# Об использовании токенов и смарт-карт в средствах электронной подписи

## Унификация аутентификации

Агафьин Сергей Сергеевич
Бородин Георгий Олегович

# Токены и смарт-карты





# Интерфейсы взаимодействия

- PCSC
  - Не формализует функциональность.
- APDU и ISO 7816
  - Транспортный интерфейс.
  - Стандарт описывает простейшую логику взамодействия.
- Cryptoki / PKCS#11
  - Популярный интерфейс.
  - Часто слишком низкоуровневый.
- Частные библиотеки производителей
  - Носители каждого производителя внедряются заново.

# Интерфейсы взаимодействия

- PCSC
  - Не формализует функциональность.
- APDU и ISO 7816
  - Транспортный интерфейс.
  - Стандарт описывает простейшую логику взамодействия.
- Cryptoki / PKCS#11
  - Популярный интерфейс.
  - Часто слишком низкоуровневый.
- Частные библиотеки производителей
  - Носители каждого производителя внедряются заново.

# Интерфейсы взаимодействия

- PCSC
  - Не формализует функциональность.
- APDU и ISO 7816
  - Транспортный интерфейс.
  - Стандарт описывает простейшую логику взамодействия.
- Cryptoki / PKCS#11
  - Популярный интерфейс.
  - Часто слишком низкоуровневый.
- Частные библиотеки производителей
  - Носители каждого производителя внедряются заново.

# Интерфейсы взаимодействия

- PCSC
  - Не формализует функциональность.
- APDU и ISO 7816
  - Транспортный интерфейс.
  - Стандарт описывает простейшую логику взамодействия.
- Cryptoki / PKCS#11
  - Популярный интерфейс.
  - Часто слишком низкоуровневый.
- Частные библиотеки производителей
  - Носители каждого производителя внедряются заново.

# Интерфейсы взаимодействия

- PCSC
  - Не формализует функциональность.
- APDU и ISO 7816
  - Транспортный интерфейс.
  - Стандарт описывает простейшую логику взамодействия.
- Cryptoki / PKCS#11
  - Популярный интерфейс.
  - Часто слишком низкоуровневый.
- Частные библиотеки производителей
  - Носители каждого производителя внедряются заново.

# Общие проблемы известных интерфейсов

1. Каждый производитель понимает по-своему.
2. Частичное следование стандарту.
3. Заказчики могут просить то, что противоречит стандарту.
4. Прикладное ПО должно разбираться в особенностях каждого токена.

# Общие проблемы известных интерфейсов

1. Каждый производитель понимает по-своему.
2. Частичное следование стандарту.
3. Заказчики могут просить то, что противоречит стандарту.
4. Прикладное ПО должно разбираться в особенностях каждого токена.

# Общие проблемы известных интерфейсов

1. Каждый производитель понимает по-своему.
2. Частичное следование стандарту.
3. Заказчики могут просить то, что противоречит стандарту.
4. Прикладное ПО должно разбираться в особенностях каждого токена.

# Общие проблемы известных интерфейсов

1. Каждый производитель понимает по-своему.
2. Частичное следование стандарту.
3. Заказчики могут просить то, что противоречит стандарту.
4. Прикладное ПО должно разбираться в особенностях каждого токена.

# Каждая задача решается отдельно

1. Существование нескольких апплетов (приложений) на одном носителе.
2. Дополнительные функциональности носителей и считывателей – хэширование, визуализация, генерация случайных чисел, поддержка защищенного канала.
3. Одна из важнейших – аутентификация. О ней и поговорим.

# Типы аутентификаций

- Контейнерная (CONT)
  - Для каждого контейнера свой собственный пароль.
- Административная (ADMIN/ROOT)
  - Создание/удаление контейнеров.
- Разблокирующая (PUK)
  - Смена/разблокировка заблокированного пароля.
- Дополнительные аутентификации.
  - Поднятие общего канала.
  - Чтение служебных данных.

- CONT + ADMIN – один пользователь на все ключи (контейнеры).
- ADMIN + PUK – суперпользовательская.

# Типы аутентификаций

- Контейнерная (CONT)
  - Для каждого контейнера свой собственный пароль.
- Административная (ADMIN/ROOT)
  - Создание/удаление контейнеров.
- Разблокирующая (PUK)
  - Смена/разблокировка заблокированного пароля.
- Дополнительные аутентификации.
  - Поднятие общего канала.
  - Чтение служебных данных.

- CONT + ADMIN – один пользователь на все ключи (контейнеры).
- ADMIN + PUK – суперпользовательская.

# Типы аутентификаций

- Контейнерная (CONT)
  - Для каждого контейнера свой собственный пароль.
- Административная (ADMIN/ROOT)
  - Создание/удаление контейнеров.
- Разблокирующая (PUK)
  - Смена/разблокировка заблокированного пароля.
- Дополнительные аутентификации.
  - Поднятие общего канала.
  - Чтение служебных данных.

- CONT + ADMIN – один пользователь на все ключи (контейнеры).
- ADMIN + PUK – суперпользовательская.

# Типы аутентификаций

- Контейнерная (CONT)
  - Для каждого контейнера свой собственный пароль.
- Административная (ADMIN/ROOT)
  - Создание/удаление контейнеров.
- Разблокирующая (PUK)
  - Смена/разблокировка заблокированного пароля.
- Дополнительные аутентификации.
  - Поднятие общего канала.
  - Чтение служебных данных.

- CONT + ADMIN – один пользователь на все ключи (контейнеры).
- ADMIN + PUK – суперпользовательская.

# Типы аутентификаций

- Контейнерная (CONT)
  - Для каждого контейнера свой собственный пароль.
- Административная (ADMIN/ROOT)
  - Создание/удаление контейнеров.
- Разблокирующая (PUK)
  - Смена/разблокировка заблокированного пароля.
- Дополнительные аутентификации.
  - Поднятие общего канала.
  - Чтение служебных данных.

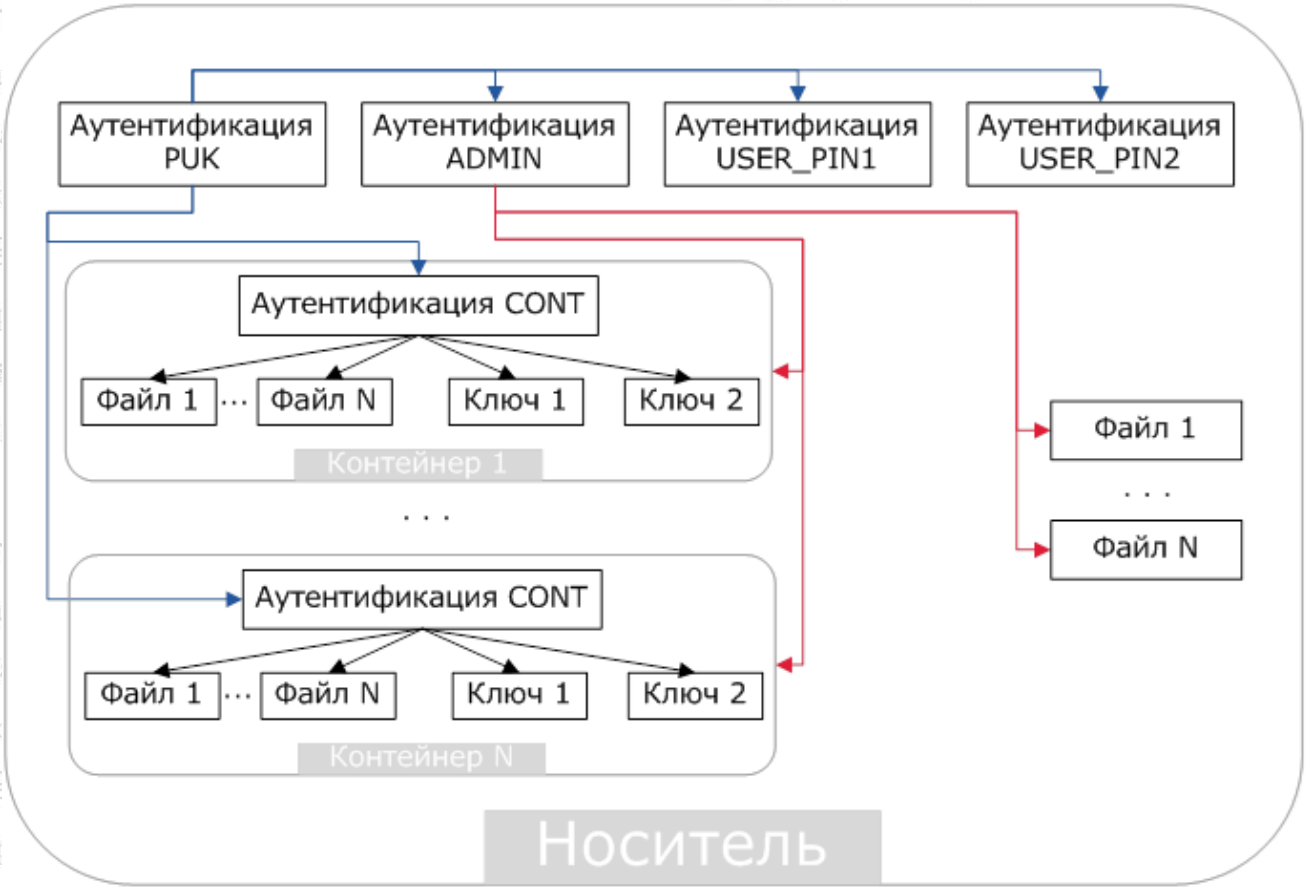
- CONT + ADMIN – один пользователь на все ключи (контейнеры).
- ADMIN + PUK – суперпользовательская.

# Типы аутентификаций

- Контейнерная (CONT)
  - Для каждого контейнера свой собственный пароль.
- Административная (ADMIN/ROOT)
  - Создание/удаление контейнеров.
- Разблокирующая (PUK)
  - Смена/разблокировка заблокированного пароля.
- Дополнительные аутентификации.
  - Поднятие общего канала.
  - Чтение служебных данных.

- CONT + ADMIN – один пользователь на все ключи (контейнеры).
- ADMIN + PUK – суперпользовательская.

# Типы аутентификаций



# Опциональные особенности аутентификация

## Общего характера

- Для смены пароля требуется предъявление текущего.
- У нескольких приложений на устройстве одна аутентификация.

## Свойства PUK

- Позволяет менять любую аутентификацию.
- Позволяет сбросить счетчики любой аутентификации.

## Свойства основной аутентификации

- Может быть установлена в особое дефолтное значение.

В теории опциональных свойств десятки!

# Опциональные особенности аутентификация

## Общего характера

- Для смены пароля требуется предъявление текущего.
- У нескольких приложений на устройстве одна аутентификация.

## Свойства PUK

- Позволяет менять любую аутентификацию.
- Позволяет сбросить счетчики любой аутентификации.

## Свойства основной аутентификации

- Может быть установлена в особое дефолтное значение.

В теории опциональных свойств десятки!

# Опциональные особенности аутентификация

## Общего характера

- Для смены пароля требуется предъявление текущего.
- У нескольких приложений на устройстве одна аутентификация.

## Свойства PUK

- Позволяет менять любую аутентификацию.
- Позволяет сбросить счетчики любой аутентификации.

## Свойства основной аутентификации

- Может быть установлена в особое дефолтное значение.

В теории опциональных свойств десятки!

# Опциональные особенности аутентификация

## Общего характера

- Для смены пароля требуется предъявление текущего.
- У нескольких приложений на устройстве одна аутентификация.

## Свойства PUK

- Позволяет менять любую аутентификацию.
- Позволяет сбросить счетчики любой аутентификации.

## Свойства основной аутентификации

- Может быть установлена в особое дефолтное значение.

В теории опциональных свойств десятки!

# Алгоритмы (способы) аутентификации

- Обычная парольная, plain (SIMPLE).
- Протокольная (SESPAKE).
- Внешняя к ПО / кнопки, биометрия (SELF).

# Алгоритмы (способы) аутентификации

- Обычная парольная, plain (SIMPLE).
- Протокольная (SESPAKE).
- Внешняя к ПО / кнопки, биометрия (SELF).

# Алгоритмы (способы) аутентификации

- Обычная парольная, plain (SIMPLE).
- Протокольная (SESPAKE).
- Внешняя к ПО / кнопки, биометрия (SELF).

# Алгоритмы (способы) аутентификации

- Обычная парольная, plain (SIMPLE).
- Протокольная (SESPAKE).
- Внешняя к ПО / кнопки, биометрия (SELF).

# Сохранение паролей / кэширование (1/2)

## Виды кэшей

- Системный кэш (реестр, жесткий диск)
  - Сохранение пароля для межпроцессного доступа.
- Глобальный кэш (в процессе)
  - Сохранение пароля между обращениями к криптосредству из одного процесса.
- Локальный кэш (в контексте)
  - Пароль сохраняется только для текущего контекста работы с ключом.
- Без кэша
  - Пароль сохраняется в токене.

# Сохранение паролей / кэширование (1/2)

## Виды кэшей

- Системный кэш (реестр, жесткий диск)
  - Сохранение пароля для межпроцессного доступа.
- Глобальный кэш (в процессе)
  - Сохранение пароля между обращениями к криптосредству из одного процесса.
- Локальный кэш (в контексте)
  - Пароль сохраняется только для текущего контекста работы с ключом.
- Без кэша
  - Пароль сохраняется в токене.

# Сохранение паролей / кэширование (1/2)

## Виды кэшей

- Системный кэш (реестр, жесткий диск)
  - Сохранение пароля для межпроцессного доступа.
- Глобальный кэш (в процессе)
  - Сохранение пароля между обращениями к криптосредству из одного процесса.
- Локальный кэш (в контексте)
  - Пароль сохраняется только для текущего контекста работы с ключом.
- Без кэша
  - Пароль сохраняется в токене.

# Сохранение паролей / кэширование (1/2)

## Виды кэшей

- Системный кэш (реестр, жесткий диск)
  - Сохранение пароля для межпроцессного доступа.
- Глобальный кэш (в процессе)
  - Сохранение пароля между обращениями к криптосредству из одного процесса.
- Локальный кэш (в контексте)
  - Пароль сохраняется только для текущего контекста работы с ключом.
- Без кэша
  - Пароль сохраняется в токене.

# Сохранение паролей / кэширование (1/2)

## Виды кэшей

- Системный кэш (реестр, жесткий диск)
  - Сохранение пароля для межпроцессного доступа.
- Глобальный кэш (в процессе)
  - Сохранение пароля между обращениями к криптосредству из одного процесса.
- Локальный кэш (в контексте)
  - Пароль сохраняется только для текущего контекста работы с ключом.
- Без кэша
  - Пароль сохраняется в токене.

# Сохранение паролей / кэширование (2/2)

Всё должно быть настраиваемо.

## Особенности/ограничения

- Носители могут не сбрасывать пароль по аппаратному событию (RESET).
- Носители могут сбрасывать пароль при следующем обращении (SELECT).
- Пароль может явно требоваться интерфейсом, передаваться в команде.
- Кэшировать PUK - плохо.

## Следствия

- Приходится кэшировать основную аутентификацию.
- Иногда приходится кэшировать PUK.

# Сохранение паролей / кэширование (2/2)

Всё должно быть настраиваемо.

## Особенности/ограничения

- Носители могут не сбрасывать пароль по аппаратному событию (RESET).
- Носители могут сбрасывать пароль при следующем обращении (SELECT).
- Пароль может явно требоваться интерфейсом, передаваться в команде.
- Кэшировать PUK - плохо.

## Следствия

- Приходится кэшировать основную аутентификацию.
- Иногда приходится кэшировать PUK.

# Сохранение паролей / кэширование (2/2)

Всё должно быть настраиваемо.

## Особенности/ограничения

- Носители могут не сбрасывать пароль по аппаратному событию (RESET).
- Носители могут сбрасывать пароль при следующем обращении (SELECT).
- Пароль может явно требоваться интерфейсом, передаваться в команде.
- Кэшировать PUK - плохо.

## Следствия

- Приходится кэшировать основную аутентификацию.
- Иногда приходится кэшировать PUK.

# Сохранение паролей / кэширование (2/2)

Всё должно быть настраиваемо.

## Особенности/ограничения

- Носители могут не сбрасывать пароль по аппаратному событию (RESET).
- Носители могут сбрасывать пароль при следующем обращении (SELECT).
- Пароль может явно требоваться интерфейсом, передаваться в команде.
- Кэшировать PUK - плохо.

## Следствия

- Приходится кэшировать основную аутентификацию.
- Иногда приходится кэшировать PUK.

# Сохранение паролей / кэширование (2/2)

Всё должно быть настраиваемо.

## Особенности/ограничения

- Носители могут не сбрасывать пароль по аппаратному событию (RESET).
- Носители могут сбрасывать пароль при следующем обращении (SELECT).
- Пароль может явно требоваться интерфейсом, передаваться в команде.
- Кэшировать PUK - плохо.

## Следствия

- Приходится кэшировать основную аутентификацию.
- Иногда приходится кэшировать PUK.

# Сохранение паролей / кэширование (2/2)

Всё должно быть настраиваемо.

## Особенности/ограничения

- Носители могут не сбрасывать пароль по аппаратному событию (RESET).
- Носители могут сбрасывать пароль при следующем обращении (SELECT).
- Пароль может явно требоваться интерфейсом, передаваться в команде.
- Кэшировать PUK - плохо.

## Следствия

- Приходится кэшировать основную аутентификацию.
- Иногда приходится кэшировать PUK.

# Заключение

- Существующие стандарты не подходят для всех возникающих задач.
- Мы предлагаем хотя бы неформально перейти на общий язык.
- Много открытых вопросов
  - Какие детали должен знать пользователь?
  - Как пользователь должен понимать разделение паролей / кэшей?
  - Нужно ли рассказывать пользователю об используемом алгоритме?

CRYPTOPRO

# Спасибо за внимание!


- sagafyin@cryptopro.ru
- borodin@cryptopro.ru