрометация секретных данных

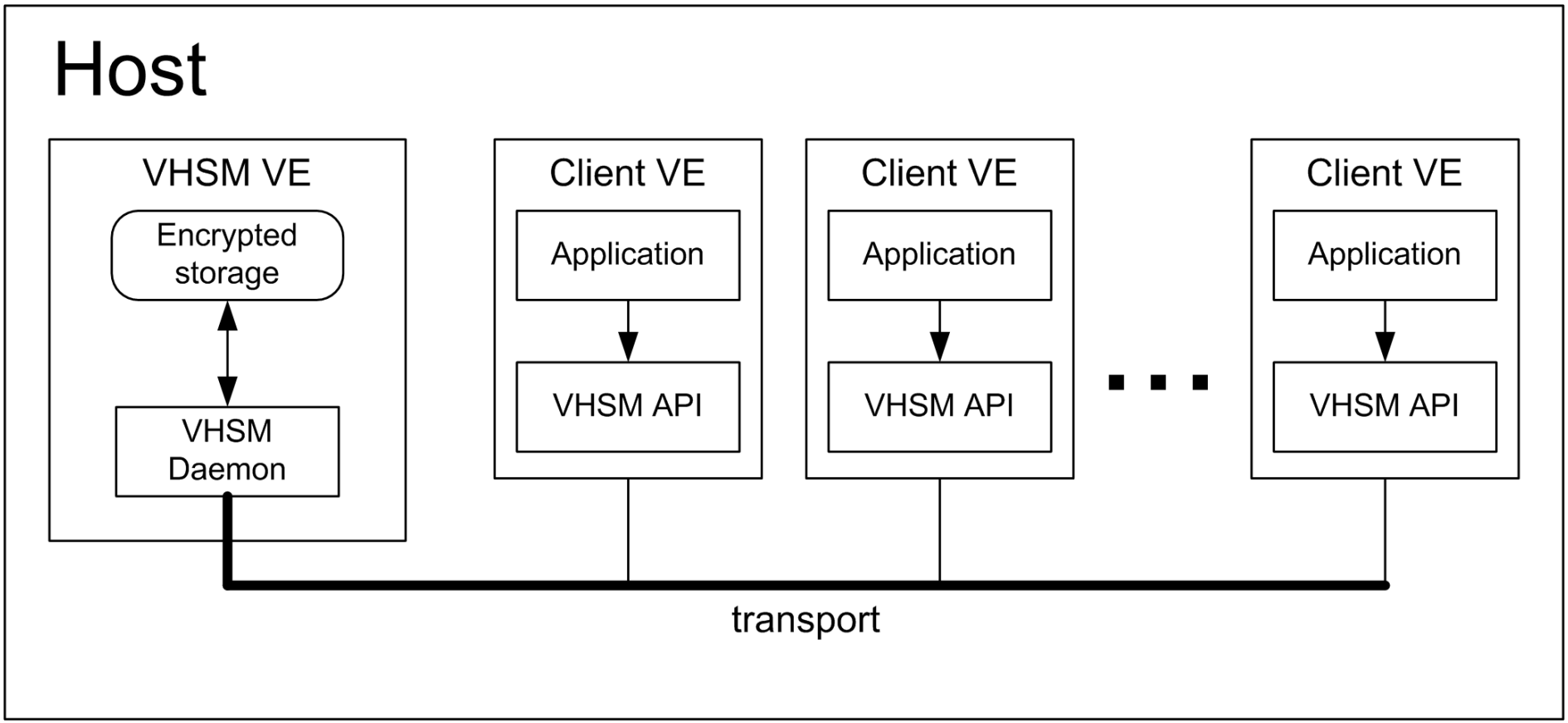
Решение:

- исключить попадание секретных данных на диск и/или в память компьютера

Цель проекта:

- разработка решения, предоставляющего функциональность HSM в виртуальном окружении

Общая архитектура решения



Основные компоненты

- **VHSM server**
  - аутентификация
  - выполнение криптографических операций с использованием секретных данных
- **Encrypted storage**
  - хранение секретных данных пользователя
- **VHSM API**
  - передача запросов на выполнение операций через транспорт
  - получение результатов операций
- **Transport**
  - пересылка сообщений
  - идентификация контейнеров
- **OpenSSL engine**
  - интерфейс между VHSM API и пользовательским приложением

VHSM server & encrypted storage

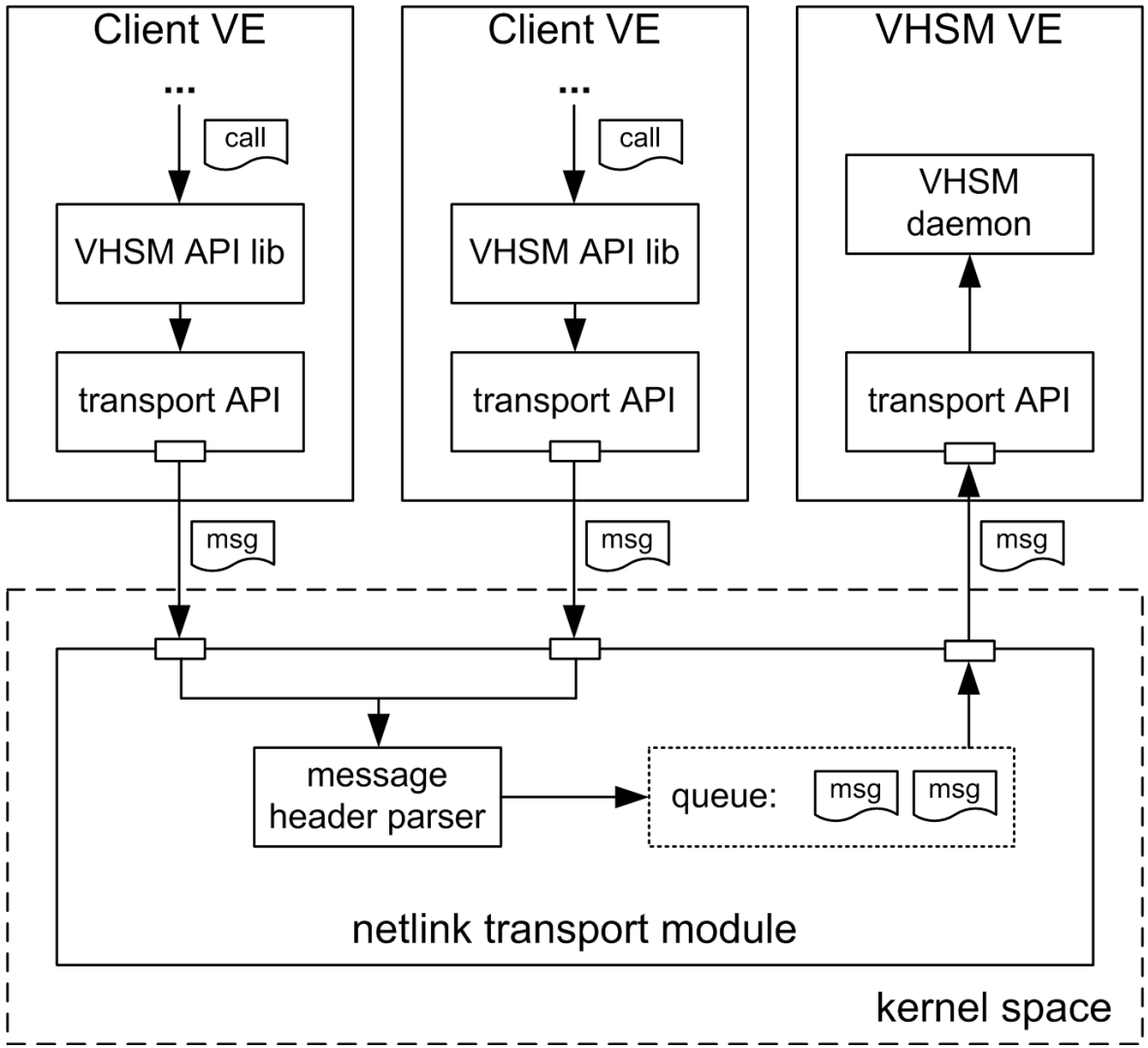
- доступ:
  - логин и пароль пользователя;
  - на основе пароля функцией PBKDF2 генерируется мастер-ключ шифрования данных пользователя;
- аутентификация:
  - 256-битный ключ аутентификации, зашифрованный с помощью мастер-ключа в режиме GCM;
- вычисление криптографических функций:
  - обращение к секретным данным по их идентификатору;
  - пользователью возвращается только результат операции;
- хранение секретных данных:
  - база данных SQLite;

VHSM API

- управление сессиями
  - открытие/завершение сессии;
  - аутентификация пользователя;
- управление ключами
  - импорт;
  - генерация;
  - удаление;
- хэширование и MAC
  - стандартные функции: `init`, `update`, `final`

Transport

- протокол — Google Protobuf
- реализован на основе netlink



OpenSSL engine

OpenSSL engine может быть использован для делегирования криптографических функций VHSM

В текущей реализации изменен алгоритм хэширования, что позволяет использовать стандартные функции для HMAC.

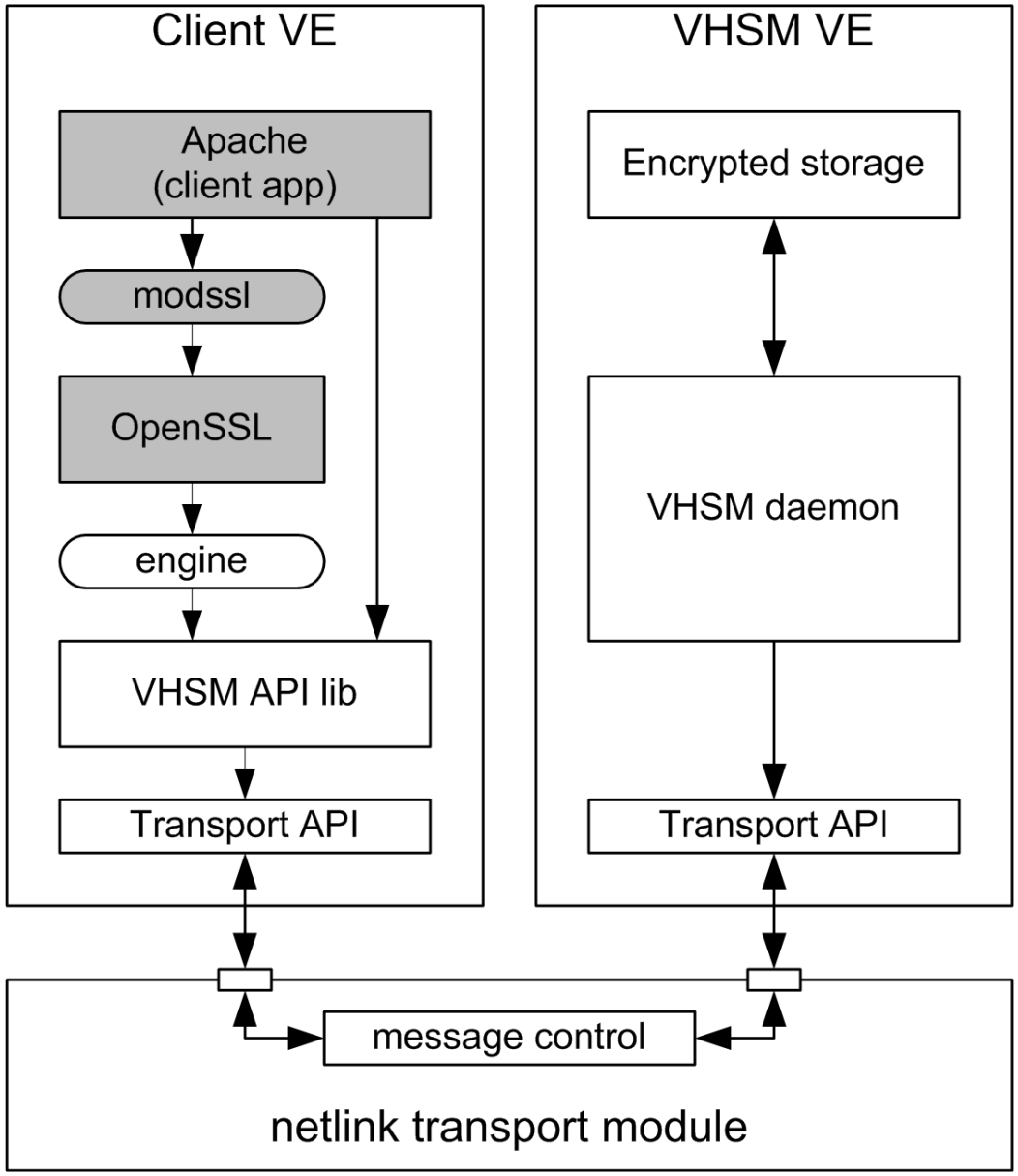
Минусы:

- алгоритм работы движка опирается на текущую реализацию функций в OpenSSL;
- возможная уязвимость при использовании конфигурационных файлов;

Плюсы:

- от конечного пользователя требуется меньше усилий для внедрения поддержки VHSM в свое приложение.

Пример использования



Итоги

Возможные направления развития проекта:

- введение ролей пользователей и уровней доступа к VHSM и хранилищу;
- расширение функциональности VHSM;
- адаптация для других виртуальных окружений.

Ссылки:

- репозиторий:
  `https://github.com/OSLL/vhsm`
- баг-трекер:
  `http://dev.osll.ru/projects/vhsm`