Проблемы встраивания криптографии в прикладные системы

Что встраивать, и как избежать ошибок?

Арина Эм

infotecs

конференция РусКрипто

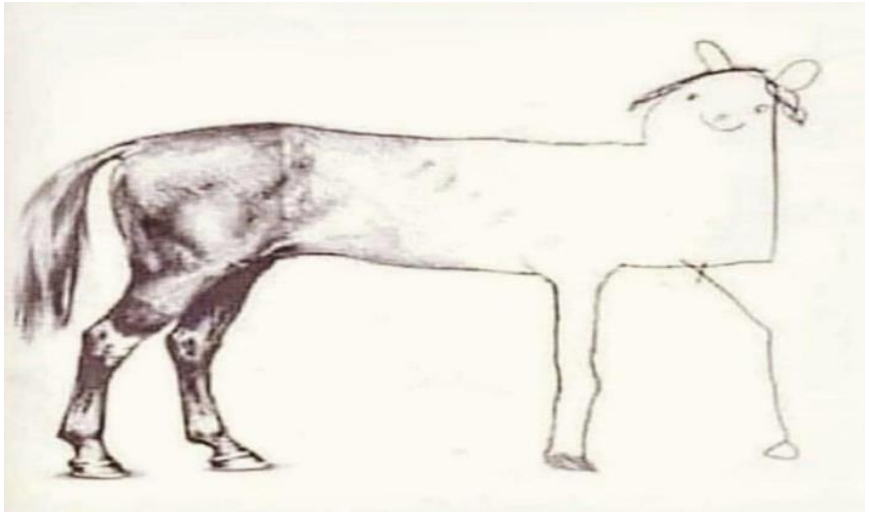
О чем поговорим

- Сложности, с которыми сталкиваются заказчики при встраивании криптографии
- Подход Инфотекс к архитектуре СКЗИ
- Как избегать ошибок

Заказчики сталкиваются с проблемами при встраивании

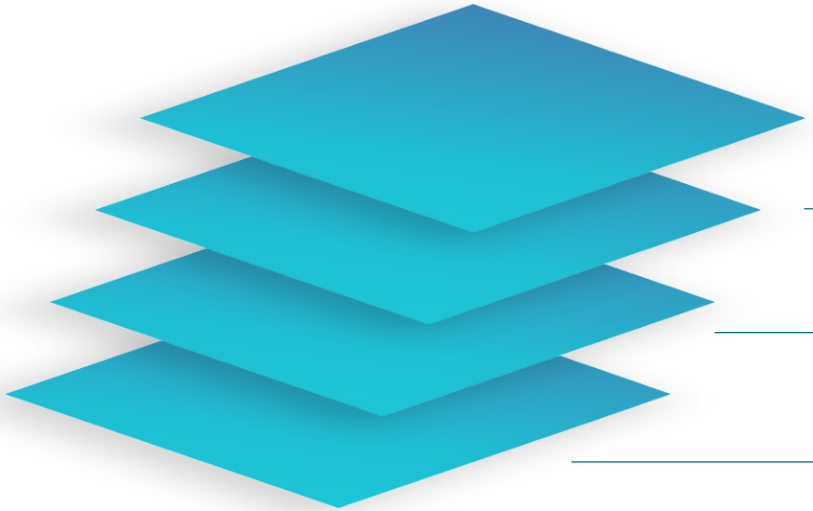
- Отсутствие опыта работы с криптографией
- Поверхностное знакомство с процессом сертификации
- Разработка без лицензии
- Использование низкоуровневых компонентов

Мы хотим избежать того, чтобы встраивание криптографии выглядело так



Как мы строим архитектуру СКЗИ

Как мы строим архитектуру СКЗИ



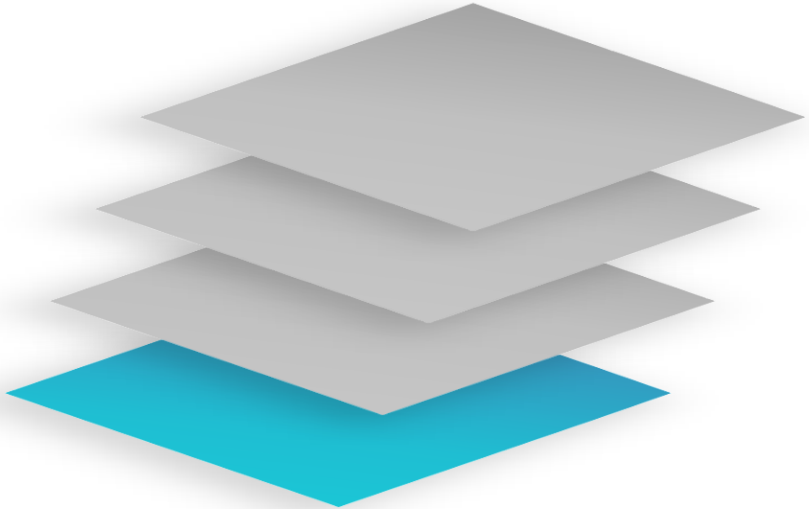
4 уровень
Прикладное ПО

3 уровень
Крипто форматы и протоколы

2 уровень
Стандартный интерфейс PKCS#11

1 уровень
Криптопримитивы

1 уровень: криптопримитивы



На этом уровне реализованы

- Математика
- Примитивы контроля функционирования и целостности ПО
- Работа с ДСЧ

Эти компоненты отдельно не распространяются и не лицензируются

Первый уровень:
Криптопримитивы

ViPNet CryptoUnderground

Не СКЗИ

Библиотека на C\C++с реализацией низкоуровневой криптографии

- Максимальная производительность
- Минимальная зависимость от ОС

Алгоритмы: ГОСТ 28147-89, Магма, Кузнечик, ГОСТ Р 34.11-2012, MD, SHA, AES, DES, ГОСТ Р 34.11-94, RSA, ECDSA, ГОСТ Р 34.10-2012, ГОСТ Р 34.10-2001

Платформы: Windows, Linux, UEFI, Байкал, Android, Аврора, iOS, macOS, Эльбрус, STM32

Архитектуры: Intel (x86, x86_64) Cortex (ARM), Байкал (ARM), Эльбрус (e2K)

Первый уровень:
Криптопримитивы

Не СКЗИ

ViPNet CrgTools

Библиотека на C++ и утилита проставления контрольных сумм

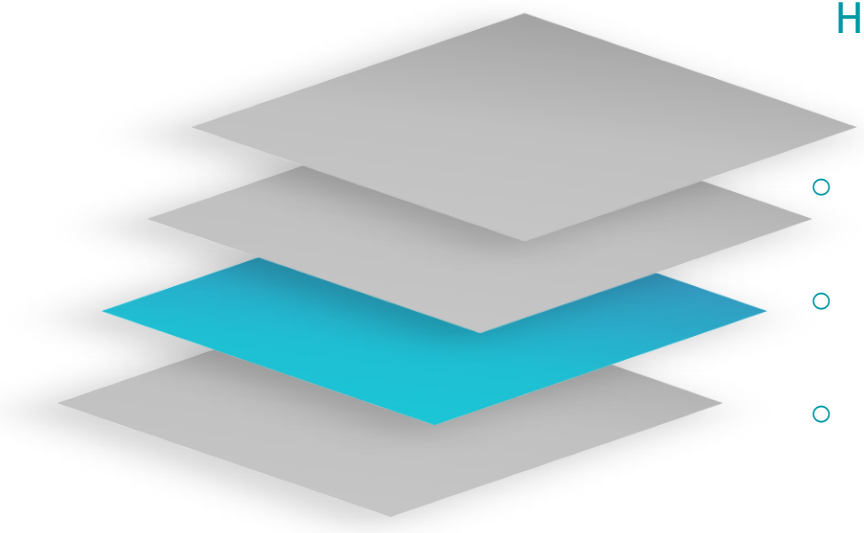
- Контроль целостности ПО
- Контроль целостности среды функционирования ПО
- Защита от случайных искажений

ViPNet UPRNG

Компонент для работы с источниками энтропии

- Накопление последовательности случайных чисел с ФДСЧ, БДСЧ
- Источник энтропии для инициализации других программных датчиков
- Использование предварительно распределенных последовательностей случайных чисел

2 уровень: интерфейс PKCS#11



На этом уровне реализованы

- Выполнение криптоопераций через интерфейс PKCS#11 с учётом методических рекомендаций ТК26
- Использование ключевых носителей разного типа, поддерживающих работу через интерфейс PKCS#11
- Использование токенов с поддержкой алгоритмов ГОСТ на неизвлекаемых долговременных ключах

Второй уровень:
Работа с ключевой информацией

Не СКЗИ

ViPNet PKCS#11 (ViPNet SoftToken)

Группа компонентов, реализующих стандартный криптографический интерфейс PKCS#11

- Соответствие мировому стандарту PKCS#11
- Кроссплатформенность
- Широкая распространенность
- Высокий уровень абстракции

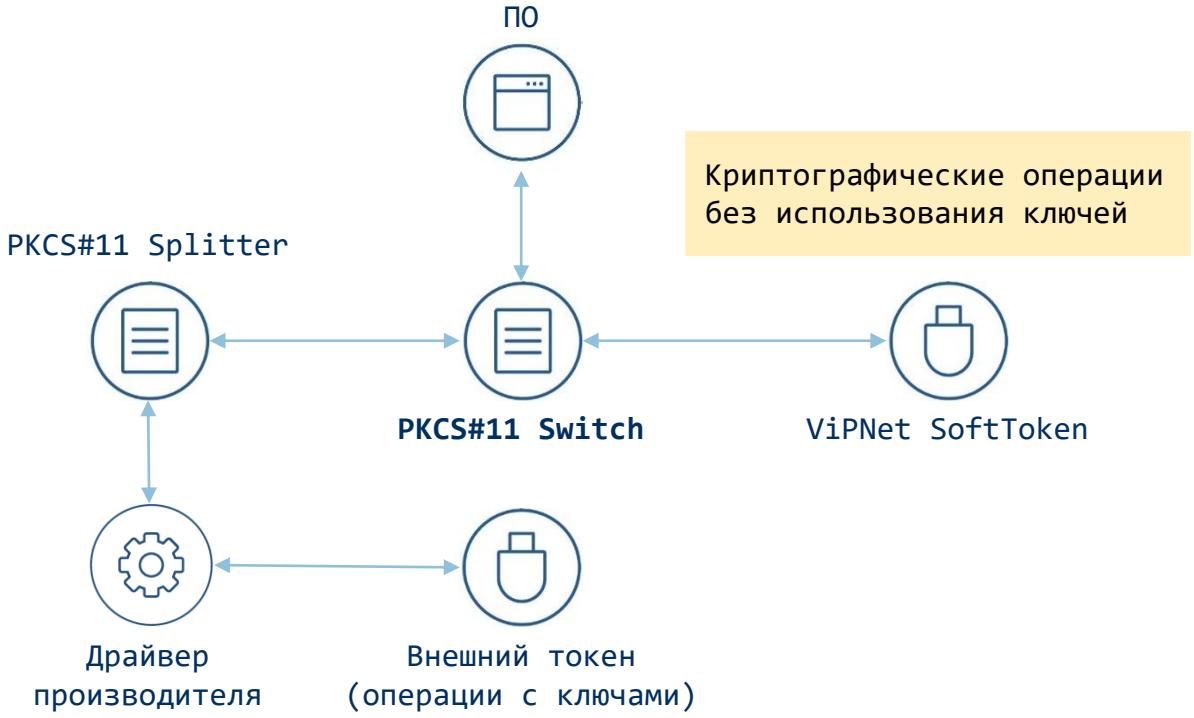
Платформы: Windows, Linux, Байкал, Эльбрус, Аврора, iOS

Компоненты ViPNet PKCS#11

PKCS#11 NGOST Storage / PKCS#11 Switch / SoftToken / PKCS#11 Splitter / PKCS#11 Proxy

Работа с токенами

Токен поддерживает криптооперации

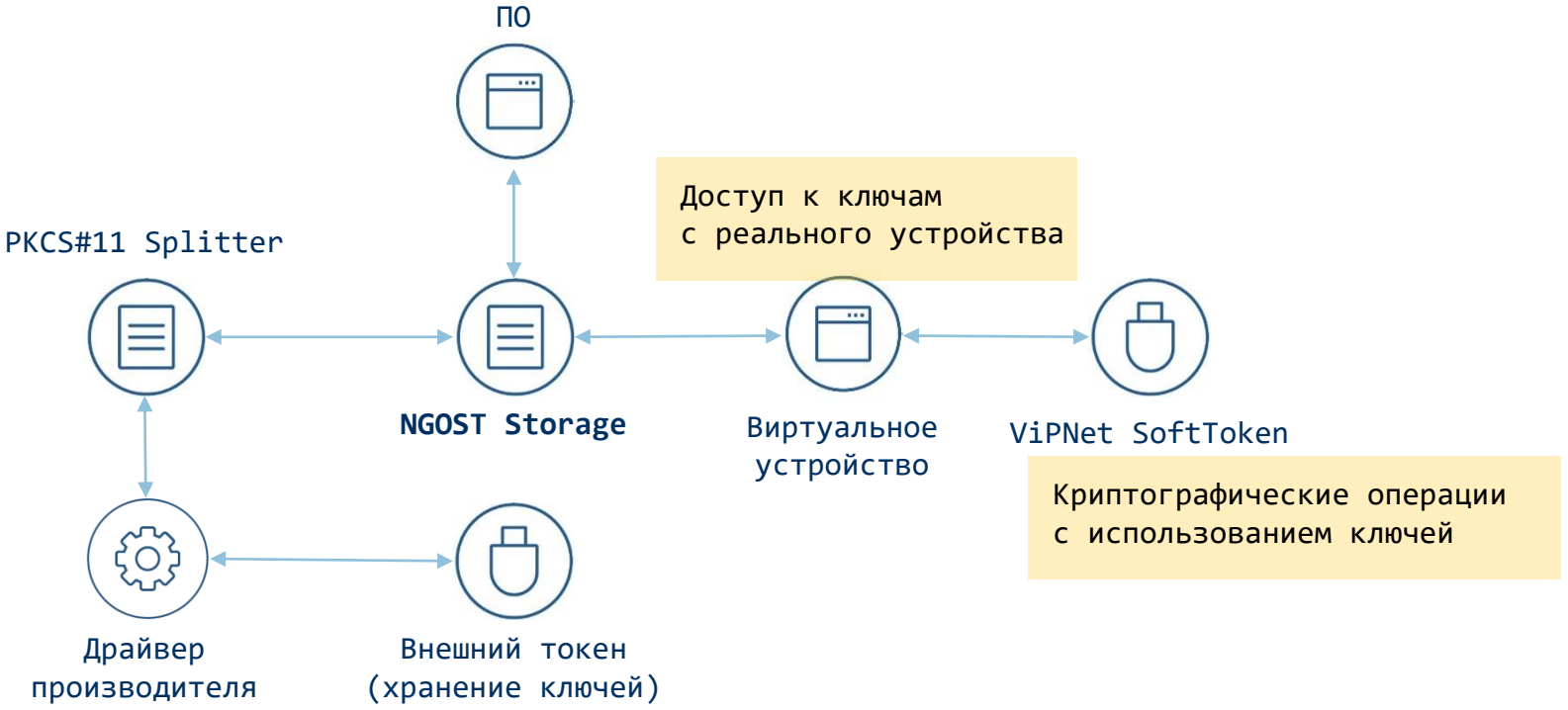


Поддерживаемые устройства

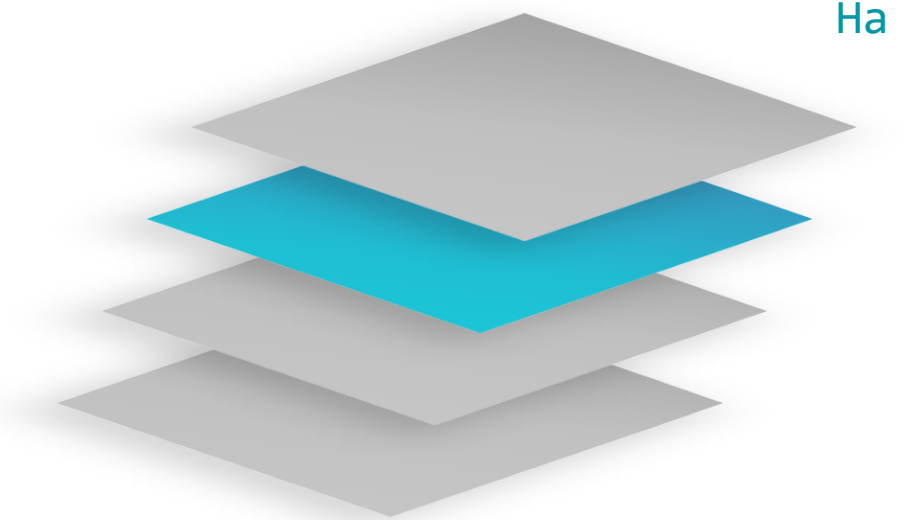
ViPNet HSM
Рутокен ЭЦП 2.0 Touch
Рутокен ЭЦП Bluetooth
Рутокен Lite
Рутокен ЭЦП 2.0, 2.0 Flash
Rutoken S
Рутокен ЭЦП 2.0 3000
Рутокен ЭЦП PKI
Рутокен ЭЦП 2.0 2100
Рутокен 2151
Рутокен ЭЦП 3.0 NFC
JaCarta LT
JaCarta PRO
JaCarta-2 SE
JaCarta-2 SF
JaCarta SF/ГОСТ
JaCarta-2 SE/PKI/ГОСТ
JaCarta-2 PKI/BIO/ГОСТ
JaCarta PKI/BIO
JaCarta PKI
JaCarta-2 ГОСТ

Работа с токенами

Токен в роли хранилища ключей



3 уровень: криптографические форматы и протоколы



На этом уровне реализованы

Третий уровень:
Реализация форматов и протоколов

СКЗИ

ViPNet CSP

Для разработки ПО под Windows

ViPNet OSSL

Для разработки мобильных
и серверных решений

ViPNet JCrypto SDK

Для разработки ПО на Java

ViPNet CryptoSmart

Для тех, кому нужен ГОСТ в блокчейне

Функции и особенности

Работа с ЭП

- ГОСТ Р 34.10-2012

Хэширование

- ГОСТ Р 34.11-2012

Шифрование

- ГОСТ Р 34.12-2015
- ГОСТ Р 34.13-2015

Защищенные соединения

- TLS 1.2
- TLS 1.3

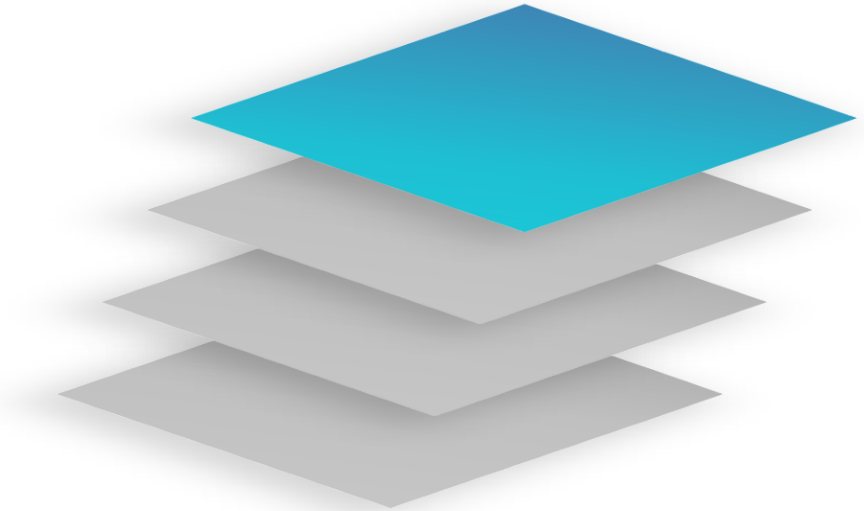
Высокоуровневые API

- CryptoAPI
- OpenSSL
- Java SDK
- GO

Поддержка ОС

4 уровень: прикладное ПО

СКЗИ или нет



На этом уровне

- Криптография встраивается в ПО
- Проводится оценка влияния/сертификация

Мы рекомендуем придерживаться хороших практик

1. Использовать белые функции
2. Выделять модуль для работы с криптографией
3. Не пересобирать данный модуль без необходимости, а использовать его в бинарном виде
4. Использовать инструменты, с которыми вендор уже провел исследования
5. Использовать сертифицированные ОС

Легкого встраивания!

Арина Эм

e-mail: Arina.Em@infotecs.ru

Подписывайтесь на наши соцсети

https://vk.com/infotecs_news

https://t.me/infotecs_news