OneSpan

Be bold. Be secure.

PSD2: WHICH STRONG AUTHENTICATION AND TRANSACTION MONITORING SOLUTIONS COMPLY?

WHITE PAPER

TABLE OF CONTENTS

About the Author

Frederik Mennes is Senior Manager Market & Security Strategy at OneSpan. He regularly advises financial institutions about the security and compliance aspects of online and mobile banking applications.

You can reach him at frederik.mennes@OneSpan.com.

INTRODUCTION

On March 13th, the European Banking Authority (EBA) published its long-awaited final draft Regulatory Technical Standards (RTS) on Strong Customer Authentication (SCA) and Common and Secure Communication (CSC) under the revised Payment Services Directive (PSD2). These requirements will become applicable on September 14, 2019.

In this whitepaper we analyze which strong authentication and transaction risk analysis solutions can comply with the requirements about SCA in the final RTS. We first provide some background about the history of the final RTS, and then discuss common authentication solutions that are used by many online banking and mobile banking applications today. Subsequently we present and discuss the most important requirements from the final draft RTS, and point out changes to the previous version of the RTS. Finally we explain which authentication solutions are most likely to meet the requirements of the final RTS.

Background

In recent years, the security of electronic payments has more and more become the subject of supranational guidelines and regulations in Europe. The initiatives for these guidelines and regulations originated from the European financial regulators as well as the European Commission.

In 2013, the SecuRe Pay forum of the European Central Bank (ECB) published its Recommendations for the security of Internet payments, as well as its (draft) Recommendations for the security of mobile payments. The former set of recommendations was republished by the European Banking Authority as the Final guidelines on the security of Internet payments. These EBA guidelines are in effect since August 1st 2015 in most member states of the European Union. The latter set of recommendations was not further developed by the EBA.

In November 2015 the Council of the European Union adopted the Revised Payment Services Directive, also known as PSD2. One of the key elements of PSD2 consists of the need to perform strong authentication of users of electronic payment services. Article 98 of PSD2 tasks the European Banking Authority with the development of more detailed requirements regarding SCA. In line with this mandate, the EBA issued its proposal for the draft Regulatory Technical Standards on SCA on August 12th 2016. This proposal was heavily influenced by the above-mentioned EBA guidelines on the security of Internet payments. Following the feedback from a large number of respondents from the payments industry, the EBA published its final draft RTS on February 23rd 2017, about 1.5 months after the planned deadline.

The final RTS was published in the Official Journal of the European Union on March 13, 2018, and becomes applicable 18 months after its entry into force, namely on September 14, 2018.

STRONG AUTHENTICATION SOLUTIONS

TWO-DEVICE-AUTHENTICATION

MY Bank
$
324760

Banking device

Authentication devices

TWO-APP-AUTHENTICATION

MY Bank
AUTHENTICATION

Banking App & Authentication App

ONE-APP-AUTHENTICATION

MY Bank
AUTHENTICATION

Banking + Authentication App

Common strong authentication solutions classification

Many online and mobile banking applications already use (strong) authentication solutions today. We divide these solutions into following categories, based on the approach in PSD2[1].

Two-device-authentication (2da)

In this case the user has two independent devices: one device to access the banking website or app, and another device to authenticate himself or a payment. The first device, which we refer to as the banking device, is typically a desktop PC, laptop, or a mobile device (e.g. phone, tablet) that runs a mobile banking app. The second device, which we call the authentication device, is usually a hardware authentication token, a combination of a smart card and smart card reader, or a dedicated app on a mobile device. The authentication device generates one-time passwords (OTPs) over transaction data.

In order to perform a payment, the user first logs on to the banking app and enters the details of the payment (e.g. beneficiary account number, amount of money). The transaction data is then transferred to the authentication device. This can happen in many ways, depending on the capabilities of the device: the user might scan a QR-code representing the transaction using the hardware token, card reader or mobile device. Alternatively the user might manually enter the transaction details into the hardware token, card reader or mobile device. Finally both devices might be connected to each other via USB or Bluetooth. The user verifies and confirms the transaction data once they are present on the authentication device. The authentication device then generates a one-time password over the transaction data, which is transferred back to the banking device. This latter transfer can again be performed in different ways, depending on the capabilities of the device. It is common that the user manually enters the OTP into the banking device.

Two-app-authentication (2aa)

In contrast to 2da, this approach does not rely on two different devices, but on two different apps running on the same mobile device. The apps interact via so-called app-to-app communication. We refer to these apps as the banking app and authentication app respectively.

When a user wants to make a payment, he opens the banking app and enters the transaction data. When the user has submitted the transaction, the banking app opens the authentication app. After verification and confirmation of the transaction data by the user, the authentication app generates an OTP linked to the transaction data and sends it back to the banking app, which submits it to the banking server for verification. Other flows than the one just described exist as well, but the precise flow is not relevant for the remainder of this text.

One-app-authentication (1aa)

In this case the user not only uses a single device, but also a single app to initiate and authenticate transactions. The user does not employ a separate authentication device or app.

Out-of-band authentication (oob)

The above categories can be combined with an out-of-band approach, whereby the OTP is not generated by the token or app, but generated by the bank and delivered via a separate channel (e.g. SMS, e-mail) to the user's device.

In case of 2da, the user's authentication device could then be a phone where he receives an SMS message. In case of 2aa or 1aa, the apps could reside on a mobile phone.

Requirements for strong customer authentication

Basic requirements

Article 97 of PSD2 requires Payment Service Providers to authenticate a user when he accesses an online payment account, when he initiates an electronic payment transaction, or when he carries out any action through a remote channel that may imply a risk of payment fraud (or other abuses).

A basic definition of "strong customer authentication" is present in article 4(30) of PSD2. It states that authentication has to be based on the use of two or more possible authentication elements, categorized as knowledge (i.e. something only the users knows, such as a password), possession (i.e. something only the user has, such as a token) or inherence (i.e. something only the user is, such as a fingerprint or face scan). Furthermore the authentication factors must be independent from each other. The SCA procedure constructed from these authentication elements must generate a one-time authentication code.

The categories of authentication solutions discussed above can meet these basic requirements:

- **2da.** The possession element is the authentication device. The knowledge or inherence element is entered onto the authentication device or banking device.
- **2aa and 1aa.** The possession element is the mobile device, which stores cryptographic keys to generate authentication codes. The knowledge or inherence element is entered onto the mobile device.
- **oob.** The possession element is the mobile phone where the user receives authentication codes. The knowledge or inherence element is entered onto the banking device (for 2da) or mobile device (for 2aa and 1aa).

Dynamic linking

In case of a payment transaction, the authentication code must be dynamically linked to the amount and the payee, meaning that this code will change if either the amount or the payee is changed during the transaction.

The requirements regarding dynamic linking have significantly changed compared to the previous version of the draft RTS. Previously the draft RTS specified that the apps or devices that are used to initiate and authenticate a payment should be segregated. This requirement ruled out the "1aa" approach above. However the EBA removed this requirement in the final draft RTS because it was "confusing" and also stated that "the independence of the elements constituting SCA does not require different devices and can be hosted on the same device". In our opinion this means that the "1aa" and "2aa" approach can be used for dynamic linking under the final RTS.

Requirement 2 of Article 5 is a very broad requirement and states that payment transaction data needs to be protected throughout all phases of authentication:

> *2. […] payment service providers shall adopt security measures which ensure the confidentiality, authenticity and integrity of each of the following:*
>
> *a. the amount of the transaction and the payee through all phases of authentication.*
> *b. the information displayed to the payer through all phases of authentication including generation, transmission and use of the authentication code.*

Requirement 2b likely aims to prevent social engineering attacks whereby a user unwittingly confirms a payment transaction after the amount and payee have been altered by a fraudster. Indeed, there is a plethora of malicious software on multi-purpose devices (such as desktop PCs and mobile devices) that is capable of altering the payment transaction data that is displayed to the payer. On mobile devices malicious software often uses so-called "overlay" windows to achieve this goal.

Specific security controls are therefore required in order to comply with Article 2(2):

- In the 2da category, a hardware token with the capability to scan a visual code (e.g. QR code or Cronto code) that contains the encrypted payment information, and to subsequently show the payment information to the payer, most likely meets the requirements. This approach is usually referred to as "What You See Is What You Sign" (WYSIWYS).
- In the 2da category, a solution consisting of an authentication app running on a mobile device that receives the payment information via a secure, encrypted channel and that displays payment information in the app to the payee most likely meets the requirements as well. However in this case the mobile app needs to be additionally protected in order to meet the independence requirements, as discussed below.
- A third option consists of following the "2aa" or "1aa" approach. In this case mobile apps need to exchange payment information via a secure channel, and also clearly show the payment information to the user. Additionally the authentication app (in case of 2aa) or the banking/auth app (in case of 1aa) should be equipped with security software that can detect malicious software and prevent it from interfering with a payment transaction. In general it is very hard for security software to provide strong guarantees that it can stop malicious software.
- In case of "oob", Requirement 2 implies that the SMS message should contain the payment information. The requirement to protect the confidentiality of the payment information could be interpreted as a need to encrypt the payment information in the SMS message. However clarification about this is required from the EBA and national competent authorities.

Protecting the possession element against cloning

Article 7 defines requirements related to the possession element, which are particularly relevant for mobile devices. The article says that:

> *"the use by the payer of elements categorized as possession shall be subject to measures designed to prevent replication of the elements".*

Mobile apps are very easy to clone if they do not contain countermeasures. Hence this requirement mandates the use of dedicated cloning countermeasures in apps. A basic countermeasure consists of including device-specific data, such as the device's IMEI or ID, into the OTP generation. A stronger countermeasure encrypts data used by the app using a cryptographic key stored inside the device's Secure Element. Another option consists of using a password or PIN to encrypt the data that is used by the app to generate an OTP.

Independence of authentication elements

Article 9 defines a number of requirements related to the independence of the various authentication elements, which is again very relevant in the context of mobile devices. We consider following requirements from the final RTS:

> *2. Where any of the elements of strong customer authentication or the authentication code is used through a multi-purpose device including mobiles phones and tablets, payment service providers shall adopt security measures to mitigate the risk resulting from the multi-purpose device being compromised.*
>
> *3. For the purposes of paragraph 2, the mitigating measures shall include, but not be limited to:*
>
> > *a. the use of separated secure execution environments through the software installed inside the multi-purpose device;*
> >
> > *b. mechanisms to ensure that the software or device has not been altered by the payer or by a third party or mechanisms to mitigate the consequences of such alteration where this has taken place.*

Requirement 3a states that software-based secure execution environments can be used. This is a clear change from the previous draft RTS, where the requirement used the wording "trusted execution environments", and which could have suggested a need for execution environments based on hardware. Hence, common mobile operating systems (e.g. Android, iOS) most likely meet the requirement of separated trusted execution environments via their sandboxing techniques.

However these sandboxing mechanisms function correctly only as long as the device is not jailbroken or rooted. The sandboxing techniques of these mobile operating systems can be further augmented using so-called "runtime application self-protection" (RASP) technology. This type of technology allows detecting whether an app runs inside an emulator or virtual machine instead of on a regular mobile device.

Requirement 3b mandates Payment Service Providers to use security controls to detect, prevent and respond to the alteration of mobile apps and devices. Again, RASP technology for mobile apps provides such security controls. More specifically, RASP technology usually provides security services to protect the confidentiality and integrity of mobile apps, to detect whether a device is rooted, to detect, whether an app runs inside a debugger or emulator, etc.

Transaction risk analysis

The draft RTS mandates the usage of transaction risk analysis (TRA) to prevent, detect and block fraudulent payments. Article 2(3) stipulates that transaction risk assessment mechanisms should be based on elements such as the amount of the payment, known fraud scenarios, signs of malware infection in the payment session, etc.

Article 18 represents a major change to the final draft RTS. In the previous version of the draft RTS, transaction risk analysis could not be used to exempt a payment from SCA. The final draft and final RTS allow payments, which are rated as low-risk by the payment service provider, to be exempted from SCA.

This exemption is however subject to a number of conditions, including:

- Transaction risk assessment should take into account additional elements such as the payment patterns of the payer, the location of the payer and payee at the time of the payment transaction, characteristics of the payer's device or software application, etc.
- The fraud rate of the payer's payment service provider determines the maximum payment amount that can be exempted from SCA. The lower the fraud rate of the payer's payment service provider, the higher the payment amount that can be exempted from SCA. Article 18(2) defines certain thresholds for fraud rates that determine the payment amount that can be exempted. In any case the maximum payment amount that can be exempted from SCA is € 500; all payments above € 500 require SCA.

Summary

Chapter 3 of the final RTS defines a number of other exemptions from SCA, besides TRA. Table 1 and Table 2 summarize when SCA must be used and when exemptions are allowed, and this for access to payment accounts and payments respectively. In these tables, "1FA" refers to authentication using a single factor, such as a password.

Table 1: Summary of requirements regarding SCA for access to payment accounts

ACCESS TO PAYMENT ACCOUNT	1FA	SCA	TRA
Balance inquiry (after first inquiry)	✓	✓	✓
Consultation of payment history of past 90 days (after first inquiry and less than 90 days since last time SCA was performed)	✓	✓	✓
Other	✗	✓	✗

Table 2: Summary of requirements regarding SCA for payments

PAYMENTS	1FA	DYNAMIC LINKING	TRA
Payment to trusted beneficiaries	✓	✓	✓
Recurring payments with same amount and payee	✓	✓	✓
Payment below € 30	✓	✓	✓
Payment in range € 30 to € 500	✗	✓	✓
Payment above € 500	✗	✓	✗

Payments below € 30 can only be exempted from SCA if the cumulative amount, or the number, of previous remote electronic payment transactions initiated by the payer since the last application of strong customer authentication does not, respectively, exceed EUR 100 or 5 consecutive individual remote electronic payment transactions.

PAYMENT FRAUD

Liability for payment fraud

Under PSD2 the SCA procedure is the responsibility of the Account Servicing PSP (ASPSP). Payment Initiation Service Providers (PISPs) must use the credentials issued by the ASPSP, unless there is a prior contractual agreement in place between the PISP and the ASPSP that the former's credentials may be used.

Article 73(1) of PSD2 states that the payer's payment service provider must refund the payer if any unauthorised payments were performed on behalf of the payer, unless the payment service provider suspects fraud:

> *[...] in the case of an unauthorised payment transaction, the payer's payment service provider refunds the payer the amount of the unauthorised payment transaction immediately, and in any event no later than by the end of the following business day, after noting or being notified of the transaction, except where the payer's payment service provider has reasonable grounds for suspecting fraud and communicates those grounds to the relevant national authority in writing.*

If payments are initiated by a PISP, article 73(2) clarifies that the ASPSP can transfer liability to the PISP:

> *Where the payment transaction is initiated through a payment initiation service provider, the account servicing payment service provider shall refund immediately, and in any event no later than by the end of the following business day the amount of the unauthorised payment transaction and, where applicable, restore the debited payment account to the state in which it would have been had the unauthorised payment transaction not taken place. If the payment initiation service provider is liable for the unauthorised payment transaction, it shall immediately compensate the account servicing payment service provider at its request [...].*

The payer always bears all the losses related to unauthorised payments if he acted fraudulently. He might also bear some losses if the unauthorised payment resulted from the use of a loss or stolen payment instrument.

These articles from PSD2 imply that the payer's payment service provider is liable for unauthorised payments even if he provides strong customer authentication in line with the RTS, unless the payer has acted fraudulently. This incentivizes payment service providers to not simply choose the SCA procedure that meets the requirements of the RTS, but to select an SCA procedure based on the payment risk.

STRONG AUTHENTICATION

Compliance of common authentication solutions

We now evaluate some of the most common authentication solutions against the above requirements from the final RTS. We differentiate between two scenarios: access to payment accounts, and payment authentication.

Access to payment accounts

Table 3 lists the authentication solutions, grouped according to the various categories (2da, 2aa, 1aa and oob), and indicates which solutions comply with the requirements for access to payment accounts. For authentication solutions in the “2da” category, only the authentication device is described. The banking device can be a laptop, desktop, mobile device, etc. as long as it is separate from the authentication device.

The solutions in the category “2da” generally comply with all requirements. This includes solutions based on hardware tokens, smart cards and smart card readers, and mobile OTP generators. Solutions based on the usage of a hardware token or smart card meet the anti-cloning and independence requirements since hardware-based protection is deemed to be sufficient. Solutions based on mobile OTP generators meet the requirements if the mobile app is protected using cloning countermeasures and RASP technology, such as root/jailbreak detection mechanisms.

The solutions in the “2aa” and “1aa” categories also comply with all requirements, but again under the condition that the mobile apps are protected using cloning countermeasures and RASP technology. Solutions in the “oob” category also generally comply.

Payment authentication

Table 4 lists the authentication solutions and indicates which solutions comply with the requirements for access to payment accounts.

The conclusions for payment authentication are largely the same as for access to payment accounts. It is noteworthy that solutions based on one-button hardware tokens, which do not support dynamic linking of transaction data, can only be used if the payment is exempted from SCA (e.g. because it is a low-risk payment).

Solutions in the “oob” category need to make sure that the SMS message contains payment information. In order to comply with the confidentiality requirement regarding Dynamic Linking, payment service providers should consider encrypting the payment information in the SMS message.

It is expected that payment service providers will select strong authentication solutions from the list of compliant solutions in line with the risk of the payments that they process. In other words payment service providers that process high-value transactions are likely to opt for more secure solutions. This is the case because payment service providers are liable for unauthorised payments even if they provide strong customer authentication, unless the payer acted fraudulently.

Table 3: Compliance of common authentication solutions with SCA requirements for access to payment accounts

	AUTHENTICATION SOLUTION	AUTHENTICATION ELEMENTS	REPLICATION PROTECTION	INDEPENDENCE	OVERALL COMPLIANCE
	2da one-button hardware token and password	Possession: hardware token Knowledge: password	✓ based on hardware	✓ token and password are independent	☺ OK
	2da PIN-protected hardware token with keypad	Possession: hardware token Knowledge: PIN		✓ based on hardware	☺ OK
	2da PIN-protected smart card	Possession: smart card Knowledge: PIN			
	2da PIN-protected hardware token with QR-code scan	Possession: hardware token Knowledge: PIN			
	2da mobile OTP generator with PIN on mobile app	Possession: mobile device Knowledge: PIN, fingerprint, etc.	✓ if cloning countermeasures used	✓ if root detection and RASP used	☺ Conditional OK
	2da mobile OTP generator with PIN on banking application	Possession: mobile device Knowledge: password			
	2aa	Possession: mobile device Knowledge: PIN, fingerprint, etc.			
	1aa	Possession: mobile device Knowledge: PIN, fingerprint, etc.			
	oob	Possession: mobile phone Knowledge: PIN, fingerprint, etc.	✓ based on SIM security	✓	☺ OK

Table 4: Compliance of common authentication solutions with SCA requirements for payments

	AUTHENTICATION SOLUTION	AUTHENTICATION ELEMENTS	DYNAMIC LINKING	REPLICATION PROTECTION	INDEPENDENCE	OVERALL COMPLIANCE
	2da one-button hardware token and password	Possession: hardware token Knowledge: password	✗	✓ based on hardware	✓ token and password are independent	☹ Except for low-risk payments
	2da PIN-protected hardware token with keypad	Possession: hardware token Knowledge: PIN	✓ if secure channel used		✓ based on hardware	☺ OK
	2da PIN-protected smart card	Possession: smart card Knowledge: PIN				
	2da PIN-protected hardware token with QR-code scan	Possession: hardware token Knowledge: PIN				
	2da mobile OTP generator with PIN on mobile app	Possession: mobile device Knowledge: PIN, fingerprint, etc.		✓ if cloning countermeasures used	✓ if root detection and RASP used	☺ Conditional OK
	2da mobile OTP generator with PIN on banking application	Possession: mobile device Knowledge: password				
	2aa	Possession: mobile device Knowledge: PIN, fingerprint, etc.				
	1aa	Possession: mobile device Knowledge: PIN, fingerprint, etc.				
	oob	Possession: mobile phone Knowledge: PIN, fingerprint, etc.	✓ if message contains payment info	✓ based on SIM security	✓ if RASP used if mobile apps involved	☺ Conditional OK

OneSpan's Solution Suite for PSD2 Compliance

OneSpan's Multi-factor Authentication Solution – Enables quick compliance through strong, customizable and easy-to-deploy, authentication options

- Complete multi-factor authentication solutions, offering fast, convenient & secure authentication for all users
- Multi-modal biometrics framework with next generation behavioural and contextual authentication options, driving stronger security and the best user experience
- Extensive hardware and software solutions to ensure full compliance across nearly every application

OneSpan's Secure Channel Technology – Ensures confidentiality, integrity and authenticity of every payment transaction

- Secure transaction data signing technology links the authentication code to the amount of the transaction,
  payee information and other key details
- Unique end-to-end protection of communication between user and the Payment Service Provider
- Unique visual validation feature provides users a quick and convenient check of each transaction
- Omni-channel user experience leveraging innovative transaction signing technology or secure push notifications

OneSpan's Mobile Application Security – Mitigates malicious attacks on mobile apps and reduces exposure to related fraud

- Provides extensive app protection features far beyond what's available through the OS or typical mobile app development security practices
- Wraps around the mobile app, enabling it to operate safely even on infected devices
- Proactively manages the real threat of sophisticated malware, preventing foreign code from altering the app, detecting real-time attacks and reporting malicious attacks to fraud management platforms

OneSpan's Secure Provisioning Tools – Dramatically reduces the risks of unauthorized use on authentication platforms

- Strong security technology for deployment, provisioning and activation of authentication solutions, preventing credential theft attacks
- Flexible and streamlined provisioning process across on- and-offline channels including: web, app, branch, IVR and ATM

OneSpan's Fraud Prevention Solution – Enables compliance with strict regulations, satisfying transaction monitoring and risk analysis requirements

- Real-time detection of sophisticated fraud drives down exposure and boosts the top line
- Processes 200+ data points to detect abnormal user behavior, suspicious transactions, and atypical navigation in the payment application
- Accurate risk scoring and threat mitigation across all common devices (e.g. mobile phones and tablets)
- Operates invisible to end users, mitigating fraud while providing the best possible user experience

For more information on PSD2 compliance,

visit OneSpan.com/psd2

[1] Vincent Haupert and Tilo Müller, On App-based Matrix Code Authentication in Online Banking, page 5, www1.cs.fau.de/content/app-based-matrix-code-authentication-online-banking

OneSpan

OneSpan enables financial institutions and other organizations to succeed by making bold advances in their digital transformation. We do this by establishing trust in people's identities, the devices they use, and the transactions that shape their lives. We believe that this is the foundation of enhanced business enablement and growth. More than 10,000 customers, including over half of the top 100 global banks, rely on OneSpan solutions to protect their most important relationships and business processes. From digital onboarding to fraud mitigation to workflow management, OneSpan's unified, open platform reduces costs, accelerates customer acquisition, and increases customer satisfaction.

CONTACT US

For more information:
info@OneSpan.com
www.OneSpan.com

Last Update August 2019.