Adyen Clientside Encryption on iOS Demo

Introduction

This app is intended as a demo to show how to submit credit-card data to the Adyen backend in a secure way from an iOS app. It may be used as a starting-point to write your own iOS app that includes Adyen payment functionality.

About the app

The functionality of the app is simple. It consists of a single screen. This screen allows the user to enter creditcard data (number, CVC, cardholder name, expiry date) and an amount and a currency.

After pressing `Submit`, the app will encrypt the credit-card data, and submit it, along with the amount and currency, to the Adyen backend. The result of the payment is shown at the bottom of the screen.

For configuration, the app supplies a settings bundle. This means that its settings can be edited in the standard iOS settings app. The parameters that can be set are: Merchant account, username, password, and public key. The values for these parameters are supplied by Adyen.

Writing your own

When writing your own app, copy the Objective-C files from the `Encryption` folder, and the header-files and static library (`libcrypto.a`) from the `OpenSSL` folder to your own project. The files in the `App` folder exist only for demo-purposes, you don't have to copy them.

When your app wants to submit credit-card data to the Adyen backend, first create an instance of the `PaymentRequest` class and populate its fields. The `card` property must be filled with a `Card` object, whose properties should also all be filled.

Next, set the `delegate` property of the `PaymentRequest` instance to some object implementing the `PaymentRequestDelegate` protocol. This delegate will be informed about the outcome of the payment.

Finally, call `[PaymentRequest submitWithMerchantAccount:username:password:publicKey:]`. The four parameters, `merchantAccount`, `username`, `password`, and `publicKey` are specific to you as a merchant, and are supplied by Adyen.

After this, the `PaymentRequest` object encrypts the credit-card data and sends all data to the Adyen backend. At the end, one of the methods on your delegate is called to inform you of the outcome.

Log output

By default, the `Encrypter` class will write debug messages to the console while performing encryption. These messages are of great help when solving problems, and Adyen will ask for them when asked for support.

However, in a production app, you will want to disable log output. To do this, open the file `Logging.h` and change the line that reads:

```
#define ENABLE_LOG 1
```

into:

```
#define ENABLE_LOG 0
```

After a recompile, no more log messages will be printed.