Миграция с плагина Рутокен Web на Рутокен Плагин

Общая информация

Плагин Рутокен Web — это браузерный плагин, созданный для работы с устройством Рутокен Web.

Рутокен Плагин — это гораздо более современный и функциональный браузерный плагин предназначенный для работы с основной линейкой устройств Рутокен, включая ЭЦП 2.0, PINPad, ЭЦП PKI.

Он активно дорабатывается и поддерживается и рекомендован для работы со всеми устройствами семейства Рутокен ЭЦП, а также ограниченно поддерживает "зеленые токены" Рутокен Web.

Развивать два одинаковых продукта нецелесообразно, поэтому доработки плагина Рутокен Web прекращены, а все новые возможности и фичи получает только Рутокен Плагин.

API плагина Рутокен Плагин включает в себя самые востребованные фичи плагина Рутокен Web, но избавлен от устаревших и малоиспользуемых функций.

Эта инструкция подготовлена специально для плавного перехода с плагина Рутокен Web на Рутокен Плагин.

Она пригодиться в том случае, если у вас уже встроена поддержка "зеленых токенов" (устройств Рутокен Web) через плагин Рутокен Web и серверная часть работает с данными, которыми оперирует плагин Рутокен Web.

Предложенная схема требует минимальных изменений в клиентском коде и абсолютно не затрагивает серверный код.

В результате описанных процедур вы получите возможность:

- использовать современный и поддерживаемый плагин (получить больше поддерживаемых платформ и браузеров)
- перейти на новые отечественные алгоритмы электронной подписи ГОСТ 2012
- перейти на мировые стандарты электронной подписи RSA (до 2048 бит)
- перейти с "сырой" подписи на полноценную подпись с атрибутами формата PKCS#7 и сертификаты X509
- использовать всю линейку продуктов семейства Рутокен ЭЦП, не ограничиваясь моделью Рутокен Web
- использовать более современную модель promise в Javascript

Схема аутентификации с использованием Рутокен Web:

1. Зарегистрированный на сайте клиент генерирует ключевую пару и сохраняет ее на сайте.
2. При авторизации на сайте запрашиваются случайные данные для подписи. Они сохраняются на сервере и отправляются клиенту.
3. Клиент получает данные, генерирует "соль", которая добавляется к данным. Результат подписывается с использованием сгенерированной в п. 1 ключевой пары и вместе с "солью" отправляется на сервер для проверки.
4. Сервер производит такую же процедуру с "солью" и сохраненными в п. 2 данными. Проверяется корректность подписи и принимается решение об успешности процесса аутентификации.

В этой схеме нет сложностей, связанных с сертификатами X509, и стандартами подписи PKCS#7 (CMS), происходит только "сырая" подпись данных и проверка корректности подписи.

Однако, при желании или необходимости, сертификаты X509 и CMS могут быть сравнительно легко подключены.

Формирование электронной подписи через плагин Рутокен Web производится по алгоритму ГОСТ Р 34.10-2001.

Схема миграции с плагина Рутокен Web на Рутокен Плагин

1. Установим Рутокен Плагин.
   Если планируется работать с устройствами Рутокен Web:
   ftp://ftp.rutoken.ru/software/Rutoken_Plugin/4.3.1.0/Windows/RutokenPlugin.msi

   Если планируется работать с устройствами семейства Рутокен ЭЦП:
   https://www.rutoken.ru/support/download/rutoken-plugin/
2. Изменим загрузчик плагина. Для этого заменим на страницах или в системе сборки скрипт rutokenweb.js на rutoken.js:
   https://www.npmjs.com/package/rutoken
   Оба эти скрипта используют Promise Если у вас установлен и работает rutokenweb.js, то зависимости должны быть. Иначе следует установить скрипт для работы с промисами, например данную библиотеку: https://github.com/jakearchibald/es6-promise
3. Установим модуль проверки совместимости плагина и браузера:
   https://www.npmjs.com/package/rutoken-browser-check
   Он дает возможность локализовать проблемы при инициализации.
   У данного скрипта есть зависимость: https://www.npmjs.com/package/bowser

4. Изменим логику инициализации работы плагина. Для этого:
   - вынесем логику в отдельный файл boot.js;
   - подключим его на всех страницах, где это необходимо;
   - удалим функцию token_refresh из подписи на onload странице.
   В общем случае, плагин получается в финальном resolve цепочки промисов. Мы передаем его конструктору нашей обертки вместе с методом получения PIN-кода.
   Этот метод нужен, чтобы эмулировать отображение окна со списком логинов и обработать выбор корректного id. Метод должен возвращать промис с id устройства и PIN-кодом, введенным пользователем.
   Все ошибки инициализации прилетают в reject промиса. Ошибки от плагина прилетают другие, поэтому то место, где они обрабатывались необходимо изменить.
5. Используем обертку над плагином rtwToCrypto.js. Код обертки:
   https://github.com/blade-runner/rutokenweb-to-plugin
   Обертка подменяет необходимые методы таким образом, что все старые сигнатуры вашего кода сохраняются и ваш код работает, но теперь уже с другим плагином и всеми устройствами семейства Рутокен ЭЦП.

Список методов, переопределенных оберткой:

- rtwIsTokenPresentAndOK;
- rtwGetNumberOfContainers;
- rtwGetContainerName;
- rtwSign;
- rtwGenKeyPair.

Если в вашем коде есть использование других функций и какой-то код не срабатывает через обертку напишите на hotline@rutoken.ru или, если вы знаете, что нужно сделать, сделайте pull request в проект обертки.