I. J. Computer Network and Information Security, **2015, 4, 1-13**
Published Online March 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2015.04.01

Server-Side Encrypting and Digital Signature Platform with Biometric Authorization

Leszek Siwik
AGH-UST University of Science and Technology, Krakow, 30-059, Poland
Email: siwik@agh.edu.pl

Lukasz Mozgowoj
Biometric Trust Information Systems, Krakow, 31-864, Poland
Email: Lukasz.mozgowoj@biotrustis.com

Abstract—The most important shortcomings of solutions based on public key infrastructure and digital signatures are: costs, ambiguous laws, and nuisance of daily use.

The purpose of this article is to discuss the motivation and benefits, as well as a presentation of concepts, high-level architecture, and demonstration of the operation of bioPKI; i.e., a server-side encryption and digital signature platform with biometric authorization. The usefulness of even the most advanced platform of any type is negligible if convenient and easy-to-implement mechanisms are not provided to integrate this solution with external systems and applications. Thus, the possibility of integrating the bioPKI platform with applications and systems supporting PKCS#11 or CryptoAPI CSP is discussed.

Index Terms—PKI, digital signature, encryption, PKCS, CSP, biometry, finger vein.

I. Introduction

Today, especially in the context of E. Snowden's revelations, it is obvious that it is not necessary to be a VIP, a top member of a government, or a corporate board member to become a victim of systematic digital surveillance.

Unfortunately, common and popular applications offering encryption and digital signatures – especially those based on asymmetric algorithms [6,12,15] and public key infrastructure fundamentals in particular [1,16] – don't provide an appropriate level of security and privacy, since it is typical for the keys to be stored directly in the file system.

Since it is not a problem in the case of the public key (it should be publicly accessible "by definition"), it can be a strong security thread regarding the private key. Obviously, it can be additionally secured by a password or PIN, but the following must be remembered: 1) it is still stored directly in the file system (always dangerous); and 2) it is as secure as strong is the password defined by the end-user (which is usually the weakest element of any security system [13]).

A more-secure approach is storing the key inside external, physically-separated hardware element. Most frequently, a dedicated smart card with a crypto processor and physically-separated space for storing keys is used in this case. This way, the private key cannot be exported or copied anyhow. Such an approach significantly reduces the risk of compromising the key, but whenever the user wants to perform a crypto-operation, (s)he must be equipped with this additional hardware element, which can be problematic and inconvenient in everyday use.

What is more, the card has to be protected by the user who (as mentioned earlier) is usually the weakest element of any security system [13].

That is why research is still being performed to propose cryptographic systems with the same (or even higher) security level as contemporary ones but without any additional devices, cards or tokens.

In project UDA-POIG.01.04.00-12-041/11-00, research has been performed to design the bioPKI platform to perform cryptographic operations and store private keys on a central (super) secure environment, and control the key access with strong biometric user authentication and authorization on the basis of recognizing the blood vessel system (the so-called finger vein).

One of the important aspects to be solved when the bioPKI platform was designed was integrating external applications, services, and systems in as easy- (and cheap-) as-possible way without any security flaws. This is also addressed in this paper.

II. Motivation

In 1976, W. Diffie and M. Hellman proposed a new schema of cryptographic key exchange [6] that has given rise to asymmetric cryptography (cryptography with a public key). It was clear then that cryptography would start a new era when it was able to provide not only classic cryptographic services and algorithms for protecting data (just encrypting) but also some kinds of additional (extra) services such as digital signature, time-stamping, or "digital notary" [16,1,15].

New algorithms, along with the appropriate cryptographic protocols, became more important when the popularity of e-services exploded. In the e-world, such elements as digital signature, digital authorization and authentication, and time-stamping are even more important than classically-understood cryptography used merely for encrypting and protecting the privacy of data transferred in public networks.

In the case of some particular e-services (e.g., e-payments, e-banking, and e-offices), such "extra functionalities" of cryptography have been a sine qua non condition of providing real (not only informative) e-services. It takes place, for instance, in the case of fully-functional access to financial and banking products and services.

Unfortunately, time passed, and the popularization of digital signature as well as the number of (e)-services using this technology were far, far below what was expected (until today, it has actually been rather marginal), and solutions such as SMS tokens became definitely more popular.

Obviously, in the "era of mobility", SMS token is very convenient since it is available for almost anyone without any additional devices, cards, etc. It also realizes the idea of two-factor authentication; i.e., "something what I know" and "something what I have."

The problem, however, is that it works only fragmentarily (mainly in the banking sector on the basis of banking laws – i.e., bilateral civil-law agreement for providing financial services signed between the financial institution and its customers).

Additionally, from a security perspective, it is enough to recall here consecutive releases of Zeus system[1] to realize how vulnerable it is to fairly-simple attacks.

Other problems are: the extremely-privileged position of mobile carriers and costs of such solutions (coming from monopolistic position of telecom operator(s)).

On the other hand, it can be said that we have both: suitable algorithms (not as risky as transmitting text messages in vulnerable GSM networks) and appropriate infrastructure (certification authorities in particular). We also have law regulations – defining the rules and formal effects of a "real" digital signature not selectively (in some industry sectors only) but globally, commonly, and for everyone. The question arises: why have such cryptographically "weak" solutions like SMS-tokens addressing additionally only part of our reality become so popular?

The answer for such a question is so complex and multi-threaded that a full diagnosis lies absolutely outside the scope of this paper. Among the most important elements, however, the following should be mentioned:

- Unclear and ambiguous law – for instance, the Polish Digital Signature Act of 18th September 2001 introduces "secure" and "more secure" signatures and, according to this act, even the footer below the e-mail can be considered as a "digital signature." It is said sometimes that the only advantage of this act is that it exists.
- The fear of technology (especially in the context of unclear law) – in common feeling "cryptography is a secret knowledge". 99% of our society know and understand neither algorithms nor cryptographic protocols. So "if I don't know and don't understand – I'm not also able to verify if it is or is not secure." The more so I'm not able to verify if it is “a secure” or maybe “a more-secure” digital signature according to the law in force. So natural doubts arise i.e., "what happens if somebody steals it from me" – e.g., will somebody be able to sell or to borrow on my house or incur debt? These are very typical concerns of typical users who conclude naturally that "just in case, it is maybe better for me not to have this digital signature/certificate."
- Complex and complicated procedure – from "John Q. Public's" perspective the procedure of issuing the certificate is complicated and, perhaps even more important – inconvenient. CA is simply one more department where you have to report that is intrinsically an important barrier (people don't like offices and clerks).
- The price – especially in relation with misunderstanding of the technology and – more so – the limited number of places and services where digital signature can be used, there is a common feeling that current prices to be paid for the certificate "just to use it once a year" is unprofitable and groundless.
- The onerousness in (occasional) usage – it is not without importance that even if somebody tries to generate his digital signature and bears the appropriate costs, what he receives is really onerous in daily and, more so, in occasional use. In a typical digital-signature solution, you have to remember that one more dongle or card you have to carry and protect. Since it is rarely used, usually if you want to use it – it turns out that you don't have it with you. It is also connected with a subjective feeling of the security level. About the theft of a phone, wallet, or credit card, one will notice within a few minutes or hours and react appropriately fast. In contrary, about the fact of the theft of the dongle or card with the signature, one realizes probably only when he has to use it again – maybe at the beginning of the next month when money transfers have to be signed, or at the end of the quarter when tax declarations have to be sent. So, it is not only ambiguous regarding the formal regulations, relatively expensive, problematic in issuing, and onerous in using, but is also perceived as a "ticking time bomb" and one more element in need of special care.

Taking all of the above into account, it is absolutely justified to ask if it is possible to propose a solution which takes all advantages from the concepts, algorithms, and protocols of asymmetric cryptography; i.e., giving not only encryption but also some "extra services" like non-repudiation, time stamping, etc., but which:

- Would be (extremely) easy to provide digital signature and crypto-services, for instance, in SaaS

[1] http://news.techworld.com/security/3415014/eurograbber-sms-trojan-steals-36-million-from-online-banks/

or IaaS model [5,11];

- Would provide not only actual but also "psychologically" realized high-level security being simultaneously as simple in daily (and occasional) use as only possible.

The attempt to propose the concept and realize a prototype of the system addressing the postulates given above was undertaken in project UDA-POIG.01.04.00-12-041/11-00; i.e.: “The server-side digital-signature platform with biometric authorization – bioPKI”. The goal of this paper is to shine a light on its proposed conception, selected elements of its top-level architecture and integration layer, and to demonstrate how it works on selected examples.

III. The Concept Of BioPKI System

The crucial goal of bioPKI project was: to propose the architecture of a server-side digital-signature platform in such a way, that any additional dongles, cards, or chips that the user must have in contemporary solutions could be eliminated preserving at least the same security level as before.

The visualization of the considered idea is presented in fig.1.

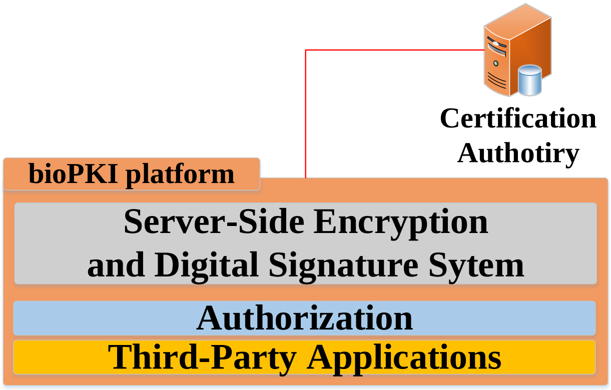


Fig. 1. The conception of the system.

As one may see, the top-level idea distinguishes:

- user applications (stations) e.g.: PC stations or third-party web services integrated with bioPKI platform;
- the bioPKI platform for realizing digital signature and(d)encryption processes. It is strongly assumed that any cryptographic operation requiring access to private key (including digital signature or decrypting) will be realized on this layer. Consequently, the strong assumption is that private keys will be stored (somehow) there and only there. Obviously, this part of the system has to be designed with the highest possible security level. Specifically, private keys never leave this (sub)system, and operations such as exporting or copying private keys will be impossible, both logically and physically;
- the authorization and authentication layer. Its main responsibility will be the clear-cut authentication of users by authorizing access to their private keys stored on the platform.

The important question is the compatibility of the centralized digital signature system with the legal system being in force. Well, it depends on the particular country and legal system. There are countries, like Poland, where according to the Polish Digital Signature Act proposed solution cannot be considered and certified as a "qualified" one – so in a legal sense, a digital signature made on the platform will not be considered as valid as a traditional, hand-written signature. Nevertheless, research on the platform like bioPKI is absolutely justified, since there are countries and legal systems where centralized, server-side digital signature is qualified as being in force. For instance:

- the idea of performing digital signature "remotely" (i.e., on the server-side) has been positively judged by the Forum of European Supervisory Authorities for Electronic Signatures [21] – the association of official authorities responsible for defining any digital signature aspects in EU member countries;
- the server-side digital signature system (authorized with SMS-tokens) has been successfully launched in Austria [18];
- scientists from Graz University of Technology published a document with a positive assessment of the server-side digital signature system for mobile devices [14]. The document confirms that signatures made remotely can be considered as "qualified" ones.

The one of two main top-level assumptions of the bioPKI platform is that the end-user doesn't have to be equipped with any additional dongles, chips, tokens, cards, etc.

There is, of course, a fundamental question whether this is possible at all; i.e., if it is possible to store private keys remotely on the server and assure the highest-possible security level without additional cards, dongles, tokens, etc. without utilizing such nonsensical or trivial approaches like SMS-tokens, logins and passwords, PINs, etc.

According to the (pre)design analysis, the decision was made to use one of the biometric authentication technology. In such an approach my “biological PIN” controlling access to my private key is always with me, and I don't need any additional dongles, cards, or tokens. It also provides not only an actual, but also a psychologically-perceived high security level – my finger, face, or eye are always with me and under my own control.

IV. The Biometry And The Security

Mechanisms and algorithms based on biometry are relatively new and, simultaneously, a promising direction of research on identification, authentication, and authorization [10, 3, 4].

From the bioPKI platform perspective, the main advantage of using biometry is eliminating any additional dongles or cards, which was one of the main assumptions of this project.

Techniques based on recognition of the blood vessel system are considered particularly interesting, since this is (much) more secure than fingerprint or face recognition, and equally secure as eye-iris recognition (yet, faster and more precise). It also has the important feature of requiring a live body, since positive authorization is possible when the blood (hemoglobin, in fact) flows through the circulatory system of a live person's finger. So any additional subsystems and algorithms for vitality detecting.

A. Identification error rates

In subsection *B*, selected biometric methods of identification are briefly discussed, focusing on the uniqueness of each feature, the immunity for preparing the pattern, and the Equal Error Rate (EER) coefficient value.

EER refers to such point on Detection Error Tradeoff (DET) curve where False Accept Rate (FAR) equals to False Reject Rate (FRR).

Generally speaking verification or identification system makes a decision by comparing the match score *s* to a threshold η.

So, taking a set of genuine and impostor match scores, FRR can be defined as the rate of genuine scores that are less than the threshold η where FAR can be defined as the rate of impostor scores that are greater than or equal to η.

So, formally FRR and FAR can be defined as [10]:

$$FAR(\eta) = p(s \geq \eta \mid \omega_0) = \int_{\eta}^{\infty} p(s \mid \omega_0) ds \qquad (1)$$

$$FRR(\eta) = p(s < \eta \mid \omega_1) = \int_{-\infty}^{\eta} p(s \mid \omega_1) ds \qquad (2)$$

where $p(s|\omega_0)$ and $p(s|\omega_1)$ are the probability density functions of the genuine and impostor scores respectively and ω_0 and ω_1 denotes impostor and genuine classes.

Intuitively, the lower is the value of ERR the better is the method of identification.

B. Selected biometric identification methods

Eye-iris – identification using the eye as a biometric feature consists in the generation of a pattern on the basis of a photo made in a special range of a grayscale. The pattern is generated by focusing on the localization of well-defined curves covering the iris in the photo. Contemporary algorithms for comparing the sample with the eye-iris pattern are pretty fast, and, what is important, this physical feature itself does not change over one's lifetime. Additionally, eye-iris is a unique technology which makes it practically impossible to cheat the scanner with an artificial sample (especially a scanner equipped with one or more vitality-detection algorithms). In this case, the ERR coefficient oscillates around 0, 01%.

Fingerprint – this method is the oldest identification technique based on biometric features. Unfortunately, identification based on the fingerprint analysis strongly depends on the condition(s) of taking a fingerprint sample. For fingers that are too dry or too wet, a comparison to stored patterns can be too inaccurate for real-world application. Additionally, without extra modules for vitality detecting, the scanner can be easily cheated. In addition, the ERR coefficient value oscillates only around 2%.

Face recognition – the next method of biometric identification is face recognition. It consists simply of comparing the taken sample with the reference face pattern. During comparison, many features are taken into consideration, such as the shape of the face, distinguishing marks, eye-span, shape and the span of the ears, shape of the nose, etc. Actually, the method is not satisfactory for real-life applications due to the relatively low quality of face-detection algorithms. Additionally, the scanner can be pretty easily cheated with a prepared photo. The ERR coefficient oscillates around 20%.

Hand geometry – this method is vulnerable on the condition of taking the sample – even the smallest change in hand location during the process can trigger a false identification. ERR oscillates around 1%.

Voice recognition – the method consists in matching a set of acoustic features found in the speech of human beings. This is classified as distinguishing for people – they depend on both anatomy and behavioral features. Unfortunately, they can change during a human's lifetime – what significantly complicates the recognition algorithm. There are also important problems with reducing background noise as well as its vulnerability to such factors as mood and health condition. The value of the ERR coefficient for voice recognition oscillates around 6%.

Finger vein – this method consists in exposing the finger to a light near the infrared band. Part of the light is absorbed by (live) hemoglobin, and the rest passes through the finger without any changes. Consequently, it is possible to generate the image of the blood vessels in a (live) finger. This method is very secure, since the system of blood vessels is not “publicly available” (for instance cannot be taken from the glass as fingerprints. It is also unique to each person and doesn't change during his lifetime. Also, it is impossible to use a finger that has been amputated or is no longer viable (i.e., alive). The value of the ERR coefficient oscillates in this case below 1% (c.a. 0.8%).

In table 1, a relative and qualitative comparison of the aforementioned biometric identification methods is presented. The comparison is made for the sake of the most important features found in each biometric identification method: i.e., accuracy and precision, efficiency, and security.

Table 1. Qualitative and relative comparison of selected biometric identification methods

Bio feature	Security level	Accuracy	Efficiency
Iris	High	High	Average
Finger print	Average	Average	Average
Face	Low	Low	Average
Hand geometry	Average	Average	Average
Voice	Low	Low	Average
Finger vein	High	High	High

As mentioned previously, considering the advantages and disadvantages of each particular biometric method – the finger vein method has been used in the bioPKI project. Thus, it is discussed in next subsection.

C. *The biometry of finger blood vessels*

Scanners recognizing the pattern of blood vessels in a finger are exposing the finger, with the use of LEDs, to light with a frequency close to the infrared band.

Part of the infrared light is absorbed by live hemoglobin, and the rest is caught by a CCD camera (see fig.2). Consequently, on the CCD matrix, the image of blood vessels (seen as dark lines) is received. The image is then normalized and the sample is generated – this is then compared to the pattern stored in the biometric data store. There are several examples of such technology on the market. In the bioPKI project, Finger Vein technology has been used [8].

The Finger Vein solution assumes three methods of scanning and analyzing the system of blood vessels; i.e., using a light reflection, light transmission, and a side-exposure approach.

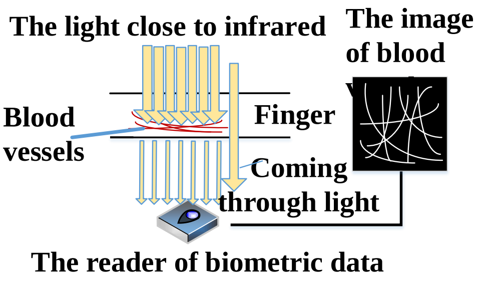


Fig. 2. The idea of blood vessels system scanning.

In the approach based on light reflection, the source of light is located on the same side as the camera. Part of the light is absorbed by hemoglobin, and the rest – reflected light – is captured by the camera. Reflection from the skin surface results in the image of the blood vessel system with low contrast. The reader in such an approach can be small and open [8].

In the approach based on light transmission, the source of light and the camera are located on opposite sides of the finger. The light which penetrates the finger is partially absorbed by hemoglobin, and the rest is captured by the camera. In this method, a high-contrast image is obtained and the reader has to be bigger and closed.

The side-exposure method is the most-advanced technique, as it combines the advantages of the previous methods to some extent. In this method, a high-contrast image is obtained, but the reader can be small and open.

The most important features of the Finger Vein solution – are as follows:

- it is "theft resistant" – biometric data used for identification is located inside live finger(s),
- every single scanning of the finger results in slightly different image of the blood vessel system (different location of the finger in the reader, different humidity, temperature, etc.),
- high accuracy – the False Rejection Rate (FRR) coefficient value is lower than 0,01% and the False Acceptance Rate (FAR) coefficient value is lower than 0,0001%,
- the system of blood vessels is unique to each person (even identical twins) and does not vary over a person's lifetime,
- the clarity of blood vessels allows for a fast analysis and efficient comparison and matching,
- it is impossible to reconstruct the image of the blood vessels system on the basis of the pattern stored in the biometric data store. In practice, the computational complexity is too high.

V. Server-Side Digital Signature System With Biometric Authorization

In this chapter, the most important technological and architectural top-level assumptions of the designed server-side digital signature system with biometric authorization are presented.

BioPKI platform has been considered as a system which allows for automation of (pre-existing) (e)services offered by such institutions as governmental offices (wills, applications, etc.), banks (wills, applications, authorization and confirmation of transactions, etc.), clinics (patient files, prescriptions, insurance verification, etc.) or drugstores (prescription realization, insurance verification, etc.).

Identifying crucial use-cases, they can be defined as in fig.3. Among the basic operation *Accessing sensitive data*, *Accessing the history of the signature* and *Sign document (transaction)* are distinguished. All of them use *Authorize (biometrically)* operation.

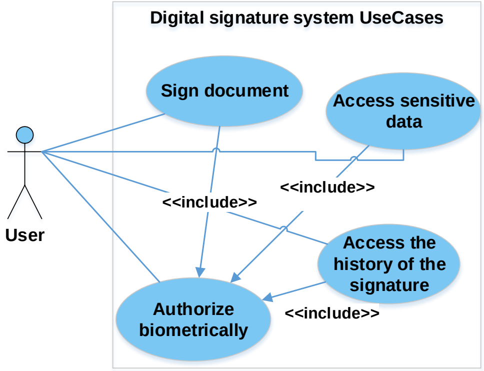


Fig. 3. Top level use-cases of bioPKI platform.

In fig.4, the top-level architectural design of the proposed server-side digital signature system with biometric authorization is presented, in accordance with the idea presented in fig.1.

There, the border(s) of the system, its crucial elements, and third party applications and systems are presented. Within the system boundaries, the following elements are distinguished:

- services; i.e., interfaces of the signature platform, which make it possible for third-party systems and applications to be (easily) integrated with the system;
- biometric authorization system, available only within the boundaries of the system;
- digital signature service and (d)e(n)cryption with biometric authorization, which provides cryptographic operations (digital signature in particular). It is assumed that any cryptographic operation requiring private key access will be performed only within a secure environment realized, among others, by physical hardware security modules with appropriate FIPS 140-2 Level 3 [2] and/or Common Criteria EAL 4+ certificates [17];
- HSM – after detailed market research modules of type nCipher Security World [24, 20] have been used. They are dedicated devices ensuring that cryptographic operations requiring access for private keys are performed in a highly-secured, physically-separated environment, so any kind of interference with both stored keys as well as performed operations is impossible.

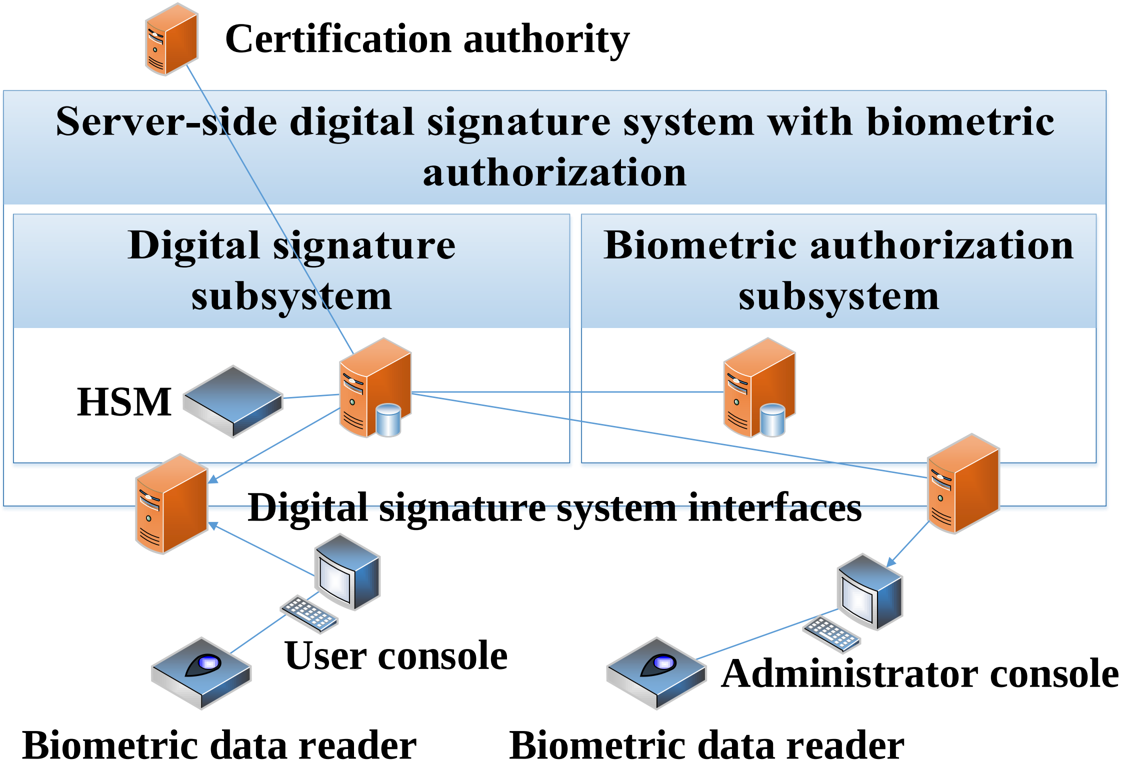

Certification authority
Server-side digital signature system with biometric authorization
Digital signature subsystem
Biometric authorization subsystem
HSM
Digital signature system interfaces
User console
Administrator console
Biometric data reader
Biometric data reader

Fig. 4. Top level bioPKI architecture.

A. The flow of basic operations

There are several types of Finger Vein readers available on the market, depending on the place where pattern matching is realized; i.e.:

- match–on–the–card – matching the biometric data is made on a smart-card connected to the reader;
- match–on–the–device – (readers of type 602/609 see fig.5) the matching process is performed inside the secured reader;
- match–on–the–server – (readers of type H1 see fig.5) the image of the blood vessel system taken in the reader is verified on the central server;
- match–on–the–host – the verification process is performed in the secure environment on the user's station.

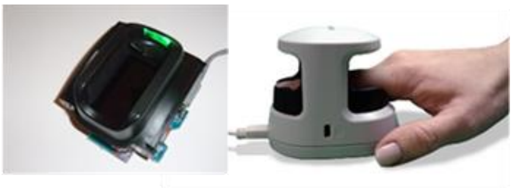


Fig. 5. FingerVein reader of type: 602/609 (left side) and H1 (right side).

The bioPKI system utilizes two of the four methods mentioned above (consequently, two families of readers);

i.e., matching data on the server and matching data inside the reader.

The flow of basic operations varies depending on the reader (consequently, the schema of authorization) used.

In fig.6, the flow of the signing operation is presented when the reader of type "match–on–the–server" is installed on the user's station.

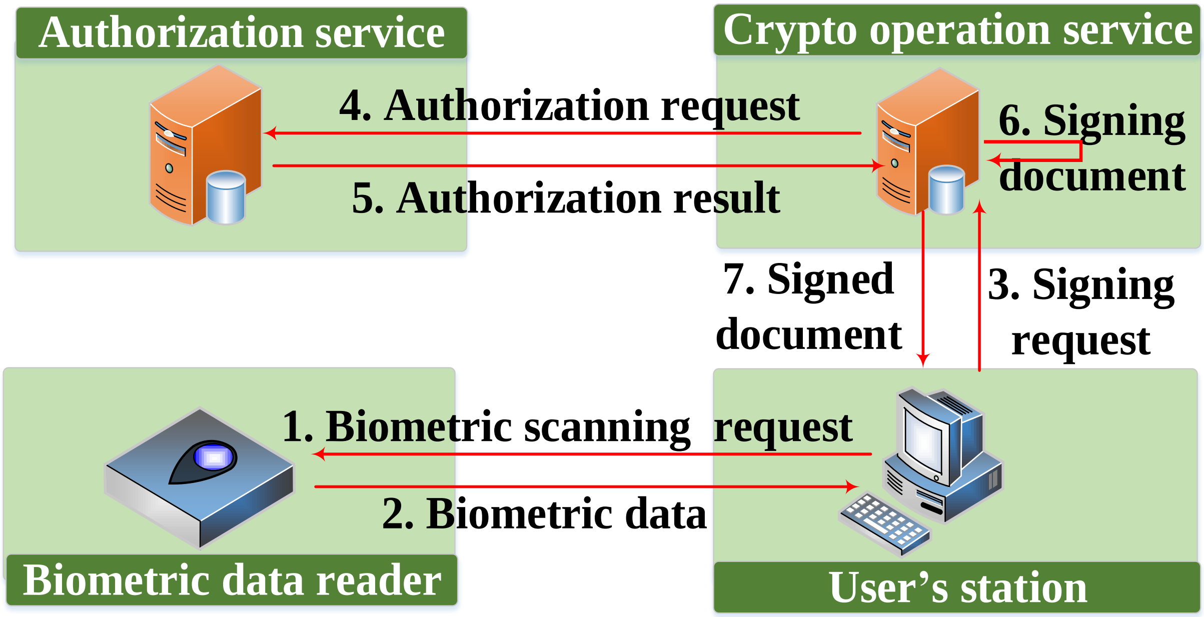


Fig. 6. The flow of the digital signature operation on the bioPKI system with match-on-the-server authorization.

In this mode of operation, the authorization server is equipped with IDs of all users registered in the system, along with their biometric data patterns. When the cryptographic operation requiring access to the private key is performed, the data taken by the reader and the one stored in the system is compared in the authorization subsystem, and (according to the results) access to the private key is allowed or rejected.

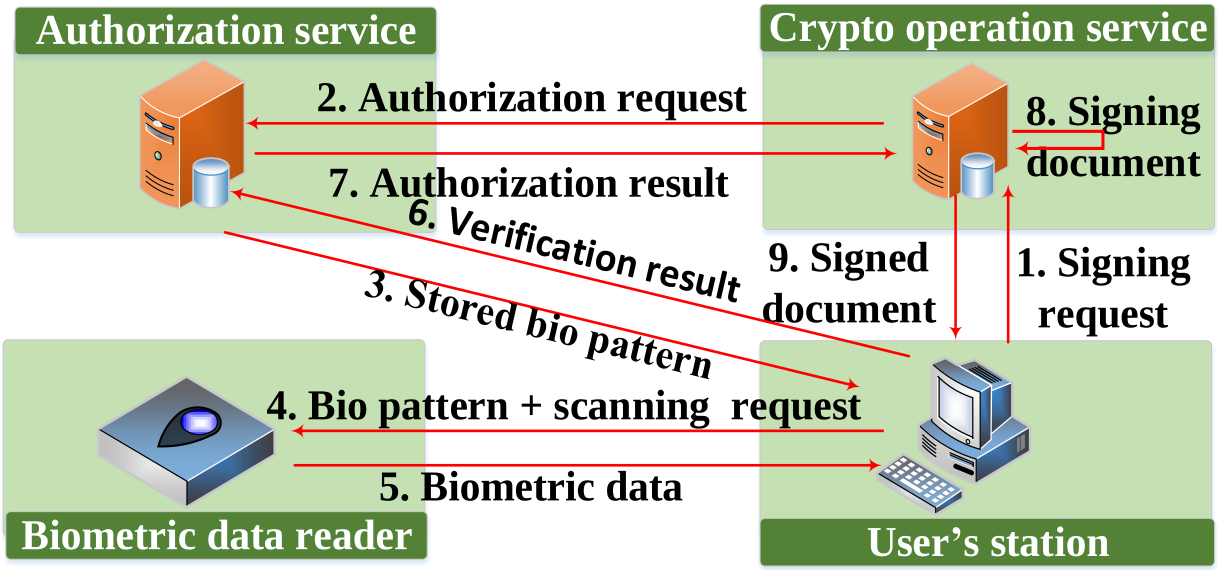


Fig. 7. The flow of the digital signature operation on the bioPKI system with match-on-the-device authorization.

In fig.7, the flow of the digital signature operation is presented when the reader of the match–on–the–device type is installed and configured.

In this case, the request for signing the document is sent first; subsequently, an authorization request is sent to the authorization service. Next, the authorization service sends the stored biometric pattern associated with the given personal ID to the reader. The reader starts the scanning process and next the reader itself matches the pattern to the sample image of the blood vessels. The result (positive or negative verification) is sent back to the authorization service and next the access to the private key is allowed or not.

VI. PKCS#11 AND CRYPTOAPI IMPLEMENTATION

The integration of external systems and third-party applications supporting the PKCS#11 and/or CryptoAPI standard is connected with creating appropriate cryptographic libraries being the implementation of these standards.

Additionally, to make it possible for external applications to use the bioPKI platform, appropriate "connectors" mapping internal calls of PKCS#11 or CryptoAPI standards on related calls of bioPKI API had to be implemented. In the next subsections, the realization for both mentioned standards is presented.

A. PKCS#11

PKCS#11 is a one of a family of standards called the Public Key Cryptography Standards (PKCS) published by RSA Laboratories [23]. PKCS#11 defines the API, which is independent from the platform on which is run, and allows for performing (cypto)operations on cryptographic tokens such as HSM or Smart Card. Logical interpretation of the "token" is the device storing the keys and/or certificates and performing cryptographic operations.

PKCS#11 isolates the application from the details of the cryptographic device – consequently, the application doesn't have to change its interface according to different devices it cooperates with [23].

Because a standard for cryptographic tokens doesn't exist, API has been created as an abstraction layer – it defines the most-frequently used cryptographic types (e.g., RSA keys, X.509 certificates, etc.) and all necessary functions for creating (generating), modifying, and deleting such objects. In fig.8, the general idea of the PKCS#11 standard is presented.

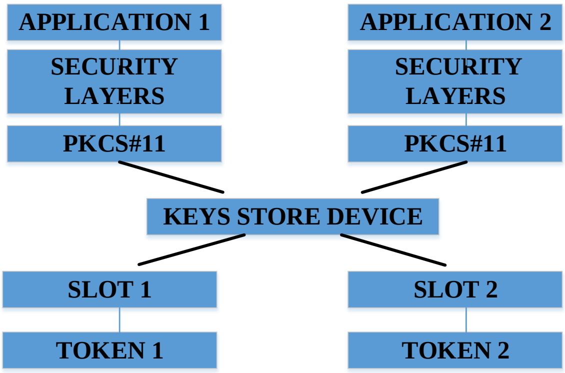


Fig. 8. The idea of PKCS#11 standard.

Creating libraries for integrating external systems and applications with the bioPKI platform required implementation of (the part) of basic functions defined by the standard; i.e., the functions responsible for initializing slots, signing data and signature verification, as well as for encrypting and decrypting data in particular.

Next, the implemented functions have been linked with elements of the biometric authorization and authentication layer (with finger-vein readers in particular) on the one hand and with the bioPKI platform itself on the other. The most important functions assumed by the standard that had to be implemented and provided are as follows:

- data encryption functions

```
CK_DEFINE_FUNCTION(CK_RV, C_EncryptInit)(
         CK_SESSION_HANDLE hSession,
         CK_MECHANISM_PTR pMechanism,
         CK_OBJECT_HANDLE hKey
      );
```

C_EncryptInit initializes the encrypting operation. *hSession* argument is a handler for the actual session, *pMechanism* indicates the mechanism of encryption to be used, and *hKey* indicates the key to be used for encryption.

```
CK_DEFINE_FUNCTION(CK_RV, C_Encrypt)(
         CK_SESSION_HANDLE hSession,
         CK_BYTE_PTR pData,
         CK_ULONG ulDataLen,
         CK_BYTE_PTR pEncryptedData,
         CK_ULONG_PTR pulEncryptedDataLen
      );
```

C_Encrypt is the function responsible for encrypting a single-data set. Consecutive arguments mean: the session handler, data to be encrypted, length of data in bytes, location of encrypted data, and length of encrypted data in bytes.

- data encryption functions

```
CK_DEFINE_FUNCTION(CK_RV, C_DecryptInit)(
         CK_SESSION_HANDLE hSession,
         CK_MECHANISM_PTR pMechanism,
         CK_OBJECT_HANDLE hKey
      );
```

C_DecryptInit is analogous to the encryption initialization function – it takes the same arguments and initializes the process of decrypting the encrypted data.

```
CK_DEFINE_FUNCTION(CK_RV, C_Decrypt)(
         CK_SESSION_HANDLE hSession,
         CK_BYTE_PTR pEncryptedData,
         CK_ULONG ulEncryptedDataLen,
         CK_BYTE_PTR pData,
         CK_ULONG_PTR pulDataLen
      );
```

C_Decrypt function is responsible for decrypting data. Analogous to **C_Encrypt**, it takes the data to be decrypted along with its length in bytes, buffer for storing decrypted data, and expected length of decrypted data in bytes.

- data signing functions

```
CK_DEFINE_FUNCTION(CK_RV, C_SignInit)(
         CK_SESSION_HANDLE hSession,
         CK_MECHANISM_PTR pMechanism,
         CK_OBJECT_HANDLE hKey
      );
```

C_SignInit function initializes digital signature operation. Analogous to previous initialization functions, it takes the session handler, data signature mechanism, and key as parameters.

```
CK_DEFINE_FUNCTION(CK_RV, C_Sign)(
         CK_SESSION_HANDLE hSession,
         CK_BYTE_PTR pData,
         CK_ULONG ulDataLen,
         CK_BYTE_PTR pSignature,
         CK_ULONG_PTR pulSignatureLen
        );
```

C_Sign is the function responsible for signing data. It takes the session handler, data to be signed, its length in bytes, localization of signed data, and length of the signature itself as arguments.

- signature verification functions

```
CK_DEFINE_FUNCTION(CK_RV, C_VerifyInit)(
         CK_SESSION_HANDLE hSession,
         CK_MECHANISM_PTR pMechanism,
         CK_OBJECT_HANDLE hKey
      );
```

C_VerifyInit function works and takes arguments similarly to previously-defined initialization functions.

```
CK_DEFINE_FUNCTION(CK_RV, C_Verify)(
         CK_SESSION_HANDLE hSession,
         CK_BYTE_PTR pData,
         CK_ULONG ulDataLen,
         CK_BYTE_PTR pSignature,
         CK_ULONG ulSignatureLen
      );
```

C_Verify function is responsible for verifying data signature. It takes arguments similar to previous functions; i.e., session handler, signed data, signature, and its length in bytes.

B. CSP (Implementation of CryptoAPI)

The Cryptographic Service Provider (CSP) is a library implementing Microsoft CryptoAPI [19].

As in the case of PKCS#11, CSP again comprises the implementation of all cryptographic operations required by the application supporting this standard.

After calling the given cryptographic function, the application supporting CSP doesn't have any knowledge of performing the operation.

Additionally, Microsoft CryptoAPI provides only the abstraction for implementation of a particular CSP. So it is, in fact, only a bridge linking the application with a particular provider.

In another words, the application calls functions of CryptoAPI, but each call is redirected to real implementation in the given CSP – it makes it possible to use CSPs as independent modules working in the same way with different applications [19].

Contrary to PKCS#11, when a cryptographic library supporting CSP standard is provided, additional configuration in the system is required to register the external cryptographic provider.

As it was in the case of the PKCS#11 standard, the integration of the bioPKI platform with external systems and applications supporting the CSP standard requires the implementation of appropriate functions assumed by the standard and linking particular cryptographic operations with the biometric layer (finger-vein readers) on the one hand and the bioPKI platform itself on the other.

Actually, the following operations are implemented and provided with bioPKI cryptographic libraries:

- function responsible for data encryption

```
BOOL CRYPTFUNC CryptEncrypt(
      HCRYPTKEY hKey,
      HCRYPTHASH hHash,
      BOOL Final,
      DWORD dwFlags,
      BYTE* pbData,
      DWORD* pdwDataLen,
      DWORD dwBufLen
      );
```

The **CryptEncrypt** function is used for encrypting data. The most important arguments are: *hKey* – encryption key, *pdData* – data to be encrypted, and *dwBufLen* – the size of the encrypted data.

- function responsible for data decryption

```
BOOL CRYPTFUNC CryptDecrypt(
      HCRYPTKEY hKey,
      HCRYPTHASH hHash,
      BOOL Final,
      DWORD dwFlags,
      BYTE* pbData,
      DWORD* pdwDataLen
      );
```

CryptDecrypt function decrypts encrypted data. As in the encrypting function, it takes key (*hKey*), (encrypted) data (*pbData*) and the size of the decrypted data (*pdwDataLen*) as arguments.

- function responsible for signing data

```
BOOL WINAPI CryptSignHash(
      HCRYPTHASH hHash,
      DWORD dwKeySpec,
      LPCTSTR sDescription,
      DWORD dwFlags,
      BYTE* pbSignature,
      DWORD* pdwSigLen
      );
```

CryptSignHash is a function responsible for signing data. It takes data to be signed, key specification, the signature itself, and its length as arguments.

- signature verification function

```
BOOL WINAPI CryptVerifySignature(
         HCRYPTHASH hHash,
         BYTE* pbSignature,
         DWORD dwSigLen,
         HCRYPTKEY hPubKey,
         LPCTSTR sDescription,
         DWORD dwFlags
       );
```

The **CryptVerifySignature** function verifies data signature. It takes signed data, signature localization and its length, as well as the public key as arguments.

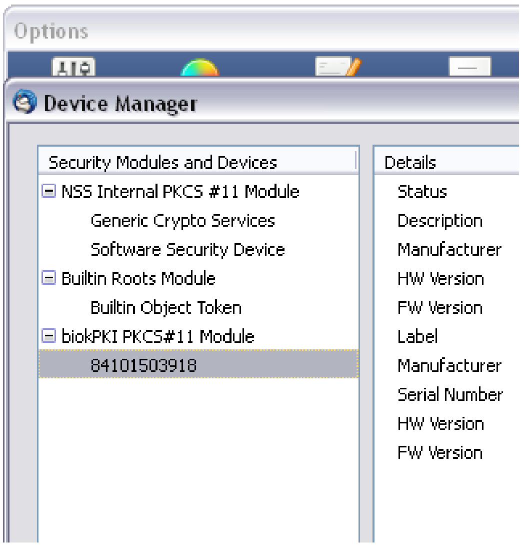
VII. Demonstration

When the CSP and PKCS#11 libraries have been implemented, it was possible to integrate external applications, web services, or middle-tiers with the bioPKI platform.

A. *Email decryption*

Below, the integration of a popular email client (i.e., Mozilla Thunderbird) with the bioPKI platform is presented. This application is compatible with the PKCS#11standard [2] so, as argued before, integration should be easy and straightforward and after that, any user should be able to encrypt or decrypt and sign their messages just by putting the finger into the finger vein reader.

The integration is as simple as typical installation of PKCS#11 cryptographic library. The process of registering the prepared library is presented in fig.9.



Options

Device Manager

Security Modules and Devices
⊟ NSS Internal PKCS #11 Module
Generic Crypto Services
Software Security Device
⊟ Builtin Roots Module
Builtin Object Token
⊟ biokPKI PKCS#11 Module
84101503918

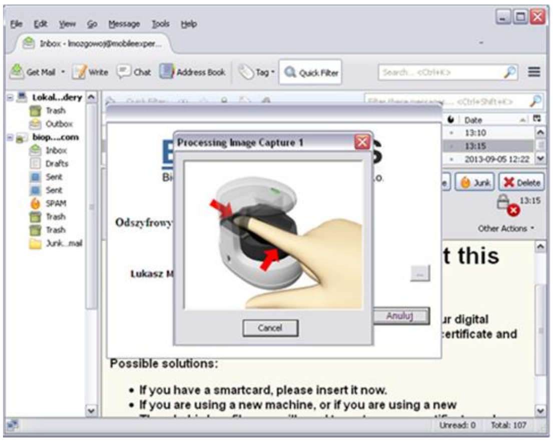
Details
Status
Description
Manufacturer
HW Version
FW Version
Label
Manufacturer
Serial Number
HW Version
FW Version

Fig. 9. PKCS#11 library registration in Mozilla Thunderbird

When the library has been installed, the user identified by hypothetical personal ID number: 84101503918 has been registered in Mozilla Thunderbird. They are existing users registered in the bioPKI platform.

When the encrypted message is received, the application itself is not able to decrypt it automatically, so the message is intercepted by the installed PKCS#11 library.

The library starts the decryption process but to complete this, access to the private key is required. Since the private key is stored on the server and is protected by the finger-vein technology, the library starts the process of blood vessel image scanning – what is presented in fig.10.



Possible solutions:

- If you have a smartcard, please insert it now.
- If you are using a new machine, or if you are using a new

Fig. 10. Message decryption in Mozilla Thunderbird

When the private key access authorization is successfully finished, the decryption process is started. It is worth remembering that the decryption process is realized remotely inside the secure environment on the bioPKI platform.

B. *Signing PDF document*

Another example is preparing a "demo banking system" and integrating it with the bioPKI platform, a popular web browser, and the Adobe Acrobat Reader application.

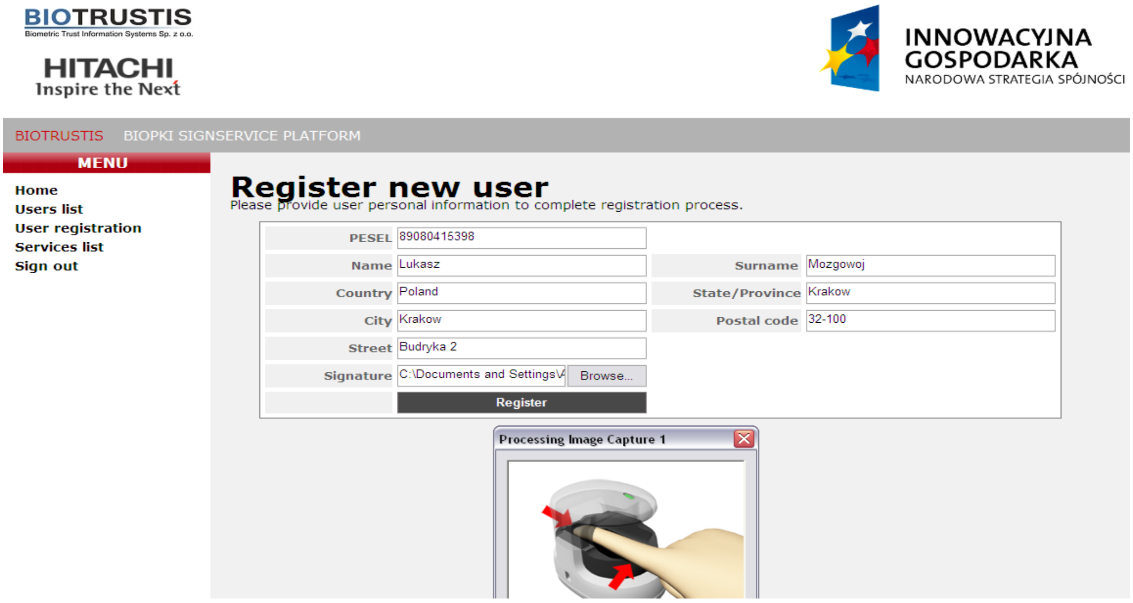
In fig.11, a sample registration form is presented. Since in this case it is run by Internet Explorer web browser this web application is integrated through CSP cryptographic libraries with the finger-vein reader.

During registration, personal data is collected, and the image of the finger blood vessel system is collected as the pattern for further matching and authorization.

When the form (for instance, a cancellation form) is completed, an appropriate PDF document is created and signed on the cryptographic service with biometric authorization.

When the signing operation is initialized, the system starts scanning the blood vessels, and the results (along with the user's ID – in this case, hypothetical personal ID number 89080415399) are transferred to the system.

[2] https://developer.mozilla.org/en-US/docs/PKCS11



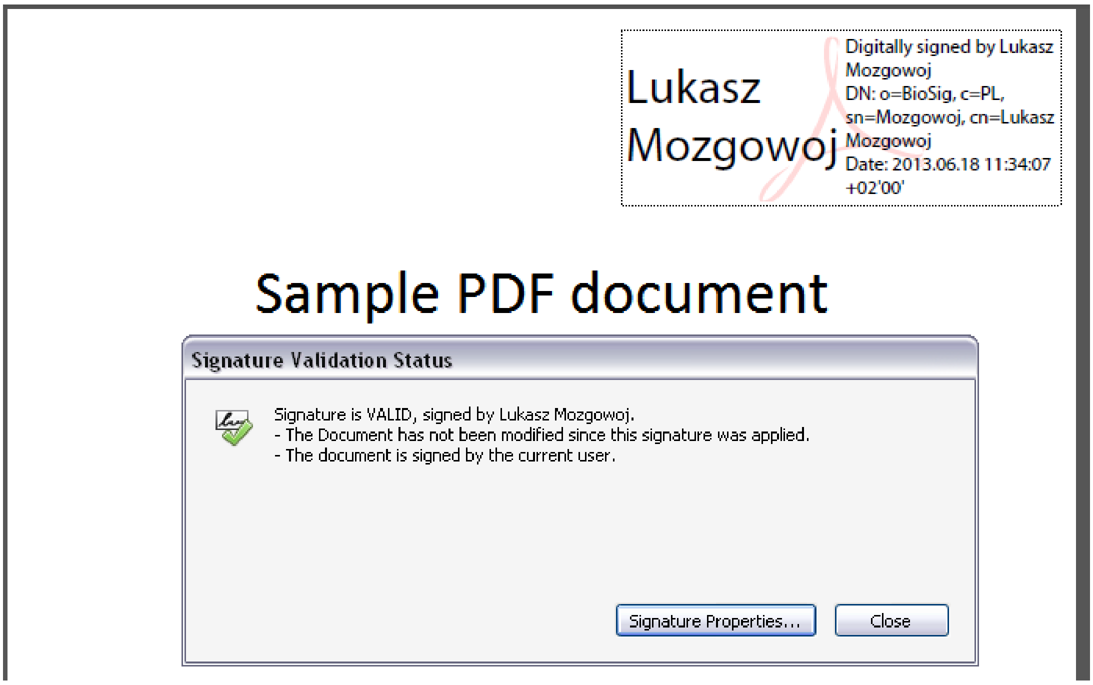
Register new user

Please provide user personal information to complete registration process.

PESEL	89080415398		
Name	Lukasz	Surname	Mozgowoj
Country	Poland	State/Province	Krakow
City	Krakow	Postal code	32-100
Street	Budryka 2		
Signature	C:\Documents and Settings\A		

Fig. 11. User registration

In fig.12, a PDF document signed on the bioPKI platform is presented. As one may see, it was signed by the user identified by the 89080415399 personal ID number.



Lukasz Mozgowoj

Digitally signed by Lukasz Mozgowoj
DN: o=BioSig, c=PL, sn=Mozgowoj, cn=Lukasz Mozgowoj
Date: 2013.06.18 11:34:07 +02'00'

Sample PDF document

Signature Validation Status

Signature is VALID, signed by Lukasz Mozgowoj.
- The Document has not been modified since this signature was applied.
- The document is signed by the current user.

Fig. 12. PDF document signed on bioPKI platform

VIII. Conclusions

When vulnerable data is transmitted in public network(s), one always runs the risk of someone intercepting, spoofing, or compromising their confidential information.

Many contemporary communication applications are equipped with built-in data encryption and protection mechanisms. Unfortunately, many of them (including the most popular ones) are based on relatively insecure solutions based on PINs, tokens, or passwords. Even the ones based on the key security often prefer usability over security and they store (private) keys directly in the file system of the station where they are used.

In this paper, the motivation, idea, basic assumptions, requirements, and top-level architecture of the server-side digital signature system with biometric authorization are discussed. Also, a sample applications and the integration layer are presented.

The most important requirements have been met: i.e.:

- since storing (private) keys is moved inside the (super) secure environment of the bioPKI platform, the end user doesn't have to care about protecting his keys;
- since key access is authorized biometrically, any other devices, cards or dongles for authorization and authentication have been eliminated (as they are awkward in everyday use).

The proposed platform has many advantages, but it would be almost useless if its integration with both: future and current systems and third-party applications was difficult, expensive, or simply impossible.

That is why when the platform was designed, the integration layer consisting of cryptographic libraries implementing CSP and PKCS#11 standards was assumed and then implemented.

Thus, any application, system, or middle-tier compatible with any of these two standards (in practice, almost any software requiring cryptographic operation and not being closed hermetic solutions) can be integrated with bioPKI easily and smoothly. In this paper it was presented on the basis of Thunderbird and Adobe Reader applications.

Acknowledgment

This work was supported by a grant UDA-POIG.01.04.00-12-041/11-00.

References

[1] Ballad B., Ballad T., Banks E., Access Control, Authentication, and Public Key Infrastructure. 1st Edition, 2010.

[2] Bement, A.L. Security requirements for cryptographic modules, Information Technology Laboratory, National Institute of Standards and Technology, 2001.

[3] Bhattacharyya D., Ranjan R., Alisherov A., Choi M., Biometric Authentication: A Review, International Journal of u- and e- Service, Science and Technology Vol. 2, No. 3, September, 2009.

[4] Boulgouris N. V., Plataniotis K.N., Micheli-Tzanakou E., Biometrics: Theory, Methods, and Applications (IEEE Press Series on Computational Intelligence). 1st Edition, 2009.

[5] Chapman, M.R., SaaS Enterpreneur. The definite guide to success in your cloud application business, Softletter, 2012.

[6] Diffie W., Hellman M.E., New Directions In Cryptography, IEEE Transactions in Information Theory 22:(6), 644-654, 1976.

[7] Heseltine T., Pears N., Austin J., Chen Z., Face Recognition: A Comparison of Appearance-Based, Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney.

[8] Himaga M., Kou K., Finger vein authentication technology and financial applications, w: Advances in Biometrics, Springer Verlag, London, 2008

[9] Huang B., Dai Y., Li R., Tang W., Li W., Finger-vein Authentication Based on Wide Line Detector and Pattern Normalization, International Conference on Pattern Recognition, 2010.

[10] Jain A.K., Ross A.A., Nandakumar K., Introduction to Biometrics. 1st Edition, 2011.

[11] McGrath, M.P. Understanding PaaS, O'Reilly Media, 2012.

[12] Menezes A.J., van Oorschot P.C., Vanstone S.A., Handbook of Applied Cryptography. Edycja V, 2005.

[13] Mitnick, K.D., Simon, W.L., Wozniak S., The art of Deception Controlling the human element of security, Wiley Publishing, 2002.

[14] Orthacker C., Centner M.,Kittl, C., Qualified Mobile Server Signature, Security and Privacy – Silver Linings in the Cloud, IFIP Advances in Information and Communication Technology, Springer Berlin Heidelberg, 2010https://online.tugraz.at/tug_online/voe_main2.getvolltext?pCurrPk=52961.

[15] Schneier, B. Applied cryptography, 2nd Edition, John Wiley and Sons, 1996.

[16] Vacca J.R., Public Key Infrastructure: Building Trusted Applications and Web Services. 1st Edition, 2004.

[17] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, version 3.1, Revision 4, CCMB-2012-09-003 https://www.niap-ccevs.org/Documents_and_Guidance/cc_docs/CCPART3V3.1R4.pdf.

[18] Government factsheet, February 2010. http://www.epractice.eu/files/eGovernment%20in%20LI%20-%20Feb%202010%20-%208.0.pdf.

[19] Microsoft, The Cryptography API, or How to Keep a Secret, 2013 http://msdn.microsoft.com/en-us/library/ms867086.aspx.

[20] Cipher Corporation Ltd., nCipher Security World – White paper,2001,http://www.cc.com.pl/pl/prods/ncipher/pdf/ncipher_security_world_wp.pdf.

[21] Public Statement on Server Based Signature Services, Forum of European Supervisory Authorities for Electronic Signatures (FESA), October 17, 2005: http://www.fesa.eu/public-documents/PublicStatement-ServerBasedSignatureServices-20051027.pdf.

[22] RSA Laboratories, PKCS #11 v2.11: Cryptographic Token Interface Standard, 2001, http://www.clizio.com/download/pkcs-11v2-11r1.pdf.

[23] Thales Security World – A secure Key management Architecture for the Thales nShield Family of Hardware Security Modules – Thales White Paper.

[24] Polish Digital Signature Act. Dz.U.01.130.1450, on the basis of European directive: EU 1999/93/EC, September 2001.

Authors' Profiles

Leszek Siwik has graduated with honors from Computer Science at the AGH-UST University of Science and Technology in 2002, next he has graduated from Department of Management at the AGH-UST in 2004. He works as an Assistant Professor at the Department of Computer Science of AGH-UST where in 2009 he obtained his Ph.D. with honors in Computer Science in artificial intelligence area. His research focuses on multi-agent systems in multi-objective optimization, security and cryptography and mobile systems.

Lukasz Mozgowoj has graduated from Computer Science at the AGH-UST University of Science and Technology in 2013. He has successfully participated in many mathematical competitions and got Minsterial scholarship for top students. He works for Biotrustis Biometric Trust Informations as IT Project Manager (R&D Department). His current work focuses on biometric authorization and authentication.

How to cite this paper: Leszek Siwik, Lukasz Mozgowoj,"Server-Side Encrypting and Digital Signature Platform with Biometric Authorization", IJCNIS, vol.7, no.4, pp.1-13, 2015.DOI: 10.5815/ijcnis.2015.04.01