Device Centric Cloud Signature Solution under eIDAS Regulation

Petru SCURTU and Victor Valeriu PATRICIU

***Abstract*—Under the new eIDAS Regulation qualified electronic signatures are equivalent, from a legal stand, to handwritten signature. Traditional signature solutions make use of cryptographic materials stored in secure devices in possession of clients, while remote or cloud signatures solutions rely on a trusted service provider which manages the private keys and produces signatures in a remote manner. This shifts the weight of dealing with the keys off clients and moves this duty to a specialist in the field. As opposed to a classical Qualified Electronic Signature, a cloud-based solution has to solve a set of specific problems: the integrity of the data submitted must be ensured, the user's intent of creating a digital signature must be demonstrated and the owner of the cryptographic key must be the only entity capable of using this cryptographic material. A device centric solution based on a simple mobile device application is proposed. This solution leverages the advancements in device technology such as the inclusion of Trusted Execution Environments (TEEs) on end user terminals. Furthermore, in comparison to similar solutions, the costs have been reduced by replacing cryptographic solutions based on SMS messages or cryptographic tokens with a device native implementation.**

***Index Terms*—Cloud Signatures, Device centric, eIDAS, TEE.**

I. INTRODUCTION

Access to Internet is one of the most fundamental needs of the modern society because a multitude of services are based on a secure connection to the World Wide Web, such as: a variety of websites used for social or economic purposes, banking services, medical services etc. Unfortunately, there is a downside to the extended use of the Internet, in the form of malwares and hacker attacks. Some information is highly sensitive and, as a result, a need for protection mechanism has evolved. Cryptographic mechanisms are used in order to insure the confidentiality of sensible data.

After taking in consideration the vulnerabilities of the above-mentioned services, the need of protection through cryptographic means becomes evident. A popular and efficient scheme of providing access to sensible, encrypted data is through a 2-step authentication system.

The vast majority of services are designed and tailored to the need of end users that are using devices such as laptops or personal computers. In recent times, mobile devices have become a major consumer of online services.

One very popular and highly used functionality is the creation of digital signatures. Due to the very sensible nature of the information used in this process, highly secure mechanism have been created. Unfortunately, these mechanisms have been tailored to the needs of fixed devices.

End users who wish to access digital signature creation services using a mobile device face a series of challenges, mostly in the form of smart card and tokens incompatibility issues with mobile devices which do not offer an USB port.

eIDAS (electronic Identification, Authentication and trust Services) is an EU regulation for electronic recognizable proof and trust services for electronic exchanges in the European Single Market. It was set up in EU Regulation 910/2014[1] of 23 July 2014 on electronic distinguishing identification and repeals directive 1999/93/EC from 13 December 1999.

Under the EU eIDAS Regulation qualified electronic signatures are, from legal point of view, equivalent to handwritten signatures. Remote qualified signing or cloud signing implies the fact that both the signature creation devices and the signature cryptographic keys are stored remotely.

The Cloud Signature Consortium (CSC) [2] is a nonprofit conglomerate formed by organizations from the economic and academy fields whose aim is better "solutions, architectures and protocols for Cloud-based Digital Signatures, also defined as "remote" Electronic Signatures".

The CSC has published a standard with the purpose of providing a common framework for cloud based digital signature solutions. This standard encourages the use of second factor authenticators such as codes received via SMS OTP [3], codes generated by hardware tokens or remote authentication using third party services.

A number of service providers have already integrated the proposed framework into their solutions, but these solutions often require additional costs or additional hardware for increased security.

A device centric solution implies that an end user will be able to access and consume a remote digital signature service via a mobile device without specialized hardware or additional costs. The advancements in mobile device technology permit the use of highly secure cryptographic solutions such as Trusted Execution Environments [4].

Promoting the end user devices as the central authentication gateway not only reduces costs but also is in accordance with the latest NIST standards for authentication [5].

Although having a device centric architecture follows the latest practices and can improve both security and costs, it also has some drawbacks. Firstly, the device becomes a single point of failure in the architecture and as a result device loss or failure entails the need for a revocation and recovery mechanism. Secondly, devices are in some scenarios vulnerable to tampering or eavesdropping.

P. SCURTU is with the Military Technical Academy "Ferdinand I", Bucharest, Romania (e-mail: petru.scurtu@mta.ro).

V. V. PATRICIU is with the Military Technical Academy "Ferdinand I", Bucharest, Romania (e-mail: victor.patriciu@mta.ro).

Digital Object Identifier 10.32754/JMT.2020.2.08

In summary, the proposed architecture makes the following contributions:

1. A device centric implementation which does not require external hardware or third-party services.
2. Leverage the advancement in device technology for improved security, such as storing the cryptographic material in a TEE.

II. Terminology and Background

A. Abbreviations and terminology

API: application programming interface
HSM: hardware security module
AdES: advanced electronic signature
QES: qualified electronic signature
QSCD: qualified signature creation device
CC: common criteria
PP: protection profile
SCAL1: sole control assurance level 1
SCAL2: sole control assurance level 2
TEE: trusted execution environment

Remote signature creation device (RSCD): a specialized device, usually an HSM, outside the control of the end user, which will handle the creation of digital signatures in his name.

Remote signing service provider (RSSP): a service provider specialized in remote signature services. It exposes a service to end users which are capable of signing documents on remote signature creation device.

Signature activation data (SAD): a special set of information, which usually contains some sort of signature of the end user, used by the remote signing service provider to guarantee the intent of signing of the end user.

Signature activation module (SAM): a specialized software component trusted with validating the signature activation data and forwarding valid requests to the signature creation device.

User device: this is the main gateway to the architecture. It is assumed that the device is able to utilize recent advances in the field of Trusted Execution Environments which will enable it to securely store cryptographic credentials within its software stack.

B. Background

The eIDAS regulation presents the idea of remote signing/server marking instead of local signing. While traditional signature solutions make use of cryptographic materials stored in secure devices in possession of clients, remote or cloud signatures solutions rely on a trusted service provider which manages the private keys and produces signatures in a remote manner. This shifts the weight of dealing with their own keys from clients to a specialist in the field.

The eIDAS legislation states that for issuing a qualified signature, a service provider must use a Qualified Signature Creation Devices (QSCD). As a novelty, cloud signing makes an addition to QSCD, in the form of a component named Signature Activation Module (SAM). The SAM is an addition to the HSM tampered resistant environment. As the name implies it is responsible for activating the HSM module in order to create a qualified signature. This process is necessary as a result of the remote nature of the signing process which requires the intent of creating a signature to be demonstrated [9]. Additionally, the hardware must be Common Criteria (CC) certified based on the eIDAS Protection Profile (PP) EN 419 241-2[6][7] "QSCD for Server Signing" to meet the requirements of such a QSCD.

The SAM will interact with aHSM that is CC-certified based on the eIDAS PP EN 419 221-5[8] "Cryptographic Module for Trust Services". The eIDAS Protection Profiles EN 419 221-5 now available provides a common certification framework.

In contrast to digital signature solutions, the architecture of a remote signing solution in compliance with the eIDAS regulation is more complex. The complexity is due in part to the remote nature of the signing process and implies the addition of a number of communication components which must solve the novel problems raised by remote signing.

Similar to a traditional signature solution, the core of the architecture is the HSM module responsible with handling the cryptographic secret keys. [10, 11] This component must be a Qualified Signature or Seal Creation Devices (QSCD), a tamper protected appliance which has been Common Criteria EAL 4+ certified against EN 419 241-2 [6,7] Protection Profile (PP).

The Signature Activation Module communicates with a component named Signer Interaction Component (SIC). The role of this component is to provide an interface, for the user, capable of communicating information related to signatures requests. The Signer Interaction Component may be capable of authenticating the end user or may delegate the authentication to a third party. A successfully authenticated user, who submits a legitimate request, will have his request forwarded to the SAM which is responsible with activating the HSM.

The Signature Activation Protocol (SAP) is defined as the interaction process between the Signer Interaction Component which is responsible for authenticating users and forwarding the signatures requests and the Signature Activation Module in the HSM module. The finality of the Signature Activation Protocol is either activating the HSM and generating a digital signature or rejecting the signature request.

The Signature Activation Protocol (SAP) describes and is responsible for the way the Signature Activation Data (SAD) will be handled by the Trust Service Provider (TSP). The signature provider must ensure the integrity of the data submitted to signing, must demonstrate and be able to identify the intent of creating a digital signature of the end user, must authenticate the end user and guarantee sole access to the signing key. The structures contained in the Signature Activation Data (SAD) and the way these are handled must ensure that these security goals are achieved.

Due to the inherit complexity of the described architecture it is not feasible for the end user to use only a client, such as a web browser. In order to guarantee sole access to the cryptographic material and to demonstrate the intent of signature we cannot on traditional authentication mechanisms. In general, multiple factor authentication solutions are employed in order to satisfy these security considerations. The downside of such solutions relies in deployment costs when using external authenticators, such as cryptographic tokens or operating cost generated, for example, by SMS providers. As such, the implementation will take into consideration mobile devices as a means to

 Digital Object Identifier 10.32754/JMT.2020.2.08

replace the cryptographic tokens in 2 factor authentications mechanism. The aforementioned mobile devices will carry the very important role of authenticating the end user and creation of the Signature Activation Data.

In addition to cost reduction, a mobile centric implementation will leverage strong authenticators, provided by biometric mechanisms, and will also enhance the security of the overall solution by making use of tamper resistant environments available on newer mobile devices.

III. Benefits, Challenges and Problems of Remote Signatures

Qualified Electronic Signatures (QES) are defined as a variant of Advanced Electronic Signatures produced by means of a qualified certificate. In the creation process a specialized highly secure hardware is used, namely a Qualified Electronic Signature Creation Device (QSCD). The resulting QES maintains all the security features from an Advanced Electronic Signature (AdES), namely:

- It will uniquely link and identify the creator of the signature.
- The creator of the signature has sole access to the signing keys.
- A data integrity feature proving that the data has not been altered after the signature was created.
- Signatures that have been tampered with will be invalidated.

As opposed to a classical Qualified Electronic Signature, a cloud-based solution that offers such signatures has to solve a set of newel and specific problems [14]:

- The signature provider must ensure the integrity of the data submitted to signing. This is increasingly difficult as the document has to travel probably through a network to reach the remote provider.
- The signature provider must demonstrate and be able to identify the intent of creating a digital signature of the end user. Simply accessing the services provided by the signature solution does not guarantee the signing intent.
- The signature provider must ensure that the valid user which is the owner of the cryptographic key is the only entity capable of using this cryptographic material
- Finally, the provider must provide an authentication framework for users. As opposed to traditional signing solutions the challenges arise from the remote nature of the accessed cryptographic material. As a result, problems related to remote authentication must be taken into consideration.

As highlighted, creating a Qualified Electronic Signature by using a remote cloud-based solution poses a set of unique challenges. On the other hand, implementing a device centric cloud signature solution could bring a series of unparallel benefits. The most important benefit is a lack of specialized hardware for end users as there is no need to deploy specialized hardware devices such as smartcards or cryptographic tokens. This also has the added benefit of reducing implementation costs.

Another advantage of a device centric implementation is the added security provided by using tamper resistant environments deployed on newer mobile devices. As a consequence, end users could also benefit from the added security and the simplified key management of a remote signing solution.

IV. Architecture Overview

The architecture aims to offer a remote signature solution, where remote signature is defined as “the creation of remote electronic signatures, where the electronic signature creation environment is managed by a trust service provider on behalf of the signatory [1].

As a result, cases in which the cryptographic material used for creating the digital signature is stored in the user device are not covered. Although still relevant, cases in which the signature is created remotely by the means of a cryptographic key stored locally, do not fit in the core definition of “remote signature”.

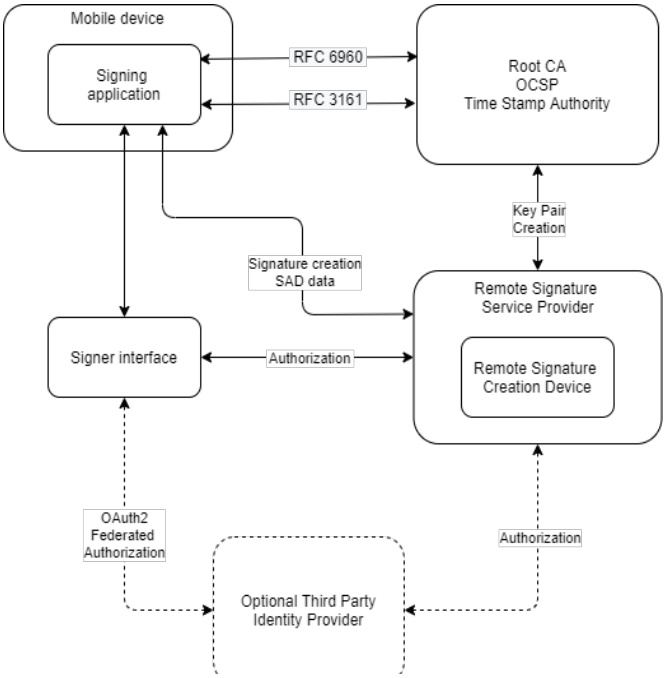

Figure 1. Remote Signature Architecture [2]

A remote signature architecture typically consists of 4 main components: [12,13]

- The user device: in a traditional implementation this is represented by a personal computer. Usually, for enhanced security, a secondary authentication factor is included in the form of a mobile devices or a cryptographic token.
- The signer interface: the role of this component is to provide an interface between the user device and the remote signature service provider. Additionally, it handles user authentication and authorization. Authentication can be done by the signer interface or be delegated to a trusted third-party identity provider.
- The remote signature service provider: it handles requests for creating digital signature on behalf of end users. It uses a qualified signature creation device which stores the cryptographic material for users.

- A PKI infrastructure: the remote signature service provider relies on a classic PKI infrastructure which uses X509 digital certificates consisting of certification authorities, time stamp authorities and revocation lists.

A device centric implementation replaces the traditional personal computer with a mobile device. Additionally, second factor authentication is achieved by using device specific functions such as biometric sensors.

Another advantage of using a mobile device as the central gateway to the remote architecture is enhanced security. By leveraging the latest developments, such as Trusted Execution Environments, highly secure applications can be deployed to the user devices.

V. PROPOSED IMPLEMENTATION

A. User registration

As stated, the complex nature of the architecture requires end users to use mobile devices in order to consume the remote signature services. These mobile devices have the role of creating the Signature Activation Data. The importance of this feature becomes apparent when the Signature Activation Module has to make a decision on whether to authorize the creation of the requested digital signature.

An immediate conclusion is that the solution needs to implement additional registration steps. As a result, in the process on enrolling a new user, the enrollment of a corresponding mobile device is mandatory.

As part of the registration process the user must submit a User ID, User account password, User email and the mobile device phone number as well as other information requested by a regular Certification Authority. The most important addition is the mobile device phone number as this information will be later used in the mobile device enrollment process.

After successfully submitting an enrollment request the user must wait for the Registration Authority to process the request. As part of this process the Signature Activation Module and the HSM must follow a series of steps which should include:

- The SAM will store the relevant information from the user profile: user ID, user password (if the module will have to authenticate the user) or information about the entity to which the user authentication will be delegated.
- The SAM will request a new key pair generation from the HSM.
- The HSM will generate a new key pair and will export the relevant CSR (certificate request).
- The Registration Authority will forward the CSR to a Certification Authority (CA) and the SAM will pair the new private key with the user details.

B. Mobile device registration

After successfully registering a user, a cryptographic secret key has been stored in the HSM, the SAM has stored the relevant user information, a link between the user information and the secret key has been created and a digital certificate has been created for the secret key.

In a traditional local signing solution, the newly created secret key can now be used by the registered user to create digital signatures. But remote signatures pose a series of unique problems such as sole access to the private key and the need to demonstrate the intent of creating a signature. As a result, a mobile device will be used to aid in the signature creation procedure.

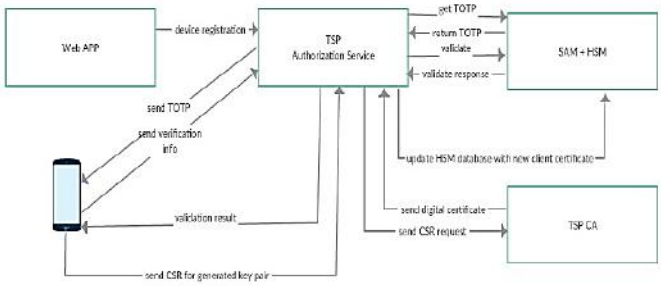

Figure 2. Mobile device registration

A solution based on a simple mobile device application is proposed. This application will aid in the mobile device registration process and also will authorize the subsequent remote signature requests. The app will not be platform dependent but will require a device which supports a Secure Element or a Trusted Execution Environment.

As part of the registration of the mobile device we must validate the ownership of the device. For this purpose, the user will be prompted to enter his credentials. After a successful authentication an TOTP token will be generated by TSP. This TOTP token will be sent to the phone number that the user declared in the enrolment process.

SMS messages are usually not a secure mode of transferring information. An alternative to sending an SMS containing the TOTP token, services such as push notifications could be used to deliver the token to the app instance. Another more robust solution might use a multi factor authentication mechanism based on multiple OTP token transferred via SMS and via email to the end user.

After successfully identifying the mobile device, the application is responsible for a key pair generation. An app instance running on a device which has a Trusted Execution Environment (TEE) has the benefit of added security for the stored secret key. The secret key will be stored in the device tampered protected environment and will be accessible only to the legitimate user. The use of the secret key will be protected by the user's touchID, faceID or device passcode using standard mobile OS functionality (iOS and Android).

The private key will be securely stored in the TEE and will authorize all the digital signature requests by signing the signature activation data. The public key will be enveloped in a CSR and sent to the TSP in order to create the corresponding digital certificate.

C. Remote signature flow

The remote signing process starts by the user logging into the application, whether it is a web-based application or a standalone app. After the user is successfully authenticated, it can request a document to be signed. In order for the remote signature provider to process the request, it will need some relevant pieces of information such as:

 Digital Object Identifier 10.32754/JMT.2020.2.08

- The user ID – the provider needs to be able to identify the user that requests the signing process.
- The user credentials – if the app does not require user to log in prior, it will need the credentials to authenticate the user. If the user is authenticated it will need information regarding the user session such as a JWT token.
- Certificate identifier – if the user has more than one private keys, it must indicate the desired key by supplying the corresponding certificate.
- Data to be signed – the data which will be signed or a representation such as a hash.
- Data to be displayed in the mobile app – the user will need to be shown a message in the mobile app so the originator of the request should build an appropriate message.

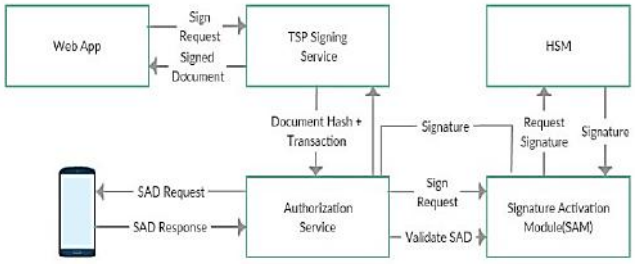

Figure 3. Remote signature flow

The TSP business application will receive the signing request with all the relevant information. It will create a request transaction for reference which will be sent to the client and the request will be forwarded to the Signature Activation Module (SAM). When the SAM receives a request, it will create an Authorization Request which should contain:

- Data to be signed remotely (hash value, hash algorithm).
- The User ID.
- The certificate alias.
- The data to be displayed on the mobile app.
- Some salt information to prevent replay attacks.
- Validity time for the request.

In order for the user to approve the request, it will need to access the cryptographic key in the TEE. Access to this key can be granted in a number of ways: be using a secret PIN, using bio-metrics (fingerprint), using remote authentication.

If access is granted to the private key, the authorization request will be signed with this private key and the result will be sent to the TSP. The TSP forwards the request to the SAM which is responsible for authorizing the signature by verifying:

- The user ID is the same.
- The centrally-held certificate alias is the same.
- The salt information, if set, is the same.
- The signature on the response can be verified by the device's authorization certificate which was set-up when the device was registered.

If the request is granted, then the corresponding representation of the document is forwarded to the HSM which will use the users secret key to sign the document. It will provide the raw signature (PKCS#1 or PKCS#7) back to the TSP application which will embed the raw signature into the PAdES, CAdES or XAdES structure within the PDF, XML or CMS structure for other types of documents. Finally, it will provide the fully signed document back to the Business Application which can optionally display this to the user.

VI. Related Work

The Cloud Signature Consortium offers a common framework for implementing remote signature solutions. The proposed common architecture supports several mechanisms for user authentication:

- 2-factor authentication using a PIN generated by a cryptographic device
- 2-factor authentication using a OTP code usually transmitted via SMS
- Delegating authentication and using a federated Oauth2 based approach by a third party

Using a 2-factor authentication mechanism is indeed a robust solution regarding security. The main downside is related to costs; business must employ some sort of external authenticator in the form of a cryptographic token or a SMS messages.

Additionally, SMS based solutions are vulnerable to attacks since the information is not protected while in transit. Token based solutions are more robust but have the downside of costs and are vulnerable to device loss.

The most robust and cost-efficient solutions are based on TOTP [15] tokens. Such solutions are implemented by Trans Sped and Intesi Group. The device scans a QR code which contains a secret key and generates a time dependent code based upon it.

The proposed solution is similar with respect to costs as it does not employ the usage of external authenticators. It also has the advantage of leveraging a TEE for cryptographic operations and enhanced security by digitally signing the transmitted data using a private key.

VII. Conclusions

As opposed to a classical Qualified Electronic Signature, a cloud based solution that offers such signatures has to solve a set of newel and specific problems: the signature provider must ensure the integrity of the data submitted, must demonstrate and be able to identify the intent of creating a digital signature of an user, must ensure that the valid user which is the owner of the cryptographic key is the only entity capable of using this cryptographic material and finally must provide an authentication framework for users.

The Signature Activation Protocol (SAP) is also responsible and describes the way the Signature Activation Data (SAD) will be handled by the Trust Service Provider (TSP). The structures contained in the Signature Activation Data (SAD) and the way these are handled must ensure that the security goals are achieved.

A solution based on a simple mobile device application is proposed. This application will aid in the mobile device registration process and also will authorize the subsequent remote signature requests. As part of the registration of the mobile device we must validate the ownership of the device. For this purpose, a scheme based on a TOTP code is used. A

more robust multi factor authentication mechanism based on multiple codes delivered via email and SMS could offer enhanced security.

After successfully identifying the mobile device, the application is responsible for a key pair generation. The private key will be securely stored in the TEE and will authorize all the digital signature requests by signing the signature activation data. The use of the secret key will be protected by the user's biometric data using standard mobile OS functionality (iOS and Android). Thus, a highly secure solution which greatly reduces costs and security risks is achieved.

REFERENCES

[1] Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32014R0910

[2] Cloud Signature Consortium, https://cloudsignatureconsortium.org/

[3] IETF, RFC2289, *A One-Time Password System*, https://tools.ietf.org/html/rfc2289

[4] Android Open Source Project, Hardware-backed Key Store, https://source.android.com/security/keystore

[5] NIST, SP 800-124 Rev. 2

[6] ETSI, EN 419_241-1 Trustworthy Systems Supporting Server Signing Part 1: General System Security Requirements

[7] ETSI, EN 419_241-2 Trustworthy Systems Supporting Server Signing Part 2: Protection Profile for QSCD for Server Signing

[8] ETSI, EN 419 221-5 "Cryptographic Module for Trust Services"

[9] ETSI, Security Week: Remote Signature Creation Services, 2018

[10] ETSI, EN 319_411-1 Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements, 2018

[11] ETSI EN 319_411-2 Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 2: Requirements for trust service providers issuing EU qualified certificates

[12] ETSI TS 119 432 V1.1.1 Electronic Signatures and Infrastructures (ESI); Protocols for remote digital signature creation

[13] Thales, The Impact of the European eIDAS Regulation, 2018

[14] Thales, CEN & ETSI standards & eIDAS Compliance, 2018

[15] IETF, RFC6238, TOTP: *Time-Based One-Time Password Algorithm*

 Digital Object Identifier 10.32754/JMT.2020.2.08