Zentrum für sichere Informationstechnologie – Austria
Secure Information Technology Center – Austria

A-1030 Wien, Seidlgasse 22 / 9
Tel.: (+43 1) 503 19 63–0
Fax: (+43 1) 503 19 63–66

A-8010 Graz, Inffeldgasse 16a
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

DVR: 1035461
ZVR: 948166612
UID: ATU60778947

CLOUD-BASED SIGNATURE SOLUTIONS: A SURVEY

Florian Reimair – florian.reimair@iaik.tugraz.at
Version 1.0, 8. October 2014

Abstract: Cloud-based signing solutions are on the rise and attempt to revolutionize business processes while integrating themselves well into cloud storage infrastructures. The combination promises faster process flows for signing a contract than the classic paper-based approach. In this survey we reviewed seven representative examples of cloud-based signature services and assessed them at the provided cryptographic features, the interfaces they offer, the authentication methods they provide and the key storage implementations used. We found that multi-factor authentication and hardware security module back-ends are common features. Interfaces range from APIs over web user interfaces to proprietary applications. Yet, there are shortcomings in flexibility and security.

Inhaltsverzeichnis

1. Introduction

The classic process of signing a contract did require a lot of resources; paper on which the contract was printed, time in which the contract was delivered to the signers, time in which the contract was returned, an archive, maintenance of the archive, and so on. Electronic signatures aim to prevent all the issues with the classic signing process. However, electronic signature merely shifts the classic signing process to an electronic one without solving the problem entirely. There is of course no need for paper and paper archiving and delivery time is short. However, eSignature raises the issue of transmission privacy and security, the clients have to have some software installed to sign and validate electronic signatures, and (digital) archiving has to be done as well. The main issue, however, arises, if clients get mobile. Laptops, tablets and Smart Phones are well-accepted devices for business processes and have to be supported by electronic signature solutions. But as the cloud revolutionized the world of data storage in terms of cost and effectiveness, cloud-based signature services are on the edge of revolutionizing signature. Cloud-based signing eliminates delivery costs and delays by shared storage and ensures transmission privacy and safety through access control mechanisms. Archiving is done by the cloud storage provider. Signature creation tends to be done by the cloud signature provider after authenticating the client by modern and therefore mobile-device-friendly methods. All in all, cloud-based signing leverages the advantages of electronic signature to a much more cost- and time-efficient solution than the previous classic pen and paper process. The demand for solutions is high.

The concept of a cloud-based signing service has already been adopted by the industry. Vendors provide different solutions with different features. This survey evaluates seven representative examples of cloud signature solutions after the provided interfaces, authentication methods, storage, and signature formats. The seven services are Adobe EchoSign, Amazons CloudHSM service, the Austrian Mobile Phone Signature, Cryptomathic Signer and their Service Gateway, Dictao Cloudcard, DocuSign and Intensi's Time4Mind signature service. Other vendors like Izenpe, ARX, SigningHub, Cryptolog, and Cryptas have similar services and methods and where therefore not reviewed separately. The surveys information base is compiled from information available to the public and therefore no in-depth crypto-analysis were possible and performed.

The results show that most of the reviewed services offer mobile-device-friendly 2-factor authentication and use hardware security modules (HSMs) for performing the cryptographic operations and key storage. Having an HSM as signature creation device, PDF, XML, and CMS Advanced electronic signature formats ETSI [6, 5, 4] are supported mostly. There are solutions that only offer the most basic electronic signatures (an image of a handwritten signature) and solutions that can only create qualified signatures, i.e. a digital signature legally equivalent to handwritten signatures. As for using the service, there are solutions offering standard APIs like PKCS#11 [2], web user-interfaces and proprietary applications available to inter-operate with the cloud signing back-end.

The survey is organized in 3 parts. Chapter 2 lists and describes the metrics in detail, chapter 3 gives detailed discussions on every reviewed product. The survey gives some informative details about the service and its vendor, sketches the inner workings of the product as far as information available to the public revealed this, and of course gives a security evaluation for every reviewed product. The survey is concluded in chapter 4 by summarizing the findings of the survey, giving an opinion on some potential shortcomings in the available products and a glimpse towards the future of cloud based signing.

2. Metrics

In order to evaluate and compare services in a fair and clear way, we defined a set of metrics. The metrics target the most important characteristics of a cryptographic service.

2.1. Cryptographic features

Cryptographic features are the most basic characteristic of a cryptographic service provider. The features determine the use cases, for which a certain service provider can be considered.

A cryptographic service provider can be classified and evaluated in much detail. Starting from the top, a cryptographic service provider can do signature, encryption, MACs and/or hashing. Symmetric and/or asymmetric cryptographic computation capabilities refine the classification. Symmetric procedures can be performed using different chaining techniques used in stream- or block ciphers, asymmetric schemes follow the classic RSA-based methods or the newer elliptic curves arithmetic. Further, crypto providers can support different padding schemes, encodings, and finally, key lengths.

This survey evaluates only the most basic classification, that is whether a cryptographic service provider can sign, encrypt, HMAC, hash, or all of them. Since the survey was targeted at web services providing cryptographic methods, the main focus in selecting the services lied upon signature and encryption. Hashing can be done easily on the client. Today's common computing devices, such as Smart-phones, Tablets, Laptops, and of course Desktop PCs, are easily capable of hash computation. With that, there is no need to transfer a whole document over the net. MACs require a shared key and are therefore not provided by the kind of service we evaluated. Thus, we only evaluate for signature and encryption capabilities.

2.2. Interfaces

The types of interfaces offered by a cryptographic service are an important characteristic from a system integration point of view. Again, the options determine the use cases, for which a certain service provider can be considered.

Interfaces can in general be classified as local interfaces and remote interfaces. Local interfaces provide access to local security modules through in-app libraries like Java Cryptographic Extension or PKCS#11, or system services like Smart-card protocol adapters. Remote interfaces allow the client application to use remote security modules. Connections are established either in a LAN scope by proprietary protocols or standardized protocols like KMIP [3] which operates on top of the network stack utilizing SSL/TLS and HTTPS. Further, an interface can be classified into stateful or stateless communication.

This survey targets central web-server-based cryptographic service providers. Therefore, the above mentioned general classification has to be somewhat adopted. We will not evaluate after interfaces that communicate with local Smart-cards for example, yet, it is perfectly normal to use an in-app library to connect to a remote server via some protocol. Therefore, the evaluation will target the communication between client and server in a general manner. In this survey, we will benchmark, how a client can use the API regardless if it is an in-app or a remote API, which technologies and protocols are used to transfer commands via the net, and how hard it is for clients to integrate the interfaces.

2.3. Authorization

Whenever cryptographic methods are involved, one of the key questions is how an operation or a key can be authorized for usage.

The best known procedure is to use a hardware Card Terminal with an integrated PIN-pad to use a removable Smart-card security module. Such a configuration makes PIN eavesdropping hard and brute-forcing the device is made hard because the device can be removed from the reader. But as this survey is about online crypto services, there is no tamper-resistant hardware device accessible by the client. So evaluating against tamper resistance on a hardware level is out of scope for this survey.

Instead, we focus on the authentication method options which authorize key usage and/or operation. Methodologies rely for example on the knowledge of the user or on

possession of some kind of token (1-factor authentication) or on a combination of knowledge and possession (2-factor authentication). Rarely, a biometric factor (i.e. fingerprints, iris-scans) is added to the authentication process and with that higher-factor authentication schemes are created. 1-factor authentication methods are much more prone against theft or eavesdropping than higher-factor authentication methods are.

One common authentication method is the simple user-name/password tuple where the user-name identifies the user and the password authenticates the user based on 1-factor knowledge. Another common authentication method merges identification and authentication into one secret (i.e. PIN authentication). More sophisticated authentication systems rely on 2-factor authentication. Popular examples are Smart-cards, where one can only use the key when he is in possession of the Smart-card and knows the PIN or authentication methods using the mobile phone as second factor of possession where a nonce is sent to the phone which is in possession of the user and the user has to provide the nonce to complete the authentication. Authorization is done by the service, be it a Smart-card or a web service. The authenticated identity is checked by the service itself and an authorization decision is made. Nowadays, 3rd party identity providers are often used to harden the authorization process.

However, in this survey we are interested in how the different service providers manage their authentication and authorization decisions and where they draw the line between convenience for the client and security.

2.4. Key Storage

In centralized cryptographic service provider structures, another crucial characteristic is how the cryptographic key material is persisted.

There are multiple options on how a cryptographic service provider, local or remote, can store (key) data. The easiest option is to store keys unencrypted on the hard disk. The ease of use comes with a huge security risk. Not only the system administrator can read and use them as they please but everyone gaining access to the server for example. The worst case scenario is of course if the private key parts are accessible by anyone over the web. More secure mechanisms involve key encryption where the encryption key is derived from user-supplied information, passwords for example. Encrypted hard disks allow for offline safety, virtualization solutions enable runtime security. Another approach utilizes special hardware security modules (HSMs) which guarantee the security of the key material throughout its lifetime by a variety of countermeasures, in software and hardware. An HSM may ask for a key secret to authorize using a key. These keys could be gathered from the user but are commonly stored on hard disk protected with file-system permissions.

Naturally, any key storage solution has its advantages and drawbacks. In the scope of the survey, we are interested in how the keys are stored and therefore where the service provider places itself between security, cost, and convenience.

2.5. Summary

In this chapter, we presented the set of metrics which we applied to the services we reviewed. We selected the metrics cryptographic features, interfaces, authorization and key storage.

As for cryptographic features, this survey evaluates if a solution can sign, encrypt, HMAC and/or hash. Since the survey targets web-based services, the interface metric evaluates the type of connections and options the solutions provide. For authorizing cryptographic primitives such as operations or keys some kind of authentication is needed. The survey evaluates the security of the methods the solutions provide and use. And last but not least, the metric of where to store the key and how to protect the key material against unauthorized access is evaluated as well.

3. Case study

In this survey, we evaluated seven crypto service providers.

3.1. Adobe EchoSign

Adobe was one among the first to adopt Cryptography as a Service (CaaS). Their solution's goal was to integrate seamlessly into existing processes and tools while making heavy use of new technology. Their Web Contracting solution promises a game-changing advantage over conventional ("analog") solutions: signed documents (i.e. PDF-documents) can be exchanged much faster than conventional mail while not breaking conventional processes. Adobe lists Sales, Marketing, Procurement, Human Resources, Legal and IT as possible deployment scenarios.

3.1.1. Inner Workings

EchoSign[1] started as a non-cert-based signature solution. There digital signatures where conventional handwritten signatures stored in some digital format. This graphical representation of a signature was linked to an email address. This form of identity binding allowed a comfortable way to check signatures. The trust in a signature and therefore a signed document was guaranteed by the service provider, namely Adobe.

A new version of EchoSign used cryptographic signature methods. Cryptographic certificates replaced the graphical representations of a signature. The quality and security of an EchoSign signature therefore were pushed by the underlying cryptographic methods coming with the PDF Advanced Electronic Signature (PAdES) standard. However, the email address remained to bind an identity to its certificate and the certificate was issued and managed by a hardware security module (HSM) operated by Adobe. Therefore, the trust in a signature remained dependent on the service provider, namely Adobe as a trusted third party.

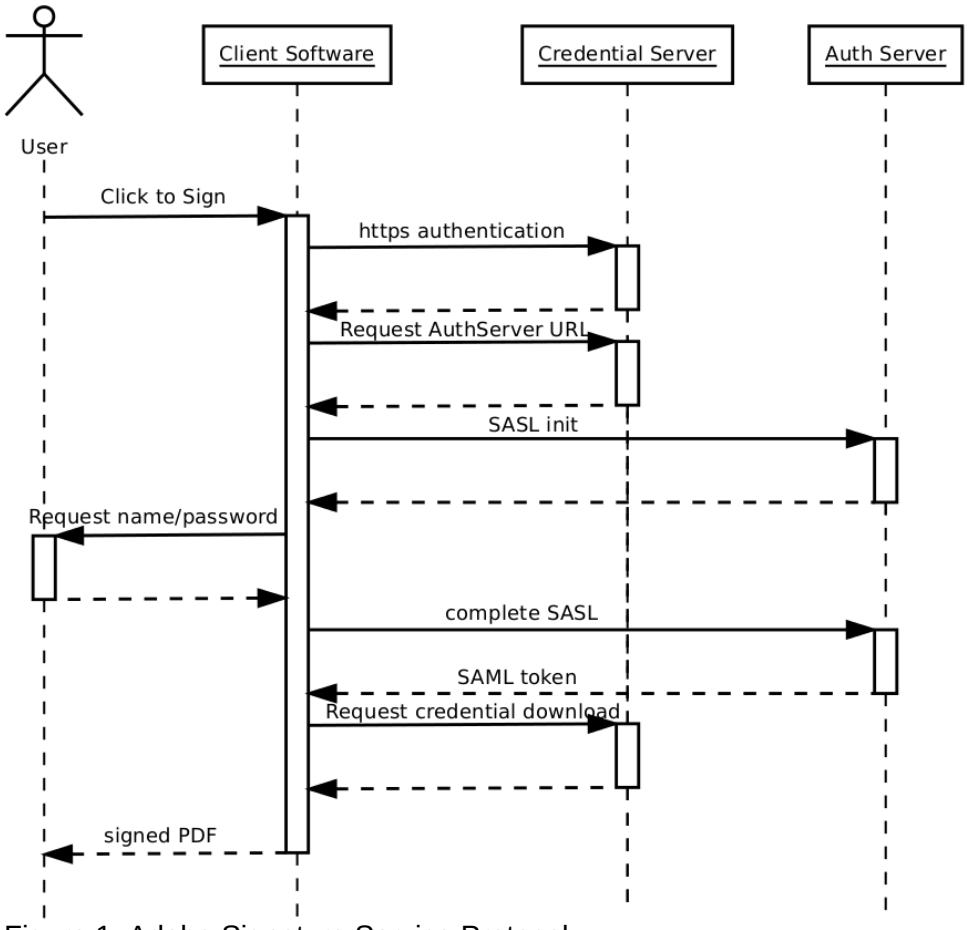

Figure 1: Adobe Signature Service Protocol

[1] https://www.echosign.adobe.com/en/home.html

Since Acrobat (Reader) 8, EchoSign can use client-supplied certificates to sign documents. With that, Smartcard HSMs can be used to sign a PDF. Server-based certificates are still available.

User identification, authentication and authorization remained a bottleneck in the EchoSign infrastructure. Adobe therefore proposed the Adobe Signature Service Protocol (ASSP), a protocol capable of handling certificate enrollment, credentials request and retrieval for client-side signing, signature request and retrieval for server-side signing and decryption of PDFs. The basic information flow of ASSP is illustrated in figure 1. ASSP uses an isolated auth server via SASL [10] to check user-supplied user-name/password authentication info to authorize credential delivery to the client software. The client software performs the cryptographic operation and returns the result, a signed PDF document for example.

Detailed specification could not be gathered for use in this survey. However, a white-paper on security [9] indicates state of the art cryptography and communication APIs follow standards and proposed standards.

3.1.2. Evaluation

We evaluate Adobes EchoSign's certificate-based version. The non-certificate based solution is completely out of scope for this survey.

Documentation and the security white-paper list signature and encryption capabilities as a service. However, encryption seems to only be available in their document management and cloud storage solution. Signatures seem to not be bound to any other service.

The white-paper [9] indicates state of the art cryptographic primitives for signature creation and encryption.

EchoSign offers a Web GUI as well as a piece of client software as an access point. While Adobe states integration into available infrastructure as one of their main goals, there is no documentation that such an integration exists. It therefore seems, that there is no API available for integrating the service in an arbitrary application.

For authentication methodology, there are no clear statements available in the documentation. However, username/password tuples are mentioned now and there. Therefore, we must assume, that there are no strong authentication methods such as multi-factor authentication methodologies or cryptographic primitives available. The support for local Smart-card HSMs via platform infrastructures (for example the Microsoft Windows Cryptographic Service Provider (MSCSP) former Microsoft Cryptographic API (MSCAPI)) allow for a reasonable control over ones key usage. That of course only applies to the client-side signing solution.

An EchoSign building block diagram indicates, that there is some central cryptographic hardware (an HSM) in use for key management. Dedicated cryptographic hardware pushes the security of the whole infrastructures security. Although, for the server-side signing solution, cryptographic certificates and their corresponding private keys are managed by Adobe itself and the cryptographic hardware is operated by Adobe as well. The HSM can therefore shield the cryptographic primitives only against external attackers. Adobe remains (needs) to be a trusted third party.

EchoSign seems to be a reasonable product meeting a lot of every-day life requirements. Standard cryptographic methods accompanied by a hardware security module indicate a high level of security. However, user-name/password based authentication for the online-signature use case does not match modern security requirements anymore. Furthermore, the ASSP protocol indicates that the credentials, i. e. the cryptographic keys, are transfered to the client software to be used to sign content. That only integrates with the concept of a central key-managing HSM if only wrapping keys used for protecting the keys of the users are managed by the HSM. Along with the identity binding, which is done solely by Adobe without any documented cryptographic methods, a rather safe conclusion is, that one is just required to to accept the service provider as a trusted third party.

3.1.3. Conclusion

EchoSign evolved from a non-certificate signing solution in the early days of cloud computing to a full-fledged document management solution incorporating online and offline signature support, encryption, and cloud storage. A client side GUI is accompanied by a Web UI and offers great flexibility.

However, given that Adobe controls the identity binding and the key management all by itself, the security of the whole system heavily relies on the trustworthiness of the service provider, i. e. Adobe, itself.

3.2. Amazon CloudHSM

In 2013, Amazon complemented their set of cloud services with the CloudHSM[2]. Amazon CloudHSM offers a hardware security module for use in the Amazon Virtual Private Cloud (Amazon VPC). Having a HSM for cryptographic operations of one's disposal can boost the security of an entire system significantly.

3.2.1. Inner Workings

The core component of the AWS CloudHSM service, the hardware security module, is accessible via PKCS#11, MS CAPI, and JCE APIs but only from within the VPC. There is no accessing the keys from the Amazon Web Service (AWS) area. AWS clients, however, can manage the HSM appliance. VPC clients can manage their keys. The general architecture is illustrated in figure 2.

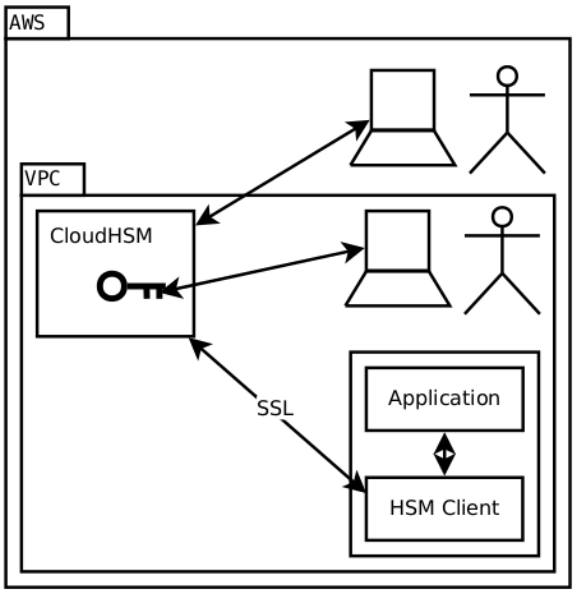

Figure 2: Amazon CloudHSM architecture

Amazon grants its VPC clients sole access to the HSMs so clients can configure their HSM appliance to meet their application needs. There is also an option for multiple HSMs in mirror mode to achieve easy to use load balancing.

Having a Luna SA HSMs from vendor SafeNet operational on-premise, one can partner up the local HSM with the one operated by Amazon. Having this mirroring service available allows for simple key and configuration backup.

3.2.2. Evaluation

Given the architecture, the Amazon CloudHSM is no service directed towards end-users. Therefore, there are no interfaces available as is. User APIs are to be provided by the application. The application itself can use the HSM via PKCS#11, MS CAPI, and JCE.

As the service is not directed towards end-users, there is no authentication procedure to discuss either. The application itself decides upon which authentication methodologies are available to the end-user. The SafeNet Luna SA HSM as a back-end HSM and

[2] https://aws.amazon.com/cloudhsm

crypto provider guarantees for a broad range of standardized cryptographic primitives and methods an application can use. Which of these are presented to the end-user again depends on the application.

Key storage has to be done by the application as well. Application can heavily rely on the HSMs features for encrypted key storage and can therefore be considered secure. As HSMs grant access to their keys only if provided the correct secret, the application has to somehow get hold of and forward the secret to the HSM. Common implementations leave the secret visible to administrators of the application. But as the secret is handled by the application, there is no definite conclusion to this.

3.2.3. Conclusion

Amazons CloudHSM service is an enabler for more secure web applications. Especially to applications which processes data that is sensitive enough so that even the web space provider cannot be trusted. For the end-user, there is no way to use the service in an easy and integrated way.

3.3. Austrian Mobile Phone Signature[4]

The Austrian Mobile Phone Signature service[3], an implementation of the Austrian citizen card concept which constitutes the official eID of Austria, provides a signature service to the user. The service meets the requirements for the creation of qualified electronic signatures which are listed in the EU Signature Directive [1].

3.3.1. Inner Workings

The Austrian Mobile Phone Signature is designed as a service to be used by web applications. A SecurityLayer [8] request issued by the application to the signature service initiates the signing process. The request holds the document and some other information. The signature server then presents a web page to the user where the user selects the key he wants to use. The user may now review the data he is about to sign. The user then authorizes the signing process by authenticating himself to the server. The signed document can be fetched from the signature server afterwards.

The service uses a HSM as cryptographic back-end.

3.3.2. Evaluation

Being a service for web-services, the Austrian Mobile Phone Signature offers the Security Layer Protocol to web applications for initiating a sign process and communicates with the user via web pages. On the one hand the service is easy to use since there is no need for any software on the client platform - any browser will do. On the other hand, an end-user cannot use the service to sign something on her own (pdfs or emails for example). The ease of use is bought by the price of low versatility. There is, however, a pdf-signer tool available for creating PDF signatures as an installable application.

Authentication-wise, the Austrian Mobile Phone Signature requires a 2-factor authentication process. Identification is done via the mobile phone number. For authentication, the service requires a password and a one-time-password (a mobile transaction number (TAN)) sent to the users phone via SMS. Authentication is therefore considered as very secure.

For signature creation, XAdES and CAdES is available as signature formats. Given the signatures created by the service are legally equivalent to handwritten signatures, only the XAdES and CAdES structures for creating the signature content are specified.

[3] https://www.buergerkarte.at

3.3.3. Conclusion

Being a certified signature service that meets the EU Signature Directive, the Austrian Mobile Phone Signature service is an easy to use service for (web) applications. Direct access for end-user is not possible due to the limited set of APIs.[4]

3.4. Crypthomatic Signer and Crypto Service Gateway

Cryptomathic Inc. was founded 1986 as a spin-off of the University of Aarhus, Denmark. The company claims that they were one of the first to commercialize cryptographic methods and focus on keeping their products at the cutting edge of technologies.

Cryptomathic[5] offers a variety of different services. First, they offer a Cryptographic Key Management System (CKMS) which is to assists its users in handling their keys. A signer service offers web-based signing capabilities and the crypto service gateway service provides a managed HSM to the customers.

3.4.1. Inner Workings

Cryptomathic provides MS CAPI and PKCS#11 interfaces which reroute cryptographic commands to a managed HSM somewhere on the Internet. Communication is secured by standardized methods such as use of the Secure Remote Password protocol (SRP), TLS and others.

The service supports a variety of formats. For signatures common formats like XAdES, PAdES, and CAdES are supported, for encryption well-known encryption schemes are available.

The user-centered approach (illustrated in figure 3 for an exemplary signing procedure) requires for a policy system that helps in managing key creation and usage as well as in permission enforcement.

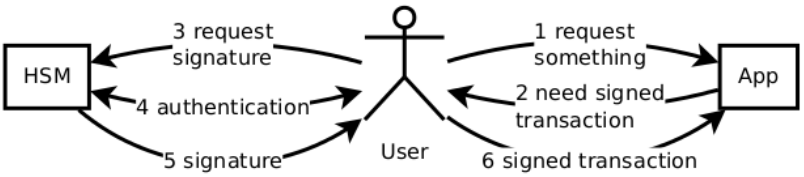

Figure 3: Cryptomathic Signer workflow example

3.4.2. Evaluation

Cryptomathics Signer as well as their Crypto Service Gateway (CSG) offer well-known and standardized cryptographic methods and primitives. Advanced Electronic Signature schemes for XML, PDF and CMS formats are available for customers of the Signer product, the CSG offers a wide range of primitives and methods by using high grade HSMs (like AEP, SafeNet, Thales, and Ultimaco) to provide the crypto service behind the scene.

Having a MS CAPI provider and PKCS#11 implementation allows easy access to cryptoservices for WindowsOS as well as others. The Microsoft CryptoAPI (MS CAPI) is provided by the MS Cryptography API: Next Generation (MS CNG) and MS Cryptographic Service Provider (MS CSP) for Windows OS versions prior to 7. Integrating a provider into the platform crypto service allows any MS Windows application which uses the operating systems crypto service (Adobe Acrobat Pro, MS Office Outlook as prominent examples) to use the Cryptomathic services. Therefore, signing email or PDF documents seems to be easy with Cryptomathics Signer.

The standardized PKCS#11 interface is widely adopted by a variety of software and platforms. The central keystore of Debian systems and the email client Mozilla Thunderbird among others can interface with PKCS#11 libraries and therefore use

[4] A-SIT, as Austrian confirmation body under the Signature Law, did a security assessment of this solution. It was confirmed by A-SIT as SSCD meeting the requirements of the Signature Directive. We therefore refrain from commenting on security aspects, as this may be seen as biased.

[5] http://www.cryptomathic.com

Smart-cards as security modules. Having a PKCS#11 library on its own, Cryptomathic allows these applications the use of their service in an integrated and comfortable way.

For authentication, Cryptomathic products offer a great range of methods. The Signer offers a proprietary authentication application, dynamic one-time-password (OTP) delivery via SMS aka mobile TAN, credit card authentication methods MasterCard CAP and Visa CodeSure as well as Open Authentication (OATH) in modes HOTP, TOTP, and OCRA. The selection covers multi-factor authentication methods as well as open standards. The service can be considered as well protected and secured.

Cryptomathics CSG in contrast offers a rather small set of authentication methods. User-name/password as well as LDAP- and RADIUS-based authentication methods are available. The service may integrate well in existing corporate infrastructures but lacks the option of multi-factor authentication. Anyhow, given the in-company use case, the solution is considered as reasonable secure.

As for key storage, Cryptomathic relies on third party HSMs build by AEP, SafeNet, Thales, and Ultimaco among others. The physical protection level therefore depends on the HSM in use. However, as every key protected by an HSM needs some sort of secret to be unlocked, and the documentation provided by Cryptomathic does not state anything regarding these secrets, one might suspect that the administrators at Cryptomathic may use the keys of customers.

3.4.3. Conclusion

The services offered by Cryptomathic seem reasonable secure and easy to use. They offer a broad range of authentication methods, a small but delicate selection of interfaces, and feature industry grade HSMs to do the key protection for them. The Signer service offers all major signature formats and therefore seems to meet a wide range of use cases. If a special use case cannot be met with Signer, Cryptomathic offers its Crypto Service Gateway.

3.5. Dictao Cloudcard

Using Smart-cards with mobile devices tends to be difficult due to the lack of and usability issues of physical card readers. The French company Dictao[6] addresses this very issue. Their core feature and goal is to replace physical SSCDs, i. e. Smart-cards, with an HSM-backed cloud service. Their Cloudcard solution ships within an all-in-one solution for companies in need for strong authentication, transaction security, and archiving/audit. Their solution is available either as a licensed product or as software as a service (SaaS).

3.5.1. Inner Workings

The workflow of Dictaos solution is as follows. The end-user asks a proprietary client software to sign some data. The software asks for authentication information. The user provides his knowledge (1 in figure 4) and authenticates the signature operation. The software adds its own authentication information (2) which is somehow bound to the device, performs some intermediate steps (3) and finally sends the information to the web-accessible service (4). The service uses the received data to recreate the user's key within a back-end HSM (5). The service now creates the signature and destroys the key afterwards (6). The signature value is returned to the end-user.

[6] https://www.dictao.com

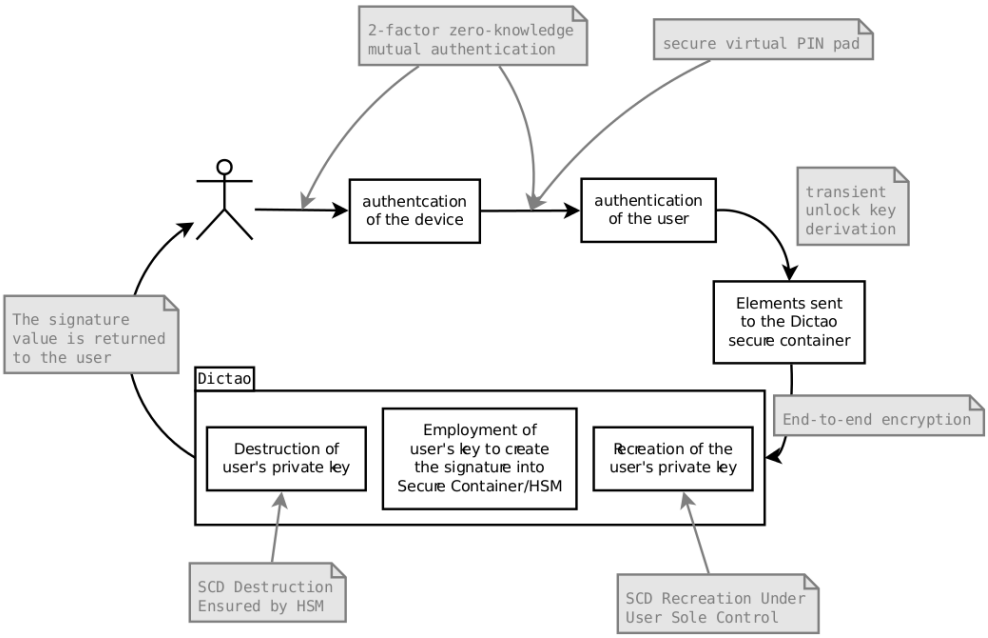

Figure 4: Dictao inner workings

3.5.2. Evaluation

To use the cloud-based SSCD service, clients have to use Dictaos client software. There seems to be no other interface or API available.

Dictao claims to perform 2-factor authentication. Yet, their client-software seems to handle both, proofing the possession of a device (the device the client-software runs on) and collecting the PIN (i. e. the knowledge factor). Other information indicate that there might as well be two independent devices for initiating the signature creation process and collecting the user PIN. As far as the publicly available information goes, Dictao seems to do better than single factor authentication, yet, they may not reach full 2-factor security.

Dictao uses an HSM as SSCD. The crypto part of their solution therefore is secure and protected. The signature service list the Advanced Electronic Signature standards for PDF and XML as supported and available for signature creation.

The key storage solution of Dictao sets it apart from other providers. Dictao does not store any private key on the cloud service. The keys are generated on the fly based on the authentication information provided by the user. After the signature operation the private key instance is destroyed. The cloud service therefore does not persist any private key and can therefore not be attacked in order to extract the private key from disk. The downside is that the information used to recreate the private key parts is visible to the client software and therefore might be eavesdropped by malware.

3.5.3. Conclusion

Dictaos solution on decoupling an SSCD electrically from the (mobile) device is an interesting approach to the well-known problem. Signature algorithms, provided interfaces and authentication methods do not differ much from other competitors. As promising as the solution sounds, a detailed security analysis may give more insight on how secure the solution really is. Such an analysis would burst the scope of this survey.

3.6. DocuSign

DocuSign[7] does electronic representation of graphical signatures, as their main business case. However, a certificate-based digital signature option can be added.

[7] https://www.docusign.com

3.6.1. Inner Workings

The user initiates the signature creation process from either a native application for mobile devices or the provided web UI. The user is then asked for authentication. In the first step, a user-name/password tuple is required. The second step of the 2-factor authentication can be selected in the GUI. Options are an SMS or voice message to an identification-bound phone number or an email to an identification-bound email address. Either message contains a one-time transaction number (TAN) which the user has to provide to finish authentication. Then the signature is created.

3.6.2. Evaluation

DocuSign offers a web UI as well as native applications for mobile devices. No other interfaces or APIs seem to be available. In order to trigger an electronic signature the user interface asks for a multi-factor authentication. Having the option of using mobile TAN as a second factor results in a high security level, the email option results in a lower security level of the authentication system.

Although there is an option for certificate-based signatures, there is no information on how one can obtain a digital certificate or where and how the related keys are stored and protected. It however seems, that keys are generated and stored by DocuSign and the users unlock the keys with their authentication information.

3.6.3. Conclusion

DocuSign claims to be one of the first to pick up on electronic signatures in a time before EC signature directives and qualified signatures. They therefore focused and focus on representing handwritten signatures in digital documents in a human readable format. They do support digital signatures as well but there is no details available to the general public.

3.7. Intesi Time4Mind Qualified Remote Signature Service

Time4Mind[8] is a product of the Italian company Intensi. Their product is a full-fledged document management solution in the cloud, offering a qualified remote signature service. The signature service supports simple electronic signatures, advanced electronic signatures and qualified electronic signatures. The security functions are not known. It supports classic digital signature features and the full protection scheme backed by the EU signature directive [1], respectively.

3.7.1. Inner Workings

There is next to no information available to the public on how the signature creation process works. However, a few keywords are: They use an HSM behind their web service, authentication is done by 2-factor authentication with password as the first factor and an OTP delivered by either SMS, or some proprietary solution like Vasco, RSA, Radius. Communication with the server is protected by SSL with client authentication and tunnels remote Java and .NET payload. SHA256 for hashing is used by RSA 1024 or more for digital signatures in the three Advanced Encryption Schemes PDF, XML, and CMS.

3.7.2. Evaluation

Intensi has a client application for users besides Java and .NET APIs for system designers. Having the communication secured by SSL and client authentication is a hint to a high level of security. Their product is therefore well connected and flexible enough to fit most use cases.

2-factor authentication with a mobile TAN option besides proprietary OTP-generators as second factor besides the classic user-name/password tuple let assume a reasonable amount of protection and a rather high security of the user authentication process.

SHA256 for hashing is state of the art as well as RSA 1024 or more is for signatures. The most common signature formats are supported as well. Backed by an HSM from vendor Thales, key and crypto operation security seem high security as well.

[8] https://www.time4mind.com

Since there is an HSM already in their infrastructure, the chances are high that an encrypted key storage solution is used with the help of the HSM.

3.7.3. Conclusion

Intensis solution seems reasonable secure. Given the lack of detailed information available to the public, there is no guarantee for that. An in-depth security analysis might result in a completely different result.

3.8. Others

Other vendors like Izenpe[9] , ARX[10] , SigningHub[11], Cryptolog[12], and PrimeSign[13] offer similar solutions to the ones discussed in detail above.

4. Conclusion

Cloud storage and outsourcing of sub-processes has become a major part of corporate infrastructure. Digital signature solutions speed up business processes and are therefore asked for by business.

The market reacted to the demand and offers a wide range of products that meet the cloud signature requirement. However, there are significant differences between the services. A classic overview is given in table 1. Please note, that the security of the reviewed solutions cannot be assessed by the information given in the table.

Most of the services feature an HSM which handles the key management and protection. The services that do not use an HSM do not feature advanced electronic signatures (digital signature) although an HSM is not required to create an advanced electronic signature. So whenever an advanced electronic signature is offered, the keys and the cryto operation itself is protected by a dedicated hardware module and therefore is considered secure. Whenever advanced digital signatures are supported, all of XML, CMS, and PDF formats are supported. As for interfaces, there are different service available for different use cases. Some offer APIs such as Java, MS CAPI or PKCS#11 interfaces. They can be integrated into existing solutions to meet existing processes, yet someone has to programmatically connect the service. Others offer web user interfaces. Web-UIs do not need any client-side software which results in great flexibility regarding client platform. Web UI providers offer all-in-one solutions. Integrating such a service into an existing process tends to be hard or impossible. Whenever a company focuses on integrating mobile devices they offer proprietary client applications to interface with their cloud service. This kind of service may be integrated in existing infrastructures, but depends strongly on the vendor and product. Several reviewed solutions require multi-factor authentication prior to authorizing the user. However, besides the state-of-the-art and reasonable secure mobile-TAN solution, other variants are implemented to proof the possession of a device. Proprietary OTP-generator devices as offered by Vasco, RSA, or Radius are one option, TAN-by-eMail is another. Whenever advanced electronic signature is offered, there tends to be a 2-factor authentication in place. A few services go with the classic user-name/password solution. Overall, most services offer reasonable secure authentication methods.

[9] http://www.izenpe.com
[10] http://www.arx.com
[11] http://www.signinghub.com
[12] http://www.cryptolog.com
[13] http://www.prime-sign.com

	Cryptograhpic Features	Interfaces	Authorization	Key Storage
Adobe EchoSign	• signing • encryption for Adobe's document service	• Web GUI	• Username/password • PIN for client-side Smart-card	• HSM-backed
Amazon CloudHSM	• Almost anything	-[14]	-[15]	• HSM
Austrian Mobile Phone Signature	• XAdES • CAdES • PAdES[16]	• Web GUI • Web API • Wrapper applications	• 2-factor Mobile TAN	• HSM-backed
Crypthomatic Signer	• XAdES • PAdES • CAdES	• MS CAPI • PKCS#11	• Proprietary • 2-factor Mobile TAN • Credit card auth	• HSM-backed
Dictao Cloudcard	• XAdES • PAdES	• Proprietary application	• 2-factor[17]	• HSM-backed • Recreates keys from user information
DocuSign	• Singing with certificates • Digital (pen-imitating) signature	• Web GUI • Native apps for mobile	• Mobile TAN • Voicemail TAN • Email TAN	No Info
Intensi Time4Mind	• XAdES • CAdES • PAdES	• SSL client authentication with remote Java or .NET	• Mobile TAN • Proprietary (Vasco, RSA, Radius)	• HSM-backed

Table 1: Feature Overview

[14] Amazon's CloudHSM has no communication with the end-user. The cloudHSM can be used by a cloud application via PKCS#11, MS CAPI or JCE
[15] An authorization scheme is upon the cloud application.
[16] Only by the desktop application.
[17] The proprietary application handles the proof of possession AND the collection of the PIN.

Vendors offer reasonable secure products with a reasonable set of features. However, there are shortcomings regarding integration and security. Integrating a web service into an existing infrastructure can be a tough job. Yet, vendors offer APIs for their products which allow a rather easy integration. However, there are next to no vendors who offer for example web-accessible user-interfaces alongside APIs. Integrating a service into an existing business process therefore mostly is about having some consensus on features and flexibility. More flexible and interchangeable connectivity options in a product may satisfy a broader range of business processes. From a security point of view, key management as well as trust relations are sub-optimal. Key management is mostly done by the vendor itself. For most of the products, the client has no sole control over the keys, i. e. how they are generated, used, and destroyed. A rouge vendor may benefit from the information he may read from his clients while claiming that he is trustworthy. There are, however, a few products which support extra-site HSMs (i.e. an HSM hosted at the client company or some third-party HSM service) and therefore render the product vendor incapable of using the keys. A desirable situation for handling trust relationships might be a multi-service-provider approach. Having different service providers for authentication, cryptography, and storage limits each of the providers to a smaller amount of data which might not be sufficient to reconstruct readable contents.

Protocols and standards for remote authentication, privilege management, and storage are subject of research for standardization [7] or already available. Service providers are yet to come. For the future, there seems to be a tendency towards service providers that focus on offering only single aspects of the all-in-one solutions we see today. Interoperability protocols and frameworks may connect multiple providers in a modular way to achieve a solution. The market might therefore see a set of products that can be connected together in a modular fashion rendering all-in-one solutions deprecated.

References

[1] Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures, December 1999.

[2] Pkcs #11: Cryptographic token interface standard, April 2009.

[3] Key Management Interoperability Protocol Specification Version 1.0. online, October 2010. URL http://docs.oasis-open.org/kmip/spec/v1.0/kmip-spec-1.0.pdf. last visited in July 2014.

[4] ETSI. TS 101903 - XML Advanced Electronic Signatures (XAdES), June 2009.

[5] ETSI. TS 101733 - CMS Advanced Electronic Signatures (CAdES), April 2013.

[6] ETSI. TS 102778 - PDF Advanced Electronic Signatures (PAdES), April 2013.

[7] European Committee for Standardization. TS 419241 - Security Requirements for Trustworthy Systems Supporting Server Signing - draft, June 2013.

[8] Arno Hollosi, Gregor Karlinger, Thomas Rössler, and Martin Centner et al. Die österreichische Bürgerkarte. online, January 2014. URL https://www.buergerkarte.at/konzept/securitylayer/spezifikation/20140114/. last visited in July 2014.

[9] Adobe Systems Incorporated. A primer on electronic document security, 2007. Whitepaper.

[10] A. Melnikov and K. Zeilenga. RFC 4422: Simple Authentication and Security Layer (SASL), June 2006.