applied system innovation

Article

Analyzing and Comparing the Performance of National Biometric eID Card in Heavy Cryptographic Applications

Gazmend Krasniqi [1], Petrit Rama [2] and Blerim Rexha [2,*]

[1] Faculty of Technical Sciences, University Ismail Qemali, 9401 Vlora, Albania; gazmend.krasniqi@ubt-uni.net
[2] Faculty of Electrical and Computer Engineering, University of Prishtina, 10000 Prishtina, Kosovo; petrit.rama@uni-pr.edu
* Correspondence: blerim.rexha@uni-pr.edu

Received: 27 August 2018; Accepted: 21 September 2018; Published: 27 September 2018

Abstract: Today, we are witnessing increased demand for more speed and capacity in the Internet, and more processing power and storage in every end user device. Demand for greater performance is present in every system. Electronic devices and their hosted applications need to be fast, but not to lose their main security features. Authentication and encryption are the main processes in the security aspect, and are required for a secure communication. These processes can be executed in different devices, among them PCs, microprocessors, microcontrollers, biometric cards or mobile devices. Biometric identity cards are becoming increasingly popular, challenging traditional PC devices. This paper compares two processing systems, the efficiency of encryption and signatures on the data executed in national identity biometric card versus PC, known also as the match-on-card versus the match-off-card. It considers how different parameters impact the process and the role they play on the overall process. The results, executed with a predefined set of test vectors, determine which processing system to use in a certain situation. Final conclusions and recommendations are given taking into consideration the efficiency and security of the data.

Keywords: cryptography; digital signature; match-on-card; match-off-card; eID biometric card

1. Introduction

Biometric protocols are being used not only to authenticate an individual to an official authority but also to encrypt or sign a message. These protocols are based on biometric traits, which are universal and unique. There are a lot of methods to implement biometric identification, and a biometric card, usually known as a national electronic identification card (eID), is one of them.

Biometric eID cards can perform authentication, encryption and data signature, because of the private parameters (keys) stored on the card. The private parameters are stored in a card, as a biometric template during the enrolment stage. This template is used for comparison between the new template and corresponding template on the card, using two processing systems: match-off-card and match-on-card. Match-on-card compares the template of the card with the fresh template, while the processing of biometric data is done on the card and never leaves the card. On the other hand, match-off-card processing is not done on the card, but in and outside a device or system. Match-off-card and match-on-card are presented in Figures 1 and 2.

Each processing system has its own advantages and disadvantages, as described in [1]. Match-off-card is faster, because of the processing power from the system, but because the biometric template leaves the card, this approach presents a security risk for the sensitive information inside the card. However, match-on-card has lower computation power, higher security because the biometric template does not leave the card, and lack of interoperability is a problem in this processing system [1]. A general introduction on the match-on-card can be found on [2], where the main advantages of this processing system are discussed, like a decentralized biometric database, data mobility, enhanced privacy and security [3].

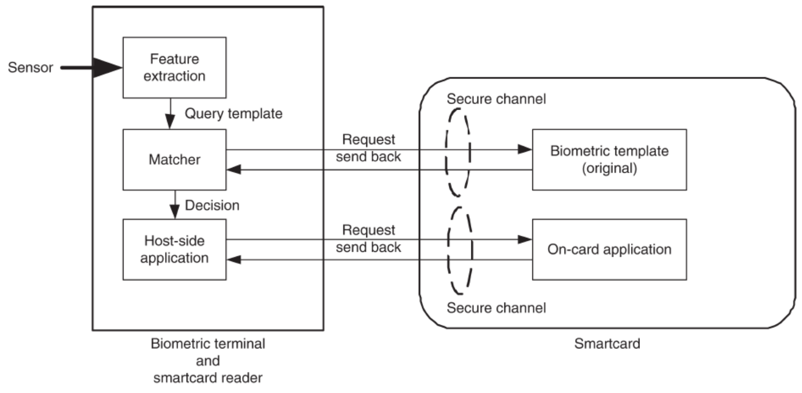


Figure 1. Match-off-card system. Reprinted with permission from [2]. © 2009 Springer US.

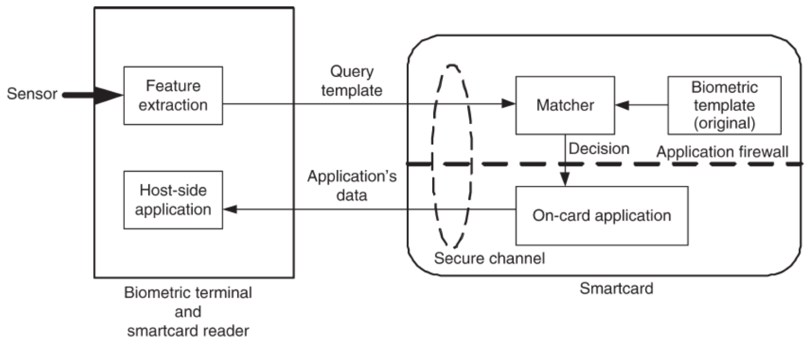


Figure 2. Match-on-card system. Reprinted with permission from [2]. © 2009 Springer US.

Encryption processed with a biometric card was proposed in [4]. The encryption is done using the biometric key, a unique key generated from the biometric template inside the card. Except for the encryption process, this key can be used for the matching and authentication process also [4].

As the technology is progressing, a need for a faster data processing system is increasing too, making the speed the main parameter for analyzing. The National Institute of Standards and Technology (NIST) has done an evaluation of the accuracy and speed of fingerprint match-on-card process. Minimum error rate, the speed of execution, and the accuracy were only a few of the parameters tested in this experiment and discussed in an extended report [5]. Fingerprint authentication using a match-on-card is also presented in [6], where the authors observed the performance of successful and unsuccessful authentication. The positive verification time is slower than the negative verification time, which is also a security issue. In this paper, performance of enrollment and the false acceptance rate of the verification process are also analyzed.

NIST has done another experiment, a feasibility study, to determine if biometric match-on-card authentication could be performed in less than 2.5 s. The protocol for this operation consisted of few steps, presenting the card to the contactless reader from the cardholder, presenting the finger to the scanner, a secure session establishment process, transmitting the encrypted fingerprint template to the card, decrypting the template, and in the end returning the matching test result. In the report, the main parameters measured are average time to establish a secure session, average time for transmission of encrypted biometric data, and average total time to perform this complete process [7].

This paper proposes a different approach than the papers mentioned above. The main parameter is the efficiency of two processing systems, match-on-card versus match-off-card. This paper will not consider the enrolment or verification stage of the biometric card, but will just use the template for encryption and data signatures, and compares this process with the same process, using the personal computer (PC) as the processing device.

A national eID biometric card stores the private and public key of its holder. Those keys will be used to encrypt and sign the data, which represent the match-on-card processing system. The speed of the processing will be compared against the match-off-card.

Moreover, this paper does not address authentication of the user or the speed which the user authenticates himself. The main experiment compares match-on-card and match-off-card efficiency. The experiment takes into consideration different parameters, in the form of file size, processing algorithms for encryption, or signature. Each parameter plays a role, and it has an impact on the whole process regarding its efficiency.

This paper is organized as follows. Section 2 describes the national biometric identity card and its security features. In Section 3 we describe the architecture of the application used to perform the experiments. In Section 4 we show the results of the experiments. We conclude the paper with Section 5, in which we describe the main results from this paper, recommendations and possible future work.

2. National Biometric Identity Cards

A biometric identity card (ID) is a credit card size format and contains personal and biometric information about its holder in printed form as well as in electronic format and is used to authenticate its bearer in the real as well in the Internet world. Such electronic ID card uses proven smart card technology to communicate to the outside world, based on recommendations and guidelines issued by the International Civil Aviation Organization (ICAO), a body run by the United Nations with a mandate for setting international travel document standards [8]. A user profile stored in the biometric eID contains a digital X.509 certificate and its corresponding private key, in compliance with ICAO Public Key Infrastructure (PKI), signed by a country-issuing certification authority (CA).

The Ministry of Internal Affairs of the government of Kosovo issued first biometric national ID cards in December 2013, thus becoming the first country supporting the new Supplemental Access Control (SAC) protocol for mutual authentication [9].

The Kosovo national biometric ID card hosts three applications, as presented in Figure 3, and it uses a SLE 78CLX1280P 16 bit crypto processor from Infineon. It has 128 kByte Electrically Erasable Programmable Read-Only Memory (EEPROM) and supports Rivest–Shamir–Adleman (RSA) 4096 key bit length, elliptic-curve cryptography (ECC) up to 521 bit and triple Data Encryption Standard (3DES) and Advanced Encryption Algorithm (AES) up to 256 bit length and the communication with outside world is done using the near field communication (NFC) protocol [10].

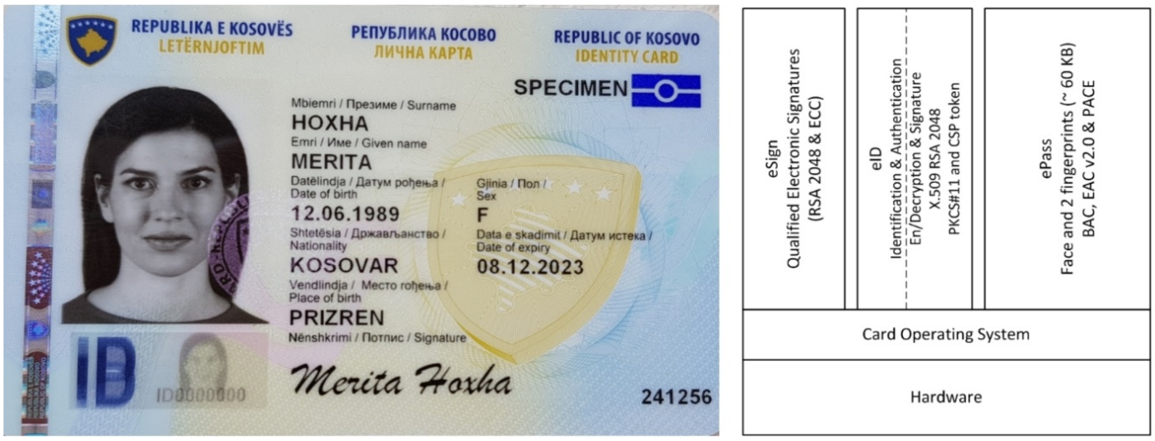

REPUBLIKA E KOSOVËS
LETËRNJOFTIM
РЕПУБЛИКА КОСОВО
ЛИЧНА КАРТА
REPUBLIC OF KOSOVO
IDENTITY CARD
SPECIMEN
Mbiemri / Презиме / Surname
HOXHA
Emri / Име / Given name
MERITA
Datëlindja / Датум рођења / Date of birth
12.06.1989
Gjinia / Пол / Sex
F
Shtetësia / Држављанство / Nationality
KOSOVAR
Data e skadimit / Датум истека / Date of expiry
08.12.2023
Vendlindja / Место рођења / Place of birth
PRIZREN
Nënshkrimi / Потпис / Signature
ID
ID0000000
241256
eSign
Qualified Electronic Signatures
(RSA 2048 & ECC)
eID
Identification & Authentication
En/Decryption & Signature
X.509 RSA 2048
PKCS#11 and CSP token
ePass
Face and 2 fingerprints (~ 60 KB)
BAC, EAC v2.0 & PACE
Card Operating System
Hardware

Figure 3. National biometric identity (ID) card and hosted apps.

The national ID card middleware communicates using the Public Key Cryptographic Standard (PKCS) #11 and Crypto Service Provider (CSP) with cryptographic interested apps. The web authentication with biometric ID card is done using X.509 certificates in two forms: (i) identity certificate or (ii) anonym certificate, whereby the corresponding 2048 bit private key never leaves the card [11]. Access to the private key is Personal Identification Number (PIN) protected, which is issued to the citizen in protected paper format. An Internet authentication scenario using a user's real and anonym profile stored in the eID card is presented in [12].

3. Preparing Testing Environment

This section offers an insight on the environment of the experiment. From this section, anybody can replicate the experiment and test the results, using different parameters with predefined test vectors.

3.1. Developing the Application

BiometricEfficiency_FIEK is an open source application for the Windows 10 operating system, developed in C# programming language using Microsoft Visual Studio 2015. BiometricEfficiency_FIEK does not install other libraries and it does not need any other prerequisites to be installed. The source code can be found in [13].

3.2. Smart Card Middleware

Staring from Windows 2000, Microsoft has integrated the usage of smart cards in Windows applications, as presented in Figure 4 [14]. *BiometricEfficiency_FIEK* uses vendor-specific Crypto Service Provider (CSP) functionality, wrapped as middleware software, to access the full functionality of biometric card cryptographic functions, such as: encrypt, decrypt, sign and verify.

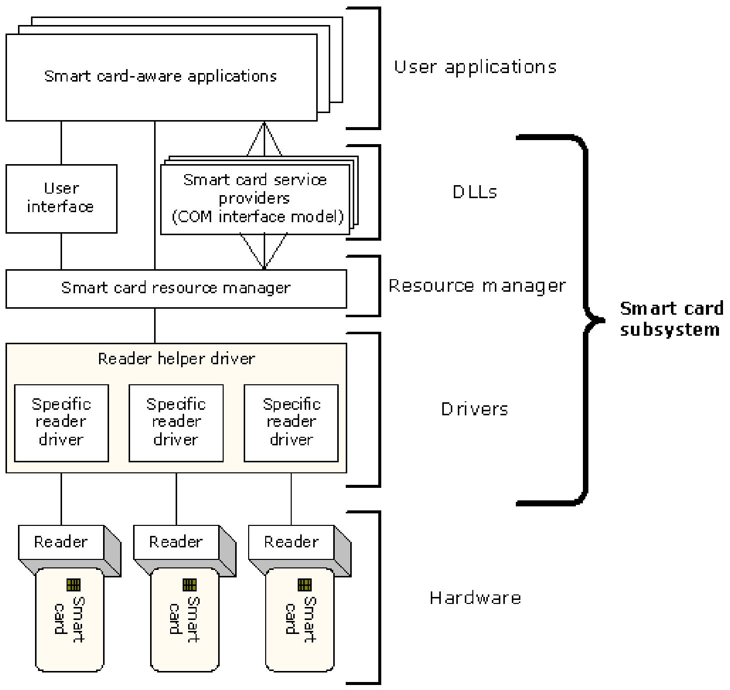


Figure 4. Smartcard Windows architecture [14].

3.3. Pseudocode

The source code is organized in helper classes, one for encryption and one helper class for digital signature.

The encryption helper class contains three methods for implementing match-on-card and match-off-card encryption using RSA and RSA CSP classes from the Microsoft NET framework. Each method initially divides the test vector in blocks, to encrypt each block, since we work with block encryption algorithms. Following methods are developed:

- **`encryptRSACSP_pc(text)`** takes one argument text of type string, which will be encrypted using the PC, using the public key stored locally on the PC.
- **`encryptRSACSP_card(text, certificate)`** takes two arguments, text of type string and certificate of type X509Certificate2. This method encrypts using the RSA CSP class, with the public key from the certificate on the eID biometric card.
- **`encryptRSA_card(text, certificate)`** also encrypts using the biometric card, but using RSA class.

```
function encryptRSACSP_pc(text)
{
    segmentLength ← 212
    loopLength ← text.Length/segmentLength+1

    RSACryptoServiceProvider rsa
    rsa.setPublicKey ← readPublicKey()

    for i←0 to loopLength do
        if (i=loopLength-1 or text.Length<segmentLength)
            copyLength ← text.Length-(i*segmentLength)
        else
            copyLength ← segmentLength

        segment ← text.Substring(i*segmentLength, copyLength);
        rsa.Encrypt(segment)
}
```

```
function encryptRSACSP_card(text, certificate)
{
    segmentLength ← 212;
    loopLength ← text.Length/segmentLength+1;

    RSACryptoServiceProvider rsa ← certificate.PublicKey.Key;
    for i ← 0 to i < loopLength do
        if (i=loopLength-1 or text.Length<segmentLength)
            copyLength ← text.Length-(i*segmentLength);
        else
            copyLength ← segmentLength;

        segment ← text.Substring(i*segmentLength, copyLength);
        rsa.Encrypt(segment);
}
```

```
function encryptRSA_card(text, certificate)
{
    segmentLength ← 212
    loopLength ← text.Length/segmentLength+1

    RSA rsa ← certificate.GetRSAPublicKey()

    for i←0 to loopLength do
        if (i=loopLength-1 or text.Length<segmentLength)
            copyLength ← text.Length-(i*segmentLength)
        else
            copyLength ← segmentLength

        segment ← text.Substring(i*segmentLength, copyLength);
        rsa.Encrypt(segment)
}
```

The signature helper class contains three methods for implementing match-on-card and match-off-card digital signature using RSA and RSA CSP classes, as:

- **`signRSACSP_pc(text)`** method is used to sign the text data, using the asymmetric algorithm RSA CSP, using the private key stored on the PC.
- **`signRSACSP_card(text, certificate)`** will be used as a method to sign the text data, using the private key in the certificate.
- **`signRSA_card(text, certificate)`** takes two arguments, one the text to sign, and the certificate, which uses the private key to digitally sign the data, with RSA class.

```
function signRSACSP_pc(text)
{
    RSACryptoServiceProvider rsa
    rsa.setPublicKey ← readPublicKey()

    rsa.SignData(text);
}
```

```
function signRSACSP_card(text, certificate)
{
    RSACryptoServiceProvider rsacsp ← certificate.PrivateKey
    rsacsp.SignData(text)
}
```

```
function signRSA_card(text, certificate)
{
    RSA rsa ← certificate.PrivateKey
    rsa.SignData(text)
}
```

3.4. Software Functionalities

BiometricEfficiency_FIEK has a simple interface, as presented in Figure 5. The application is used to encrypt or sign data, using the processing power of the PC or the processing power of a national biometric card. The main purpose of the application is to measure the efficiency, or the processing time of both those processing methods: match-off-card and match-on-card systems. This application is primarily used for experimental purposes, not for encryption or signing data.

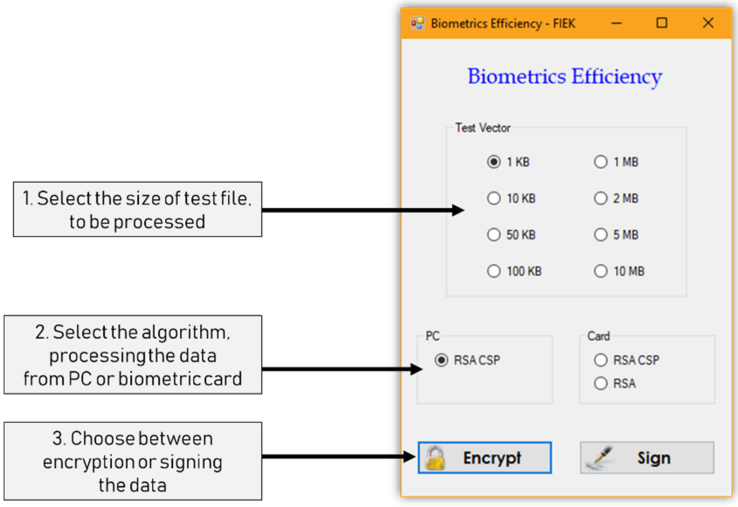
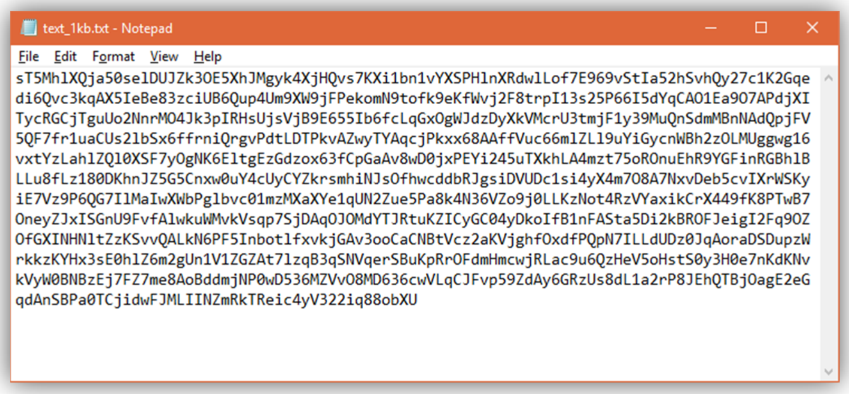

1. Select the size of test file, to be processed
2. Select the algorithm, processing the data from PC or biometric card
3. Choose between encryption or signing the data
Biometrics Efficiency - FIEK
Biometrics Efficiency
Test Vector
[✓ 1 KB]
10 KB
50 KB
100 KB
1 MB
2 MB
5 MB
10 MB
PC
[✓ RSA CSP]
Card
RSA CSP
RSA
Encrypt
Sign

Figure 5. The interface of *BiometricEfficiency_FIEK* app.

The test data used here are random text files of different lengths, which offers the opportunity to study the impact of length or size of the file on the processing time.

The basic steps for encrypting or signing data, with one of the processing systems are:

1. The first step is to choose the size of the text file which will be encrypted or signed. The test vector consists of eight text files with a random text of different size, 1 KB, 10 KB, 50 KB, 100 KB, 1 MB, 2 MB, 5 MB, 10 MB. Each file will have the different impact on the processing time, which will be discussed later. The text is random text, as presented in Figure 6.
2. The second step is to choose the processing system and the algorithm for processing the data. The app offers both processing systems: match-off-card using a PC as outside processing system and match-on-card using a national eID biometric card as a processing system. The PC interface implements only the RSA CSP [15] from the NET framework as the only processing algorithm. Whereas the national eID biometric card offers two processing algorithms: RSA [16] and RSA CSP. This comparison is the main experiment conducted in this paper, it measures the processing time of the two processing systems.
3. The third and last step is to choose if the user wants to encrypt or sign the selected data, with the selected algorithm and selected processing system. Each experiment is run 10 times and the results are written in a text file. This text file shows the execution time for each of 10 runs and the best time, worst time and the average time from the experiment.



```
sT5Mh1XQja50selDUJZk3OE5XhJMgyk4XjHQvs7KXi1bn1vYXSPH1nXRdwlLof7E969vStIa52hSvhQy27c1K2Gqe
di6Qvc3kqAX5IeBe83zciUB6Qup4Um9XW9jFPekomN9tofk9eKfWvj2F8trpI13s25P66I5dYqCA01Ea907APdjXI
TycRGCjTguUo2NnrMO4Jk3pIRHsUjsVjB9E655Ib6fcLqGxOgWJdzDyXkVMcrU3tmjF1y39MuQnSdmMBnNAdQpjFV
5QF7fr1uaCUs2lbSx6ffrniQrgvPdtLDTPkvAZwyTYAqcjPkxx68AAffVuc66mlZL19uYiGycnWBh2zOLMUggwg16
vxtYzLahlZQ10XSF7yOgNK6EltgEzGdzox63fCpGaAv8wD0jxPEYi245uTXkhLA4mzt75oROnuEhR9YGFinRGBhlB
LLu8fLz180DKhnJZ5G5Cnxw0uY4cUyCYZkrsmhiNJsOfhwcddbRJgsiDVUDc1si4yX4m7O8A7NxvDeb5cvIXrWSKy
iE7Vz9P6QG7IlMaIwXWbPglbvc01mzMXaXYe1qUN2Zue5Pa8k4N36VZo9j0LLKzNot4RzVYaxikCrX449fK8PTwB7
OneyZJxISGnU9FvfAlwkuWMvkVsqp7SjDAqOJOMdYTJRtuKZICyGC04yDkoIfB1nFASta5Di2kBROFJeigI2Fq9OZ
OfGXINHNltZzKSvvQALkN6PF5InbotlfxvkjGAv3ooCaCNBtVcz2aKVjghfOxdfPQpN7ILLdUDz0JqAoraDSDupzW
rkkzKYHx3sE0hlZ6m2gUn1V1ZGZAt7lzqB3qSNVqerSBuKpRrOFdmHmcwjRLac9u6QzHeV5oHstS0y3H0e7nKdKNv
kVyW0BNBzEj7FZ7me8AoBddmjNP0wD536MZVvO8MD636cwVLqCJFvp59ZdAy6GRzUs8dL1a2rP8JEhQTBjOagE2eG
qdAnSBPa0TCjidwFJMLIINZmRkTReic4yV322iq88obXU
```

Figure 6. The 1 KB text file.

4. Experimental Results

All the experiments conducted in this paper are done using *biometricefficiency_FIEK*, as described in Section 3, and its source code is freely available on GitHub.

The first step is to select the size of the text file for the experiment. The user can select between 1 KB, 10 KB, 50 KB, 100 KB, 1 MB, 2 MB, 5 MB and 10 MB. Those files, each will have impact on both processes in different ways, which affect the efficiency or time needed for processing.

The second step, which is the main step is to select the algorithm. By selecting the algorithm, we select also the processing system. The match-off-card implements only RSA CryptoServiceProvider, and the processing is done on a PC. On the other hand, the match-on-card implements two algorithms RSA and RSA CryptoServiceProvider, and the national eID biometric card will be used as the processing system. The main test case in this paper will compare the efficiency of the match-off-card and match-on-card, for different data size, algorithms and processes.

The final step is to choose the process, encryption or signing. Both can be processed on a PC or card and will influence differently the processing time.

As explained above, different parameters will have an impact on the efficiency. All these parameters will be grouped in three test cases. The first test case will compare two NET framework classes RSA and RSA CSP, in the encryption process using match-on-card technology. The second test case will compare the efficiency of the match-off-card and match-on-card processing system in the encryption, using RSA CSP class. The third test case will compare again match-off-card and match-on-card processing system, but now in the signing process, again using the RSA CSP class.

4.1. RSA vs. RSA Crypto Service Provider (CSP)—Encryption with National eID Biometric card

The first experiment will compare the RSA class and RSA Crypto Service Provider class from the NET framework, in the encryption process, using the match-on-card processing system on national eID biometric card.

The experiment will include all test vectors, and for each text file, the experiment will be executed 10 times. This will serve the accuracy of the experiment and will help to generate the average time, best time and worst time of execution.

The experimental results, for all test vectors are shown in Table 1 and graphically in Figure 7.

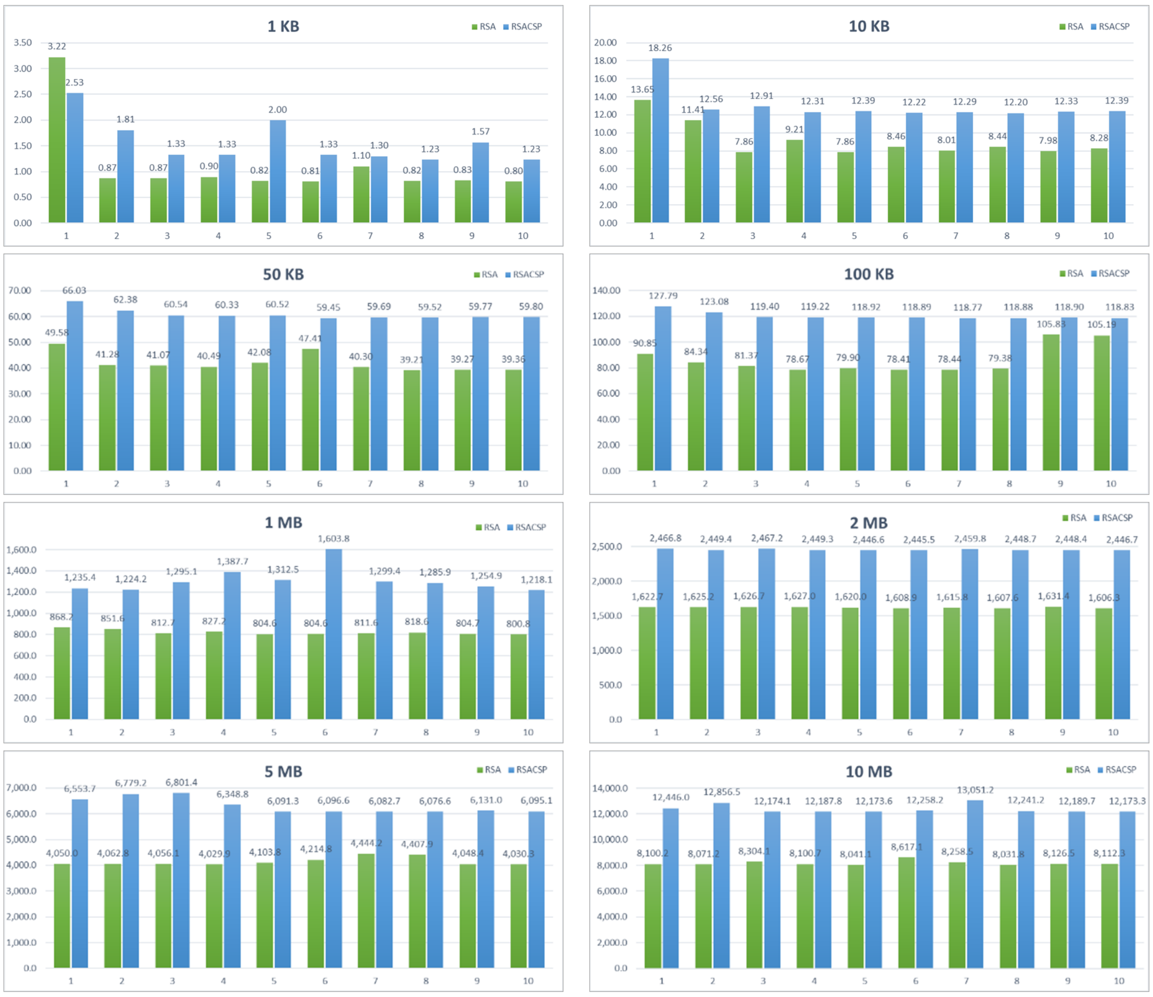


Figure 7. Graphical results of the experiment.

Table 1. Results RSA vs. RSA Crypto Service Provider (CSP).

Size	**1 KB**		**10 KB**		**50 KB**		**100 KB**		**1 MB**		**2 MB**		**5 MB**		**10 MB**	
#.	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP
1	3.22	2.53	13.65	18.26	49.58	66.03	90.85	127.79	868.2	1235.4	1622.7	2466.8	4050.0	6553.7	8100.2	12,446.0
2	0.87	1.81	11.41	12.56	41.28	62.38	84.34	123.08	851.6	1224.2	1625.2	2449.4	4062.8	6779.2	8071.2	12,856.5
3	0.87	1.33	7.86	12.91	41.07	60.54	81.37	119.40	812.7	1295.1	1626.7	2467.2	4056.1	6801.4	8304.1	12,174.1
4	0.90	1.33	9.21	12.31	40.49	60.33	78.67	119.22	827.2	1387.7	1627.0	2449.3	4029.9	6348.8	8100.7	12,187.8
5	0.82	2.00	7.86	12.39	42.08	60.52	79.90	118.92	804.6	1312.5	1620.0	2446.6	4103.8	6091.3	8041.1	12,173.6
6	0.81	1.33	8.46	12.22	47.41	59.45	78.41	118.89	804.6	1603.8	1608.9	2445.5	4214.8	6096.6	8617.1	12,258.2
7	1.10	1.30	8.01	12.29	40.30	59.69	78.44	118.77	811.6	1299.4	1615.8	2459.8	4444.2	6082.7	8258.5	13,051.2
8	0.82	1.23	8.44	12.20	39.21	59.52	79.38	118.88	818.6	1285.9	1607.6	2448.7	4407.9	6076.6	8031.8	12,241.2
9	0.83	1.57	7.98	12.33	39.27	59.77	105.83	118.90	804.7	1254.9	1631.4	2448.4	4048.4	6131.0	8126.5	12,189.7
10	0.80	1.23	8.28	12.39	39.36	59.80	105.19	118.83	800.8	1218.1	1606.3	2446.7	4030.3	6095.1	8112.3	12,173.3

Two things characterize this experiment. Firstly, as depicted in Figure 7, one can see from the first experiment that more time (expressed in milliseconds) will be spent at the beginning of an experimental cycle. This can be seen especially in the 1 KB and 10 KB experiment, where the processing time is greater in the first cycle. This because the time needed to load the data on the memory, as basic concepts from the memory organization of smart card memory [16]. After that, the time needed for processing is shorter.

The second thing to notice is the change in processing time, when we increase the size of data. The processing time will be increased when we process a larger amount of data, but the class RSA performs better than the class RSA CSP. The difference in processing time increases with each larger data set, as shown in Table 2 and Figure 8.

In this case, we can conclude that the class RSA is more efficient than RSA CSP, especially when we have a large amount of data.

Table 2. Average time of RSA vs. RSA CSP.

	Average Time (ms)		
Size	**RSA**	**RSA CSP**	**Diff**
1 KB	1.10	1.56	41.72%
10 KB	9.12	12.99	42.45%
50 KB	42.00	60.80	44.76%
100 KB	86.24	120.27	39.46%
1 MB	820.46	1311.71	59.88%
2 MB	1619.16	2452.83	51.49%
5 MB	4144.82	6305.64	52.13%
10 MB	8176.34	12,375.17	51.35%

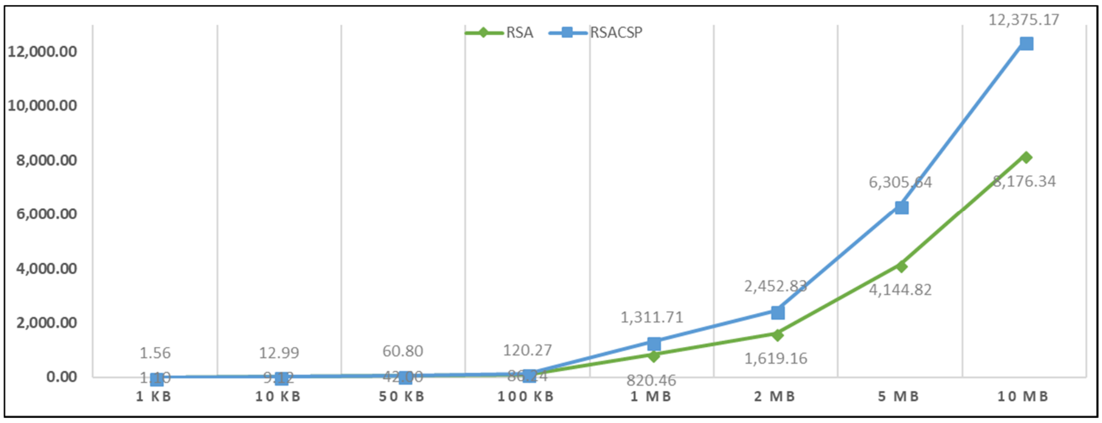


Figure 8. Graphical results of RSA vs. RSA CSP.

4.2. Personal Compuer (PC) vs. Card—Encryption Using RSA CSP Class

In the second experiment we will compare the two processing systems, match-off-card vs. match-on-card. As stated above, match-off-card processing will be done using a PC, Intel Core i5 5200U CPU 2.20 GHz, 8 GB of RAM, Windows 10 64-bit operating system and using the RSA CSP class from NET framework. However, the match-on-card uses the smart card reader to transfer the information between the national eID biometric smart-card and the PC.

As in the first experiment, this experiment will include all eight text files and the experiment will be executed 10 times for each file. The experimental results for all test vectors are shown in Table 3 and Figure 9.

Table 3. Results of personal computer (PC) vs. card.

Size	**1 KB**		**10 KB**		**50 KB**		**100 KB**		**1 MB**		**2 MB**		**5 MB**		**10 MB**	
#	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP
1	3.16	2.53	12.55	18.26	60.67	66.03	120.60	127.79	1217.8	1235.4	2479.8	2466.8	6603.3	6553.7	13,160.6	12,446.0
2	1.48	1.81	20.58	12.56	59.56	62.38	119.04	123.08	1214.6	1224.2	2438.2	2449.4	6419.6	6779.2	12,201.2	12,856.5
3	1.33	1.33	12.12	12.91	59.65	60.54	118.85	119.40	1216.3	1295.1	2434.1	2467.2	6257.3	6801.4	12,180.8	12,174.1
4	1.29	1.33	12.07	12.31	68.15	60.33	119.35	119.22	1216.6	1387.7	2440.3	2449.3	6359.9	6348.8	12,173.6	12,187.8
5	1.28	2.00	12.05	12.39	59.38	60.52	118.84	118.92	1223.2	1312.5	2431.2	2446.6	6083.2	6091.3	12,155.7	12,173.6
6	1.24	1.33	12.54	12.22	59.50	59.45	118.75	118.89	1219.8	1603.8	2434.7	2445.5	6084.8	6096.6	12,931.1	12,258.2
7	1.25	1.30	12.57	12.29	59.45	59.69	118.80	118.77	1220.2	1299.4	2434.5	2459.8	6126.9	6082.7	12,235.5	13,051.2
8	1.24	1.23	12.11	12.20	60.55	59.52	119.10	118.88	1217.1	1285.9	2432.5	2448.7	6106.4	6076.6	12,215.9	12,241.2
9	1.23	1.57	12.06	12.33	60.70	59.77	119.27	118.90	1217.0	1254.9	2429.7	2448.4	6079.7	6131.0	12,179.4	12,189.7
10	1.23	1.23	11.99	12.39	59.44	59.80	118.84	118.83	1217.6	1218.1	2517.4	2446.7	6076.0	6095.1	12,272.7	12,173.3

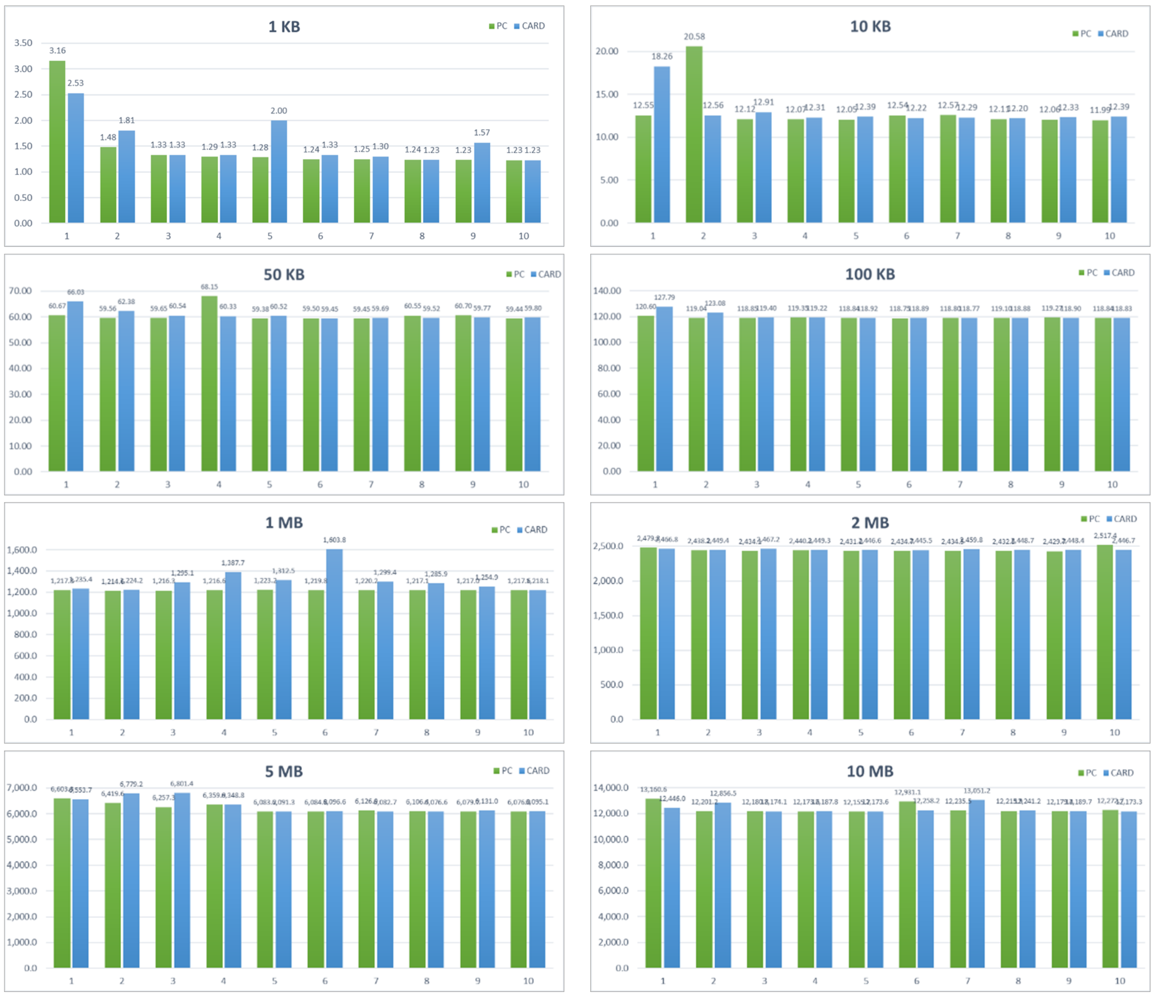


Figure 9. Graphical results of the experiment.

As with the previous experiment, more time will be needed at the beginning of each experiment, for the same reason as before. PCs also have internal memory, and more time will be needed for data to load and stored there [17].

The main conclusion from this experiment that we can draw is that the processing time increases exponentially with the size of a file, as shown in Table 4 and Figure 10. But, we cannot draw a conclusion as to which processing time is more efficient, since both systems perform roughly the same. So, neither of the match-on-card or match-off-card processing systems performs better and both can be used for encryption of information.

Table 4. Average time of PC vs. card.

Average Time (ms)			
Size	**PC**	**Card**	**Diff**
1 KB	1.47	1.56	6.20%
10 KB	13.06	12.99	−0.59%
50 KB	60.70	60.80	0.16%
100 KB	119.14	120.27	0.95%
1 MB	1218.03	1311.71	7.69%
2 MB	2447.24	2452.83	0.23%
5 MB	6219.72	6305.64	1.38%
10 MB	12,370.65	12,375.17	0.04%

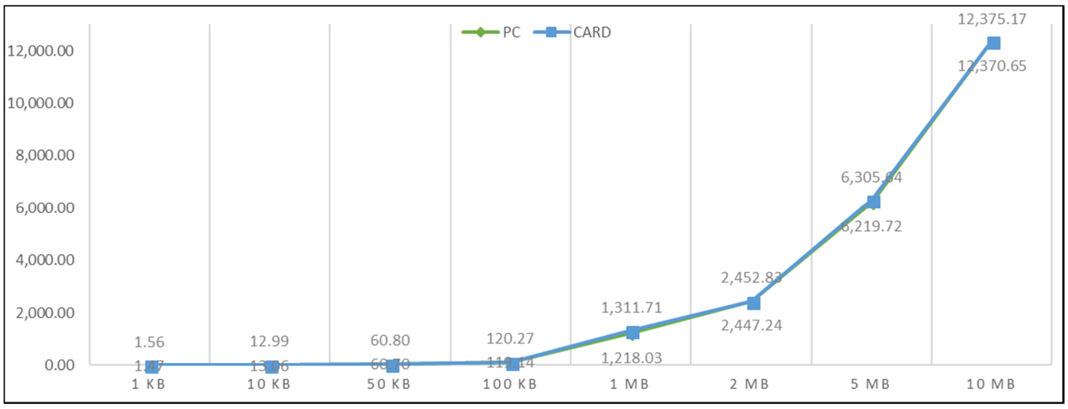


Figure 10. Average time of PC vs. card.

4.3. PC vs. Card—Signing Using RSA CSP Class

The third and last experiment compares again two processing systems, match-off-card vs. match-on-card, using the signing process. We use the same devices, PC for the match-off-card and national eID bio-metric card for match-on-card.

The results of the experiment, for all eight text files and 10 cycles for each file, are shown in Table 5 and Figure 11.

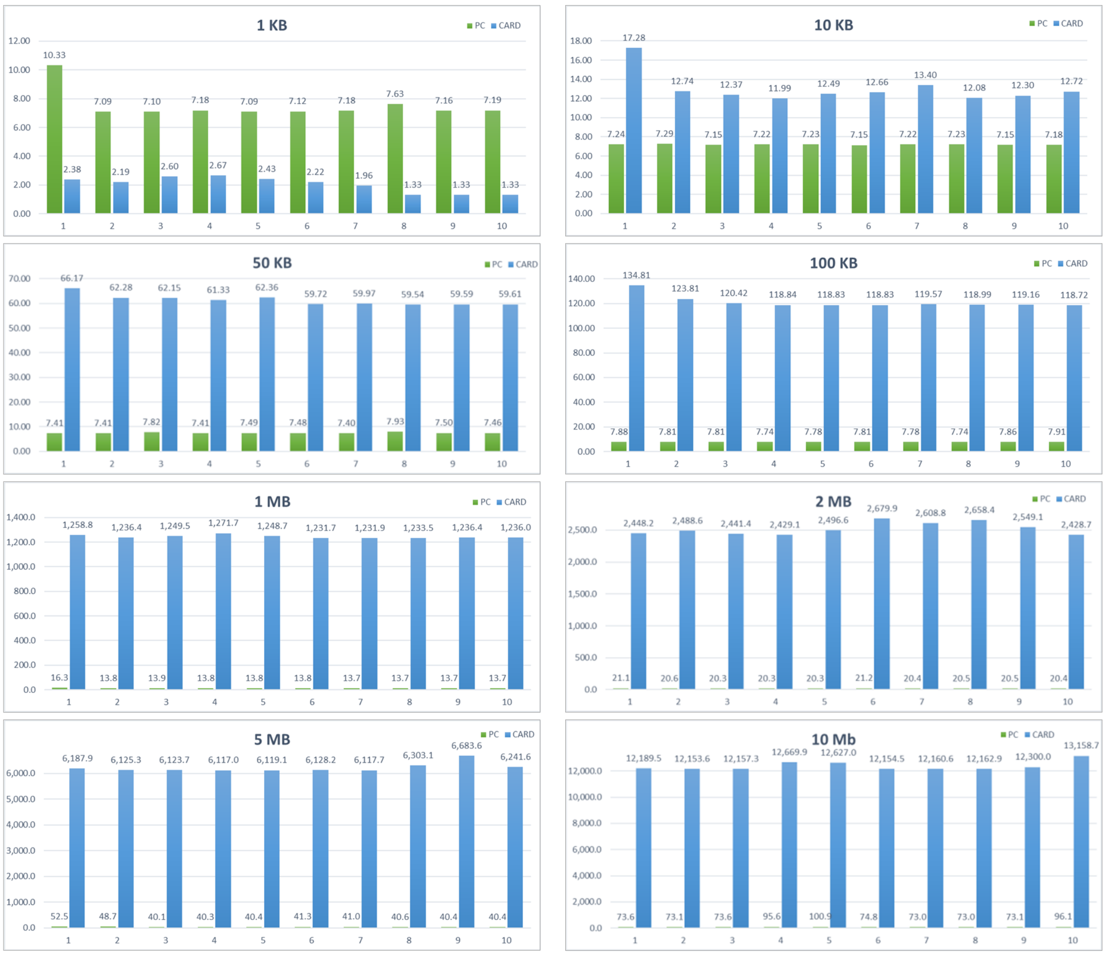


Figure 11. Graphical results of the experiment for signing RSA vs. card.

Table 5. Signing RSA vs. card.

Size	**1 KB**		**10 KB**		**50 KB**		**100 KB**		**1 MB**		**2 MB**		**5 MB**		**10 MB**	
#	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP	RSA	CSP
1	10.33	2.38	7.24	17.28	7.41	66.17	7.88	134.81	16.3	1258.8	21.1	2448.2	52.5	6187.9	73.6	12,189.5
2	7.09	2.19	7.29	12.74	7.41	62.28	7.81	123.81	13.8	1236.4	20.6	2488.6	48.7	6125.3	73.1	12,153.6
3	7.10	2.60	7.15	12.37	7.82	62.15	7.81	120.42	13.9	1249.5	20.3	2441.4	40.1	6123.7	73.6	12,157.3
4	7.18	2.67	7.22	11.99	7.41	61.33	7.74	118.84	13.8	1271.7	20.3	2429.1	40.3	6117.0	95.6	12,669.9
5	7.09	2.43	7.23	12.49	7.49	62.36	7.78	118.83	13.8	1248.7	20.3	2496.6	40.4	6119.1	100.9	12,627.0
6	7.12	2.22	7.15	12.66	7.48	59.72	7.81	118.83	13.8	1231.7	21.2	2679.9	41.3	6128.2	74.8	12,154.5
7	7.18	1.96	7.22	13.40	7.40	59.97	7.78	119.57	13.7	1231.9	20.4	2608.8	41.0	6117.7	73.0	12,160.6
8	7.63	1.33	7.23	12.08	7.93	59.54	7.74	118.99	13.7	1233.5	20.5	2658.4	40.6	6303.1	73.0	12,162.9
9	7.16	1.33	7.15	12.30	7.50	59.59	7.86	119.16	13.7	1236.4	20.5	2549.1	40.4	6683.6	73.1	12,300.0
10	7.19	1.33	7.18	12.72	7.46	59.61	7.91	118.72	13.7	1236.0	20.4	2428.7	40.4	6241.6	96.1	13,158.7

This experiment is very interesting and one can draw few conclusions. Only in the 1 KB test case has the national biometric card performed better than PC. In the other seven text files, the PC has performed better. From Table 6 and Figure 12, we notice that match-off-card processing time increases very little when we increase the size of the test files. This is not the case in the match-on-card processing system, where the processing time increases exponentially when we increase the file size.

Table 6. Average time for signing PC vs. card.

Average Time (ms)			
Size	**PC**	**Card**	**Diff**
1 KB	7.51	2.04	−72.77%
10 KB	7.21	13.00	80.45%
50 KB	7.53	61.27	713.49%
100 KB	7.81	121.20	1451.75%
1 MB	14.01	1243.48	8772.78%
2 MB	20.57	2522.90	12,165.59%
5 MB	42.56	6214.72	14,500.78%
10 MB	80.68	12,373.40	15,237.16%

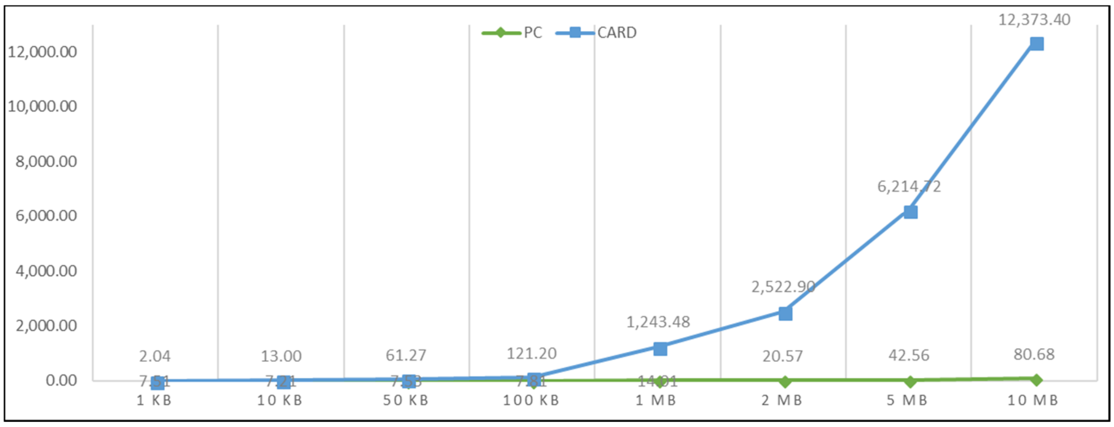


Figure 12. Average time for signing RSA vs. card.

So, from this experiment we can conclude that, in the signing process, overall match-on-card is more efficient than a match-off-card, especially for larger files. Match-on-card can still be used, in cases when we have small files to process.

Another form of data representation could be used, especially for representing Figures 8, 10 and 12 using the Weierstrass–Mandelbrot function as in [18,19], which will be future work.

5. Conclusions

Match-on-card and match-off-card are two processing systems used today for security processing. In this paper, match-on-card uses a national biometric card, with very advanced hardware architecture, to process the data, whereas a PC is used as a device in the match-off-card. Between them are many advantages and disadvantages, each playing a significant role when choosing them as the processing device.

As shown within experimental results, there are few cases where, usually when handling a small amount of data, a biometric card has a better performance; even for a very small amount of data, less than 1 kB, the biometric card outperforms the PC, as presented in Figure 12. With an increased amount of data, the performance of the biometric card decreases, as was expected, due to limited hardware resources of the biometric card, as described in Section 2.

Future work will add more functionality to *biometricefficiency_FIEK* app, such as using elliptic curve algorithms, verifying the digital signature, and adding more encryption algorithms such as the AES.

Author Contributions: G.K., P.R. and B.R. designed the application. P.R. developed the application. G.K. and P.R. conceived and designed the experiments. G.K. performed the experiments; B.R. and P.R. analyzed the data. G.K., P.R. and B.R. wrote the paper.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. *Biometrics for Payment Applications the SPA Vision on Financial Match-on-Card*; Smart Payment Association (SPA): Munich, Germany, November 2013.
2. Pang, C.T.; Yun, Y.W.; Xudong, J. On-Card Matching. In *Encyclopedia of Biometrics*; Springer: New York, NY, USA, 2009.
3. Smart Cards and Biometrics. In *A Smart Card Alliance Physical Access Council White Paper*; Smart Card Alliance: Princeton, NJ, USA, March 2011.
4. Bringer, J.; Chabanne, H.; Pointcheval, D.; Zimmer, S. An Application of the Boneh and Shacham Group Signature Scheme to Biometric Authentication. In Proceedings of the 3rd International Workshop on Security (IWSEC '08), Kagawa, Japan, 25–27 November 2008.
5. Grother, P.; Salamon, W.; Watson, C.; Indovina, M.; Flanagan, P. *MINEX II Performance of Fingerprint Match-on-Card Algorithms Phase II/III Report—NIST Interagency Report 7477*; Information Access Division—National Institute of Standards and Technology: Gaithersburg, MD, USA, 2009.
6. Security and Performance Evaluation Platform of Biometric Match on Card. In Proceedings of the International Conference on Mobile Applications and Security Management (ICMASM), Sousse, Tunisia, 22–24 June 2013.
7. Cooper, D.; Dang, H.; Lee, P.; MacGregor, W.; Mehta, K. *Secure Biometric Match-on-Card Feasibility Report*; NIST Interagency Report 7452; National Institute of Standards and Technology: Gaithersburg, MD, USA, November 2007. Available online: https://csrc.nist.gov/publications/detail/nistir/7452/final (accessed on 22 August 2018).
8. *ICAO Doc9303, Machine Readable Travel Documents*, 7th ed.; Available online: https://www.icao.int/publications/Documents/9303_p3_cons_en.pdf (accessed on 22 August 2018).
9. Rexha, B.; Imeraj, D.; Shabani, I. Using efficient TRNGs for PSEUDO profile in national eID card. *Int. J. Recent Contrib. Eng. Sci.* **2018**, *6*, 57–73. [CrossRef]
10. I. AG, Technical Details for SLE 78CLX1280P. Available online: http://www.infineon.com/ (accessed on 22 August 2018).
11. Giesecke & Devrient GmbH. Help files and technical notes for HIGHSEC eID App. Available online: https://mpb.rks-gov.net/eID.html (accessed on 22 August 2018).
12. Rexha, B.; Qerimi, E.; Neziri, V.; Dervishi, R. *Using eID Pseudonymity and Anonmity for Strengthing User Freedom in Internet*; Time for a European Internet; Central and Eastern European e | Dem and e | Gov Days 2015 Independence Day: Budapest, Hungary, 2015.
13. Krasniqi, G.; Rama, P.; Rexha, B. Source code of application developed and hosted by GitHub. Available online: https://github.com/petritrama-unipr/BiometricEfficiency_FIEK (accessed on 20 July 2018).
14. Microsoft. Smart Card Authentication. Available online: https://docs.microsoft.com/en-us/windows/desktop/secauthn/smart-card-authentication (accessed on 22 August 2018).
15. Microsoft. RSACryptoServiceProvider Class. NET Framework 4.7.2. Available online: https://msdn.microsoft.com/en-us/library/system.security.cryptography.rsacryptoserviceprovider(v=vs.110).aspx (accessed on 20 July 2018).
16. Rankl, W.; Effing, W. *Smart Card Handbook*; John Wiley & Sons Ltd.: London, UK, 2003.

17. Stallings, W. *Operating Systems: Internals and Design Principles*; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 2012.
18. Guariglia, E. Entropy and Fractal Antennas. *Entropy* **2016**, *18*, 1–17. [CrossRef]
19. Guariglia, E. Spectral Analysis of the Weierstrass-Mandelbrot Function. In Proceedings of the 2nd International Multidisciplinary Conference on Computer and Energy Science, Split, Croatia, 12–14 July 2017.